



ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ



# Дисциплина: "Компьютерная схемотехника и архитектура компьютеров"

## ЛЕКЦИЯ № 7

### Цифровые компьютеры

доцент кафедры информационных систем  
к.т.н., доцент Голубничий Дмитрий Юрьевич

# ОСНОВНЫЕ ТЕРМИНЫ АРХИТЕКТУРЫ ЭВМ

**Архитектура вычислительной машины (Архитектура ЭВМ, англ. *Computer architecture*)** — концептуальная структура вычислительной машины, определяющая проведение обработки информации и включающая методы преобразования информации в данные и принципы взаимодействия технических средств и программного обеспечения.

В основе построения электронных вычислительных машин лежат два принципа, позволяющие автоматизировать процесс вычислений:

- ❑ **Принцип программного управления вычислительным процессом**, разработанный английским ученым Чарльзом Бебиджем еще в 1833 году
- ❑ **Принцип программы, хранящейся в памяти**, предложенный американским математиком Джоном фон Нейманом в 1945 году.

В настоящее время наибольшее распространение в ЭВМ получили 2 типа архитектуры: *принстонская (неймановская)* и *гарвардская*. Обе они выделяют 2 основных узла ЭВМ: **центральный процессор и память ПК**.

Различие заключается в структуре памяти: в принстонской архитектуре программы и данные хранятся в одном массиве памяти и передаются в процессор по одному каналу, тогда как гарвардская архитектура предусматривает отдельные хранилища и потоки передачи для команд и данных.

# УРОВНИ АБСТРАКЦИИ АРХИТЕКТУРЫ ЭВМ

- ❖ **Архитектура аппаратных средств** – состав электронных, электрических и механических узлов и блоков, их взаимосвязь и взаимодействие.
- ❖ **Архитектура аппаратно-программной границы.** Устанавливает границу между аппаратным обеспечением (электронными логическими схемами и микропрограммами) и операционной системой (программным обеспечением управления функционирования компьютера).
- ❖ **Архитектура микропрограммного управления.** Задаёт границу выполнения операций с помощью логических электронных схем и микропрограмм (набор микроинструкций, реализующих выполнение отдельных этапов машинных операций).
- ❖ **Архитектура программного обеспечения.** Устанавливает уровни разделения используемых языков, программ, программных модулей и приемов их построения.
- ❖ **Архитектура процессора.** Устанавливает организацию процессора и интерфейса между процессором и вторыми устройствами компьютера.
- ❖ **Архитектура каналов связи.** Определяет взаимодействие ядра компьютера (процессора, оперативной памяти) с периферийными устройствами.
- ❖ **Архитектура системы.** Разделяет функции обработки данных, которые выполняются системой и внешней средой (например, пользователями). При этом связь системы с внешней средой осуществляется через наборы интерфейсов: аппаратных (видео подсистема, периферийные устройства, средства связи) и программных (языки программирования, системные программы).

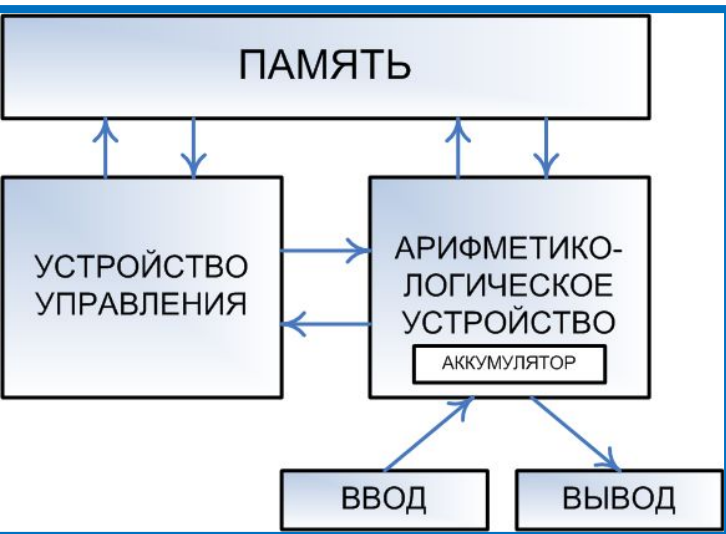
# АРХИТЕКТУРА НЕЙМАНОВСКИХ И ПОСТНЕЙМАНОВСКИХ МАШИН

*Неймановский принцип программного управления* используется в качестве основного принципа построения ЭВМ.

Этот принцип состоит в следующем:

- ❑ информация кодируется в двоичной форме и разделяется на единицы (элементы) информации — **слова**;
- ❑ разнотипные слова информации различаются по способу использования, но не способами кодирования;
- ❑ слова информации размещаются в ячейках памяти машины и идентифицируются номерами ячеек, которые называются **адресами слов**;
- ❑ алгоритм представляется в форме последовательности управляющих слов — **команд**, которые определяют наименование операции и слова информации, участвующие в операции. Алгоритм, представленный в терминах машинных команд, называется программой;
- ❑ выполнение вычислений, предписанных алгоритмом, сводится к последовательному выполнению команд в порядке, однозначно определяемом программой. **Первой** выполняется команда, заданная пусковым адресом программы. Обычно это адрес первой команды программы. Адрес **следующей** команды однозначно определяется в процессе выполнения текущей команды и может быть либо адресом следующей по порядку команды, либо адресом любой другой команды. Процесс вычислений продолжается до тех пор, пока не будет выполнена команда, предписывающая прекращение вычислений.

# АРХИТЕКТУРА НЕЙМАНОВСКИХ МАШИН



Принцип совместного хранения команд и данных в памяти компьютера



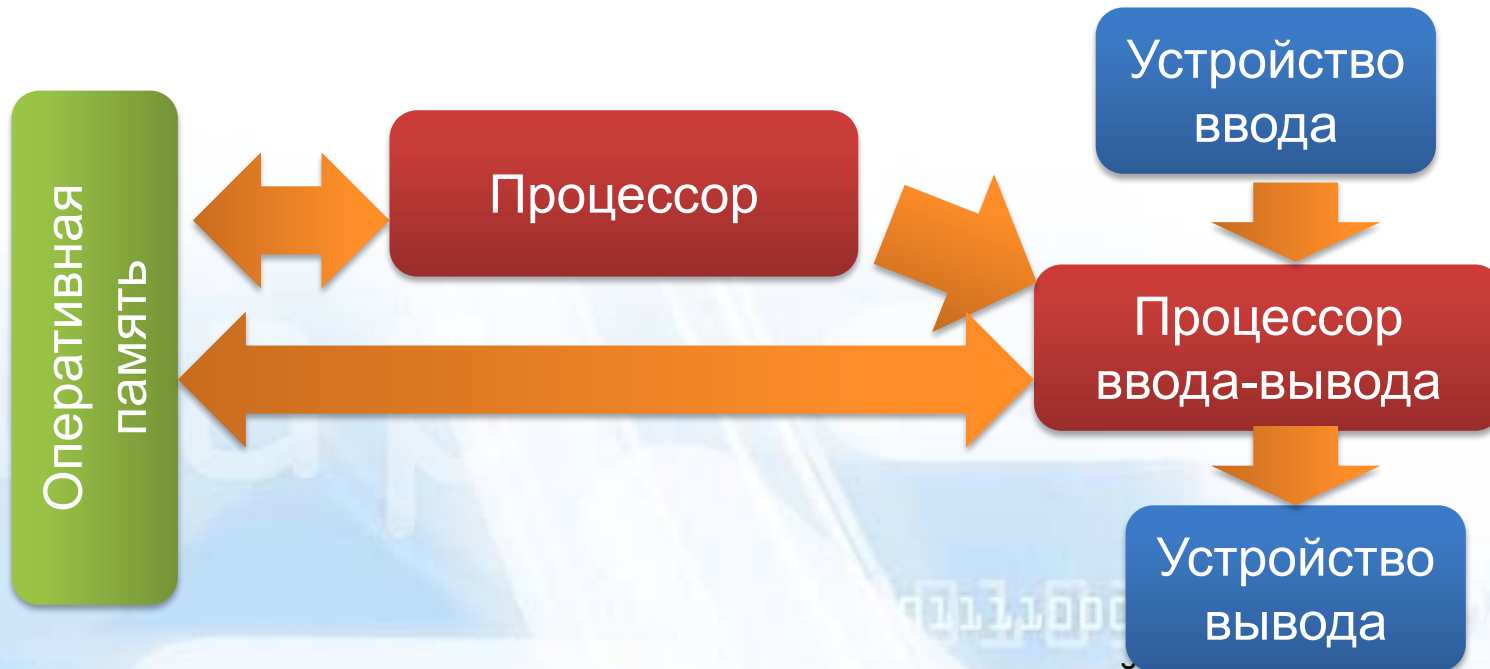
Джон фон Нейман



Основным **недостатком** неймановской архитектуры есть то, что повышение производительности вычислительной машины возможно только за счет повышения тактовой частоты. Это повышение ограничено уровнем развития технологии изготовления элементной базы компьютеров.

# НЕДОСТАТКИ НЕЙМАНОВСКОЙ АРХИТЕКТУРЫ И ПУТИ ИХ УСТРАНЕНИЯ

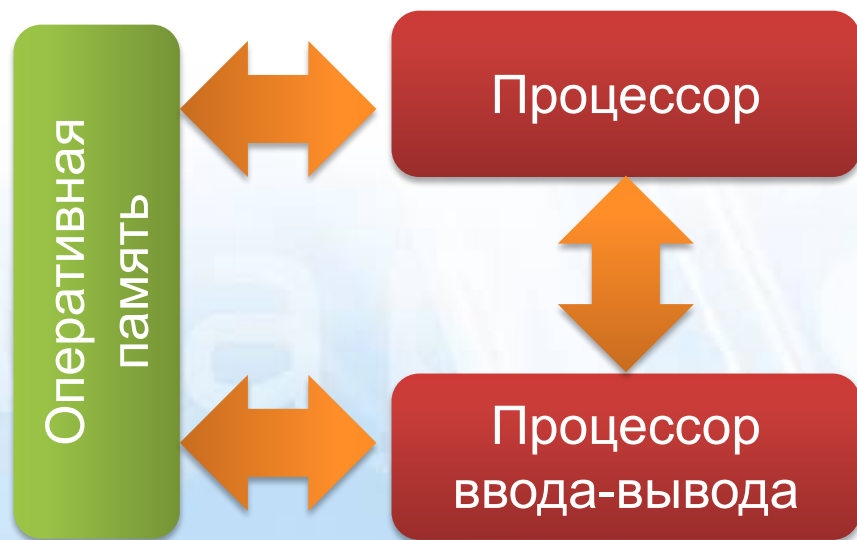
1. *Использование процессора для управления процессами ввода-вывода информации уменьшает общую производительность компьютера*, так как устройства ввода-вывода значительно медленнее по отношению к процессору и большую часть времени процессор простаивает при выполнении операций ввода-вывода.



В состав компьютера включают специализированный процессор, выполняющий исключительно только операции ввода-вывода и работающий параллельно с основным процессором. За счет такого архитектурного решения во время выполнения ввода-вывода информации не прекращается выполнение основной программы.

# НЕДОСТАТКИ НЕЙМАНОВСКОЙ АРХИТЕКТУРЫ И ПУТИ ИХ УСТРАНЕНИЯ

2. *Выполнение всех типов команд программы поочередно одна за одной приводит к необходимости загружать центральный процессор и унижает общую производительность компьютера.* При этом количество длинных (за продолжительностью выполнения) команд в программах, как правило, значительно меньше количества коротких команд.



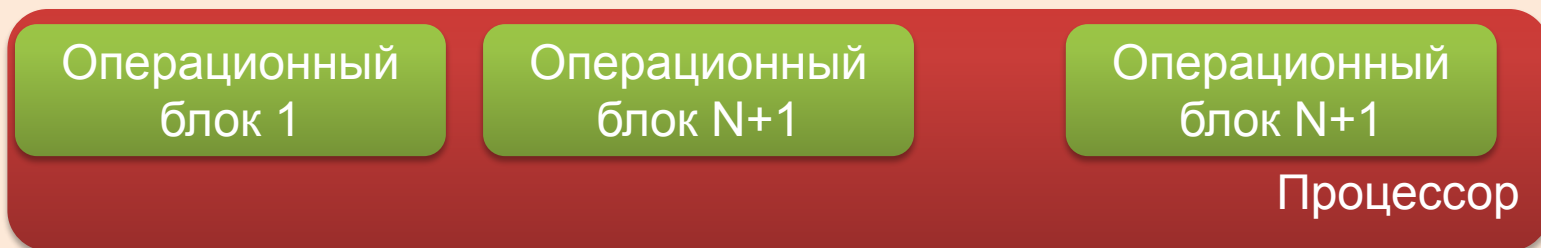
В состав компьютера могут быть включены и вторые специализированные процессоры, на пример, устройства формирования исполнительных адресов операндов, матричные вычислители и прочие.

Все эти специализированные процессоры работают параллельно с центральным процессором, за счет чего значительно повышается общая производительность компьютера.

В состав компьютера включают специализированный процессор, который выполняет длинные операции и работает параллельно с центральным процессором. Он выполняет только короткие команды и осуществляет общее управление вычислительным процессом. Специализированный процессор длинных команд называют **сопроцессором**.

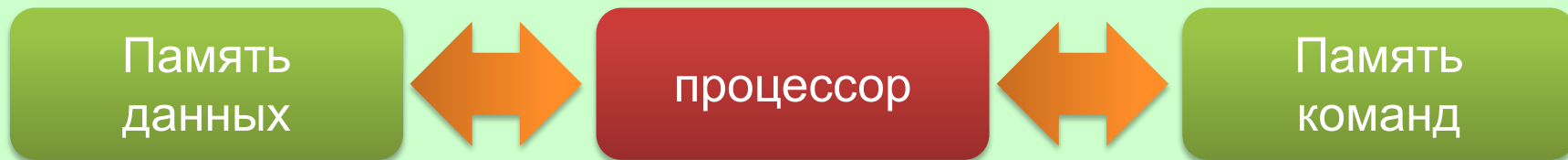
# НЕДОСТАТКИ НЕЙМАНОВСКОЙ АРХИТЕКТУРЫ И ПУТИ ИХ УСТРАНЕНИЯ

**3. Одновременно в процессоре выполняется только одна операция.** При этом оборудование процессора используется малоэффективно, так как в каждое время работает только один функциональный блок, а последние блоки простаивают.



В компьютере используют **конвейерный процессор**, который позволяет одновременно выполнять несколько операций путем максимальной загрузки всех блоков процессора.

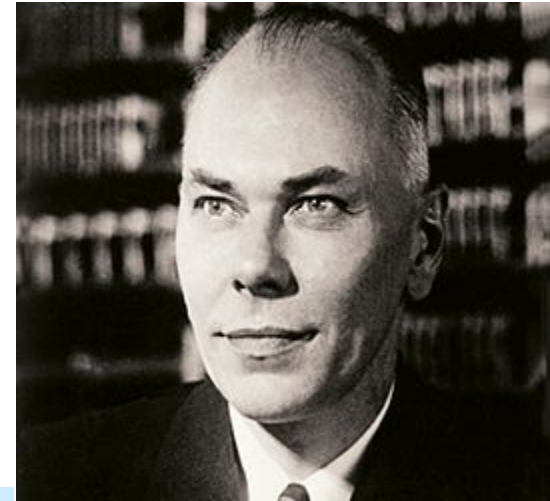
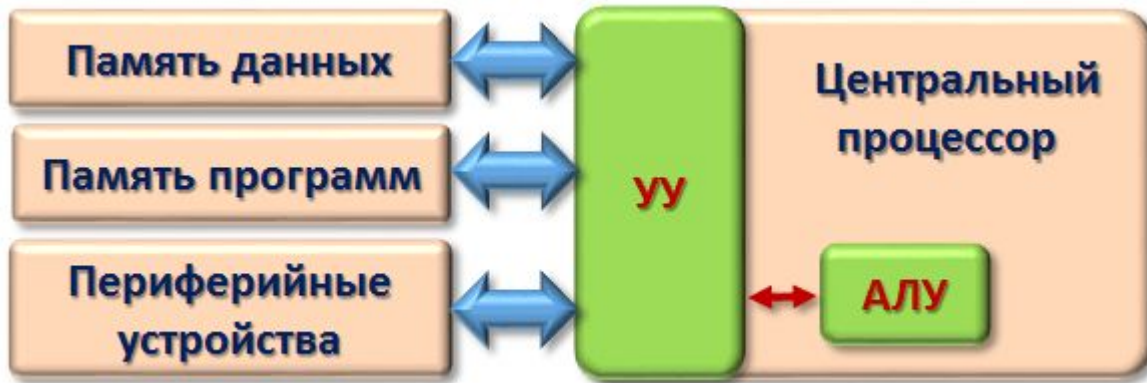
**4. Использование единой памяти для хранения информации не позволяет одновременно и взаимно независимо считывать команды и данные.** Также это приводит к значительной фрагментации последовательностей команд и данных, что затрудняет их поиск.



Общую память разделяют на две: память для хранения команд, память для хранения данных. Чтение с этих двух устройств осуществляется взаимозависимо и может проводиться одновременно.



# ГАРВАДСКАЯ АРХИТЕКТУРА



Говард Хатауэй Эйкен

**Гарвардская архитектура** — архитектура ЭВМ, отличительными признаками которой являются:

1. Хранилище инструкций и хранилище данных представляют собой разные физические устройства;
2. Канал инструкций и канал данных так же физически разделены.

**Недостатки** Гарвардской архитектуры связаны с необходимостью проведения большего числа шин, а также с фиксированным объемом памяти, выделенной для команд и данных, назначение которой не может оперативно перераспределяться в соответствии с требованиями решаемой задачи. Поэтому **приходится использовать память большего объема**, коэффициент использования которой при решении разнообразных задач оказывается более низким, чем в системах с Принстонской архитектурой.

## Отличие от архитектуры фон Неймана

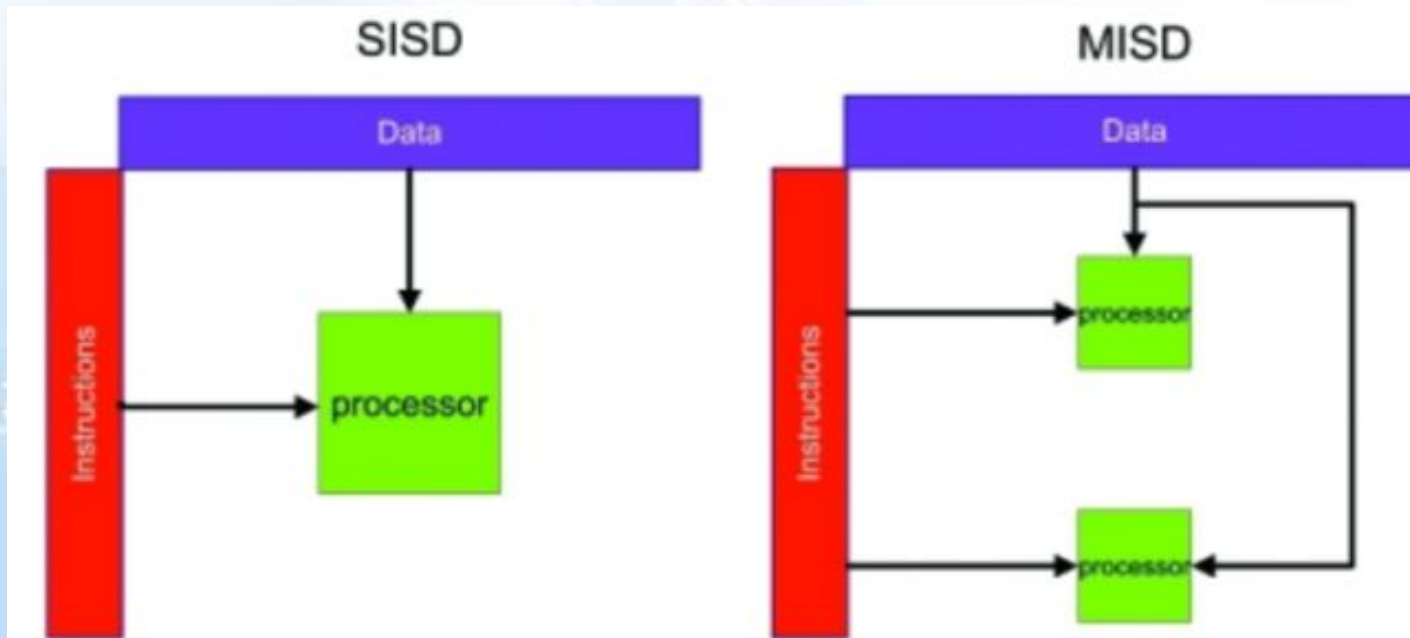
- ❑ В чистой архитектуре фон Неймана процессор в каждый момент времени может либо читать инструкцию, либо читать/записывать единицу данных из/в памяти. Оба действия одновременно происходить не могут, поскольку инструкции и данные используют один и тот же поток (*шину*).
- ❑ В компьютере с использованием гарвардской архитектуры процессор может считывать очередную команду и оперировать памятью данных одновременно и без использования кэш-памяти. Таким образом, компьютер с гарвардской архитектурой при определенной сложности схемы быстрее, чем компьютер с архитектурой фон Неймана, поскольку потоки команд и данных расположены на отдельных физически не связанных между собой аппаратных каналах.
- ❑ Исходя из физического разделения шин команд и данных, разрядности этих шин (следовательно, и ад пространства) могут различаться и *физически* не могут пересекаться.

# КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ ПО АРХИТЕКТУРЕ

Обобщенная классификация архитектур аппаратных средств компьютеров базируется на понятиях потока команд – *I* и потока данных – *D* в вычислительной структуре. При этом различают: одинарный поток – *S*; множественный поток – *M*.

**SISD** - *одинарный поток команд и одинарный поток данных.* Управление осуществляет одинарная последовательность команд, каждая с которых обеспечивает выполнение одной операции и дальше передает управление следующей команде.

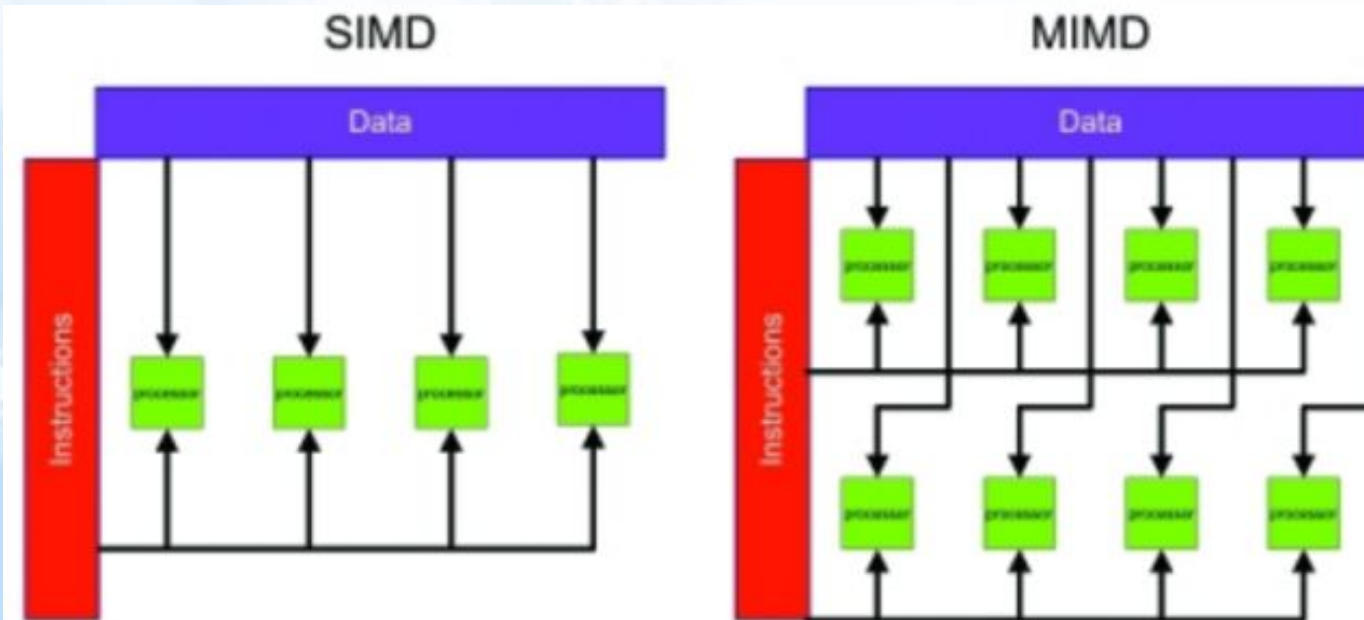
**MISD** - *множественный поток команд и одинарный поток данных.* Имеет также название конвейера обработки данных. Она представляет цепочку последовательно соединенных процессоров (микропроцессоров), управляющих параллельным потоком команд. На вход конвейера из памяти подается одинарный поток данных. Конвейерную архитектуру ЭВМ предложил академик С.А. Лебедев в 1956 году.



# КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ ПО АРХИТЕКТУРЕ

**SIMD - одинарный поток команд и множественный поток данных.** Процессор таких машин имеет матричную структуру, в узлах которой включено большое количество сравнительно простых быстродействующих процессорных элементов. Одинарный поток команд вырабатывает одно общее устройство управления. При этом все процессорные элементы выполняют одновременно одну и ту же команду, но над разными операндами, которые доставляются из памяти множественным потоком.

**MIMD - множественные потоки команд и данных.** К таким структурам относятся многопроцессорные и многомашинные вычислительные системы. Гибкость MIMD структур позволяет организовать совместную работу ЭВМ или процессоров за распараллеленную программой при решении одного сложного задачи или отдельную работу всех ЭВМ при одновременном решении множества задач независимых программ.



# ОСНОВНЫЕ ХАРАКТЕРИСТИКИ КОМПЬЮТЕРОВ

**Коэффициент готовности** – комплексный показатель ПК:

$$S = \frac{P \cdot k_H}{C_{ЭВМ} + C_{эсп.}}$$

где  $P$  – производительность компьютера;

$k_H$  – комплексный коэффициент эксплуатационной надежности (использование), что характеризует потерю производительности через нарушение надежности функционирования компьютера;

$C_{ЭВМ}$  – стоимость компьютера;

$C_{эсп.}$  – стоимость эксплуатации компьютера.

**Производительностью** компьютера называют скорость выполнения команд (операций):

$$P = \frac{\sum_{i=1}^h k_i}{h \sum_{i=1}^h k_i t_i}$$

где  $k_i$  – весовой коэффициент, который определяет относительную частоту выполнения команд  $i$ -го типа;

$t_i$  – время выполнения команд  $i$ -го типа;

$h$  – число типов команд, которые используются в компьютере.

Более простой образ определения **производительности компьютера** – количество коротких операций типа регистр-регистр, которые выполняются за одну секунду.

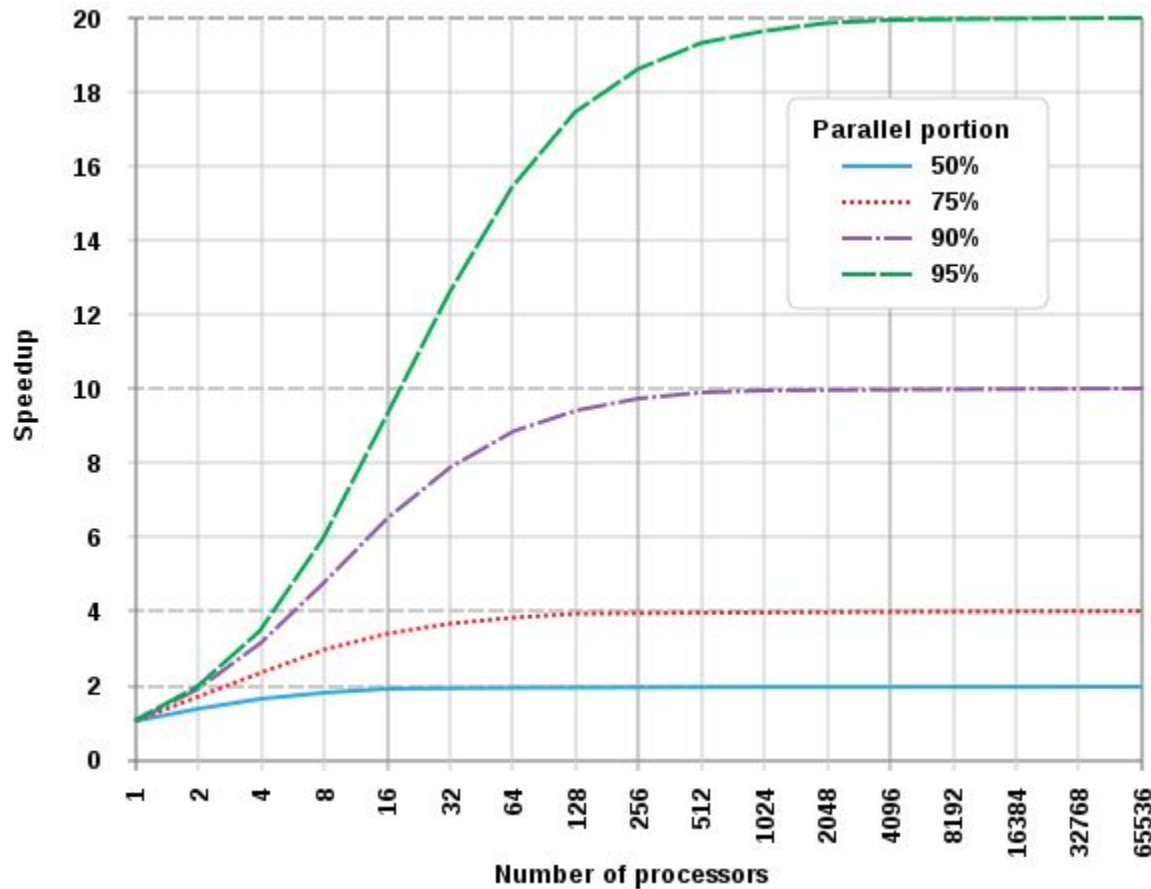
**Производительность суперкомпьютеров** определяют в мегафлопах за секунду (Мфл/с) (количеством миллионов операций с плавающей точкой за секунду).

# ЗАКОН АМДАЛА

Увеличение числа ядер никогда не дает идеального увеличения

Пусть необходимо решить некоторую вычислительную задачу. Предположим, что её алгоритм таков, что доля  $\alpha$  от общего объёма вычислений может быть получена только последовательными расчётами, а, соответственно, доля  $1 - \alpha$  может быть распараллелена идеально (то есть время вычисления будет обратно пропорционально числу задействованных узлов  $p$ ). Тогда ускорение, которое может быть получено на вычислительной системе из  $p$  процессоров, по сравнению с однопроцессорным решением не будет превышать величины  $S$ :

$$S_p = \frac{1}{\alpha + \frac{1 - \alpha}{p}}$$



Производительность любой параллельной программы ограничена частью кода, не поддающейся распараллеливанию.

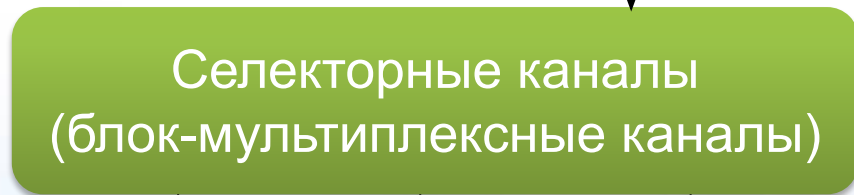
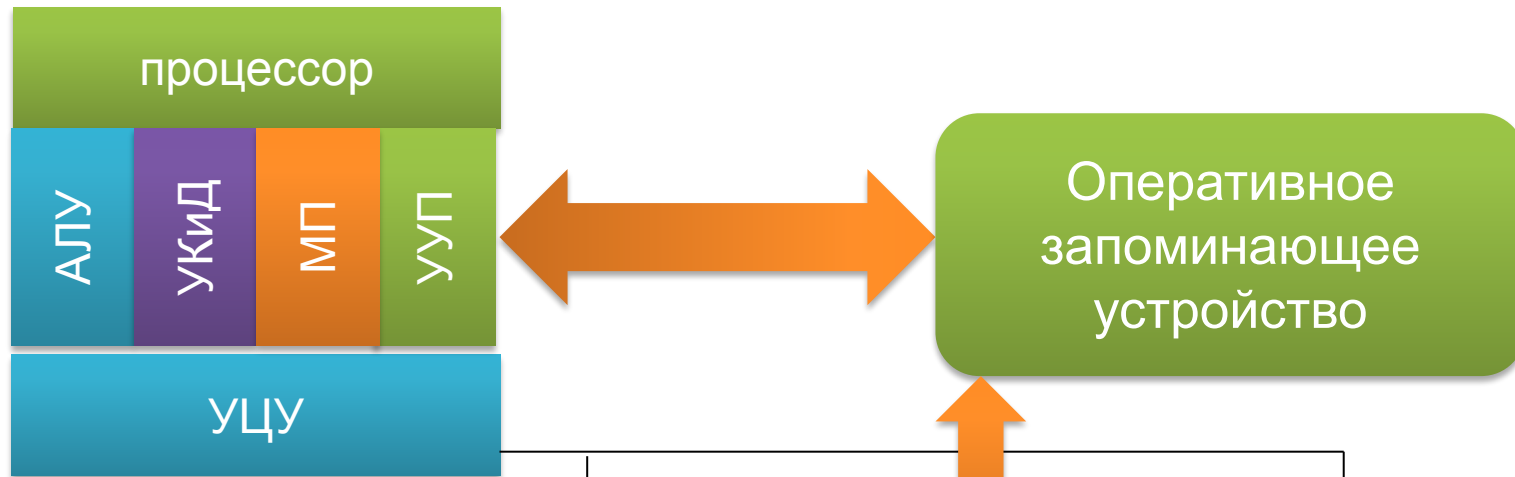
# НАЗНАЧЕНИЕ УСТРОЙСТВ БОЛЬШИХ ЭВМ

Большие универсальные ЭВМ являются высокопроизводительными машинами коллективного пользования, которые работают в качестве майнфреймов вычислительных сетей в режиме мультипрограммной обработки информации и имеют развитую систему ввода-вывода информации

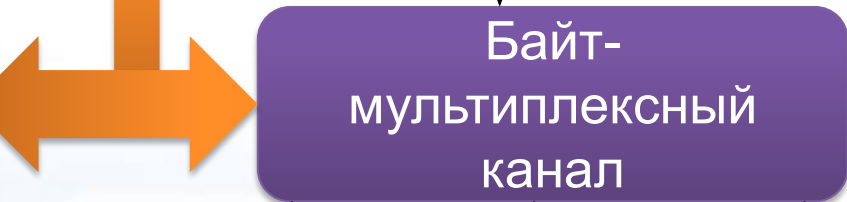
## Отличительные особенности больших универсальных ЭВМ:

- Развитая мощная система ввода-вывода, основанная на использовании специализированных процессоров (каналов) ввода-вывода и стандартизированных асинхронных интерфейсов для подключения большого количества разнообразных периферийных устройств. В качестве периферийных устройств могут использоваться и интеллектуальные абонентские пункты в виде персональных компьютеров.
- В качестве ядра, как правило, выступает многопроцессорная система очень большой мощности и большая по объему оперативная память.
- Для управления используется многопользовательские операционные системы с развитой системой администрирования.
- Для эксплуатации ЭВМ через ее большой аппаратный объем и сложность привлекается специальный штат инженеров и системных программистов. В целях наибольшего использования возможностей и ресурсов организуется круглосуточное использования ЭВМ.
- При проектировании и изготовлении машин такого класса мало внимания уделяют стоимости реализации высоких показателей. Из-за аппаратной сложности и малого количества образцов стоимость их составляет большие значения.

# АРХИТЕКТУРА БОЛЬШИХ ЭВМ



Быстродействующие периферийные ус-ва



Медленнодействующие периферийные ус-ва

# АРХИТЕКТУРА БОЛЬШИХ ЭВМ

Процессор является центральным элементом компьютера и непосредственно осуществляет преобразование информации при управлении программами. Для этого он имеет:

- ❑ **арифметико-логическое устройство (АЛУ)** выполняет арифметические и логические операции над двоичными и двоично-десятичными числами;
- ❑ **устройство центрального управления (УЦУ)** обеспечивает микропрограммное управление всего процессора, обработку прерываний и отсчет времени;
- ❑ **устройство управления памятью (УУП)** обеспечивает связь процессора и каналов ввода-вывода с оперативным запоминающим устройством (ОЗУ), разрешение конфликтов при обращении к памяти и передаваемую буферизацию информации;
- ❑ **устройство контроля и диагностики (УКиД)** обеспечивает текущий контроль функционирования компьютера, а также проверку его работоспособности при инициализации системы.
- ❑ **ОЗУ** предназначено для оперативного сохранения программ и данных.
- ❑ **Мультиплексный (байт-мультиплексный) канал** является специализированным процессором ввода-вывода и обеспечивает ввод/вывод информации из медленно действующих **периферийных устройств** (ПУ). Он работает в мультиплексном режиме, т. е. после чтения / записи одного байта информации из одного периферийного устройства возможен обмен байтом информации по другим более приоритетным устройствам.
- ❑ **Селекторные (блок-мультиплексные) каналы** предназначены для работы с быстро действующими периферийными устройствами, например, с устройствами внешней памяти. Этот канал работает в селекторном режиме, то есть, если начался обмен информацией с одним устройством, то он не может быть прерван другим устройством.
- ❑ Все периферийные устройства подключаются к каналу через свои **устройства управления (УУ)**, обеспечивающих стандартное подключение разнотипных устройств к каналам



# МИНИ И МИКРО ЭВМ. ПРИНЦИП ОБЩЕЙ ШИНЫ

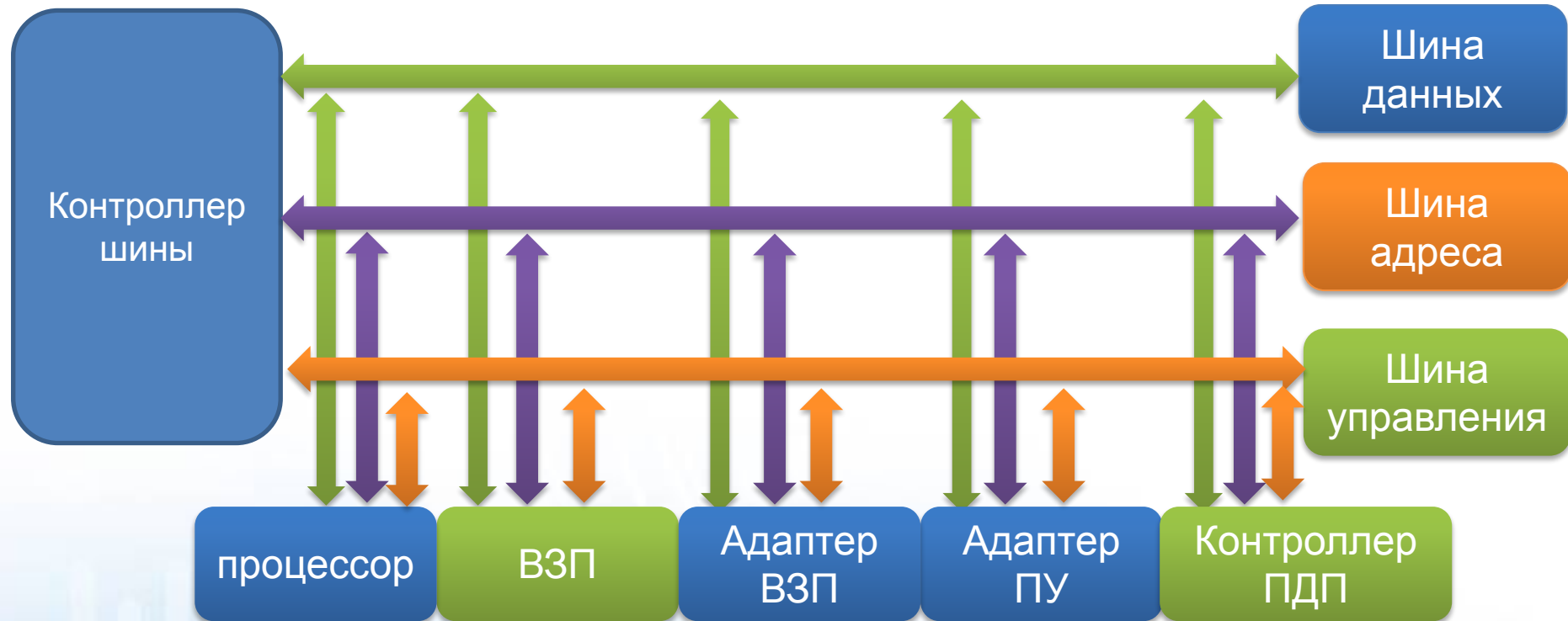
**Основное назначение мини и микро ЭВМ** - это работа в качестве управляющих устройств в автоматизированных системах управления механизмами и технологическими процессами, работающих в режиме реального времени. Такие управляющие устройства называют также контроллерами.

В качестве отдельных управляющих устройств такие контроллеры входят в состав крупных универсальных машин или персональных компьютеров.

## **По месту использования такие ЭВМ должны иметь следующие особенности:**

- уменьшена разрядность представления данных, малый набор команд и форматов представления данных. Часто используются только форматы чисел с фиксированной точкой и операционные команды алгебраического сложения и сдвига;
- специфический набор периферийного оборудования, включающий цифро-аналоговые преобразователи (ЦАП) для формирования управляющих аналоговых сигналов и аналого-цифровые преобразователи (АЦП) для ввода в компьютер исходной аналоговой информации;
- должна предусматриваться возможность стандартизованного подключения к контроллеру датчиков информации (цифровых и аналоговых) и исполнительных механизмов технологического оборудования;
- повышенные требования к эксплуатационной надежности и простоты эксплуатации;
- низкая стоимость, упрощенный порядок установления и использования;
- отсутствие системного программного обеспечения, так как такие специализированные системы работают под управлением своей уникальной программы, которая называется драйвером.

# МИНИ И МИКРО ЭВМ. ПРИНЦИП ОБЩЕЙ ШИНЫ



Основой этой архитектуры является общая шина, однотипно соединенные модули компьютера между собой. Этот системный интерфейс состоит из трех шин: **шины адреса**, **шины данных** и **шины управления**.

**Контроллер прямого доступа к памяти** (ПДП) выполняет роль управляющего устройства при обмене данными между любым **периферийным устройством** и оперативной памятью и представляет в данной структуре основу системы ввода-вывода информации. При этом **процессор** освобождается от управления процедурами ввода-вывода.

Блоки **внешней запоминающей памяти** (ВЗП) и **периферийные устройства** (ПУ) подключаются к системному интерфейсу через **адаптеры**, обеспечивающие единый интерфейс и единые процедуры управления разнотипными устройствами.

# МИНИ И МИКРО ЭВМ. ПРИНЦИП ОБЩЕЙ ШИНЫ

Такая архитектура носит название *открытой архитектуры*, так как позволяет практически безгранично расширять количество подключенных к ней устройств.

Она также является *модульной архитектурой*, так как отдельные устройства, которые включены в компьютер, имеют характер самостоятельного модуля и могут подключаться и отключаться к общей шине без изменения ее характеристик и работоспособности. С другой стороны каждое устройство строится как отдельный агрегат по своим принципам согласно назначению, но обеспечивает стандартное внешнее подключение к общей шине.

*Модульная открытая архитектура характерна вообще для большинства современных компьютерных систем.*

# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

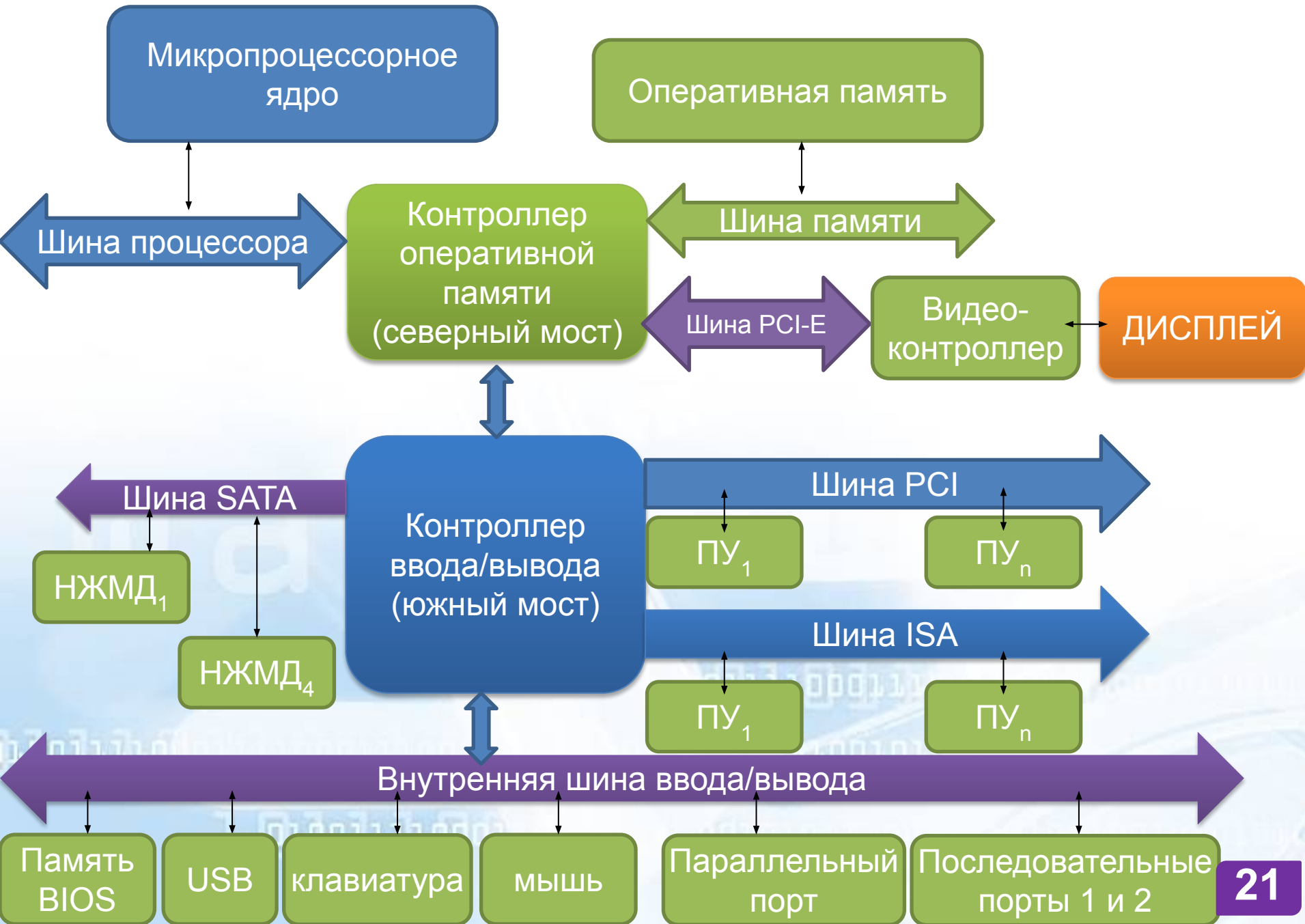
Типовая архитектура персональных компьютеров строится по открытой архитектуре по типу общей шины с использованием свойств модульности структуры.

Желание повысить производительность привело к усложнению структуры машины за счет использования не одной а целого набора шин, имеющих различные показатели по разрядности и пропускной способности. При этом система шин имеет четко выраженную иерархическую структуру. Наиболее мощные шины расположены в ядре ЭВМ, а менее мощные служат для связи ядра ЭВМ с периферийными устройствами.

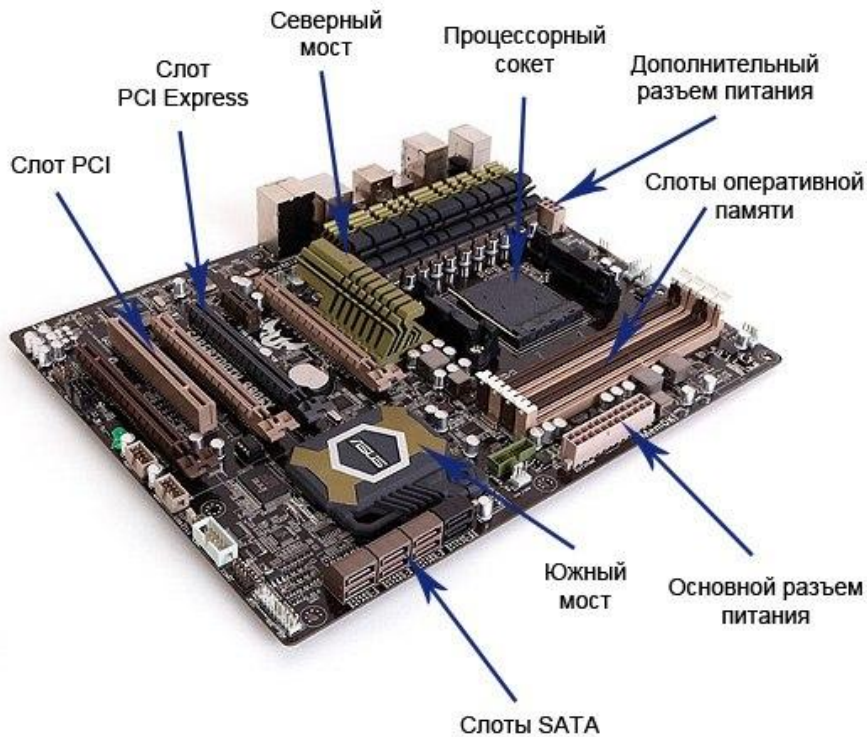
## Отличительными свойствами персональных компьютеров является:

- высокая надежность и простота эксплуатации, не требует от пользователя глубоких знаний по ее построению и работы;
- широкая возможность изменения конфигурации в зависимости от требований пользователя и класса задач, которые решаются с помощью компьютера;
- простота и стандартизация процесса подключения к компьютеру периферийных устройств;
- развитая операционная система с мощной справочной системой и доступным с точки зрения пользователя интерфейсом между машиной и человеком;
- стандартизация программного обеспечения, что позволяет выполнять одни программные продукты на компьютерах разных производителей;
- низкая стоимость компьютеров, что может делать их доступными для широкого круга пользователей.

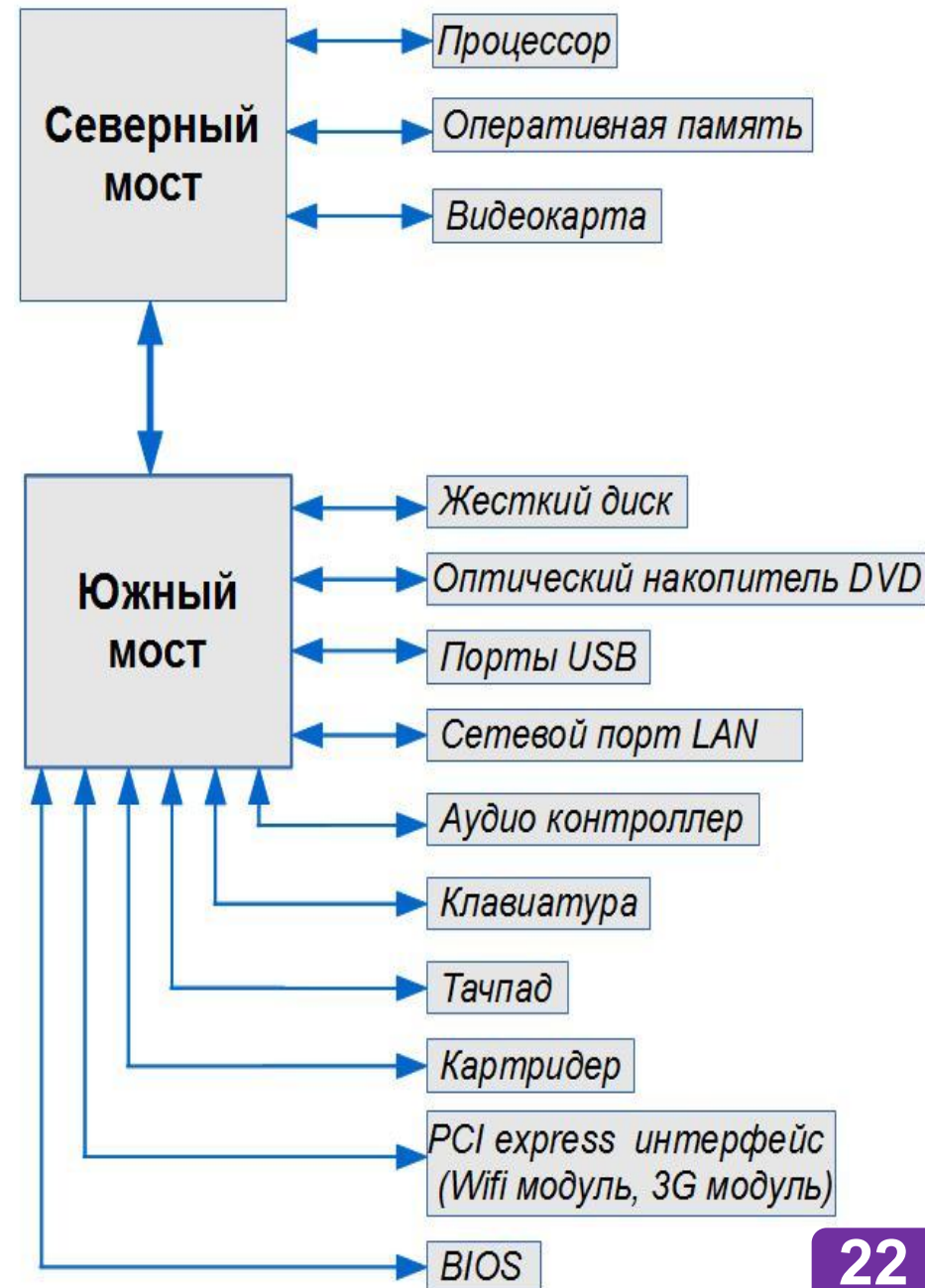
# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА



# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА



**Южный мост (South bridge)** - это функциональный контроллер, известен как контроллер ввода-вывода или **ICH** (In/Out Controller Hub). В современных ноутбуках южный мост входит в состав **PCN** (Platform Controller Hub) или **FCH** (Functional Controller Hub)



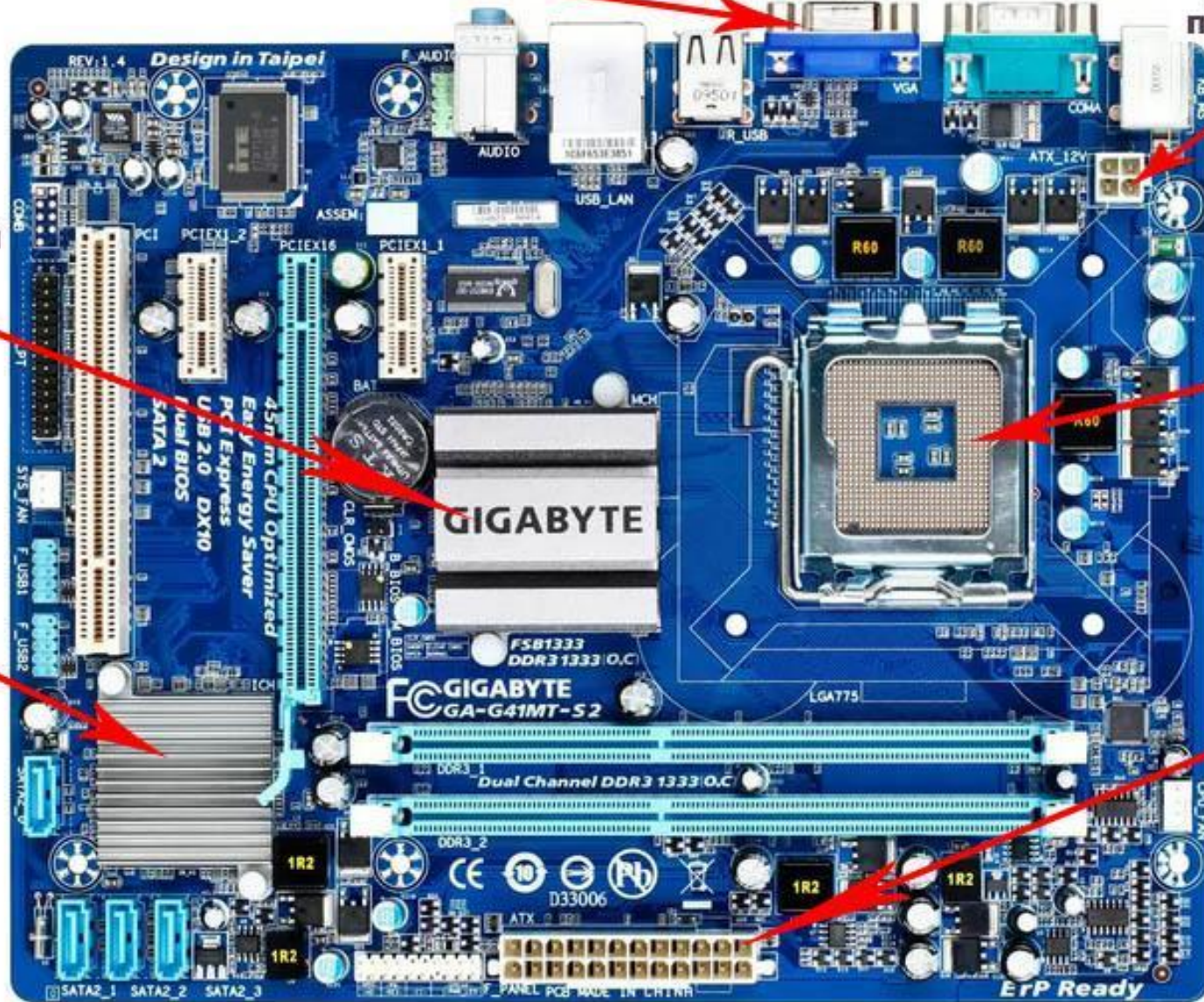
# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Разъем для подключения монитора

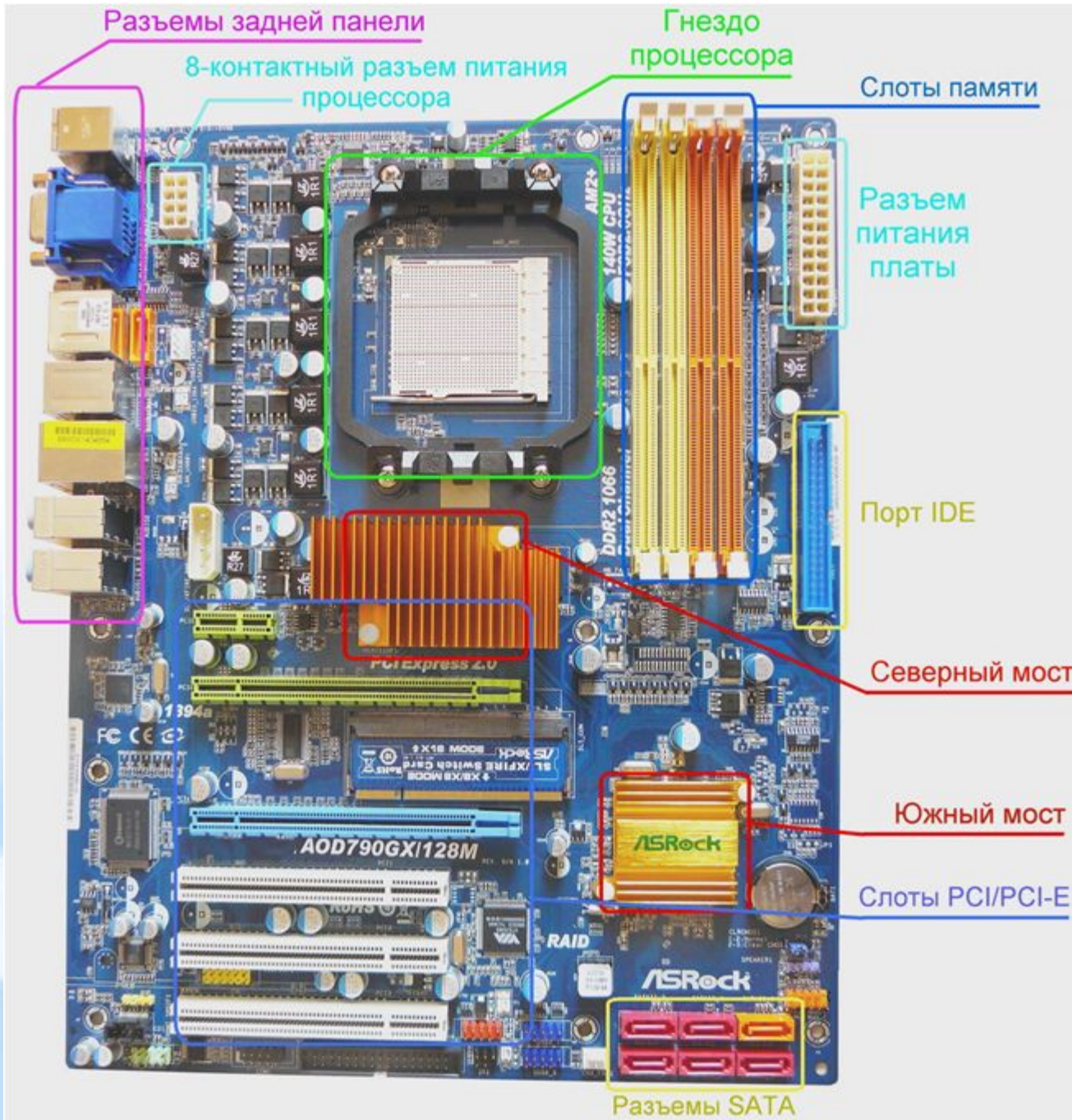
Питание процессора 12V

Разъем центрального процессора

Питание материнской платы



# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА





# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА



# ТИПОВАЯ АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Основой компьютера служит **материнская плата**, на которой расположены две микросхемы коммутации шин и контроллеров ввода-вывода информации.

Первая микросхема называется **северным мостом** и содержит контроллер оперативной памяти, включающей контроллер прямого доступа к ОП, и коммутатор, или мост между системной шиной процессора, шиной оперативной памяти, шиной для подключения видео контроллера типа AGP, и периферийной шиной PCI. **Южный мост** с подключенными к нему устройствами составляет ядро машины.

Вторая микросхема называется **южным мостом** и служит для подключения к ядру компьютера разнообразных периферийных устройств. На этой микросхеме расположены также контроллеры накопителей на жестких и гибких дисках, а также внутренняя периферийная шина ввода-вывода, которая используется для подключения медленно действующих периферийных устройств, как клавиатура, мышь, последовательные и параллельные порты, порты USB, а также другие устройства.

Соединения и согласование работы различных шин компьютера обеспечивается либо с помощью мостов между шинами, или путем использования концентраторов.

Наличие шин с разной пропускной способностью и разрядностью говорит о том, что между ними обязательно есть развязка, посредством которой могут использоваться шинные формирователи и согласительные буферы (буфер адреса, буфер данных).

Согласование передачи информации между шинами в чипсетах типа Intel строится по принципу использования концентратора. В чипсетах типа Apollo эта задача решается с помощью мостовых схем.