

# Вычислительные Машины, Системы и Сети





# Основные характеристики ЭВМ

**Электронная вычислительная машина (компьютер)** - комплекс технических и программных средств, предназначенный для автоматизации подготовки и решения задач пользователей.

**Структура** - совокупность элементов и их связей. Различают структуры технических, программных и аппаратно-программных средств.

## Группы характеристик ЭВМ

- *технические и эксплуатационные характеристики ЭВМ* (быстродействие и производительность, показатели надежности, достоверности, точности, емкость оперативной и внешней памяти, габаритные размеры, стоимость технических и программных средств, особенности эксплуатации и др.);
- *характеристики и состав функциональных модулей базовой конфигурации ЭВМ; возможность расширения состава технических и программных средств; возможность изменения структуры;*
- *состав программного обеспечения ЭВМ и сервисных услуг* (операционная система или среда, пакеты прикладных программ, средства автоматизации программирования).



# Технические и эксплуатационные характеристики ЭВМ

## Быстродействие

характеризуется числом команд, выполняемых ЭВМ за одну секунду.

Для измерения быстродействия используются следующие единицы измерения:

- **MIPS** (Mega Instruction Per Second) - миллион целочисленных операций в секунду;
- **MFLOPS** (Mega Floating Operations Per Second) - миллион операции над числами с плавающей запятой в секунду;
- **GFLOPS** (Giga Floating Operations Per Second) - миллиард операций над числами с плавающей запятой в секунду;
- **TFLOPS** (Tera Floating Operations Per Second) - триллион операций над числами с плавающей запятой в секунду.

## Производительность

объем работ, осуществляемых ЭВМ в единицу времени. Например, можно определять этот параметр числом задач, выполняемых за определенное время.

**Однако сравнение по данной характеристике ЭВМ различных типов может вызвать затруднения.**



# Технические и эксплуатационные характеристики ЭВМ

## Емкость запоминающих устройств

Емкость памяти измеряется количеством структурных единиц информации, которое может одновременно находиться в памяти.

Наименьшей структурной единицей информации является бит - одна двоичная цифра. Как правило, емкость памяти оценивается в более крупных единицах измерения - байтах (байт равен восьми битам). Следующими единицами измерения служат

**1 Кбайт =  $2^{10}$  = 1024 байта, 1 Мбайт = 1024 Кбайта, 1 Гбайт = 1024 Мбайта, 1 Тбайт = 1024 Гбайта.**

## Надежность

Способность ЭВМ при определенных условиях выполнять требуемые функции в течение заданного периода времени (стандарт ISO (Международная организация стандартов) 23 82/14-78).

## Точность

Возможность различать почти равные значения (стандарт ISO - 2382/2-76). Точность получения результатов обработки в основном определяется разрядностью ЭВМ, а также используемыми структурными единицами представления информации (байтом, словом, двойным словом).

## Достоверность

Свойство информации быть правильно воспринятой. Достоверность характеризуется вероятностью получения безошибочных результатов. Заданный уровень достоверности обеспечивается аппаратно-программными средствами контроля самой ЭВМ.



# Классификация средств ЭВТ

По принципу действия вычислительные машины делятся на три больших класса: аналоговые (**АВМ**), цифровые (**ЦВМ**) и гибридные (**ГВМ**).

Критерием деления вычислительных машин на эти три класса является форма представления информации, с которой они работают.

**АВМ** - вычислительные машины, которые работают с информацией, представленной в непрерывной (аналоговой) форме, т.е. в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).

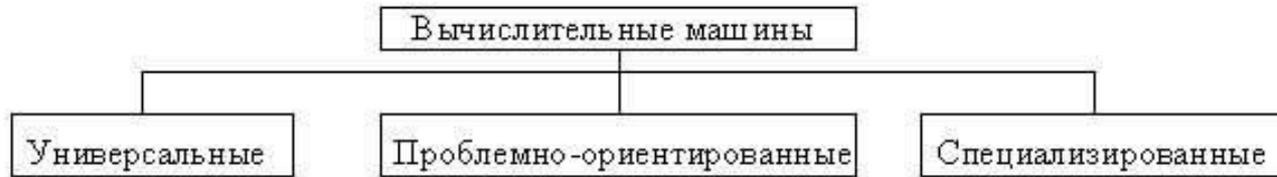
**ЦВМ** - вычислительные машины, предназначенные для работы с информацией, представленной в дискретной, цифровой форме.

**ГВМ** - вычислительные машины комбинированного действия, работают с информацией, представленной в цифровой и в аналоговой форме; они совмещают в себе достоинства АВМ и ЦВМ.

Гибридные вычислительные машины целесообразно использовать для решения задач управления сложными быстродействующими техническими комплексами.



# Классификация ЭВМ по назначению



- **Универсальные** ЭВМ предназначены для решения самых различных видов задач: научных, инженерно-технических, экономических, информационных, управленческих и других задач. В качестве универсальных ЭВМ используются различные типы компьютеров, начиная от супер-ЭВМ и кончая персональными ЭВМ. Решаемые на этих компьютерах задачи отличаются сложностью алгоритмов и объемами обрабатываемых данных.
- **Проблемно-ориентированные** ЭВМ служат для решения более узкого круга задач, связанных, как правило, с управлением технологическими объектами; регистрацией, накоплением и обработкой относительно небольших объемов данных; выполнением расчетов по относительно несложным алгоритмам. На проблемно-ориентированных ЭВМ, в частности, создаются всевозможные управляющие вычислительные комплексы.
- **Специализированные** ЭВМ используются для решения еще более узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация ЭВМ позволяет четко специализировать их структуру, во многих случаях существенно снизить их сложность и стоимость при сохранении высокой производительности и надежности их работы.



# Классификация ЭВМ по производительности



- **Большие ЭВМ** за рубежом часто называют мэйнфреймами (Mainframe). Они поддерживают многопользовательский режим работы (обслуживают одновременно от 16 до 1000 пользователей).
- **Малые ЭВМ (мини-ЭВМ)** - надежные, недорогие и удобные в эксплуатации компьютеры, обладающие несколько более низкими по сравнению с мэйнфреймами возможностями. В многопользовательском режиме поддерживают 16-512 пользователей.
- **Супер-ЭВМ** — мощные, высокоскоростные вычислительные машины (системы) с производительностью от сотен миллионов до триллионов операций с плавающей запятой в секунду. Супер-ЭВМ выгодно отличаются от больших универсальных ЭВМ по быстродействию числовой обработки, а от специализированных машин, обладающих высоким быстродействием в сугубо ограниченных областях, возможностью решения широкого класса задач с числовыми расчетами.
- **Микро-ЭВМ** (Персональные компьютеры и рабочие станции) – наибольшей популярностью в настоящее время пользуется ПК архитектурного направления (платформы) IBM с микропроцессорами фирмы Intel. Данное направление имеет большое количество клонов, т.е. аналогичных компьютеров, выпускаемых различными фирмами США, Западной Европы, России, Японии и др.

# **Общие принципы построения современных ЭВМ**



- Принцип программного управления
- Принцип конвейеризации
- Принцип децентрализации построения и управления



# Принцип программного управления

## Основной принцип построения всех современных ЭВМ - программное управление.

“Алгоритм - предписание, задающее последовательность действий по переработке данных в результат решения задачи”.

“Программа (для ЭВМ) - описание данных задачи и последовательность инструкций по их обработке, которую должен выполнить компьютер”.

Принцип программного управления может быть осуществлен различными способами. Стандарт - способ, описанный **Дж. фон Нейманом** в 1945 г. при построении еще первых образцов ЭВМ.

Все вычисления, предписанные алгоритмом решения задачи, должны быть представлены в виде программы, состоящей из последовательности управляющих слов-команд. Каждая команда содержит указания на конкретную выполняемую операцию, место нахождения (**адреса**) операндов и ряд служебных признаков. **Операнды** - переменные, значения которых участвуют в операциях преобразования данных.

Список всех переменных (входных данных, промежуточных значений и результатов вычислений) - еще один неотъемлемый элемент любой программы.



# Принцип программного управления

Для доступа к программам, командам и операндам используются их адреса. В качестве адресов выступают номера ячеек памяти ЭВМ, предназначенных для хранения объектов. Информация (командная и данные: числовая, текстовая, графическая и т.п.) кодируется двоичными цифрами **0** и **1**. Поэтому различные типы информации, размещенные в памяти ЭВМ, практически неразличимы, идентификация их возможна лишь при выполнении программы, согласно ее логике, по контексту.

Каждый тип информации имеет форматы - структурные единицы информации, закодированные двоичными цифрами 0 и 1. Обычно все форматы данных, используемые в ЭВМ, состоят из целого числа байтов.

Последовательность битов в формате, имеющая определенный смысл, называется **полем**. Например, в каждой команде программы различают поле кода операций, поле адресов операндов. Применительно к числовой информации выделяют знаковые разряды, поле значащих разрядов чисел, старшие и младшие разряды.

Последовательность, состоящая из определенного принятого для данной ЭВМ числа байтов, называется **словом**. В настоящее время длина слова ПК – 4 байта.

В качестве структурных элементов информации различают также полуслово, двойное слово и др.

# Принцип конвейеризации



Уже в первых ЭВМ для увеличения их производительности широко применялось совмещение операций. При этом последовательные фазы выполнения отдельных команд программы (формирование адресов операндов, выборка операндов, выполнение операции, отсылка результата) выполнялись отдельными функциональными блоками. В своей работе они образовывали своеобразный **конвейер**, а их параллельная работа позволяла обрабатывать различные фазы целого блока команд. Этот **принцип** получил дальнейшее развитие в ЭВМ следующих поколений.



# Принцип децентрализации построения и управления

Децентрализация построения и управления привела к появлению таких стандартов как **модульность построения** и **иерархия управления**.

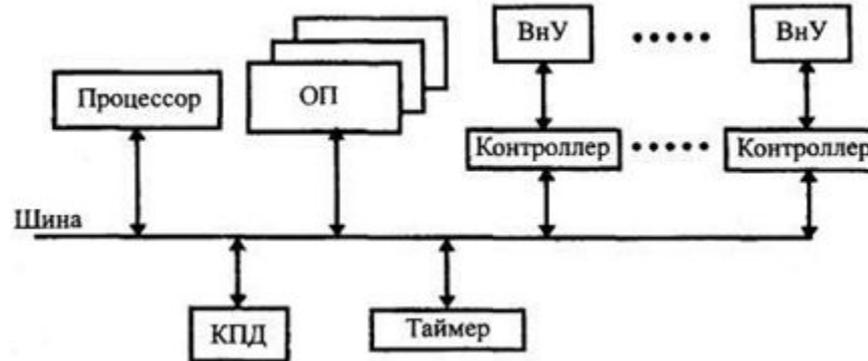
Модульность построения предполагает выделение в структуре ЭВМ достаточно автономных, функционально и конструктивно законченных устройств (процессор, модуль памяти, накопитель на жестком или гибком магнитном диске).

Сильносвязанные устройства АЛУ и УУ получили название процессор, т.е. устройство, предназначенное для обработки данных. В схеме ЭВМ появились также дополнительные устройства, которые имели названия: процессоры ввода-вывода, устройства управления обменом информацией, каналы ввода-вывода (КВВ). Последнее название получило наибольшее распространение применительно к большим ЭВМ. Здесь наметилась тенденция к децентрализации управления и параллельной работе отдельных устройств, что позволило резко повысить быстродействие ЭВМ в целом.



# Принцип децентрализации построения и управления

**В персональных ЭВМ произошло дальнейшее изменение структуры**



•Соединение всех устройств в единую машину обеспечивается с помощью общей шины, представляющей собой линии передачи **данных, адресов, сигналов управления и питания**. Единая система аппаратных соединений значительно упростила структуру, сделав ее еще более децентрализованной. Все передачи данных по шине осуществляются под управлением сервисных программ.

•Ядро ПК образуют процессор и основная память (ОП), состоящая из оперативной памяти и постоянного запоминающего устройства (ПЗУ). ПЗУ предназначается для записи и постоянного хранения наиболее часто используемых программ управления. Подключение всех внешних устройств (ВнУ), дисплея, клавиатуры, внешних ЗУ и других обеспечивается через соответствующие адаптеры - согласователи скоростей работы сопрягаемых устройств или контроллеры - специальные устройства управления периферийной аппаратурой. Контроллеры в ПК играют роль каналов ввода-вывода. В качестве особых устройств следует выделить таймер - устройство измерения времени и контроллер прямого доступа к памяти (КПД) - устройство, обеспечивающее доступ к ОП, минуя процессор.



# Принцип децентрализации построения и управления

- Модульная конструкция ЭВМ делает ее открытой системой
- В современных ЭВМ **принцип децентрализации и параллельной работы** распространен как на периферийные устройства, так и на сами ЭВМ (процессоры).
- Появились отдельные специализированные процессоры, например сопроцессоры, выполняющие обработку чисел с плавающей точкой, матричные процессоры и др.
- Модульность структуры ЭВМ требует **стандартизации и унификации** оборудования, номенклатуры технических и программных средств, средств сопряжения - интерфейсов, конструктивных решений, унификации типовых элементов замены, элементной базы и нормативно-технической документации. Все это способствует улучшению технических и эксплуатационных характеристик ЭВМ, росту технологичности их производства.
- **Децентрализация управления** предполагает *иерархическую организацию структуры ЭВМ*. Централизованное управление осуществляет устройство управления главного, или центрального, процессора. Подключаемые к центральному процессору модули (контроллеры и КВВ) могут, в свою очередь, использовать специальные *шины или магистрали для обмена управляющими сигналами, адресами и данными*. Инициализация работы модулей обеспечивается по командам центральных устройств, после чего они продолжают работу по собственным программам управления. Результаты выполнения требуемых операций представляются ими “вверх по иерархии” для правильной координации всех работ.



# Иерархический принцип построения и управления для подсистемы памяти

- Пользователю желательно иметь оперативную память **большой емкости и высокого быстродействия**. Однако в данной структуре маршрутизации информации **на верхних уровнях памяти организуется обращение к более низкому современному уровню с одновременным копированием этой информации на более быстрый уровень**. Это перемещение блоков информации на верхний уровень позволяет экономить время при повторных обращениях к ней.
- Следующий уровень образует кэш-память. Она представляет собой буферное запоминающее устройство, предназначенное для хранения активных страниц памяти объемом в Кбайты. Время обращения к данным составляет **10-20 нс**, при этом могут использоваться специальные высокоскоростные алгоритмы выборки данных. Кэш-память предназначена для ускорения выполнения команд программы и **децентрализованном управлении ими**.
  - Программы пользователей и данные к ним размещаются в оперативном запоминающем устройстве (емкость - миллионы машинных слов, время выборки - до **100 нс**). Часть машинных программ, обеспечивающих алгоритмическое управление вычислениями и используемых наиболее часто, может размещаться в постоянном запоминающем устройстве (ПЗУ).
  - На более низких уровнях иерархии находятся внешние запоминающие устройства на магнитных носителях: на жестких и гибких магнитных дисках, магнитных лентах, магнитооптических дисках и др. Их отличает более низкое быстродействие и очень большая емкость **превышающей емкость ОЗУ**.

**Функциональная и структурная**

**организация ЭВМ**



# Архитектурные концепции вычислительных систем



**АРХИТЕКТУРА СИСТЕМЫ** включает три важнейших вида структур: **ФИЗИЧЕСКУЮ, ЛОГИЧЕСКУЮ** и **ПРОГРАММНУЮ**.

Каждая из этих структур определяется набором элементов и характером их взаимосвязи. Связь структур друг с другом образует **АРХИТЕКТУРУ** рассматриваемой **СИСТЕМЫ** (вычислительной сети, терминального комплекса, вычислительной машины, полупроводникового кристалла) .

Элементы **ФИЗИЧЕСКОЙ СТРУКТУРЫ** - технические объекты (полупроводниковые кристаллы или части вычислительных машин или комплексы вычислительных машин).

Элементы **ЛОГИЧЕСКОЙ СТРУКТУРЫ** - функции, определяющие основные операции.

Элементы **ПРОГРАММНОЙ СТРУКТУРЫ** - взаимосвязанные программы.

**АРХИТЕКТУРА СИСТЕМЫ** - описание взаимосвязи большого числа различного типа элементов.

**Наиболее распространенные в настоящее время архитектуры микропроцессоров:**

**фон-Неймановская** вычислительная архитектура (или архитектура общей шины);

**Гарвардская** вычислительная архитектура, которая внедряется в цифровые сигнальные процессоры (ЦСП/DSP), начиная с конца 1970-х;

**СуперГарвардская** вычислительная архитектура (в ЦСП с 1990-х);



# Фон-Неймановская архитектура

Вычислительное устройство машины с архитектурой фон-Неймана выбирает сначала команду из памяти, затем данные, обрабатывает их, и полученный результат вновь передает в память.

Машины фон-Неймана хранят программу и данные **в одной и той же области памяти**. В машинах этого типа команды содержат указание на то, что выполнить, и адреса данных, подлежащих обработке. Внутри этой машины имеются три основных функциональных блока.

Арифметико-логическое устройство (**АЛУ**) выполняет самые важные операции: умножение, сложение, вычитание и многие другие.

Другой блок – **память**, в которой хранятся команды и обрабатываемые данные.

Третий блок - **устройство ввода/вывода**, которое управляет потоком внешних данных и команд. Эти блоки, реализованные в виде устройств, соединены между собой с помощью трех шин: управления (**ШУ**), данных (**ШД**) и адреса (**ША**).



## **Гарвардская архитектура**

Гарвардская машина, как и фон-Неймановская, имеет арифметико-логическое устройство и устройство ввода/вывода. Единственное отличие гарвардской архитектуры состоит в том, **что память программ и память данных разделены, и они используют физически разделенные линии передачи.** Это позволяет подобной машине пересылать команды и данные одновременно.

Есть еще варианты этих архитектур:

**модифицированная Гарвардская архитектура и СуперГарвардская архитектура.**

**подавляющее большинство современных компьютеров основано на архитектуре фон Неймана.**

# Связь между функциональной и логической структурой ЭВМ



**Существуют два взгляда на построение и функционирование ЭВМ:**

- взгляд пользователя (не интересуется технической реализацией ЭВМ и озабочен только получением некоторого набора функций и услуг, обеспечивающих эффективное решение его задач);**
- взгляд разработчика ЭВМ (озабочен технической реализацией необходимых пользователю функций).**

**С учетом этого обстоятельства и вводятся понятия "функциональная и логическая структура" компьютера.**

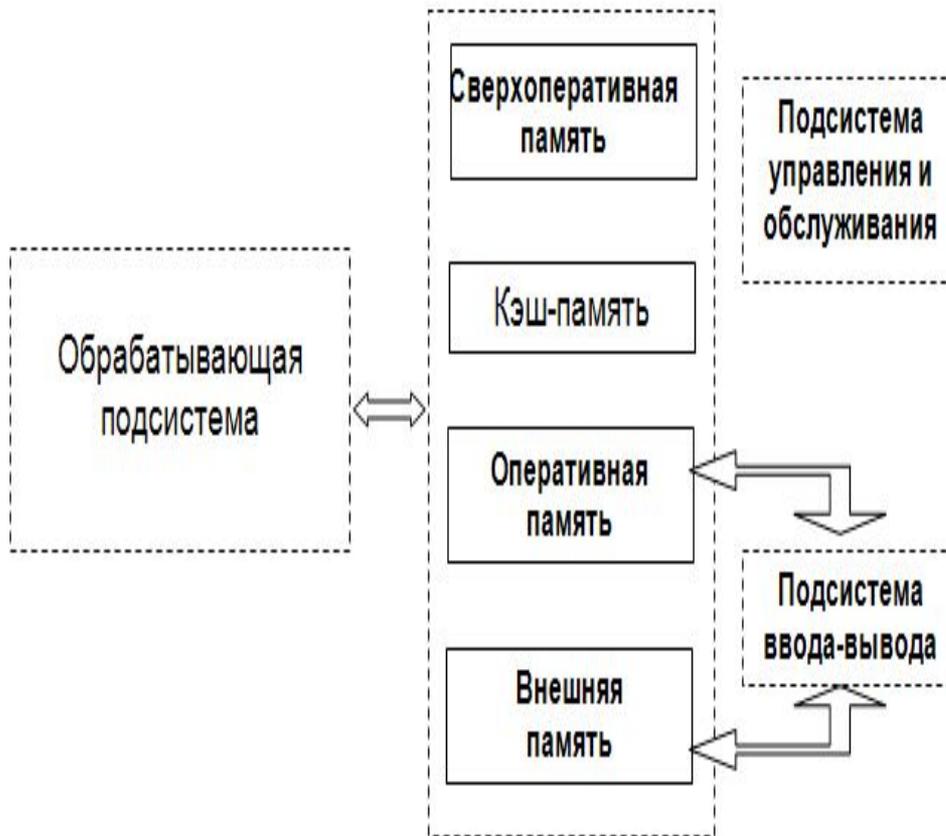
# Связь между функциональной и логической структурой ЭВМ



**Функциональная структура ЭВМ** – это реализуемые ей набор функций и услуг. При проектировании ЭВМ создается абстрактная модель, описывающая функциональные возможности машины и предоставляемые ею услуги. Предусматриваемые абстрактной моделью функции ЭВМ реализуются на основе реальных, физических средств (устройств, блоков, узлов, элементов), взаимодействующих между собой.

**Логическая структура ЭВМ** - состав, порядок и принципы взаимодействия основных функциональных частей машины (без учета их реализации).

# Обобщенная структура ЭВМ

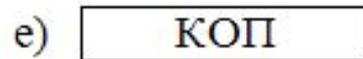
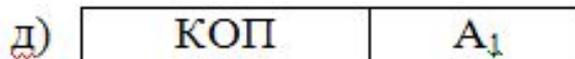
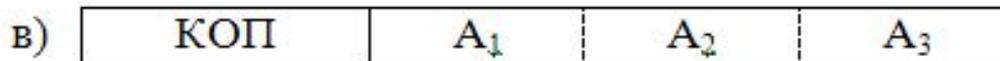
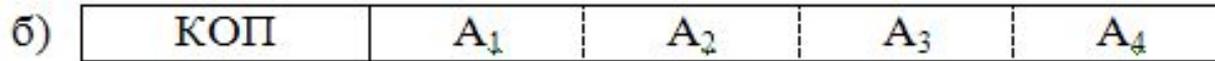


Основной тенденцией в развитии структуры ЭВМ является разделение функций системы и максимальная специализация подсистем для выполнения этих функций.

- Обобщенная структура ЭВМ:
- Обработка подсистема;
  - подсистема памяти;
  - подсистема ввода-вывода;
  - подсистема управления и обслуживания



# Структура и форматы команд ЭВМ



$A_1$ ,  $A_2$  - адреса операндов,  $A_3$  - адрес результата,  $A_4$  - адрес следующей команды (принудительная адресация команд).

Такая структура приводит к большей длине команды. Можно установить, что после выполнения данной команды, расположенной по адресу  $K$  (и занимающей  $L$  ячеек), выполняется команда из  $(K+L)$ -й ячейки.

В таком случае отпадает необходимость указывать в команде в явном виде адрес следующей команды.

# Архитектуры процессоров, основанные на различных системах команд



В настоящее время наиболее распространенными являются процессоры двух архитектур, в основе которых лежат различные системы команд:

- **CISC**- процессоры;
- **RISC**- процессоры.

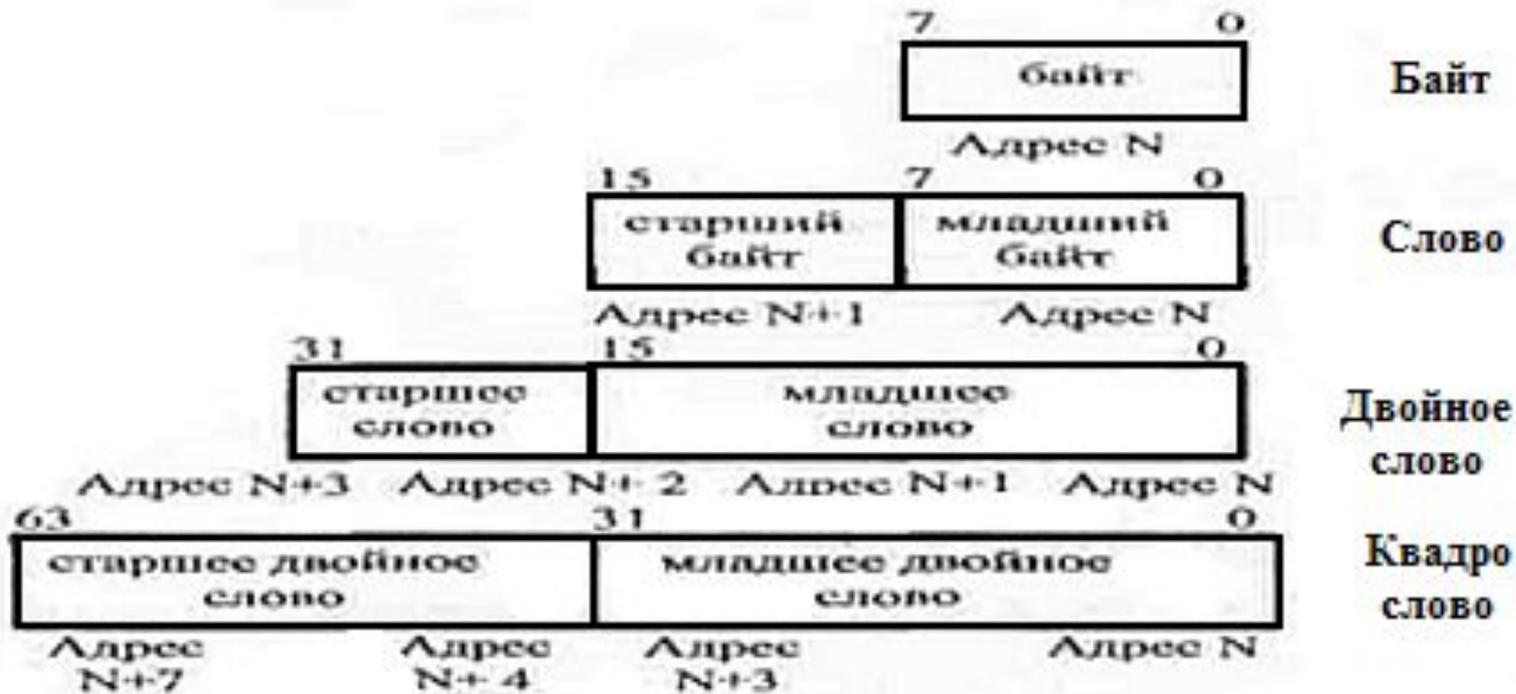
**CISC** - процессоры с расширенной системой команд (**Complex Instruction Set Computer**) имеют:

- широкий набор команд, позволяющих в зависимости от класса машины обрабатывать числа с плавающей точкой, числа с фиксированной точкой или текстовую информацию;
- полный набор способов адресации.

**RISC** - процессоры с ограниченной системой команд (**Reduced Instruction Set Computer**) имеют следующие особенности:

- выделены короткие команды обработки информации, имеющие регистровую адресацию и трехадресный формат; все команды этой группы имеют одинаковую длину и аппаратно выполняются;
- длинные команды реализуются либо на программном уровне на базе более простых команд, либо в специальном процессоре;
- упрощены форматы команд и использованы простейшие способы адресации.

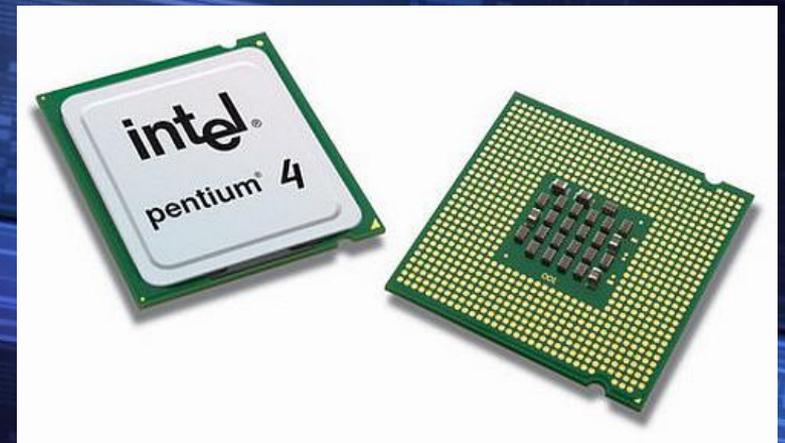
# Типы данных





# Функциональная и логическая организация центрального процессора

ПРОЦЕССОР - УСТРОЙСТВО,  
НЕПОСРЕДСТВЕННО  
ОСУЩЕСТВЛЯЮЩЕЕ  
ОБРАБОТКУ ДАННЫХ И  
ПРОГРАММНОЕ УПРАВЛЕНИЕ  
ЭТИМ ПРОЦЕССОМ.  
РАЗЛИЧАЮТ ПРОЦЕССОРЫ  
ЦЕНТРАЛЬНЫЕ,  
СПЕЦИАЛИЗИРОВАННЫЕ, ВВОДА-  
ВЫВОДА, ПЕРЕДАЧИ ДАННЫХ И  
КОММУНИКАЦИОННЫЕ





# Назначение центрального процессора

**Центральный процессор  
дешифрует и выполняет команды  
программы,  
организует обращение к памяти,  
инициализирует работу периферийных  
устройств,  
воспринимает и обрабатывает  
запросы прерываний, поступающих из  
других устройств и внешней среды**



# Основные характеристики процессоров

1. Система команд процессора – набор поддерживаемых данным процессором команд, который обуславливает программную совместимость разных процессоров. В настоящее время существует две основные группы систем команд:
  - а) **CISC**-архитектура (Complex Instruction Set Computer – расширенный набор команд);
  - б) **RISC**-архитектура (Reduced Instruction Set Computer – ограниченный набор команд).
2. Способы адресации.
3. Разрядность:
  - регистров;
  - шин:
4. Наличие конвейера – конвейеризация выполнения команд – это возможность выполнения сразу нескольких команд (на уровне микрокоманд). Конвейеров может быть несколько .
5. Наличие кэш-памяти
6. Тактовая частота.



# Выполнение команд в процессоре. Микропрограммы

Выполнение отдельной команды программы в процессоре разделяется на более малые этапы – микрокоманды или микрооперации.

**Машинный такт (тактовый интервал)** - интервал времени, в течение которого выполняется одна микрооперация.

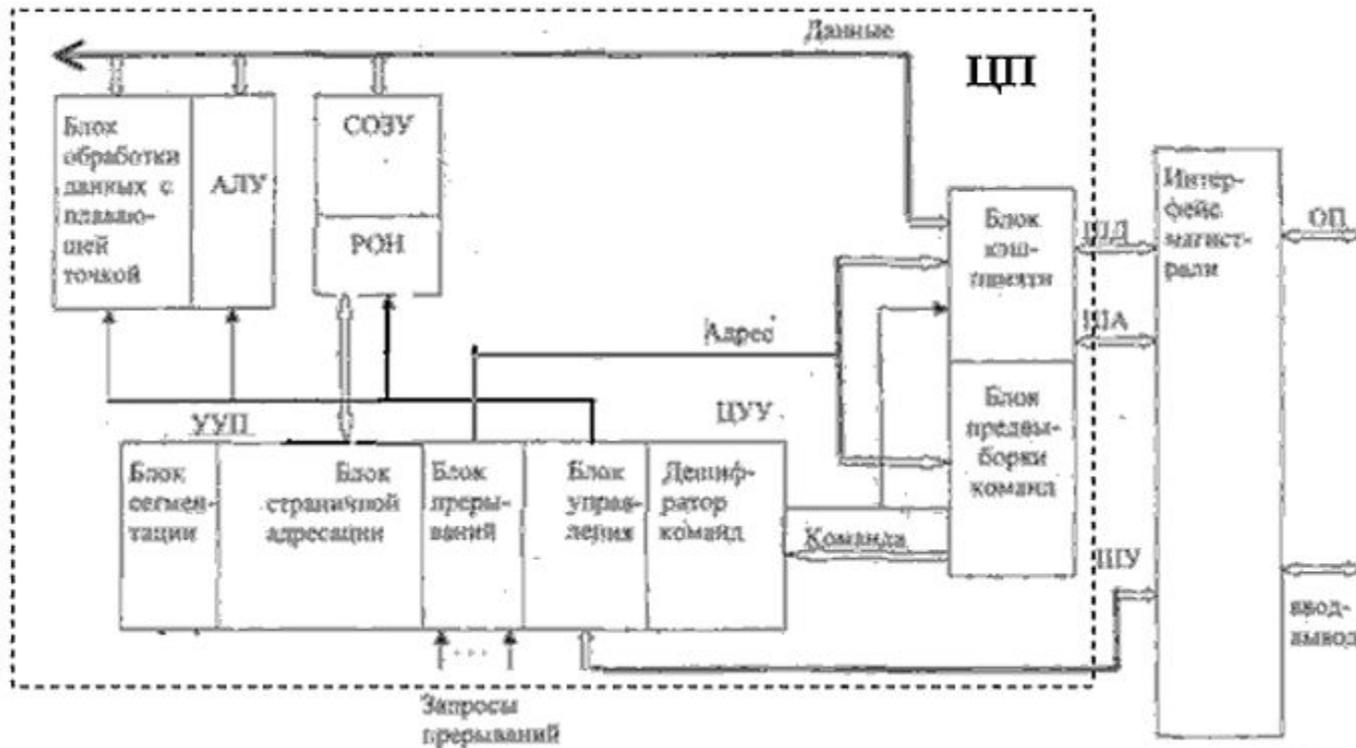
Длительность машинных тактов задается специальной схемой синхронизации, работающей от генератора импульсов.

Последовательность микроопераций, реализующая данную команду, образует микропрограмму данной команды или данной операции.



# Структура центрального процессора и принципы его работы

Существует обязательный (стандартный) минимальный набор устройств для каждого типа центрального процессора. Он включает в себя: регистры общего назначения, устройства выполнения стандартного набора операций и устройства управления вычислительным процессом.





## Назначение и организация УУ

**Устройство управления (УУ)** - комплекс средств автоматического управления процессом передачи и обработки информации. УУ вырабатывает управляющие сигналы (УС), необходимые для выполнения всех операций, предусмотренных системой команд, а также координирует работу всех узлов и блоков ЭВМ.

**УУ** можно считать преобразователем **первичной командной информации, представленной программой решения задачи, во вторичную командную информацию, представляемую управляющими сигналами.**

**Микропрограммное управление** – это вид иерархического управления работой ЭВМ, при котором каждая команда машины является обращением к последовательности микрокоманд (**микропрограмме**). Для реализации этого принципа в УУ используется блок микропрограммного управления (БМУ).



# Устройство управления микропрограммного типа

Каждая машинная команда реализуется путем выполнения определенной микропрограммы, интерпретирующей алгоритм выполнения данной операции. Совокупность микропрограмм, необходимая для реализации системы команд ЭВМ, хранится в специальной **памяти микропрограмм**.

Каждая **микропрограмма** состоит из определенной последовательности **микрокоманд**, которые после выборки из памяти преобразуются в **набор управляющих сигналов**. Эти сигналы воздействуют на все блоки **ЦП (АЛУ и др.)**, обеспечивающие выполнение очередной команды и переход к следующей.

Таким образом, использование в составе центрального устройства управления **БМУ** приводит к **двухуровневому принципу управления** процессом обработки данных.

- **Первый уровень** - это система команд ЭВМ (программное управление),
- **второй** - микропрограммное управление.



# Процедура выполнения команд. Конвейер команд

Стандартные фазы работы **ЦП** включают в себя:

- выборку команды,
- выборку операндов,
- выполнение команды и запись результатов,
- обработку прерывания,
- изменение состояния процессора и системы в целом.

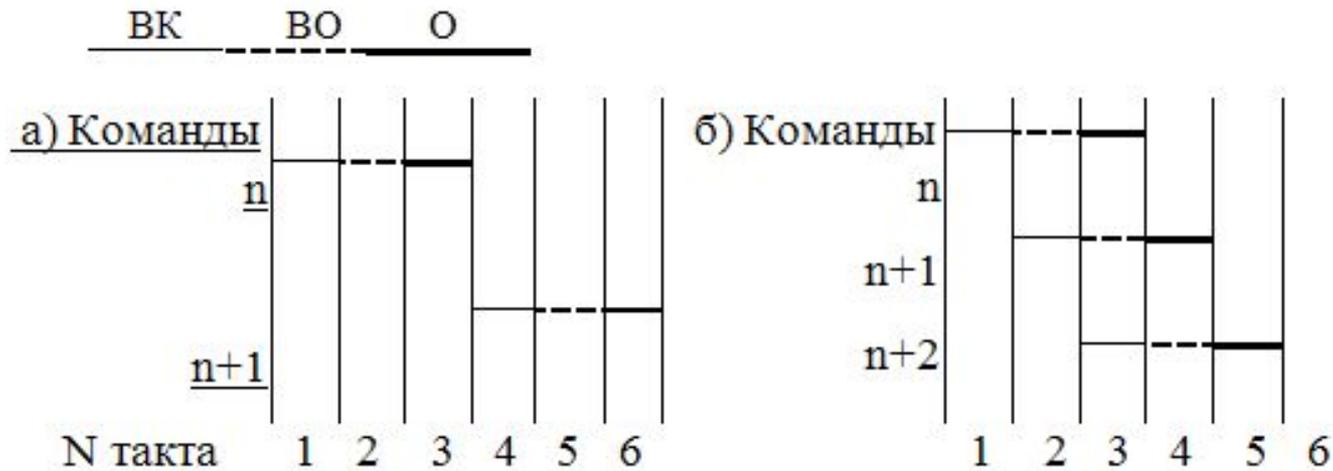
**Выборка команд (ВК)** - передача содержимого счетчика адреса команд в регистр адреса памяти, считывание команды из основной памяти в регистр команды, модификация содержимого счетчика команд для выборки следующей команды.

**Выборка операнда (ВО)** - вычисление адреса и обращение в основную память или к регистру локальной памяти (СОЗУ). Операнд считывается и принимается в регистр АЛУ.

**Выполнение команды (О)** - инициализация (кодом операции) цикла работы устройства управления, которое, в свою очередь, управляет работой АЛУ, регистров и схем сопряжения. Результат выполнения передается в локальную или основную память, и процессор переходит к выборке и выполнению следующей команды.



# Процедура выполнения команд. Конвейер команд



**а) последовательное выполнение команд,**

**б) совмещенное выполнение команд (конвейеризация)**



# Организация системы прерывания программ

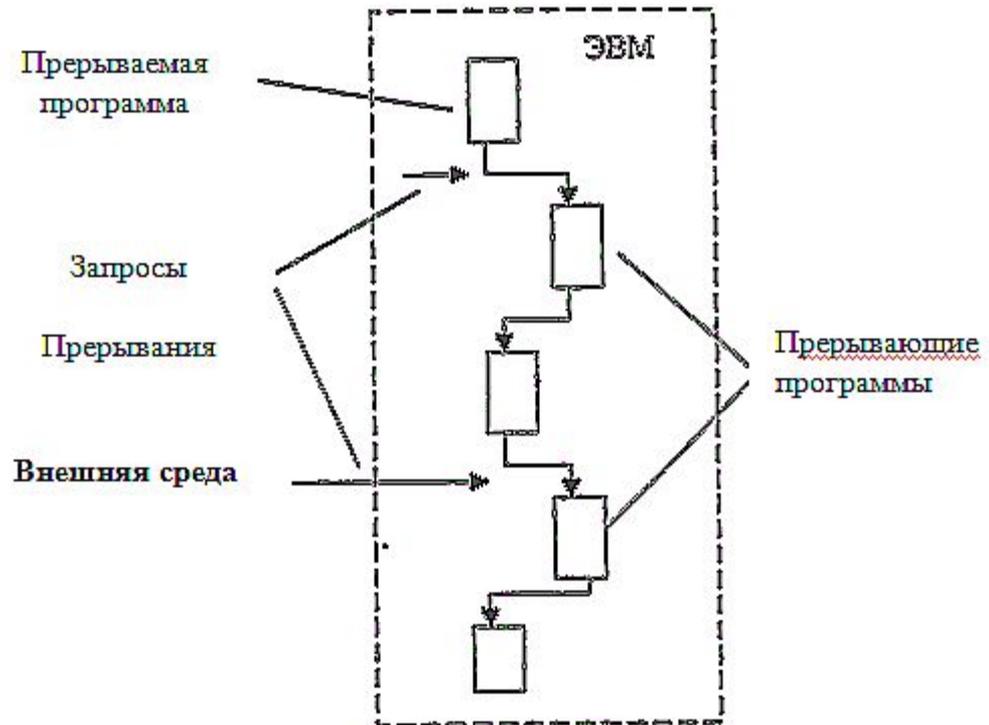
Во время выполнения программы внутри машины или во внешней среде могут возникать события, требующие немедленной реакции на них со стороны машины.

Машина прерывает обработку текущей программы и переходит к выполнению некоторой другой программы, специально предназначенной для данного события. По завершению этой программы ЭВМ возвращается к выполнению прерванной программы.

Этот процесс называется **прерыванием программ**

Каждое событие, требующее прерывания, сопровождается сигналом, который называют **запросом прерывания**.

Программу, затребованную запросом прерывания, называют **прерывающей программой**.





# Иерархическая структура памяти ЭВМ

**Сверхоперативная  
память**

**Кэш-память**

**Оперативная память**

**Внешняя память**

Минимальная адресуемая единица хранения информации в памяти – 1 байт = 8 бит.

Память ЭВМ организуется в виде иерархической структуры запоминающих устройств, обладающих различным быстродействием и емкостью. Чем выше уровень, тем выше быстродействие соответствующей памяти, но меньше ее емкость.

Система управления памятью обеспечивает обмен информационными блоками между уровнями.

Обычно **первое обращение** к блоку информации вызывает его **перемещение с низкого медленного уровня на более высокий**. Это позволяет при последующих обращениях к данному блоку осуществлять его выборку с более быстродействующего уровня памяти.



# Иерархическая структура памяти ЭВМ

Упрощенно память делят на **внутреннюю** и **внешнюю**. Применительно к РС их можно определить так:

- **внутренняя память** — электронная (полупроводниковая) память, устанавливаемая на системной плате или на платах расширения;
- **внешняя память** — память, реализованная в виде устройств с различными принципами хранения информации и обычно с подвижными носителями. Сюда входят устройства магнитной (дисковой и ленточной) памяти, оптической и магнитооптической памяти.

**Для процессора непосредственно доступна внутренняя память**, доступ к которой осуществляется по адресу, заданному программой. Для внутренней памяти характерен одномерный (линейный) адрес. Внутренняя память подразделяется на *оперативную*, информация в которой может изменяться процессором в любой момент времени, и *постоянную*, информацию которой процессор может только считывать. Обращение к ячейкам оперативной памяти может происходить в любом порядке (как по чтению, так и по записи), и оперативную память называют памятью с произвольным доступом - **Random Access Memory (RAM)** - в отличие от постоянной памяти (**Read Only Memory, ROM**).



# Иерархическая структура памяти ЭВМ

Внешняя память адресуется более сложным образом:

**каждая ее ячейка имеет свой адрес внутри некоторого блока, который, в свою очередь, имеет многомерный адрес.**

Во время физических операций обмена данными блок может быть считан или записан **только целиком**. В случае одиночного дискового накопителя адрес блока будет трехмерным: **номер поверхности (головки), номер цилиндра и номер сектора**. В современных накопителях этот трехмерный адрес часто заменяют линейным номером - логическим адресом блока, а его преобразованием в физический адрес занимается внутренний контроллер накопителя. Поскольку дисковых накопителей в компьютере может быть множество, в адресации дисковой памяти участвует и **номер накопителя**, а также **номер канала интерфейса**. С такой сложной системой адресации процессор справляется только с помощью программного драйвера, в задачу которого входит копирование некоторого блока данных из оперативной памяти в дисковую и обратно. Дисковая память является внешней памятью с прямым доступом, что подразумевает возможность обращения к блокам (**но не ячейкам**) в произвольном порядке.



# Иерархическая структура памяти ЭВМ

Важные параметры подсистемы памяти:

- **объем хранимой информации.** Максимальный (в принципе неограниченный) объем хранят ленточные и дисковые устройства со сменными носителями, за ними идут дисковые накопители, и затем оперативная память;
- **время доступа** - усредненная задержка начала обмена полезной информацией относительно появления запроса на данные. Минимальное время доступа имеет оперативная память, за ней идет дисковая и после нее - ленточная;
- **скорость обмена при передаче потока данных** (после задержки на время доступа). Максимальную скорость обмена имеет оперативная память, за ней идет дисковая и после нее - ленточная;
- **удельная стоимость хранения единицы данных** - цена накопителя (с носителями), отнесенная к единице хранения (байту или мегабайту). Минимальную стоимость хранения имеют ленточные устройства со сменными носителями, их догоняют дисковые накопители, а самая дорогая - оперативная память.



# Иерархическая структура памяти ЭВМ

- Внутренняя и внешняя память используются различными способами.
- Внутренняя (оперативная и постоянная) память является хранилищем программного кода, который непосредственно может быть исполнен процессором. В ней же хранятся и данные, также непосредственно доступные процессору (а, следовательно, и исполняемой программе).
  - Внешняя память обычно используется для хранения файлов, содержимое которых может быть произвольным. Процессор (программа) имеет доступ к содержимому файлов только опосредованно через отображение их (полное или частичное) в некоторой области оперативной памяти.



# Виртуальная память

Суть ее заключается в том, что программам предоставляется виртуальное пространство оперативной памяти, по размерам превышающее объем физически установленной оперативной памяти. Это виртуальное пространство разбито на **страницы фиксированного размера**, а в физической оперативной памяти в каждый момент времени присутствует только часть из них. Остальные страницы хранятся на диске, откуда операционная система может их **«подкачать»** в физическую на место предварительно выгруженных на диск страниц. Для прикладной программы этот процесс «не виден» (если только она не критична ко времени обращения к памяти). Для пользователя этот процесс заметен по работе диска даже в тот момент, когда не требуется обращение к файлам. Последствиями большого увеличения объема доступной памяти является снижение средней производительности памяти и некоторый расход дисковой памяти на так называемый **файл подкачки**. Размер виртуальной памяти не может превышать размера диска (файл подкачки на нескольких дисках обычно не размещают).



# Внутренняя память процессора

Стандарт современной архитектуры ЭВМ - организация **регистров общего назначения** в виде сверхоперативного ЗУ с прямой адресацией (короткие адреса регистров размещаются в команде). В машинах с одноадресными командами один из общих регистров выделяется в качестве **аккумулятора** - регистра, в котором находится один из операндов и в который помещается результат операции. Регистр-аккумулятор в явном виде в команде не адресуется, используется подразумеваемая адресация.

**Стек** - группа последовательно пронумерованных регистров (аппаратный стек) или ячеек памяти, снабженных указателем стека (обычно регистром), в котором автоматически при записи и считывании устанавливается номер (адрес) первой свободной ячейки стека (вершина стека). При операции записи заносимое в стек слово помещается в свободную ячейку стека, а при считывании из стека извлекается последнее поступившее в него слово. Таким образом, в стеке реализуется правило «последний пришел - первый ушел» - магазинная адресация.

**Стек** и стековая адресация широко используется при организации переходов к подпрограммам и возврата из них, а также в системах прерывания.



# Организация кэш-памяти

При обращении к блоку данных, находящемуся на оперативном уровне, его копия пересылается в сверхоперативную буферную память (**СБП**). Последующие обращения производятся к копии блока данных, находящейся в СБП. Поскольку время выборки из сверхоперативной буферной памяти (**5 нс**) много меньше времени выборки из оперативной памяти, это приводит к уменьшению времени обращения.

**Принцип локальности обращений** - при выполнении какой-либо программы (практически для всех классов задач) большая часть обращений в пределах некоторого интервала времени приходится на ограниченную область адресного пространства ОП, причем обращения к командам и элементам данных этой области производятся многократно.

Буферная память **не является программно доступной**.

За единицу информации при обмене между основной памятью и кэш-памятью принята **строка**, (набор слов, выбираемый из оперативной памяти при одном к ней обращении). Хранимая в оперативной памяти информация представляется, таким образом, совокупностью строк с последовательными адресами. Кэш не может хранить копию всей основной памяти, поскольку его объем во много раз меньше объема основной памяти. Он хранит лишь **ограниченное количество блоков данных** и **каталог** - список их текущего соответствия областям основной памяти. В любой момент времени строки в кэш-памяти представляют собой копии строк из некоторого их набора в ОП.



# Принципы организации оперативной памяти

Оперативная память - следующий уровень иерархии памяти.

Для оценки производительности (быстродействия) основной памяти используются два основных параметра: **задержка** и **полоса пропускания**.

Задержка памяти традиционно оценивается двумя параметрами: **временем доступа** и **длительностью цикла памяти**.

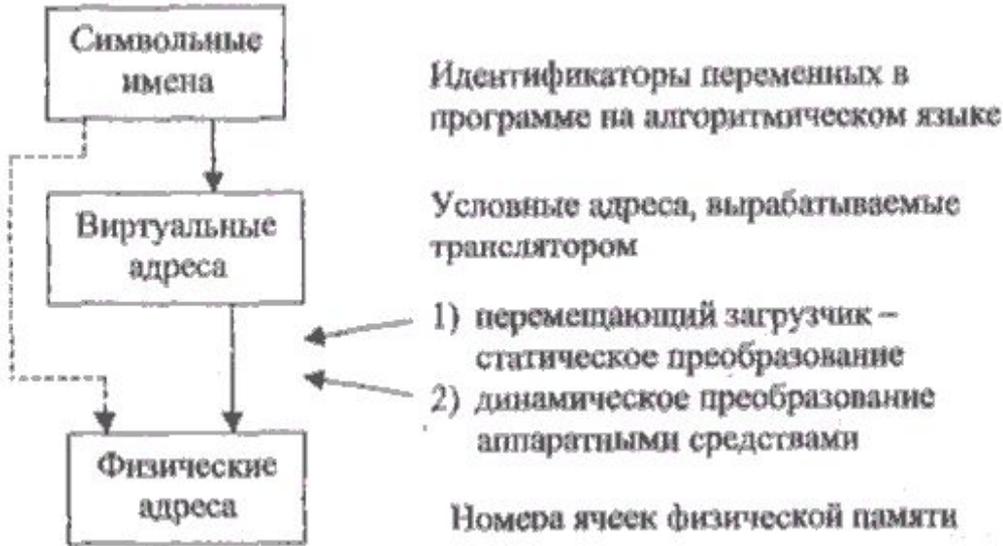
**Цикл памяти** (цикл обращения к памяти) - минимальное время между двумя последовательными обращениями к оперативной памяти.

Внутренняя память современных компьютеров реализуется на микросхемах **статических** и **динамических** запоминающих устройств с произвольной выборкой. Микросхемы статических ЗУ (СЗУ) имеют меньшее время доступа и не требуют циклов регенерации (восстановления) информации. Микросхемы динамических ЗУ (ДЗУ) характеризуются большей емкостью и меньшей стоимостью, но требуют схем регенерации и имеют значительно большее время доступа. У статических ЗУ время доступа совпадает с длительностью цикла.

Для микросхем, использующих примерно одну и ту же технологию, емкость **ДЗУ** по грубым оценкам в 4 - 8 раз превышает емкость **СЗУ**, но последние имеют в 8 - 16 раз меньшую длительность цикла и большую стоимость. По этим причинам в основной памяти практически любого компьютера используются полупроводниковые микросхемы ДЗУ (для построения кэш-памяти при этом применяются СЗУ).



# Методы управления памятью



## Типы адресов

Для идентификации переменных и команд используются **символьные имена** (метки), **виртуальные адреса** и **физические адреса**.

*Символьные имена* присваивает пользователь при написании программы на алгоритмическом языке или ассемблере.

*Виртуальные адреса* вырабатывает транслятор, переводящий программу на машинный язык.

Так как во время трансляции в общем случае неизвестно, в какое место ОП будет загружена программа, то транслятор присваивает переменным и командам виртуальные (условные) адреса, начиная с нулевого адреса. Совокупность виртуальных адресов процесса (программы) называется **виртуальным адресным пространством**. Каждый процесс имеет **собственное виртуальное адресное пространство**. Максимальный размер виртуального адресного пространства ограничивается разрядностью адреса, присущей данной архитектуре компьютера, и обычно не совпадает с объемом физической памяти, имеющимся в компьютере.

*Физические адреса* соответствуют номерам ячеек ОП, где в действительности расположены или будут расположены переменные и команды. Переход от виртуальных адресов к физическим может осуществляться двумя способами.



# Переход от виртуальных адресов к физическим

- **Первый способ** - замену виртуальных адресов на физические делает специальная системная программа - **перемещающий загрузчик**. Перемещающий загрузчик на основании имеющихся у него исходных данных о начальном адресе физической памяти, в которую предстоит загружать программу, и информации, предоставленной транслятором об адресно-зависимых константах программы, выполняет загрузку программы, совмещая её с заменой виртуальных адресов физическими.
- **Второй способ** - программа загружается в память в неизменном виде в виртуальных адресах, при этом ОС фиксирует смещение действительного расположения программного кода относительно виртуального адресного пространства. Во время выполнения программы при каждом обращении к ОП выполняется преобразование виртуального адреса в физический.



# Методы управления памятью

Два класса методов:

- методы распределения ОП без использования внешней памяти (дискового пространства);
- методы распределения памяти с использованием дискового пространства;





# Методы управления без использования внешней памяти

- **Распределение памяти фиксированными разделами**

Самый простой способ управления - разделение ее на несколько разделов (сегментов) фиксированной величины (**статическое распределение**). Очередная задача, поступающая на выполнение, помещается либо в общую очередь, либо в очередь к некоторому разделу. Подсистема управления памятью сравнивает размер программы и свободных разделов памяти; выбирает подходящий раздел; осуществляет загрузку программы и настройку адресов.

**Достоинство** - простота реализации, **недостаток** – жесткость (заданное число сегментов, т. е. выполняемых задач).

- **Распределение памяти разделами переменной величины**

Память машины не делится заранее на разделы. Сначала вся память свободна. Каждой вновь поступающей задаче выделяется необходимая ей память. Если достаточный объем памяти отсутствует, то задача не принимается на выполнение и стоит в очереди. После завершения задачи память освобождается, и на это место может быть загружена другая задача. Таким образом, в произвольный момент времени оперативная память представляет собой случайную последовательность занятых и свободных участков (разделов) произвольного размера.

**Достоинство** - метод обладает гораздо большей гибкостью, **недостаток** - **фрагментация** памяти. Фрагментация - это наличие большого числа несмежных участков свободной памяти очень маленького размера (фрагментов). Ни одна из программ не может поместиться ни в одном из участков, хотя суммарный объем фрагментов может превысить требуемый объем памяти.

- **Перемещаемые разделы**

Метод борьбы с фрагментацией - перемещение всех занятых участков в сторону старших либо в сторону младших адресов так, чтобы свободная память образовывала единую свободную область. Эта процедура называется «сжатием». Сжатие может выполняться либо при каждом завершении задачи, либо только тогда, когда для вновь поступившей задачи нет свободного раздела достаточного размера.



# Организация виртуальной памяти

**Виртуальным** называется такой ресурс, который для пользователя (пользовательской программы) представляется обладающим свойствами, которыми он в действительности не обладает.

*Например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную ОП. Пользователь пишет программы так, как будто в его распоряжении имеется однородная оперативная память большого объема, но в действительности все данные, используемые программой, хранятся на нескольких разнородных запоминающих устройствах, обычно в ОП и на дисках, и при необходимости частями перемещаются между ними.*

**Виртуальная память** - это совокупность программно-аппаратных средств, которая решает следующие задачи:

- размещает данные в запоминающих устройствах разного типа, например, часть программы в ОП, а часть на диске;
- перемещает по мере необходимости данные между запоминающими устройствами разного типа, например, подгружает нужную часть программы с диска в ОП;
- преобразует виртуальные адреса в физические.

Все эти действия выполняются автоматически средствами операционной системы.

Наиболее распространенными реализациями виртуальной памяти являются **страничное, сегментное и странично-сегментное** распределение памяти, а также **свопинг**.



# Организация виртуальной памяти

## Страничное распределение

- Виртуальное адресное пространство каждого процесса делится на части одинакового, фиксированного для данной системы размера, называемые **виртуальными страницами**. В общем случае размер виртуального адресного пространства не является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.
- Вся оперативная память машины также делится на части такого же размера, называемые **физическими страницами** (или блоками).
- При загрузке процесса часть его виртуальных страниц помещается в оперативную память, а остальные - на диск.

## Сегментное распределение

- При страничной организации виртуальное адресное пространство процесса делится механически на равные части. При этом на одной виртуальной странице может находиться разная по смыслу информация, например, код программы и данные для нее. Гораздо полезнее делить информацию на порции, соответствующие ее смыслу (сегменты кода, данных). Например, можно запретить обращаться с операциями записи и чтения в кодový сегмент программы, а для сегмента данных разрешить только чтение. Кроме того, разбиение программы на «осмысленные» части делает принципиально возможным разделение одного сегмента несколькими процессами. Например, если два процесса используют одну и ту же математическую подпрограмму, то в оперативную память может быть загружена только одна копия этой подпрограммы.
- Недостатком данного метода распределения памяти является фрагментация на уровне сегментов и более медленное по сравнению со страничной организацией преобразование адреса.

## Странично-сегментное распределение

- Как видно из названия, данный метод представляет собой комбинацию страничного и сегментного распределения памяти и, вследствие этого, сочетает в себе достоинства обоих подходов. Память делится на сегменты, а внутри сегмент разбивается на страницы.



# СВОПИНГ

Для загрузки процессора на 90 % достаточно всего **трех счетных задач**. Однако для того, чтобы обеспечить такую же загрузку **интерактивными задачами**, выполняющими интенсивный ввод-вывод, потребуются **десятки таких задач**.

Необходимым условием для выполнения задачи является загрузка её в оперативную память, объем которой ограничен. В этих условиях был предложен метод организации вычислительного процесса, называемый **свопингом**. При свопинге некоторые процессы (обычно находящиеся в состоянии ожидания) временно выгружаются на диск. Планировщик операционной системы не исключает их из своего рассмотрения, и при наступлении условий активизации некоторого процесса, находящегося в области свопинга на диске, этот процесс перемещается в оперативную память. Если свободного места в оперативной памяти не хватает, то выгружается другой процесс.

При свопинге, в отличие от рассмотренных ранее методов реализации виртуальной памяти, процесс перемещается между памятью и диском целиком, т.е. в течение некоторую времени процесс может полностью отсутствовать в оперативной памяти.



# Методы защиты памяти

**Подсистема защиты памяти** представляет собой комплекс аппаратно-программных средств для предотвращения:

- взаимного искажения одновременно находящихся в ОП программ
- несанкционированного доступа к любой хранящейся в ОП информации.

В общем случае защита осуществляется:

- при записи для предотвращения искажения информации, не относящейся к выполняемой в данный момент программе,
- при считывании для исключения возможности использования информации, не принадлежащей данному пользователю, т.е. для предотвращения несанкционированного доступа к информации.

Независимо от принятых принципов построения подсистемы защиты памяти в основе её функционирования **заложена проверка всех адресов**, поступающих для обращения к ОП. В результате такой проверки формируется сигнал управления:

- разрешающий обращение к ОП, если адрес относится к выделенной для данной программы области памяти,
- запрещающий выполнение данной команды в противном случае.

Реализация идеи защиты памяти в любом случае не должна сопровождаться заметным снижением производительности машины и не требовать больших аппаратных затрат.



# Методы ускорения процессов обмена между ОП и ВЗУ

Эффективная скорость обмена между оперативным и внешним уровнями памяти в значительной степени определяется **затратами на поиск секторов или блоков** в накопителе ВЗУ. Для уменьшения влияния затрат времени поиска информации на скорость обмена используют традиционные методы буферизации.

Метод буферизации заключается в использовании так называемой **дисковой кэш-памяти**. Дисковый кэш уменьшает среднее время обращения к диску. Это достигается за счет того, что копии данных, находящихся в дисковой памяти, заносятся в полупроводниковую память. Когда необходимые данные оказываются находящимися в кэше, время обращения значительно сокращается. За счет исключения задержек, связанных с позиционированием головок, время обращения может быть уменьшено в **2 - 10** раз.

Дисковый кэш может быть реализован программно или аппаратно



# *Принципы организации подсистемы ввода-вывода*



# Проблемы организации подсистем ввода-вывода

Производительность и эффективность использования ЭВМ в очень большой степени определяются составом ее периферийных устройств (ПУ), их техническими данными и способом организации их совместной работы с ядром (процессором и основной памятью) ЭВМ.

Связь устройств ЭВМ друг с другом осуществляется с помощью интерфейсов.

**Интерфейс - совокупность линий и шин, сигналов, электронных схем и алгоритмов (протоколов), предназначенную для осуществления обмена информацией между устройствами.**

От характеристик интерфейсов во многом зависят производительность и надежность ЭВМ.



# Проблемы организации подсистем ввода-вывода

При разработке систем **ввода-вывода** должны быть решены следующие проблемы:

1. Должна быть обеспечена возможность реализации машин с переменным составом оборудования (машин с переменной конфигурацией). В первую очередь, с различным набором периферийных устройств с тем, чтобы пользователь мог выбирать состав оборудования (конфигурацию) машины в соответствии с ее назначением, легко дополнять машину новыми устройствами.
2. Для эффективного и высокопроизводительного использования оборудования ЭВМ должны реализовываться параллельная во времени работа процессора над программой и выполнение ПУ процедур ввода-вывода.
3. Для пользователя необходимо упростить и стандартизировать программирование операций ввода-вывода, обеспечить независимость программирования ввода-вывода от особенностей того или иного ПУ.
4. Необходимо обеспечить автоматическое распознавание и реакцию ядра ЭВМ на многообразие ситуаций, возникающих в ПУ (готовность устройства, отсутствие носителя, различные нарушения нормальной работы и др.)



# Основные пути решения проблем

- **Модульность.** Средства современной ВТ проектируются на основе **модульного принципа**. Отдельные устройства выполняются в виде конструктивно законченных модулей, которые могут сравнительно просто в нужных количествах и номенклатуре объединяться, образуя ЭВМ.
- **Унифицированные** (не зависящие от типа ПУ) **форматы данных**, которыми ПУ обмениваются с ядром ЭВМ, в том числе унифицированный формат сообщения, которое ПУ посылает в ядро о своем состоянии.
- **Унифицированный интерфейс**, т.е. унифицированный по составу и назначению набор линий и шин, унифицированные схемы подключения, сигналы и алгоритмы (протоколы) управления обменом информацией между ПУ и ядром ЭВМ.
- **Унифицированные формат и набор команд процессора для операций ввода-вывода.** Операция ввода-вывода с любым ПУ представляет для процессора просто операцию передачи данных независимо от особенностей принципа действия данного ПУ, типа его носителя и т.п.



# Способы организации передачи данных

В подсистемах ввода-вывода ЭВМ используются два основных способа организации передачи данных между памятью и ПУ:

- **Программно-управляемая передача** - осуществляется при непосредственном участии и под управлением процессора, который при этом выполняет специальную подпрограмму ввода-вывода.
- **Прямой доступ к памяти (ПДП)** - это такой способ обмена данными, который обеспечивает автономно от ЦП установление связи и передачу данных между ОП и ПУ.



# Программно-управляемая передача

При программно-управляемой передаче данных ЦП на все время этой передачи отвлекается от выполнения основной программы. Пересылка данных слишком проста, процессор не загружен эффективно. Но при этом ЦП приходится для каждой единицы передаваемых данных (байт, слово) выполнять довольно много инструкций (буферизация данных, преобразование форматов, подсчет количества переданных данных, формирование адресов в памяти и т.п.). В результате **скорость передачи данных** при пересылке **блока данных** под управлением процессора оказывается мала.

**Поэтому для быстрого ввода-вывода блоков данных и разгрузки ЦП от управления операциями ввода-вывода используют прямой доступ к памяти.**



# Прямой доступ к памяти

ПДП освобождает процессор от управления операциями ввода-вывода, позволяет осуществлять параллельно во времени выполнение процессором программы с обменом данными между ОП и ПУ, производить обмен со скоростью, ограничиваемой только пропускной способностью ОП или ПУ.





# Прямой доступ к памяти

Прямой доступ к памяти управляет контроллер ПДП, который выполняет следующие функции:

- Управление инициируемой процессором или ПУ передачей данных между ОП и ПУ.
- Задание размера блока данных, который подлежит передаче и области памяти, используемой при передаче.
- Формирование адресов ячеек ОП, участвующих в передаче.
- Подсчет числа единиц данных (байт, слов), передаваемых от ПУ в ОП или обратно, и определение момента завершения заданной операции ввода-вывода.

Контроллер ПДП имеет более высокий приоритет в занятии цикла памяти по сравнению с процессором. Управление памятью переходит к контроллеру ПДП, как только завершится цикл ее работы, выполняемый для **текущей** команды процессора. ПДП обеспечивает **высокую скорость** обмена данными за счет **аппаратного, а не программного** управления обменом.

В современных ЭВМ используется как программно-управляемая передача данных, так и прямой доступ к памяти. **Программно-управляемый обмен** сохраняют для операций ввода-вывода **отдельных байт (слов)**, которые выполняются быстрее, чем при ПДП, так как исключаются потери времени на программно-управляемую установку начальных состояний регистров и счетчиков контроллера ПДП (инициализация).



# Унификация средств обмена и интерфейсы ЭВМ

Объединение отдельных подсистем ЭВМ в единую систему основывается на многоуровневом принципе с унифицированным сопряжением между всеми уровнями - **стандартным интерфейсом**.

Стандартные интерфейсы – это такие интерфейсы, которые приняты и рекомендованы в качестве обязательных отраслевыми или государственными стандартами, различными международными комиссиями, а также крупными зарубежными фирмами.

Однозначной классификации интерфейсов нет. Выделяют следующих четыре классификационных признака интерфейсов:

- способ соединения компонентов системы (радиальный, магистральный, смешанный);
- способ передачи информации (параллельный, последовательный, параллельно-последовательный);
- принцип обмена информацией (асинхронный, синхронный);
- режим передачи информации (двусторонняя поочередная передача, односторонняя передача).



# Унификация средств обмена и интерфейсы ЭВМ

- **Радиальный интерфейс** позволяет всем модулям работать независимо, но имеет максимальное количество шин.  
**Магистральный интерфейс** (общая шина) использует принцип разделения времени для связи между ЦМ и другими модулями. Он сравнительно прост в реализации, но лимитирует скорость обмена.
- **Параллельные интерфейсы** позволяют передавать одновременно определенное количество бит или байт информации по многопроводной линии. **Последовательные интерфейсы** служат для последовательной передачи по двухпроводной линии.
- В случае **синхронного интерфейса** моменты выдачи информации передающим устройством и приема ее в другом устройстве должны синхронизироваться, для этого используют специальную линию синхронизации. При **асинхронном интерфейсе** осуществляется передача по принципу "запрос-ответ". Каждый цикл передачи сопровождается последовательностью управляющих сигналов, которые вырабатываются передающим и приемным устройствами. Передающее устройство может осуществлять передачу данных (байта или нескольких байтов) только после подтверждения приемником своей готовности к приему данных.



# Классификация интерфейсов по назначению

В ЭВМ и вычислительных системах можно выделить несколько уровней сопряжения:

- машинные системные интерфейсы;
- локальные шины;
- интерфейсы периферийных устройств (малые интерфейсы);
- межмашинные интерфейсы.

**Машинные** (внутримашинные) системные интерфейсы предназначены для организации связей между составными компонентами ЭВМ на уровне обмена информацией с центральным процессором, ОП и контроллерами (адаптерами) ПУ.

**Локальной шиной** называется шина, электрически выходящая непосредственно на контакты микропроцессора, и предназначенная для увеличения быстродействия видеоадаптеров и контроллеров дисковых накопителей. Она обычно объединяет процессор, память, схемы буферизации для системной шины и ее контроллер, а также некоторые вспомогательные схемы. Типичный пример – шина PCI.

Назначение **интерфейсов периферийных устройств** состоит в выполнении функций сопряжения контроллера (адаптера) с конкретным механизмом ПУ.

**Межмашинные интерфейсы** используются в вычислительных системах и сетях.



# Типы и характеристики стандартных шин

**ISA (Industry Standard Architecture).** Старый промышленный стандарт. Позволила связать все устройства системного блока и обеспечила простое подключение новых устройств через стандартные разъемы (слоты). Пропускная способность до **5,5 Мбайт/с**.

**EISA (Extended ISA).** Расширение стандарта ISA. Пропускная способность до **32 Мбайт/с**. После 2000 года выпуск материнских плат с разъемами ISA/EISA и устройств, подключаемых к ним, прекратился.

**PCI (Peripheral Component Interconnect).** Стандарт подключения внешних компонентов. По своей сути это интерфейс локальной шины, связывающей процессор с оперативной памятью, в которую врезаны разъемы для подключения внешних устройств. PCI появилась как высокоскоростная шина с пропускной способностью **132 Мбайт/с** при частоте шины 33 МГц. Более поздние версии поддерживают частоту до 66 МГц и обеспечивают производительность **264 Мбайт/с** для 32-разрядных данных и **528 Мбайт/с** для 64-разрядных данных.

Важное нововведение этого стандарта - поддержка механизма **plug-and-play** (после физического подключения внешнего устройства к разъему шины PCI происходит автоматическая конфигурация этого устройства).

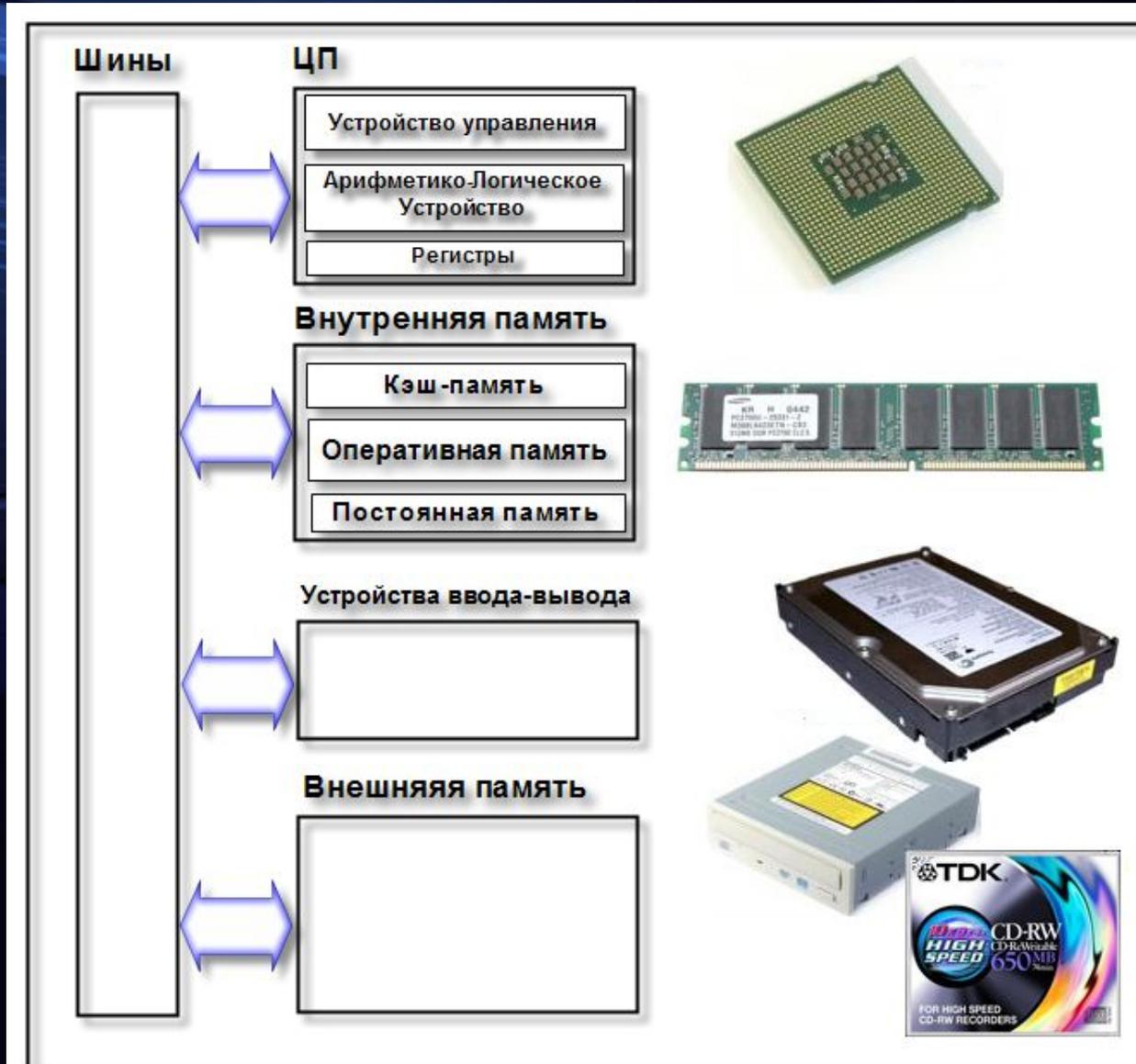
**FSB (Front Side Bus).** Эта шина работает на частоте 100-133 МГц и имеет пропускную способность до **800 Мбайт/с**. За шиной PCI осталась лишь функция подключения новых внешних устройств.

**AGP (Advanced Graphic Port).** Специальный шинный интерфейс для подключения видеоадаптеров. Частота этой шины – 33, 66 МГц и более, пропускная способность до **1066 Мбайт/с**.

**PCI Express.** Шина PCI Express, ранее известная как шина ввода/вывода третьего поколения, призвана заменить шину PCI и взять на себя задачу по связи компонентов внутри компьютера на ближайшие несколько лет. В отличие от PCI, шина PCI Express является последовательной. То есть она использует небольшое число контактов. В то же время, частота работы шины намного выше частоты PCI, что обеспечивает высокую пропускную способность. PCI Express является масштабируемой и позволяет легко нарастить пропускную способность, объединяя несколько линий PCI Express. Спецификация описывает пять различных типов слотов: x16, x8, x4, x2 и x1.

**USB (Universal Serial Bus).** Стандарт универсальной последовательной шины определяет новый способ взаимодействия компьютера с периферийным оборудованием. Он разрешает подключать до **256** разных устройств с последовательным интерфейсом, причем устройства могут подсоединяться цепочкой. Производительность шины USB небольшая и составляет **1,55 Мбит/с**, этого вполне достаточно для таких устройств как мышка и клавиатура. Среди преимуществ этого стандарта - возможность подключать и отключать устройства в "горячем режиме" (то есть без перезагрузки компьютера), и возможность объединения нескольких компьютеров в простую сеть без использования специального аппаратного и программного обеспечения.

# Принципиальное устройство компьютера



Шины

ЦП

Устройство управления

Арифметико-Логическое  
Устройство

Регистры

Внутренняя память

Кэш-память

Оперативная память

Постоянная память

Устройства ввода-вывода

Внешняя память





# Подсистемы

- подсистема управления и обслуживания;
- обрабатывающая подсистема;
- подсистема памяти;
- подсистема ввода-вывода.

# Подсистема управления и обслуживания



## Материнская плата

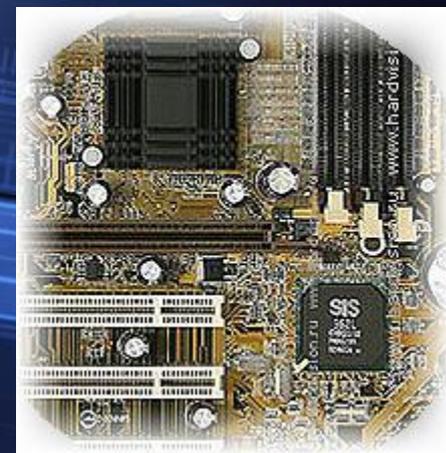
Основной компонент системы. Она объединяет все компоненты компьютера; определяет работу процессора и памяти, позволяет распределить все информационные потоки компьютера, и управляет питанием каждого компонента.



## Чипсет (Chipset) - набор микросхем.

Функции: обмен и регулировка потоков данных между процессором и всеми устройствами, находящимися на плате - памятью, системными шинами, интегрированными устройствами (видео- и аудиоконтроллеры), контроллерами жестких дисков и т.д.

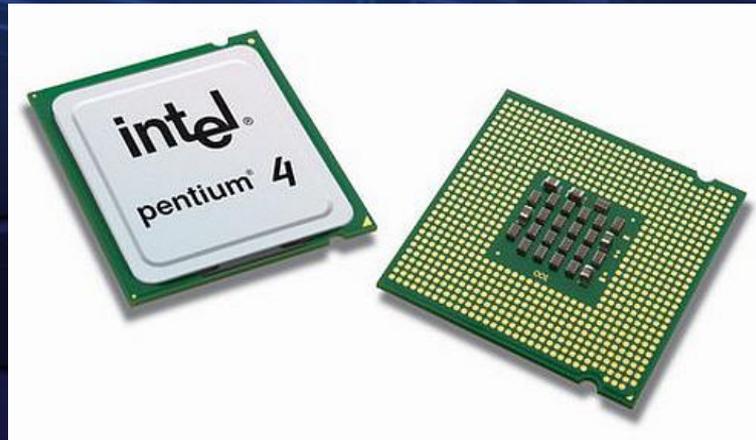
Параметры производительности и функциональности материнской платы напрямую связаны с тем чипсетом, который используется на конкретной плате.



# Обрабатывающая подсистема



- **Центральный процессор** выполнен в виде интегральной микросхемы, называемой микропроцессором. В ПК IBM PC используются микропроцессоры фирмы INTEL, а также совместимые с ними микропроцессоры других фирм.



Микропроцессор обычно характеризуется своим типом и тактовой частотой. Она указывает, сколько элементарных операций выполняется в секунду. Частота измеряется в мегагерцах.

# Подсистема памяти



**Память** - совокупность устройств, служащих для запоминания, хранения и выдачи информации.

Различают внутреннюю и внешнюю память.

В постоянной памяти (**ROM**) помещаются программы, необходимые для запуска компьютера и важнейших компонент операционной системы.

Оперативная память (**RAM**) хранит программный код и данные при работе компьютера. Данные и программа загружаются в оперативную память, откуда процессор и берет их для обработки (непосредственно, либо через кэш-память). В нее же записывают полученные результаты.

Кэш-память (**CASH**) – быстродействующая сверхоперативная память, которая уменьшает количество обращений к оперативной памяти. Быстродействие этой памяти намного больше, чем у оперативной, а объём её – меньше.

Внешняя память обычно используется для хранения файлов, содержимое которых может быть произвольным.