

---

# Архитектура ЭВМ

---

## Содержание

1. Понятие архитектуры ЭВМ
2. Классическая архитектура ЭВМ. Принципы фон Неймана
3. Схема ПК

---

# Понятие и основные виды архитектуры ЭВМ

- *Архитектура компьютера* — логическая организация и структура аппаратных ресурсов вычислительной системы и программного обеспечения.
  - Открытая
  - Закрытая



---

# Понятие и основные виды архитектуры ЭВМ

- **Открытая архитектура** — архитектура компьютера, периферийного устройства или же программного обеспечения, на которую опубликованы спецификации, что позволяет другим производителям разрабатывать дополнительные устройства к системам с такой архитектурой.
-

- ***Стандарты*** – определенные правила, которых придерживаются производители для обеспечения совместимости своих продуктов с продуктами других производителей.

### ***В области компьютерных технологий***

стандарты определяют физические и функциональные характеристики:

- оборудования ПК;
- сетевого и коммуникационного оборудования;
- операционных систем;
- программного обеспечения.

---

# Понятие и основные виды архитектуры ЭВМ

- ***Закрытая архитектура*** - это архитектура, спецификации которой не опубликованы, либо в них не предусмотрено подключение устройств и дополнительных плат.
-

---

# Виды стандартов

- ❑ Стандарты отдельных фирм
  - ❑ Стандарты специальных комитетов и объединений, создаваемых несколькими фирмами
  - ❑ Национальные стандарты
  - ❑ Международные стандарты
-

# Понятие и основные виды архитектуры ЭВМ

**1946 г.** «Предварительное рассмотрение логической конструкции электронно-вычислительного устройства»:

1. обосновывается использование двоичной системы для представления чисел.
  2. принцип «хранимой программы».
- **Архитектура Фон-Неймана** - подразумевает физическое отделение процессорного модуля от устройств хранения программ и данных.



Джон Фон-Нейман

# Понятие и основные виды архитектуры ЭВМ

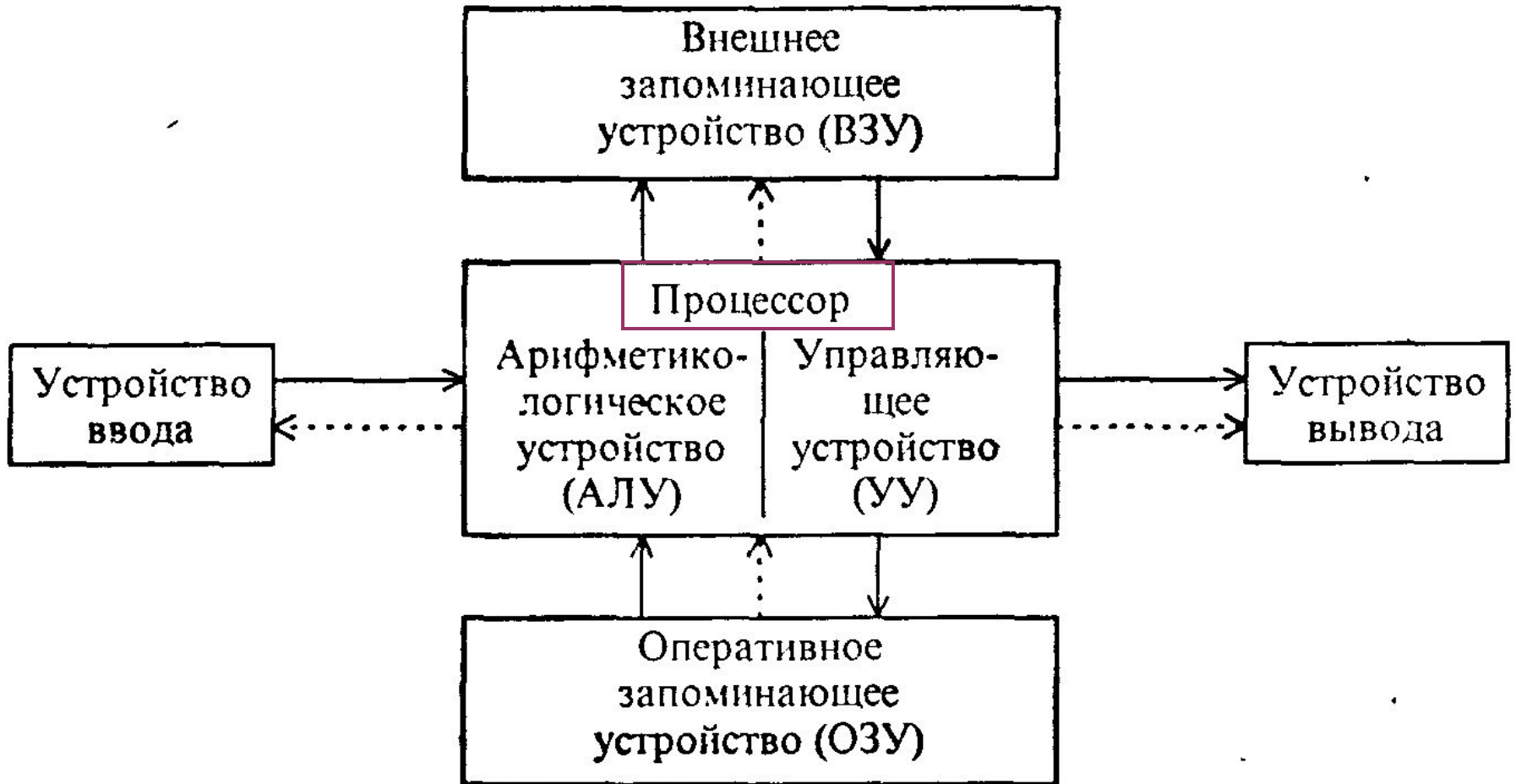
## *Принципы Фон-Неймана*

- Принцип использования двоичной системы счисления для представления данных и команд.
- Принцип однородности памяти.
  - Как программы (команды), так и данные хранятся в одной и той же памяти (и кодируются в одной и той же системе счисления — чаще всего двоичной). Над командами можно выполнять такие же действия, как и над данными.
- Принцип адресуемости памяти.
  - Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.
- Принцип программного управления.
  - Программа состоит из набора команд, которые выполняются процессором друг за другом в определенной последовательности.
- Принцип последовательного программного управления
  - Все команды располагаются в памяти и выполняются последовательно, одна после завершения другой.

Компьютеры, построенные на этих принципах, относят к типу фоннеймановских.



# Архитектура Фон-Неймана

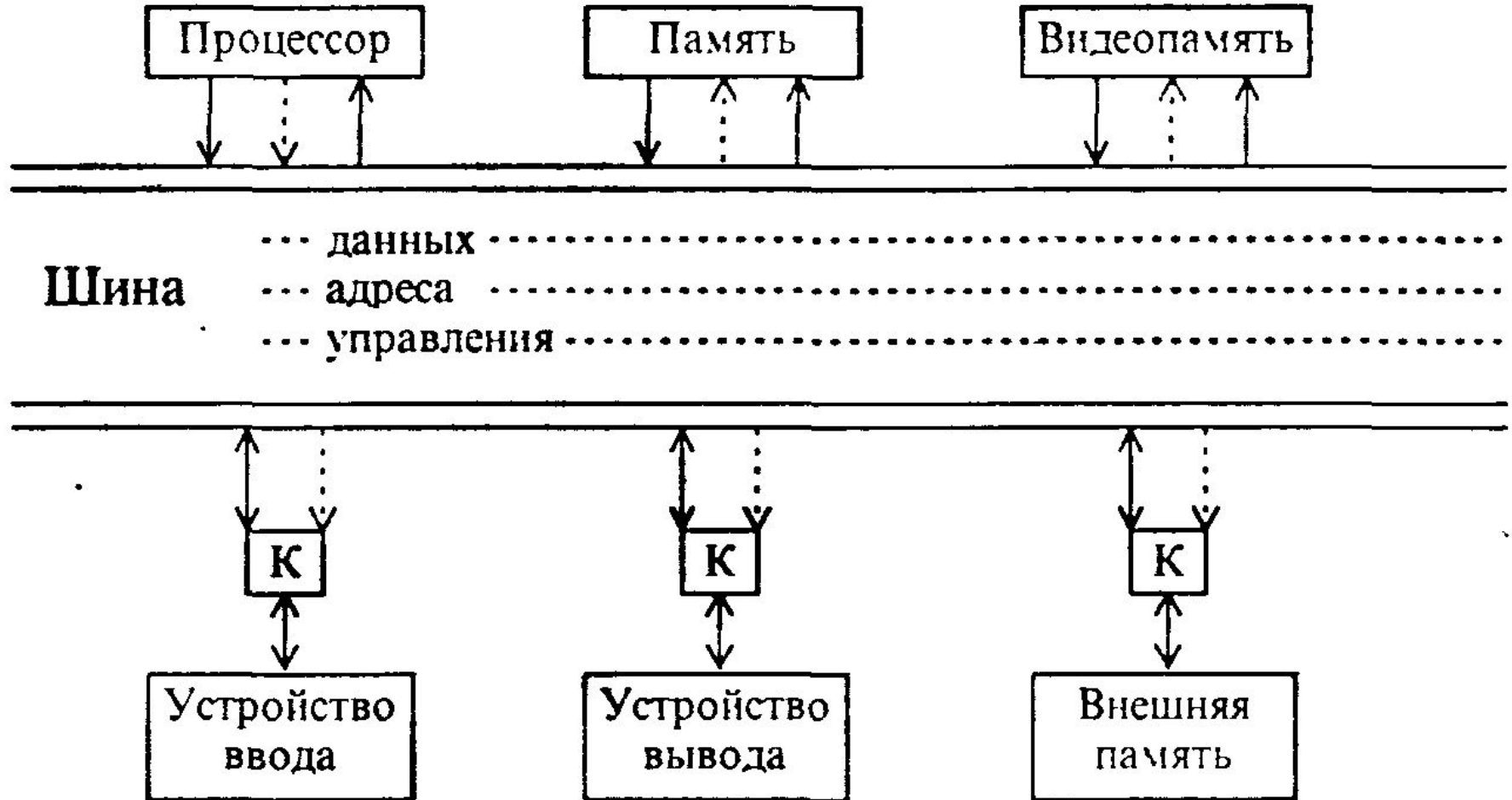


---

*Существенное противоречие* между высокой скоростью обработки информации внутри машины и медленной работой устройств ввода-вывода.

*Контроллер* - специализированный процессор, управляющий работой «вверенного ему» внешнего устройства по специальным встроенным программам обмена.

---



Шина, связывающая только два устройства, называется *портом*.

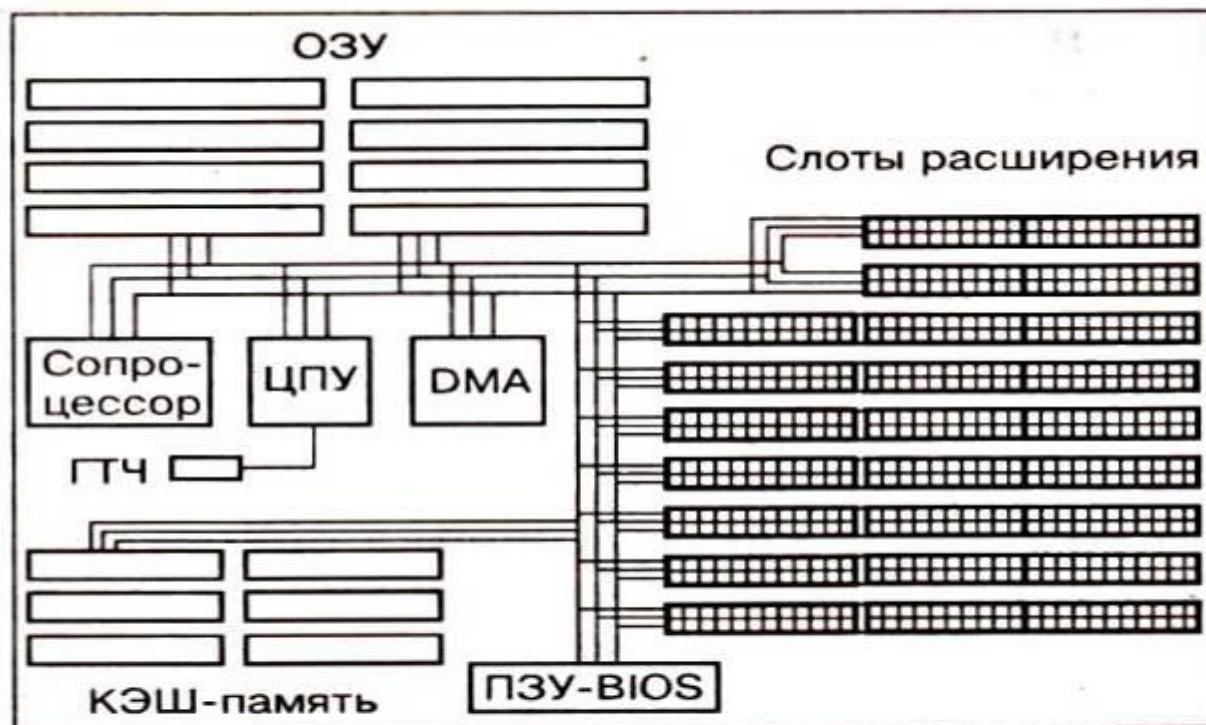
---

# *Системная шина*

## **Назначение:**

Это главная магистраль, по которой происходит обмен информацией между процессором и памятью и их связь с периферийными устройствами.

---



Схематичное представление системной шины на материнской плате.

Обозначения: ЦПУ — центральный процессор; ГТЧ — генератор тактовой частоты; DMA — устройство прямого доступа к памяти

---

## *Основные пользовательские характеристики:*

- ☺ **Разрядность** – количество бит информации, параллельно «проходящих» через неё;
  - ☺ **Пропускная способность** – количество бит информации, передаваемых по шине за секунду.
-

---

Разрядности шины данных, шины адреса, шины управления, как правило не совпадают.

*Пример:*

Компьютеры с процессором 80286 имеют 16-разрядную шину данных, компьютеры семейства Pentium – 64 разрядную шину данных.

---

---

# *Шина данных*

- По этой шине данные передаются между различными устройствами в любом направлении.
  - Разрядность шины данных определяется разрядностью процессора, т.е. количеством двоичных разрядов, которые могут обрабатываться и передаваться процессором одновременно.
  - Разрядность процессоров постоянно увеличивается по мере развития компьютерной техники.
-



# *Шина адреса*

- Выбор устройства или ячейки памяти, куда пересылаются или откуда считываются данные по шине данных, производит процессор.
- Каждое устройство или ячейка оперативной памяти имеет свой адрес.
- Адрес передается по адресной шине, причем сигналы по ней передаются в одном направлении – от процессора к оперативной памяти и устройствам (однонаправленная шина).

*Разрядность адресной шины* определяет доступное адресное пространство, т.е. количество однобайтовых ячеек оперативной памяти, которые могут иметь уникальные адреса.

Если разрядность адресной шины равна  $n$ , то максимальный адрес, который может быть по ней передан –  $2^n$ .

Очевидно, количество байтов оперативной памяти не должно превышать  $2^n$ , иначе байты с большими адресами не будут использоваться.

---

*Пример:*

Компьютеры с процессором 80286 имеют 24-разрядную адресную шину и могут адресовать память объемом

$$2^{24} = 16\,777\,216 \text{ байт} = 16 \text{ Мб.}$$

Компьютеры семейства Pentium – 32 разрядную адресную шину и могут адресовать память объемом 4 Гб.

---

---

# *Шина управления*

- По шине управления передаются сигналы, определяющие характер обмена информацией по магистрали.
  - Сигналы управления показывают, какую операцию – считывание или запись информации из памяти – нужно производить, синхронизируют обмен информацией между устройствами и т.д.
-

---

Для определения пропускной способности шины необходимо умножить разрядность шины на тактовую частоту, которая, как и для процессора, определяется генератором тактовой частоты.

*Пример:*

Для 16-разрядной шины при тактовой частоте 8,33 МГц пропускная способность равна:

$16 \text{ бит} \times 8,33 \text{ МГц} = 16,66 \text{ Мбайт/с.}$

---

---

# Современные тенденции развития архитектуры ЭВМ

- Расширяется и совершенствуется набор внешних устройств
  - Компьютеры перестают быть однопроцессорными
-

## Внутренняя архитектура ЭВМ



### Схема архитектуры ПК, основанной на магистрально-модульном принципе

Обозначения: НГМД — накопитель на гибких магнитных дисках (дискетод флоппи-диска); Винчестер (НЖМД) — накопитель на жестких магнитных дисках

---

Для того чтобы устройства работали в комплексе, нужны специальные программы управления устройством (для каждого устройства – своя). Такие программы называются *драйверами*.





# Понятие и основные виды архитектуры ЭВМ

- *Гарвардская архитектура* – подразумевает физическое разделение хранения и одновременную передачу для обработки данных и команд.
- *Модифицированная гарвардская архитектура* - идея использовать общую шину данных и шину адреса для всех внешних устройств, а внутри процессора использовать шину данных, шину команд и две шины адреса ⇒ *современные сигнальные процессоры.*

---

# Понятие и основные виды архитектуры ЭВМ

## *Однокристальные ЭВМ*

*(микроконтроллеры)*. В них одна шина адреса и данных применяется и внутри кристалла.

В этих микросхемах более важным параметром является не быстродействие, а **надежность** работы устройства.

Надежность достигается невозможностью случайного стирания программы в процессе ее выполнения

---

# Понятие и основные виды архитектуры ЭВМ

- *Гибридные модификации архитектур* – сочетают преимущества гарвардской архитектуры и архитектуры Фон-Неймана.

*Например:* современные варианты процессоров **ARM** обладают отдельной кэш-памятью для инструкций и данных, что позволяет им за один такт получать одновременно как команду, так и данные для её выполнения, то есть процессорное ядро, формально, является гарвардским, но с программной точки зрения выглядит как фон-Неймановское, что упрощает написание программ.

## Внешняя архитектура ЭВМ

Базовый комплект  
персонального компьютера



---

# *Понятие архитектуры ЭВМ*

- Под архитектурой ЭВМ понимают описание устройства и работы компьютера, достаточное для пользователя и программиста.
  - Понятие архитектуры не включает в себя технические детали организации ЭВМ, электронные схемы и т.д.
  - Понятие архитектуры отражает движение информации в компьютере.
-

---

# *Понятие архитектуры ЭВМ*

- Толковый словарь по вычислительным системам предлагает следующее определение термина:
  - «Архитектура ЭВМ используется для описания принципа действия, конфигурации и взаимного соединения основных логических узлов ЭВМ».
-

---

# *Понятие архитектуры ЭВМ*

- Учебник А.В.Могилева дает следующее определение:
  - «Архитектура — это наиболее общие принципы построения ЭВМ, реализующие программное управление работой и взаимодействием основных ее функциональных узлов».
-

---

# *Классическая архитектура ЭВМ.*

## *Принципы фон Неймана*

- Американский математик Джон фон Нейман в 1946 г. в классической статье «Предварительное рассмотрение логической конструкции электронно-вычислительного устройства» совместно с Г.Голдстайном и А.Берксом предложил идею принципиально новой ЭВМ. Выдвинутые идеи актуальны и сегодня.
-



---

# *Принципы фон Неймана*

## 1. Программное управление работой ЭВМ.

Программа состоит из команд.

- Все команды образуют систему команд машины.
  - Команды программы последовательно считываются из памяти и выполняются.
  - Адрес очередной команды хранится в счетчике команд.
-

---

# *Принципы фон Неймана*

## 2. Принцип хранимой программы.

- Команды представляются в числовой форме и хранятся в той же памяти, что и данные.
-

---

# *Принципы фон Неймана*

## 3. Принцип условного перехода.

- Можно нарушить естественную последовательность команд в программе.
  - Используется в командах безусловного и условного переходов
-

---

# *Принципы фон Неймана*

4. Использование двоичной системы счисления для представления информации в ЭВМ.

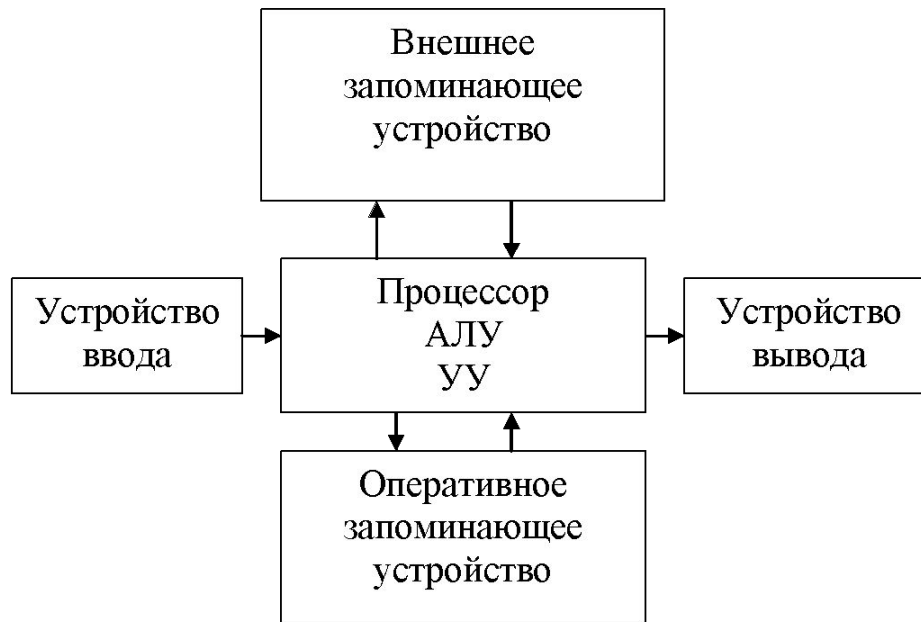
- Ее просто реализовать технически для выполнения арифметических и логических операций.
  - Ранее ЭВМ обрабатывали числа в десятичном виде.
-

---

# *Принципы фон Неймана*

- Принцип иерархичности ЗУ.
  - 1 уровень — Быстродействующее ОЗУ — небольшой емкости для операндов и команд, участвующих в счете в данный момент,
  - 2 уровень — внешнее ЗУ большей емкости.
  - Иерархичность ЗУ в ЭВМ это компромисс между емкостью и быстрым доступом к данным.
-

# Схема фон Неймана



---

# Устройства

- Процессор. Программно-управляемое устройство, обрабатывает данные и управляет работой компьютера.
  - Состоит из устройства управления (УУ) и арифметико-логического устройства (АЛУ).
  - УУ управляет работой компьютера, взаимодействием компонентов друг с другом.
  - АЛУ исполняет арифметические и логические операции.
-

# Устройства

- Оперативное запоминающее устройство.
- Хранит информацию, с которой компьютер работает в данное время: программу, исходные данные, промежуточные и конечные результаты счета.
- Эта память небольшого объема, энергозависима.

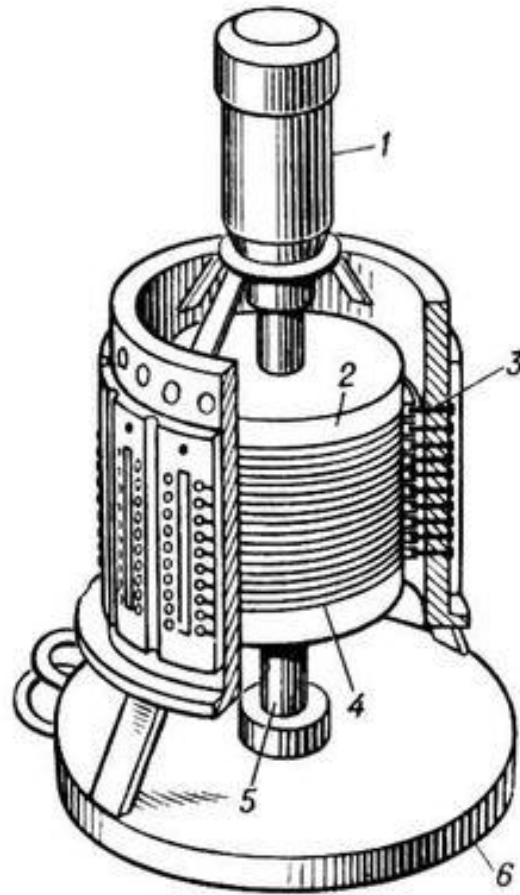


---

# Устройства

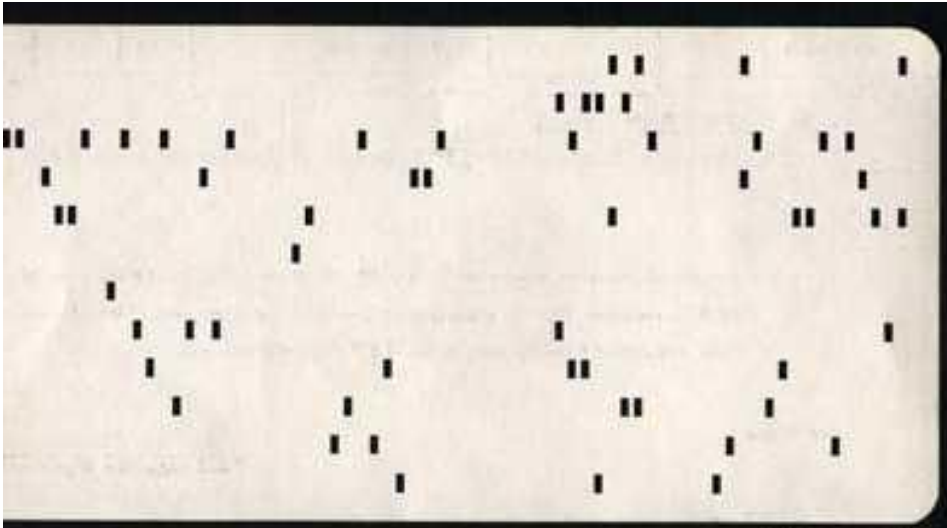
- Внешнее запоминающее устройство.
  - Это были магнитные устройства для долговременного хранения информации.
  - Большого объема, более медленные.
  - Магнитные барабаны, ленты, диски.
-

**Магнитный барабан** 1 электродвигатель 2 цилиндр барабан  
3 магнитные головки 4 дорожки 5 ось магнитного барабана  
6 станина корпус



- 
- Устройства ввода информации.
  - Перфокарты,
  - перфоленты,
  - клавиатура.
-

# Перфокарты, перфоленга

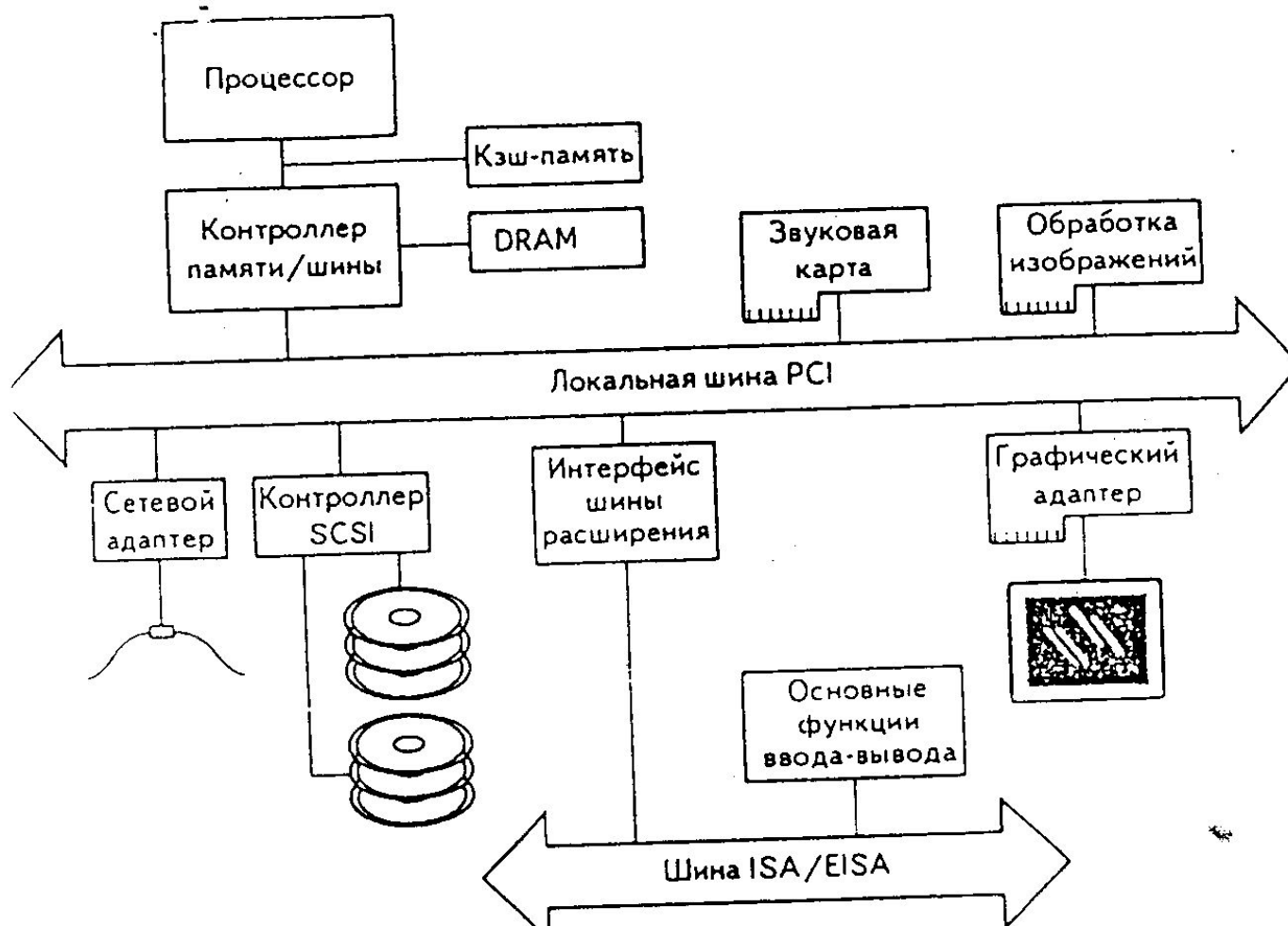


---

## **3. Схема микрокомпьютера 4 поколения**

- В архитектуре персональных машин реализован магистрально модульный принцип:
  - Все устройства выполнены в виде самостоятельно работающих модулей
  - Для связи всех устройств компьютера используют шину, магистраль, по которой передаются данные, адреса и управляющие сигналы.
-

# Схема ПК 4 поколения



# Архитектура персонального компьютера

- **Архитектура 2, 3, 4 разграничивают определённые уровни внутри программного обеспечения.**
- Если программы, реализующие прикладные задачи, написаны на языках программирования, не входящих в число тех, которые представлены в распоряжение пользователя, то можно говорить об архитектуре уровня, назначение которого, определение указанных языков. Трансляторы таких языков взаимодействуют с более низкими уровнями программного обеспечения, обозначенными на абстрактной модели архитектуры как 3 и 4.
- Из-за отсутствия лучшего термина о всех трех уровнях 2-4 ~~будем говорить как об архитектуре программного~~ обеспечения.

# Архитектура персонального компьютера

- **Архитектура уровня 5**

- Уровни отражают границу между системным программным и аппаратным обеспечением (термин “аппаратное обеспечение” используется для обозначения как микропрограмм, так и электронных логических схем).
- Микропрограмма – это записанная в памяти программа, которая физически управляет передачей всех символов и данных в физических компонентах системы, таких, как шины, регистры, сумматоры или процессор.



# Архитектура персонального компьютера

- **Архитектура уровня 6**
- Представляет собой интерфейс микропрограммы т.е. обеспечивает согласование потока данных и управляющих сигналов с форматом микрокоманд внутри каждого процессора. Архитектуру уровней 6 и 8 также часто называют архитектурой процессора или организацией процессора.

# Архитектура персонального компьютера

- **Архитектура уровня 7**

- Уровень 7 определяет, какие функции реализует центральные процессоры выполняющие программы, а какие процессоры ввода – вывода (т.е. каналы).
- Архитектура другого уровня определяет разграничение функций между процессорами ввода – вывода и контроллерами (устройствами управления) внешних устройств. В свою очередь можно разграничить функции, реализуемые контроллерами и самими устройствами ввода – вывода (терминалами, модемами, накопителями на магнитных дисках и магнитных лентах). Архитектура уровней 7, 9 и 10 может быть названа архитектурой физического ввода - вывода.

# Архитектура персонального компьютера

- **Архитектура уровня 8**

- Уровень 8 (интерфейс между процессором и основной памятью) . Функции каждого процессора и контроллера внешнего устройства могут быть распределены между микропрограммами и логическими схемами.
- Последняя разновидность архитектуры, в явном виде не показанная может быть определена как мультипроцессорная архитектура. Такая архитектура предусматривает распределение функций между группой процессоров.

# Архитектура ПК

Информационный обмен между устройствами компьютера



## Виды процессоров

1. Центральный процессор
2. Графический процессор
3. Физический процессор
4. Цифровой сигнальный процессор
5. Сетевой процессор
6. Звуковой п



## Центральный процессор



В современном компьютере может быть один или несколько *Центральных процессоров* и *Графический процессор*. *Центральный процессор (ЦП)* является наиболее распространённым термином. Зачастую под термином *процессор* подразумевается именно *Центральный процессор*. В англоязычной литературе для обозначения центрального процессора используются термины *CPU* или *Central Processing Unit*, что дословно можно перевести как основное вычислительное устройство. Вычислительная система, в которой работает несколько центральных процессоров и единое адресное пространство, называется многопроцессорной.

## Графический процессор



В отношении *Графического процессора (ГП)* в англоязычной литературе используется термин *Graphics Processing Unit* (англ.: *GPU*). Графический процессор выполняет специфические функции по обработке графической информации. Он обычно монтируется на видеокарте или материнской плате. Как правило, в литературе центральный и графический процессоры обозначают сокращённо термином *процессор*, однако из контекста документа ясно о каком конкретном виде процессора идёт речь.

## Физический процессор



*Физический процессор* (англ.: *Physics Processing Unit, PPU*) – специализированный процессор, предназначен для выполнения математических вычислений при моделировании различных физических процессов, таких как расчёт динамики тел, обнаружение столкновений и пр.





## Цифровой сигнальный процессор



*Цифровой сигнальный процессор* (сигнальный микропроцессор, СМП; процессор цифровых сигналов, ПЦС) — специализированный микропроцессор, предназначенный для цифровой обработки сигналов (обычно в реальном масштабе времени). Данное понятие в англоязычной литературе обозначается термином *Digital signal processor (DSP)*



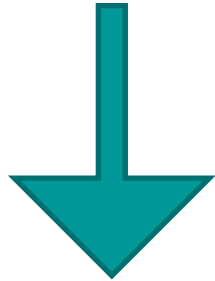
## Сетевой процессор



*Сетевой процессор* (англ.: *network processor*) – это микропроцессор, размещаемый в сетевых устройствах, выполняющий специализированные операции, которые востребованы при передаче данных по сетям. Как правило, сетевой процессор размещается в сетевом устройстве: сетевых платах, маршрутизаторах, коммутаторах и пр.



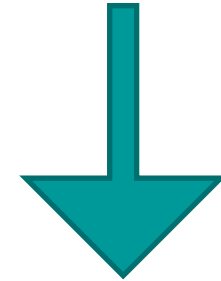
# Память компьютера



## ВНУТРЕННЯЯ

**Внутренняя память** – это электронное устройство, которое хранит информацию, пока питается энергией.

**Внешняя память** – это различные магнитные носители, для сохранения которых, не требуется постоянное электропитание.



## ВНЕШНЯЯ





---

32 Gb

## ОЗУ

(оперативное запоминающее устройство)

предназначена для хранения информации, к которой приходится часто обращаться, и обеспечивает режимы ее записи, считывания и хранения.

---

При выключении ПК оперативная память

# Внутренняя память —

совокупность  
специальных

электронных ячеек,  
каждая из которых может  
хранить конкретную  
комбинацию из 8 битов,  
имеющих два состояния:

**0** — выключено,

**1** - включено.

Принцип  
организации  
внутренней памяти

Но м е р а б а й т о	Биты							
	0	0	0	1	0	1	0	0
1	1	1	0	0	1	0	1	1
2	1	0	1	0	0	1	0	1
3	0	1	1	0	0	1	1	0

Данные и программы в памяти компьютера хранятся в виде двоичного кода.

# Свойства внутренней памяти:

- **Дискретность:** компьютерная память состоит из битов.
- **Адресуемость:** занесение информации в память, а также извлечение ее из памяти, производится по адресам. Порядковый номер байта называется его адресом.





Магнитные  
дорожки



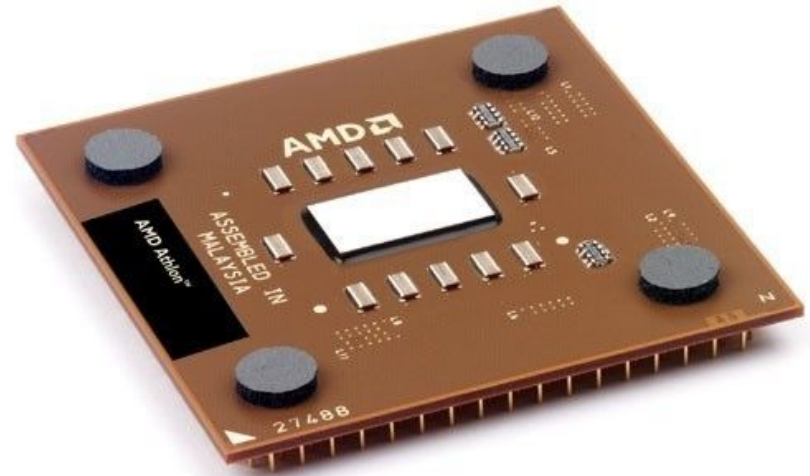
Магнитные  
головки

## Постоянная память (ПЗУ — постоянное запоминающее устройство (винчестер))

обычно содержит такую информацию, которая не должна меняться в ходе выполнения микропроцессором различных программ. Постоянная память энергонезависима.

---

## Кеш – память – (cache memory)



запоминающее устройство с малым временем доступа увеличивает скорость работы диска, но повышает риск потери данных в случае внезапного краха системы.

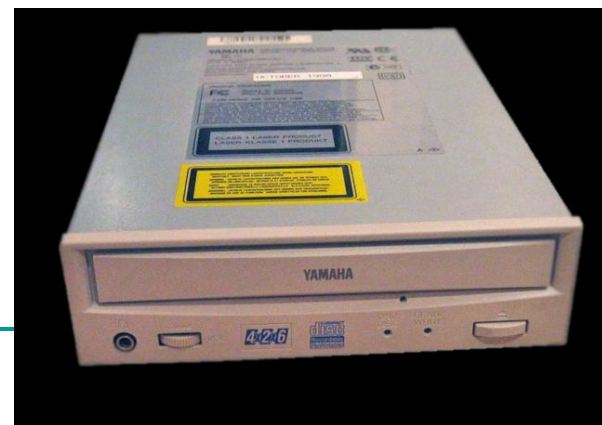
---



# Оптические диски

С появлением программ занимающих десятки и сотни мегабайт гибкие и даже жесткие диски перестали удовлетворять нашим потребностям. Появились принципиально новые носители **лазерные диски** получившие название **CD-ROM** (Compact Disk – Read Only Memory (компакт диск – только для чтения). Информация на такие диски записывалась только один раз. Стереть или перезаписать её невозможно.

Позже были изобретены перезаписываемые лазерные диски – **CD-RW**. На них информацию можно



# Оптические диски

Наибольшей информационной ёмкостью из сменных носителей обладают лазерные диски типа DVD. Объём информации, находящийся на DVD измеряется гигабайтами (4,7 Gb). Они бывают перезаписываемые DVD-RW и не перезаписываемые DVD-R (DVD+R).



**Blu-ray Disc (BD)** - формат оптических дисков последнего поколения. **Стандарт Blu-ray** - продукт совместной разработки группы компаний по производству бытовой электроники и компьютеров, которые вошли в Ассоциацию Blu-ray дисков. Он имеет информационную емкость на спой (25 гигабайт)



# FLASH'ка!



**Флеш-память** (*flash memory*) — разновидность полупроводниковой технологии электрически перепрограммируемой памяти.

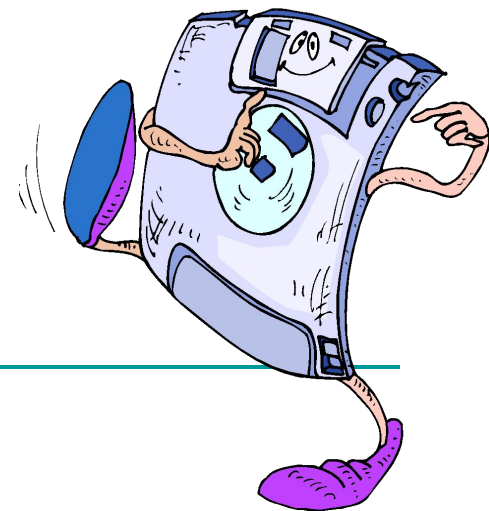
Благодаря компактности, дешевизне, механической прочности, большому объему, скорости работы и низкому энергопотреблению флеш-память широко используется в цифровых портативных устройствах и носителях информации!

## Функции внешней памяти:

1. прием информации от других устройств,
2. запоминание информации,
3. передача информации по запросу в другие устройства.

## Основные характеристики внешней памяти:

1. Объем
2. Время доступа
3. Скорость передачи информации



# Управление памятью

- **Оперативная память** – важнейший ресурс вычислительной системы, требующий управления со стороны ОС. Причина – процессы и потоки хранятся и обрабатываются в оперативной памяти.
- Память распределяется между приложениями и модулями самой операционной системы.
- **Функции ОС по управлению оперативной памятью:**
  - Отслеживание наличия свободной и занятой памяти;
  - Контроль доступа к адресным пространствам процессов;
  - Вытеснение кодов и данных из оперативной памяти на диск, когда размеров памяти недостаточно для размещения всех процессов, и возвращение их обратно;
  - Настройка адресов программы на конкретную область физической памяти;
  - Защита выделенных областей памяти процессов от взаимного вмешательства.
- Часть ОС, которая отвечает за управление памятью, называется **менеджером памяти**.

# Физическая организация памяти

- Запоминающие устройства компьютера разделяют, как минимум, на два уровня: *основную* (главную, *оперативную*, *физическую*) и вторичную (внешнюю) память.
- *Основная память* представляет собой упорядоченный массив однобайтовых ячеек, каждая из которых имеет свой уникальный адрес (номер). Процессор извлекает команду из *основной памяти*, декодирует и выполняет ее. Для выполнения команды могут потребоваться обращения еще к нескольким ячейкам *основной памяти*.
- Вторичную память (это главным образом диски) также можно рассматривать как одномерное линейное *адресное пространство*, состоящее из последовательности байтов. В отличие от *оперативной памяти*, она является энергонезависимой, имеет существенно большую емкость и используется в качестве расширения *основной памяти*.

# Иерархия памяти

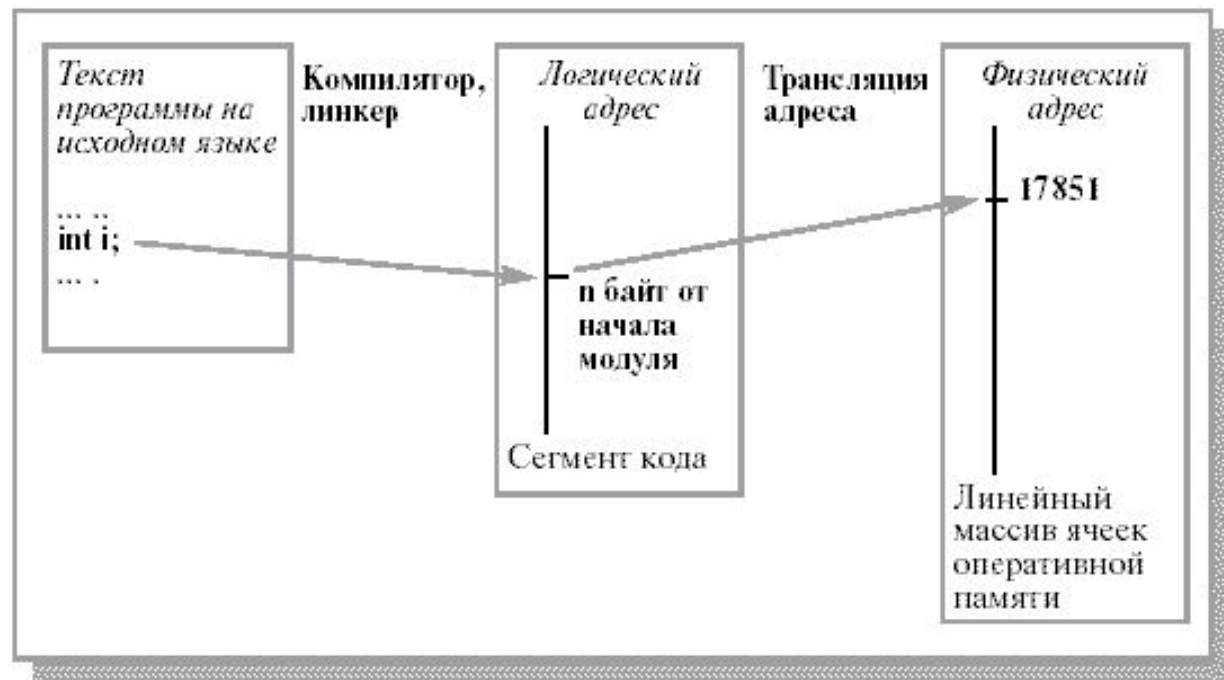


# Представление потоков в оперативной памяти

- Для идентификации переменных и команд программы используются разные типы адресов:
  - **Символьные** (имена переменных, функций и т.п.);
  - **Виртуальные** – условные числовые значения, вырабатываемые компиляторами;
  - **Физические** – адреса фактического размещения в оперативной памяти.



# Связывание адресов



# Виртуальное пространство

- Совокупность виртуальных адресов называется *виртуальным адресным пространством*. Диапазон возможных адресов виртуального пространства у всех процессов одинаков.
- Совпадение виртуальных адресов различных процессов не должно приводить к конфликтам и операционная система отображает виртуальные адреса различных процессов на разные физические адреса.
- Разные ОС по разному организуют виртуальное адресное пространство:
  - **Линейная организация** – пространство представляется непрерывной линейной последовательностью адресов (по другому плоская структура адресного пространства).
  - **Сегментная организация** – пространство разделяется на отдельные части. В этом случае, помимо линейного адреса, может быть использован виртуальный адрес (сегмент, смещение).

# Виртуальное адресное пространство

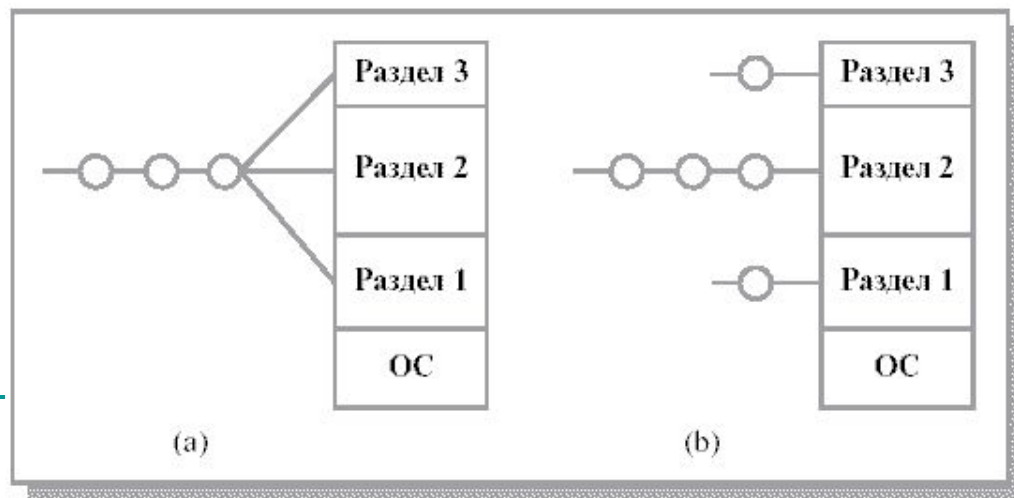
- В виртуальном адресном пространстве выделяют две непрерывные части:
  - Системная – для размещения модулей общих для всей системы (размещаются коды и данные ядра ОС, другие служебные модули);
  - Пользовательская – для размещения кода и данных пользовательских программ.
- Системная область включает в себя область, подвергаемую страничному вытеснению, и область, на которую страничное вытеснение не распространяется. В последней располагаются системные процессы, требующие быстрой реакции или постоянного присутствия в памяти. Остальные сегменты подвергаются вытеснению, как и пользовательские приложения.

# Алгоритмы распределения памяти



# Схема с фиксированными разделами

- Схема основана на предварительном разбиении общего адресного пространства на несколько разделов фиксированной величины.
- Процессы помещаются в тот или иной раздел.
- Связывание физических и логических адресов процесса происходит на этапе его загрузки.

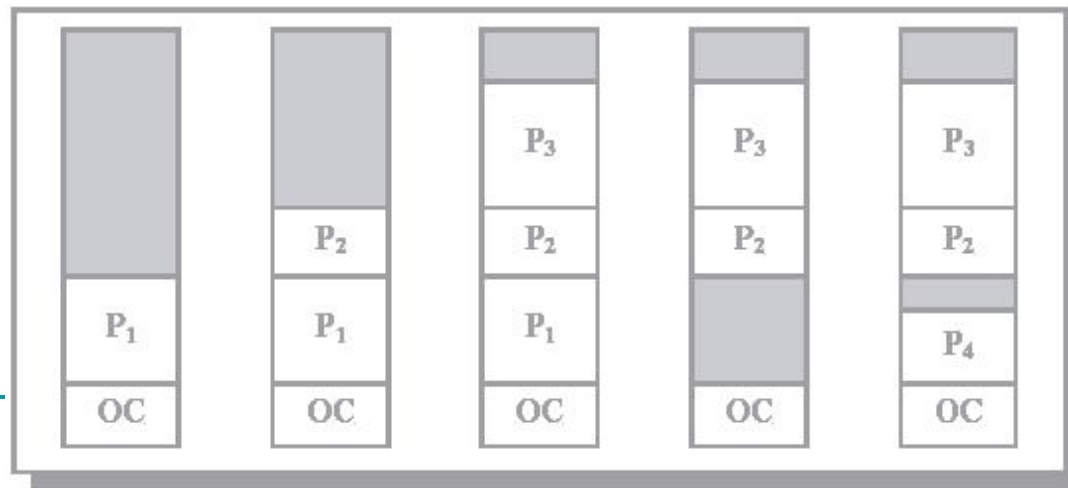


# Динамическое распределение. Свопинг.

- В системах с разделением времени возможна ситуация, когда память не в состоянии содержать все пользовательские процессы.
- В таких случаях используется свопинг (swapping) – перемещению процессов из главной памяти на диск и обратно целиком. Частичная выгрузка процессов на диск осуществляется в системах со страничной организацией (paging).
- Выгруженный процесс может быть возвращен в то же самое *адресное пространство* или в другое. Это ограничение диктуется методом *связывания*. Для схемы *связывания* на этапе выполнения можно загрузить процесс в другое место памяти.

# Схема с переменными разделами

- Типовой цикл работы менеджера памяти состоит в анализе запроса на выделение свободного участка (раздела), выборе его среди имеющихся в соответствии с одной из стратегий (первого подходящего, наиболее подходящего и наименее подходящего), загрузке процесса в выбранный раздел и последующих изменениях таблиц свободных и занятых областей.
- Аналогичная корректировка необходима и после завершения процесса. *Связывание адресов* может осуществляться на этапах загрузки и выполнения.



# Страничная организация

- В случае страничной организации памяти (или paging) как логическое *адресное пространство*, так и физическое представляются состоящими из наборов блоков или *страниц* одинакового размера.
- При этом образуются логические *страницы* (page), а соответствующие единицы в *физической памяти* называют страничными кадрами (page frames). *Страницы* (и страничные кадры) имеют фиксированную длину, обычно являющуюся степенью числа 2, и не могут перекрываться.
- Каждый кадр содержит одну *страницу* данных. При такой организации *внешняя фрагментация* отсутствует, а потери из-за *внутренней фрагментации*, поскольку процесс занимает целое число *страниц*, ограничены частью последней *страницы* процесса.

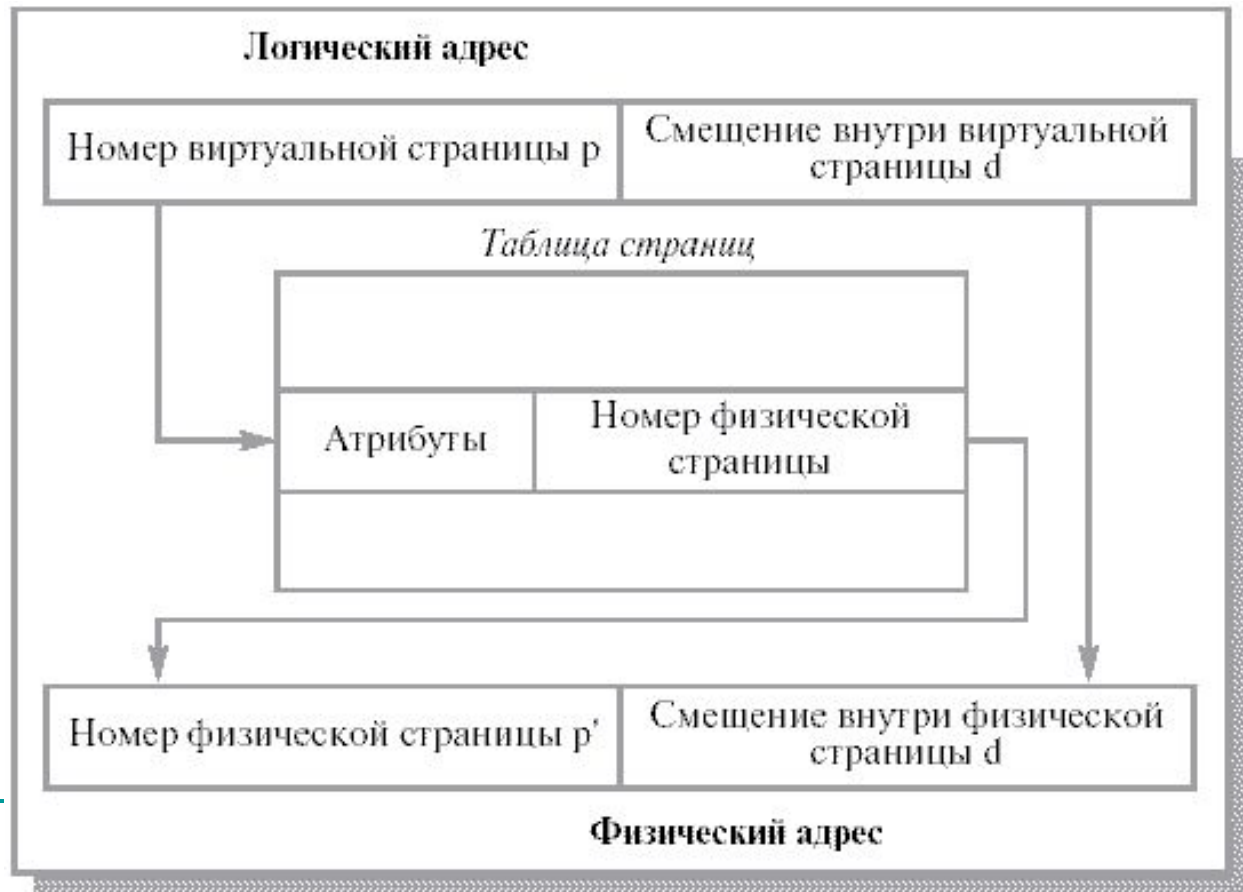


---

# Связь логического и физического адресов

- Логический адрес в страничной системе – упорядоченная пара  $(p, d)$ , где  $p$  – номер *страницы* в виртуальной памяти, а  $d$  – смещение в рамках *страницы*  $p$ , на которой размещается адресуемый элемент.
  - Разбиение *адресного пространства* на *страницы* осуществляется вычислительной системой незаметно для программиста.
  - Адрес является двумерным лишь с точки зрения операционной системы, а с точки зрения программиста *адресное пространство* процесса остается линейным.
-

# Схема адресации при страничной организации



# Сегментная и сегментно-страничная организация памяти

- *Сегменты*, в отличие от *страниц*, могут иметь переменный размер.
- Каждый *сегмент* – линейная последовательность адресов, начинающаяся с 0. Максимальный размер *сегмента* определяется разрядностью процессора (при 32-разрядной адресации это 2<sup>32</sup> байт или 4 Гбайт).
- Размер *сегмента* может меняться динамически (например, *сегмент* стека). В элементе таблицы *сегментов* помимо физического адреса начала *сегмента* обычно содержится и длина *сегмента*.
- Логический адрес – упорядоченная пара  $v=(s,d)$ , номер *сегмента* и смещение внутри *сегмента*.

---

# Виртуальная память

- Разработчикам программного обеспечения часто приходится решать проблему размещения в памяти больших программ, размер которых превышает объем доступной оперативной памяти.
  - Развитие архитектуры компьютеров и расширение возможностей операционной системы по управлению памятью позволило переложить решение этой задачи на компьютер. Одним из подходов стало появление *виртуальной памяти* (virtual memory).
-

# Концепция работы с виртуальной памятью

- Информация, с которой работает активный процесс, должна располагаться в оперативной памяти.
- В схемах *виртуальной памяти* у процесса создается иллюзия того, что вся необходимая ему информация имеется в основной памяти.
  - во-первых, занимаемая процессом память разбивается на несколько частей, например страниц;
  - во-вторых, логический адрес (логическая страница), к которому обращается процесс, динамически транслируется в физический адрес (физическую страницу);
  - и наконец, в тех случаях, когда страница, к которой обращается процесс, не находится в физической памяти, нужно организовать ее подкачку с диска.
- Для контроля наличия страницы в памяти вводится специальный *бит присутствия*, входящий в состав атрибутов страницы в *таблице страниц*.

---

# Кэширование данных

- Для ускорения доступа к данным используется принцип кэширования. В вычислительных системах существует иерархия запоминающих устройств:
    - нижний уровень занимает емкая, но относительно медленная дисковая память;
    - оперативная память;
    - верхний уровень – сверхоперативная память процессорного кэша.
  - Каждый уровень играет роль кэша по отношению к нижележащему.
-

---

# Кэширование данных

- Каждая запись в кэш-памяти об элементе данных включает в себя:
    - Значение элемента данных;
    - Адрес, который этот элемент данных имеет в основной памяти;
    - Дополнительную информацию, которая используется для реализации алгоритма замещения данных в кэше и включает признак модификации и актуальности данных.
-

# Организация кэш-памяти

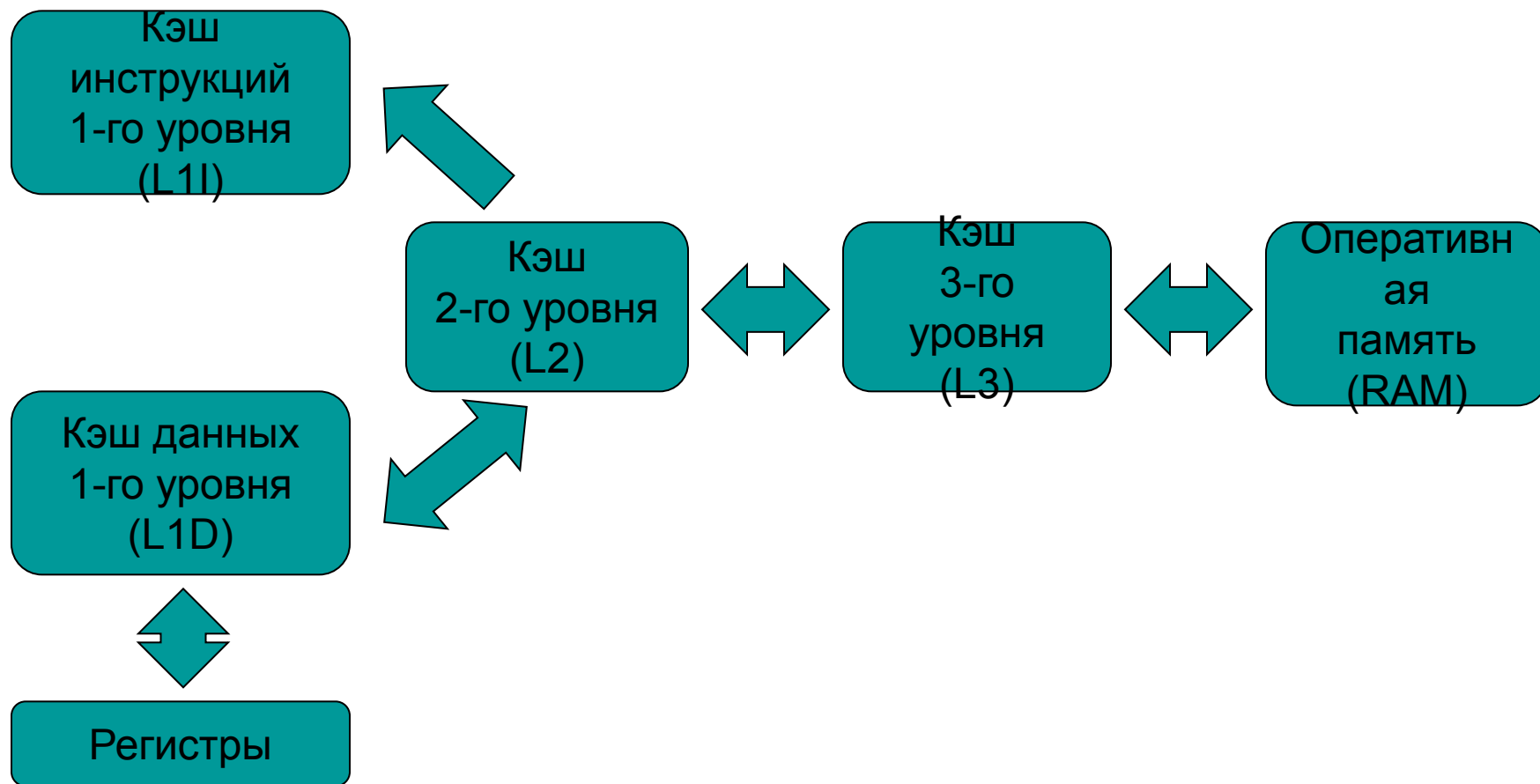
**Кэш-память** это высокоскоростная память небольшого размера с прямым доступом. Она предназначена для временного хранения фрагментов кода и данных. Кэш-память охватывает все адресное пространство памяти, но в отличие от оперативной памяти, она не адресуема и невидима для программиста.

## Схема построения кэш-памяти

- Кэш-память построена на принципе локальности ссылок во времени и в пространстве.
- Кэш-контроллер загружает копии программного кода и данных из ОП в кэш-память блоками, равными размеру строки за один цикл чтения. Процессор читает из кэш-памяти по словам.
- Кэш-контроллер перехватывает запросы процессора к основной памяти и проверяет, есть ли действительная копия информации в кэш-памяти.



# Схема иерархического построения памяти



---

# Алгоритмы отображения

- Прямой (**direct mapping**).
  - Ассоциативный (**full associative mapping**).
  - Множественно-ассоциативный (**set-associative mapping**).
-

---

# Алгоритмы записи

- Сквозная запись (**Write Through (WT)**).
  - Сквозная запись с буферизацией (**Write Combining**).
  - Обратная запись (**Write Back (WB)**).
-

---

# Алгоритмы замещения кэш-строк

- **Least Recently Used (LRU)**
  - **Most Recently Used (MRU)**
  - **Pseudo-Least Recently Used (PLRU)**
-

---

# Каким должен быть размер линии кэш-памяти?

- Размер линии должен быть как минимум в ширину канала памяти
  - **Большой размер**
    - Более эффективное использование канала памяти при последовательном доступе
    - Позволяет уменьшать «ассоциативность» кэша и количество линий
  - **Маленький размер**
    - Более эффективное использование канала памяти при произвольном доступе
    - Заполнение можно делать за одну транзакцию к памяти
-

---

# Какими должны быть основные параметры кэша?

- Размер кэша

- Большой, чтобы вместить рабочие данные
- Маленький, для быстрого доступа

- Степень ассоциативности кэша

- Большая, чтобы избегать пробуксовки
- Маленькая, для быстрого доступа

- Размер строки кэша

- Большой, чтобы использовать локальность
  - Большой, чтобы уменьшить теги
  - Маленький (доля полезных данных в кэше больше, если данные в памяти распределены произвольным образом)
-

---

# **Эффективное использование иерархии памяти**

- **Объем обрабатываемых данных**
  - **Обход данных**
-