

## Лекция 1. Назначение периферийных устройств и интерфейсов

Предмет, цели и задачи курса «ИПУ»,  
рекомендации по литературе – см. Программу курса

[12\\_100229\\_1\\_102149.pdf](#)

[12\\_100229\\_1\\_102149.pdf](#)

Архитектура ПЭВМ и ее подсистемы ввода-вывода.

Сигналы, протоколы, транзакции, арбитраж.

Классификация периферийных устройств.

Классификация интерфейсов, основные характеристики.



Проф. Дудкин Александр Арсентьевич, д.т.н., проф.

doudkin@newman.bas-net.by

Объединенный институт проблем информатики Национальной академии наук Беларуси (ОИПИ НАН Беларуси),  
зав. лабораторией идентификации систем

Другие дисциплины: СиФО ЭВМ, АП ВМиС

Научные интересы:

автоматизация логического проектирования дискретных устройств

цифровая обработка изображений и распознавание образов

системы компьютерного зрения

К.т.н. - Синтез быстродействующих дискретных устройств на программируемых логических матрицах

Д.т.н. - Методы обработки и анализа цифровых изображений топологических слоев интегральных микросхем

# Предмет, цели и задачи курса

Цель учебной дисциплины: теоретическая и практическая подготовка, обеспечивающая получение знаний о составе и принципах функционирования ПУ ВС, методах их управления на основе широко используемых периферийных интерфейсов; получение практических навыков разработки аппаратно-программных решений доступа к периферийным устройствам; получение навыков использования периферийных интерфейсов к решению практических задач.

Задачи дисциплины:

- подготовка специалиста, уверенно владеющего разнообразными возможностями, предоставляемыми современными компьютерными технологиями для построения и управления периферийным оборудованием вычислительных систем;
- обучение студентов принципам передачи и приема информации, в том числе с использованием беспроводных интерфейсов;
- формирование навыков организации вычислительных систем с применением современного периферийного оборудования и интерфейсов

# Предмет, цели и задачи курса

## **знать:**

принципы передачи и приема информации через параллельный и последовательный порты;

принципы работы с портами устройств ПЭВМ;

принципы управления блоками и узлами ПЭВМ;

структуру периферийных устройств и интерфейсов;

принципы функционирования периферийных устройств;

## **уметь:**

программировать и взаимодействовать с периферийными устройствами на уровне интерфейсов;

выбирать требуемый интерфейс передачи данных в зависимости от условий;

решать прикладные задачи подключения периферийных устройств к ПЭВМ;

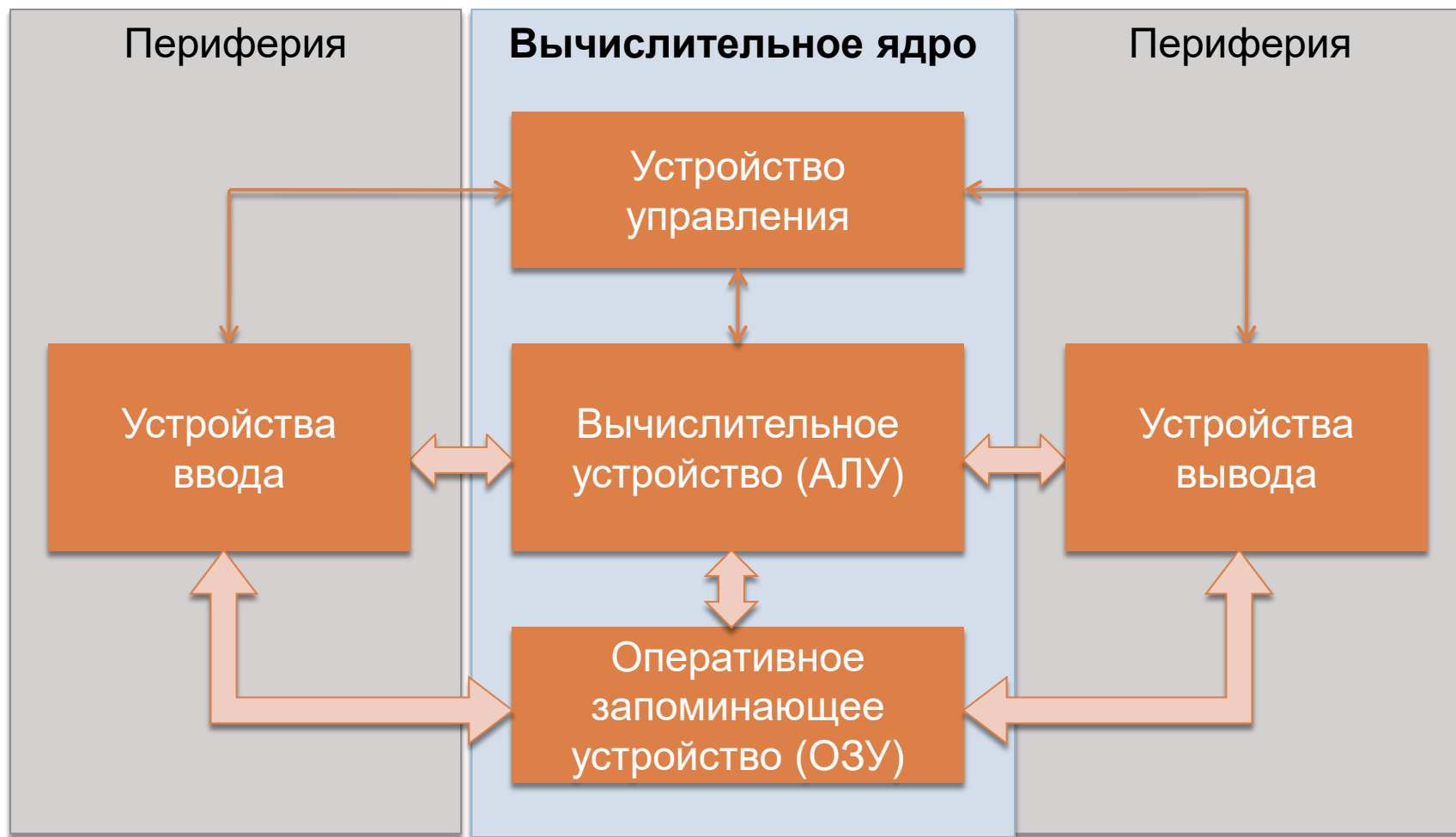
## **владеть:**

навыками сопряжения основных видов периферийных устройств и ПЭВМ;

протоколами обмена данных;

навыками организации интерфейсов на физическом уровне.

# Архитектура ПЭВМ



# Архитектура ПЭВМ

## Принципы концепции ВМ фон Неймана

- Двоичного кодирования.
- Однородности памяти.
- Адресуемости памяти.
- Последовательного программного управления.
- Жесткости архитектуры.

## Архитектура ЭВМ фон Неймана

- последовательно адресуемая единственная память линейного типа для хранения программ и данных;
- команды и данные различаются через идентификатор неявным способом лишь при выполнении операций (позволяют обращаться с командой как с данными, например, для ее модификации);
- назначение данных определяется лишь логикой программы, так как в памяти машины набор бит может представлять собой как десятичное число с фиксированной точкой, так и строку символов.

# Состав ПЭВМ: Ядро и периферия

В составе ПЭВМ (или узла более сложной системы) можно выделить *вычислительное ядро* и его *периферию*.

**Ядро** обычно состоит из АЛУ, выполняющего также некоторые из задач управления, и ОЗУ.

В современных ВМ большинство принципов фон Неймана не соблюдены:

- микропроцессоров может быть несколько, и каждый обладает поддержкой многопоточности (содержит реальные или виртуальные симметричные вычислительные модули),
- суперскалярность (выполнение нескольких инструкций одновременно), MISD, MIMD, переупорядочивания команд, поддержкой SIMD-инструкций.
- ОЗУ – это комбинация контроллера памяти и микросхем памяти.

# Периферия

Все устройства, не входящие в вычислительное ядро (ядра), относятся к **периферийным**. Они могут располагаться снаружи / внутри корпуса ЭВМ, а также входить в состав основных микросхем системы.

Основная задача периферийных устройств – поставка данных на обработку, а также вывод их за пределы вычислительного ядра. Данная задача охватывает такие процессы, как оцифровка и преобразование данных в электрическую форму (из оптической, механической, электромагнитной и т.д.), регистрация различных внешних воздействий, преобразование данных, сохранение на внешних носителях, изготовление «твердой копии» на бумаге, передача по каналам связи, отображение в графической форме на экране и т.д.

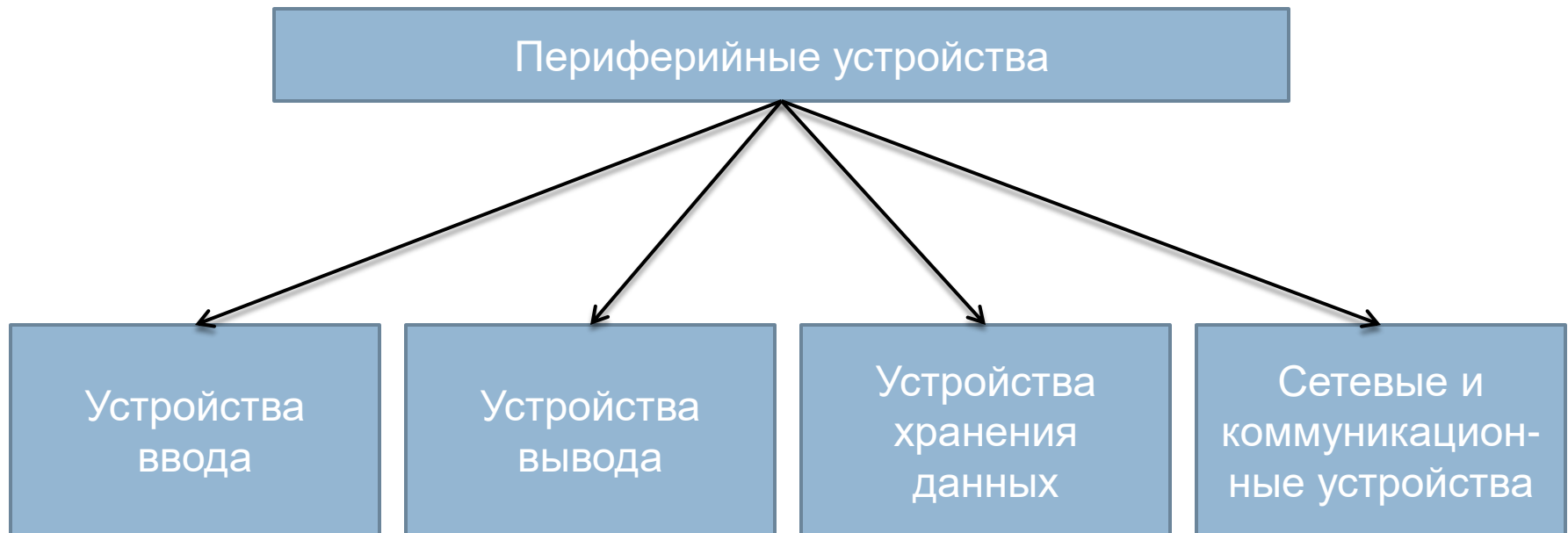
Можно выделить отдельный класс устройств *управления и обслуживания системы (system management and control)*, которые по назначению неправомерно относить к периферийным. Однако по принципу действия они являются именно периферийными.



# Классификация ПУ

Первоначальное деление периферийных устройств по назначению на устройства ввода и вывода неполно описывает их особенности.

Общепринятым и более полным является деление на 4 базовых класса:



# Классификация ПУ

**Устройства ввода** предназначены для преобразования информации некоторой физической природы в электрические сигналы, пригодные для обработки ядром системы.

**Устройства вывода** оформляют информацию, обработанную ядром системы, таким образом, что она становится пригодной для обработки человеком или другой системой.

**Устройства хранения данных** обеспечивают хранение и последующую загрузку машинного кода и/или данных. По сути они расширяют объем оперативной памяти системы, но, в отличие от нее, не обеспечивают непосредственного доступа со стороны процессора.

**Сетевые и коммуникационные устройства** (Network & Communication) выполняют передачу данных между вычислительными системами, минуя промежуточные носители информации. Как правило, передача выполняется на большие расстояния, не сравнимые с размерами самой ЭВМ, и требует затрат на кодирование для защиты от искажений, помех, потери, злонамеренного перехвата и т.п.

# Другие критерии классификации

ПУ можно также классифицировать по другим признакам, например, по **конструктивному исполнению**:

- **Внешние** – имеющие свой корпус и (зачастую) отдельный источник питания (если питание по интерфейсу не предусмотрено или его мощности недостаточно).
- **Внутренние** – расположенные внутри корпуса системы и питающиеся от системного блока питания или интерфейса.
- **Встроенные** – расположенные на системной (материнской) плате или являющиеся частью одной из микросхем на этой плате.

Общепринятой является классификация по **основной функции ПУ**. При этом в одном физическом корпусе могут объединяться несколько устройств различного класса (монитор со встроенными колонками и камерой, клавиатура с IP-телефоном, МФУ с принтером, сканером и факс-аппаратом и т.д.).

Каждому классу устройств присущ свой набор характеристик.

# Определение понятия «шина»

ПУ и системные компоненты ЭВМ соединяются друг с другом посредством средств подключения, организованных по иерархическому принципу.

Средства (аппаратные и программные), используемые для соединения двух компонентов или систем, называются **интерфейсом**.

Система шин,

иерархия шин оптимизирована под определенный вид коммуникаций.

# Определение понятий «шина», «транзакция шины». Целевое назначение шин. Иерархия шин.

Чтобы охарактеризовать конкретную шину, нужно описать:  
совокупность сигнальных линий;  
физические, механические и электрические характеристики шины;  
используемые сигналы арбитража, состояния, управления и синхронизации;  
правила взаимодействия подключенных к шине устройств (протокол шины).

Операции на шине называют *транзакциями*. Основные виды транзакций - *транзакции чтения* и *транзакции записи*. Если в обмене участвует устройство ввода/вывода, можно говорить о *транзакциях ввода и вывода*, по сути эквивалентных транзакциям чтения и записи соответственно. Шинная транзакция включает в себя две части: посылку адреса и прием (или посылку) данных.

# Параметры, характеризующие шину



- совокупность сигнальных линий;
- физические, механические и электрические характеристики шины;
- используемые сигналы арбитража, состояния, управления и синхронизации;
- правила взаимодействия подключенных к шине устройств (протокол шины).

# Типы шин

Важным критерием, определяющим характеристики шины, может служить ее целевое назначение.

- шины «процессор-память»;

Шина переднего плана (FSB - Front-Side Bus, обеспечивает непосредственную связь между ЦП и ОП). Это Системная шина, но эффективнее – отдельная шина (для связи ЦП-кэш) - *шина заднего плана* — BSB (Back-Side Bus).

- шины ввода/вывода;

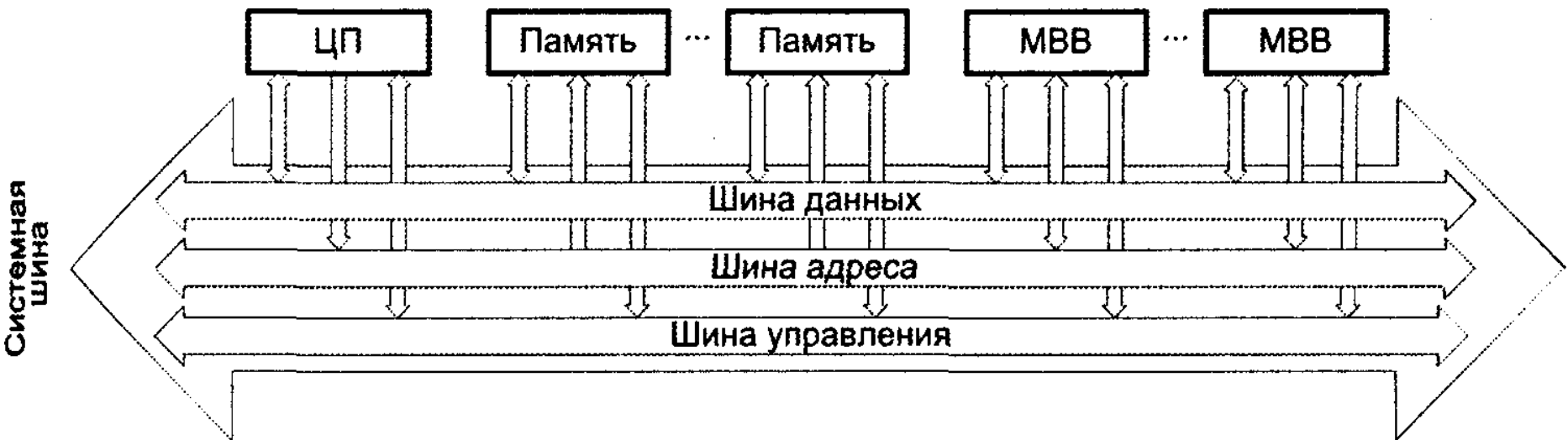
не требуют от шины высокой пропускной способности (PCI, SCSI).

# Структура системной шины

С целью снижения стоимости некоторые ВМ имеют общую шину для памяти и устройств ввода/вывода. Такая шина часто называется системной.

*Системная шина* служит для физического и логического объединения всех устройств ВМ.

3 функциональные группы

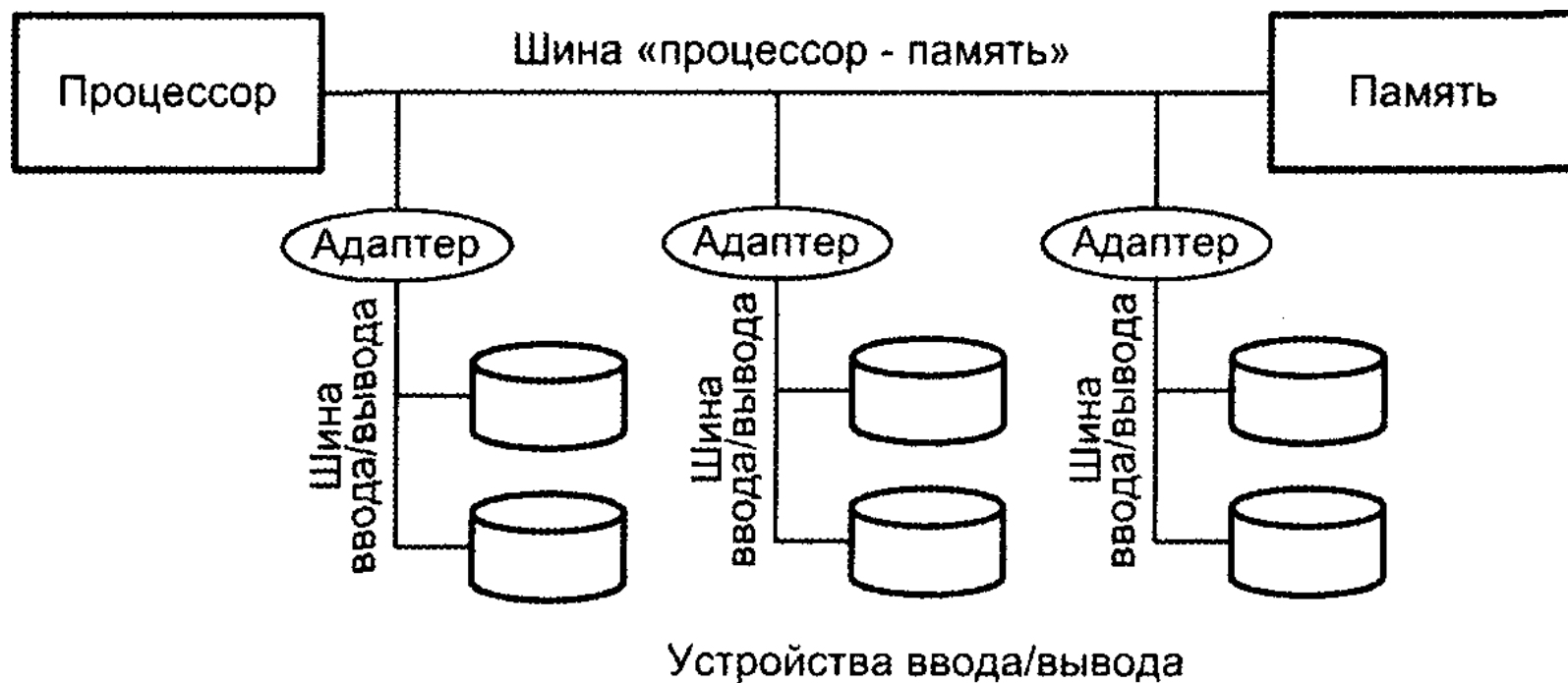
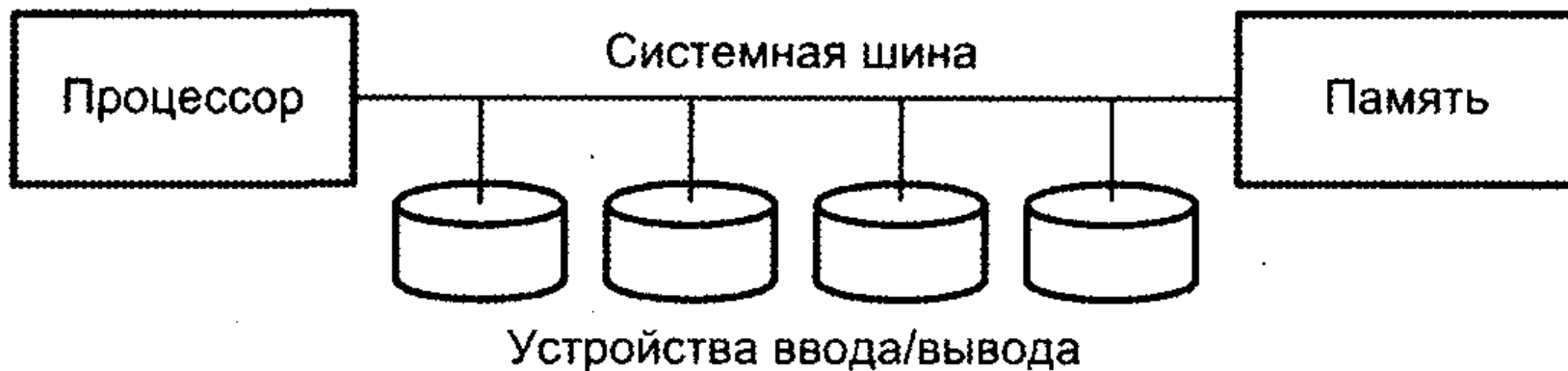


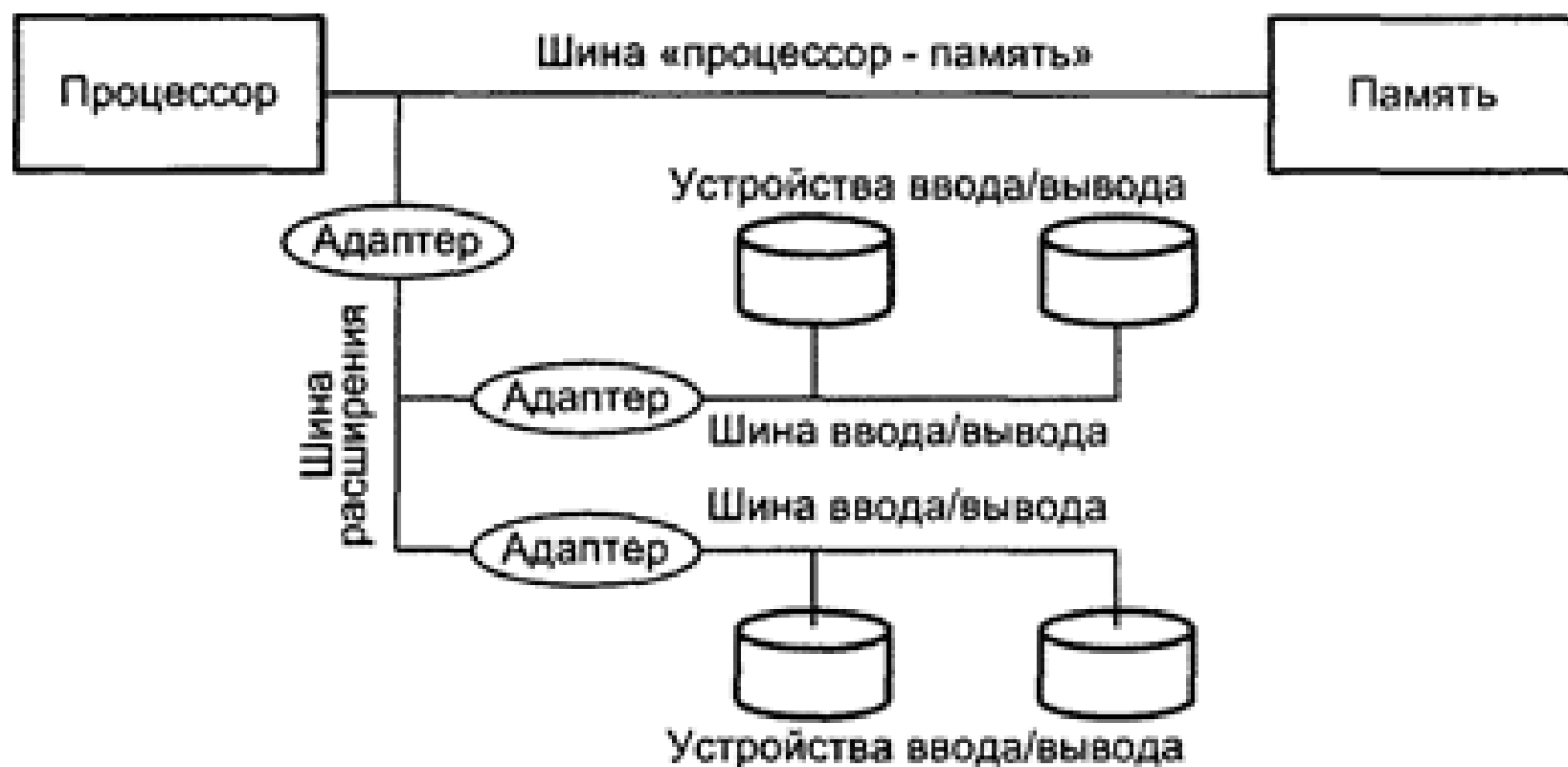


# Иерархия шин

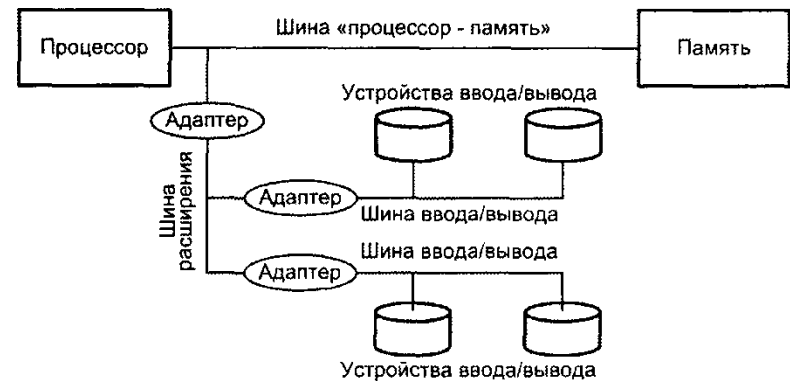
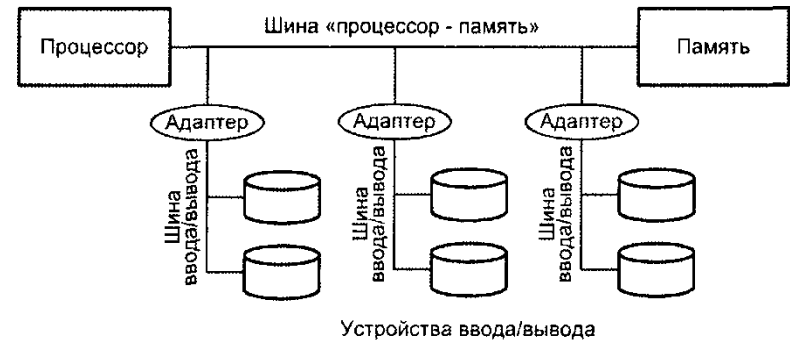
Если к шине подключено большое число устройств, ее пропускная способность падает, поскольку слишком частая передача прав управления шиной от одного устройства к другому приводит к ощутимым задержкам. По этой причине во многих ВМ предпочтение отдается использованию нескольких шин, образующих определенную иерархию. Сначала рассмотрим ВМ с одной шиной.

# Иерархия шин





# Вычислительная машина с 1, 2, 3 шиной



Понятие протокола шины.  
Последовательность действий в процедуре  
квитирования установления связи.  
Синхронные и асинхронные шины,  
достоинства и недостатки. Методы  
повышения эффективности шин.

**Протоколы** - строго заданная процедура или совокупность правил, определяющая способ выполнения определенного класса функций соответствующими СВТ.

Практически любой интерфейс содержит больше или меньше элементов протокола, определяемых процедурными и функциональными интерфейсами.

# Режимы интерфейсов

- режим 1 - однонаправленный ввод/вывод с квитированием;
- режим 2 - двунаправленный ввод/вывод с квитированием.

Квитирование, как известно, позволяет вести асинхронный обмен с учетом готовности абонента к передаче, т.е. иметь переменный темп обмена соответственно возможностям внешнего устройства.

# Синхронные шины

Синхронные шины достаточно удобно использовать благодаря дискретным временным интервалам, но здесь все же есть некоторые проблемы. Например, если процессор и память способны закончить передачу за 3.1 цикла, они вынуждены продлить ее до 4 циклов, поскольку неполные циклы запрещены.

Еще хуже то, что если однажды был выбран определенный цикл шины и в соответствии с ним были разработаны память и карты ввода/вывода, то в будущем трудно делать технологические усовершенствования. Если синхронная шина соединяет ряд устройств, один из которых работают быстро, а другие медленно, шина подстраивается под самое медленное устройство, а более быстрые не могут использовать свой полный потенциал.

# Асинхронные шины

асинхронные шины, то есть шины без задающего генератора.

Здесь ничего не привязывается к генератору. Когда задающее устройство устанавливает адрес и любой другой требуемый сигнал, он выдает специальный сигнал, который мы будем называть MSYN (Master SYNchronization). Когда подчиненное устройство получает этот сигнал, оно начинает выполнять свою работу настолько быстро, насколько это возможно. Когда работа закончена, устройство выдает сигнал SSYN(Slave SYNchronization).



# Транзакции

Операции на шине называют *транзакциями*.

Основные виды транзакций - *транзакции чтения* и *транзакции записи*.

Если в обмене участвует устройство ввода/вывода, можно говорить о *транзакциях ввода и вывода*, по сути эквивалентных транзакциям чтения и записи соответственно. Шинная транзакция включает в себя две части: посылку адреса и прием (или посылку) данных.

Когда два устройства обмениваются информацией по шине, одно из них должно инициировать обмен и управлять им. Такого рода устройства называют *ведущими* (bus master). В компьютерной терминологии «ведущий» — это любое устройство, способное взять на себя владение шиной и управлять пересылкой данных. Ведущий не обязательно использует данные сам. Он, например, может захватить управление шиной в интересах другого устройства. Устройства, не обладающие возможностями инициирования транзакции, носят название *ведомых* (bus slave).

# Арбитраж

В принципе к шине может быть подключено несколько потенциальных ведущих, но в любой момент времени активным может быть только один из них: если несколько устройств передают информацию одновременно, их сигналы перекрываются и искажаются. Для предотвращения одновременной активности нескольких ведущих в любой шине предусматривается процедура допуска к управлению шиной только одного из претендентов (арбитраж). В то же время некоторые шины допускают широковещательный режим записи, когда информация одного ведущего передается сразу нескольким ведомым (здесь арбитраж не требуется). Сигнал, направленный одним устройством, доступен всем остальным устройствам, подключенным к шине.


# Схемы приоритетов

Известны разные схемы приоритетов.

Каждому потенциальному ведущему присваивается определенный уровень приоритета, который может оставаться неизменным (**статический** или *фиксированный* приоритет)


либо изменяться по какому-либо алгоритму (**динамический** приоритет).

Основной недостаток *статических приоритетов* в том, что устройства, имеющие высокий приоритет, в состоянии полностью блокировать доступ к шине устройств с низким уровнем приоритета. Системы с *динамическими приоритетами* дают шанс каждому из запросивших устройств рано или поздно получить право на управление шиной, то есть в таких системах реализуется принцип равнодоступности.




Наибольшее распространение получили следующие алгоритмы  
**динамического** изменения приоритетов:

*простая циклическая смена приоритетов,  
циклическая смена приоритетов с учетом последнего запроса,  
смена приоритетов по случайному закону,  
схема равных приоритетов,  
алгоритм «наиболее давнего» использования.*



В алгоритме *простой циклической смены приоритетов* после каждого цикла арбитража все приоритеты понижаются на один уровень, при этом устройство, имевшее ранее низший уровень приоритета, получает наивысший приоритет.


В схеме *циклической смены приоритетов с учетом последнего запроса* все возможные запросы упорядочиваются в виде циклического списка. После обработки очередного запроса обслуженному ведущему назначается низший уровень приоритета. Следующее в списке устройство получает наивысший приоритет, а остальным устройствам приоритеты назначаются в убывающем порядке, согласно их следованию в циклическом списке. В обеих схемах циклической смены приоритетов каждому ведущему обеспечивается шанс получить шину в свое распоряжение, однако большее распространение получил второй алгоритм.



При смене приоритетов по случайному закону после очередного цикла арбитража с помощью генератора псевдослучайных чисел каждому ведущему присваивается случайное значение уровня приоритета.

В схеме *равных приоритетов* при поступлении к арбитру нескольких запросов каждый из них имеет равные шансы на обслуживание. Возможный конфликт разрешается арбитром. Такая схема принята в асинхронных системах.


В алгоритме «*наиболее давнего*» использования после каждого цикла арбитража наивысший приоритет присваивается ведущему устройству, которое дольше чем другие не использовало шину.



Помимо рассмотренных существует несколько алгоритмов смены приоритетов, которые не являются чисто динамическими, поскольку смена приоритетов происходит не после каждого цикла арбитража. К таким алгоритмам относятся *алгоритм очереди* и *алгоритм фиксированного кванта времени*.

В *алгоритме очереди* запросы обслуживаются в порядке очереди, образовавшейся к моменту начала цикла арбитража. Сначала обслуживается первый запрос в очереди, то есть запрос, поступивший раньше остальных. Аппаратная реализация алгоритма связана с определенными сложностями, поэтому используется такой алгоритм редко.


В *алгоритме фиксированного кванта времени* каждому ведущему для захвата шины в течение цикла арбитража выделяется определенный квант времени. Если ведущий в этот момент не нуждается в шине, выделенный ему квант остается не использованным. Такой метод наиболее подходит для шин с синхронным протоколом.



Арбитраж запросов на управление шиной может быть организован по *централизованной* или *децентрализованной* схеме. Выбор конкретной схемы зависит от требований к производительности и стоимостных ограничений.

При **централизованном арбитраже** в системе имеется специальное устройство – центральный арбитр, которое ответственно за предоставление доступа к шине только одному из запросивших доступ ведущих устройств. Это устройство, называемое иногда центральным контроллером шины, может быть самостоятельным модулем или частью центрального процессора. Наличие на шине только одного арбитра означает, что в централизованной схеме имеется единственная точка отказа. В зависимости от того, каким образом ведущие устройства подключены к центральному арбитру, возможны *параллельные* и *последовательные* схемы централизованного арбитража.





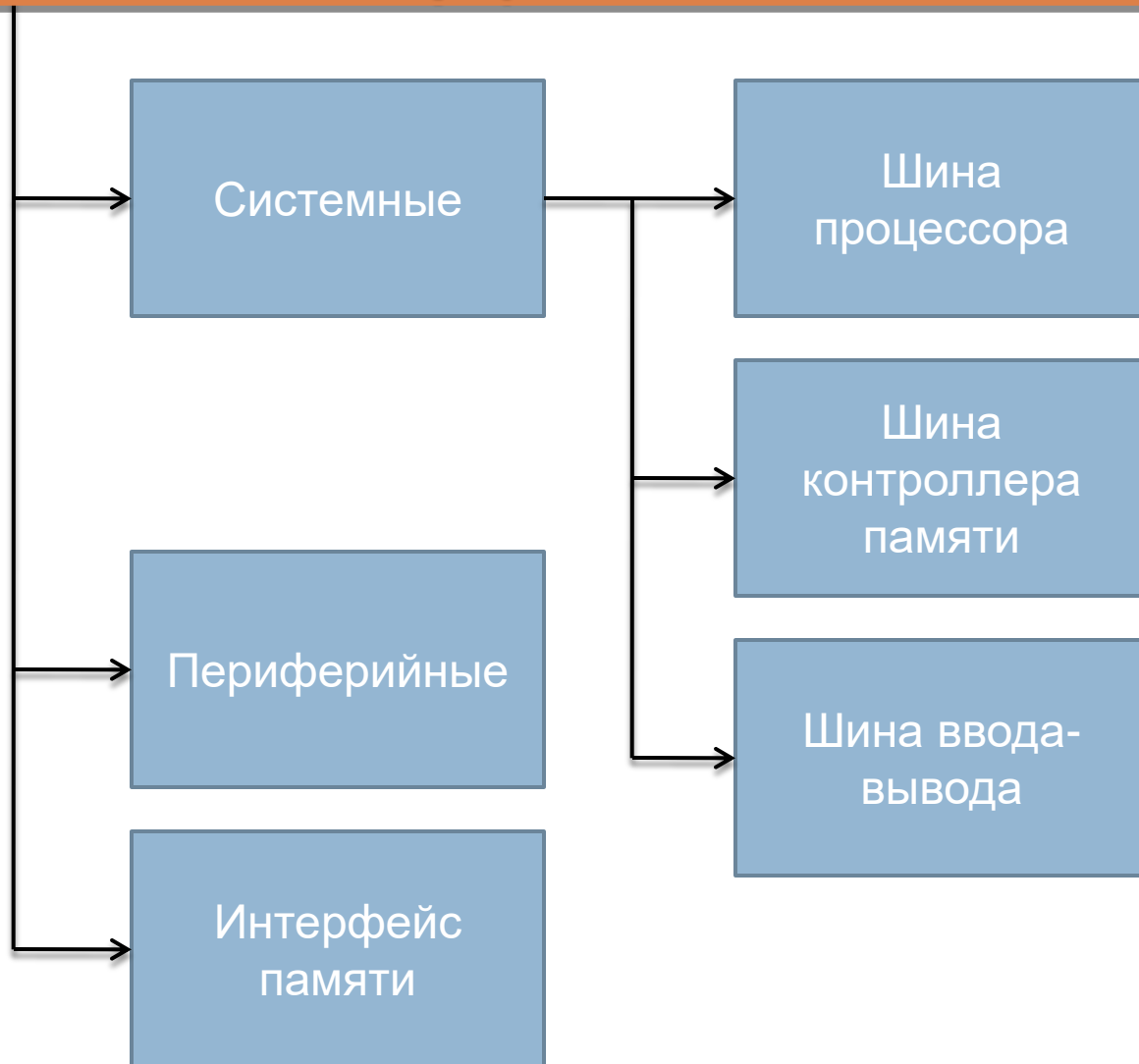
При **децентрализованном** (или *распределенном*) **арбитраже** единый арбитр отсутствует. Вместо этого каждый ведущий содержит блок управления доступом к шине, и при совместном использовании шины такие блоки взаимодействуют друг с другом, разделяя между собой ответственность за доступ к шине. По сравнению с централизованной схемой децентрализованный арбитраж менее чувствителен к отказам претендующих на шину устройств. В целом схемы децентрализованного арбитража потенциально более надежны, поскольку отказ контроллера шины в одном из ведущих не нарушает работу с шиной на общем уровне. Тем не менее должны быть предусмотрены средства для обнаружения неисправных контроллеров. Основным недостатком децентрализованных схем заключается в относительной сложности логики арбитража, которая должна быть реализована в аппаратуре каждого ведущего.

# Интерфейсы

ПУ и системные компоненты ЭВМ соединяются друг с другом посредством средств подключения, организованных по иерархическому принципу.

Средства (аппаратные и программные), используемые для соединения двух компонентов или систем, называются **интерфейсом**.

# Интерфейсы ЭВМ



# Системные интерфейсы

**Системные интерфейсы** образуют единую логическую *системную шину*, по которой информация передается в виде данных, пригодных для обработки, снабженных адресами в общем адресном пространстве системы (физические адреса, с которыми оперирует процессор).

Системная шина может быть разделена несколько шин, имеющих различную физическую природу и протоколы передачи данных (на уровне электрических и/или логических сигналов).

Как правило, все системные интерфейсы имеют электрическую природу и реализованы в виде дорожек на печатных платах (или линий внутри микросхем).

При этом *шина ввода-вывода* отвечает за обмен данными с контроллерами внутренних периферийных устройств и контроллерами периферийных интерфейсов.

# Классификация интерфейсов

В рамках данного курса предусматривается изучение периферийных интерфейсов, а также системной шины ввода-вывода, которая обеспечивает соединение между ПУ и компонентами ядра системы.

По способу кодирования и передачи данных интерфейсы делятся:

- **Параллельные**, характеризующиеся разрядностью (количеством бит одного машинного слова, передаваемых в один момент времени);
- **Последовательные**, характеризующиеся количеством агрегированных каналов передачи данных (количеством бит различных машинных слов, передаваемых одновременно, не обязательно синхронно и с одной скоростью).

# Классификация интерфейсов

По *направлению передачи*:

- **Однонаправленные** (симплексные);
- **Двунаправленные** (дуплексные);
- **С возможностью изменения направления передачи** (полудуплексные).

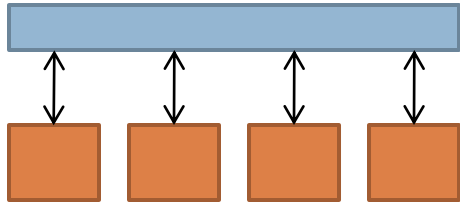
Современные последовательные интерфейсы обычно обеспечивают дуплекс за счет работы двух встречно направленных симплексных каналов. При этом зачастую в одну сторону передаются данные, а в другую – пакеты квитирования и управления потоком.

По *физическому явлению*, используемому для кодирования информации:

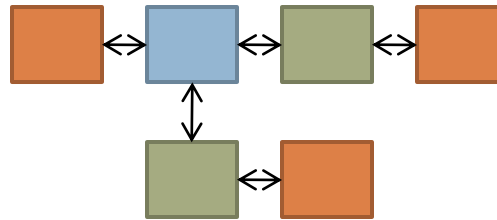
- **Электрические** (с управлением током или напряжением);
- **Оптические** (оптоволоконные);
- **Беспроводные** (радио).

# Топология интерфейса

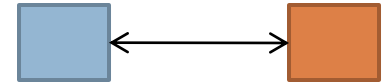
Шина



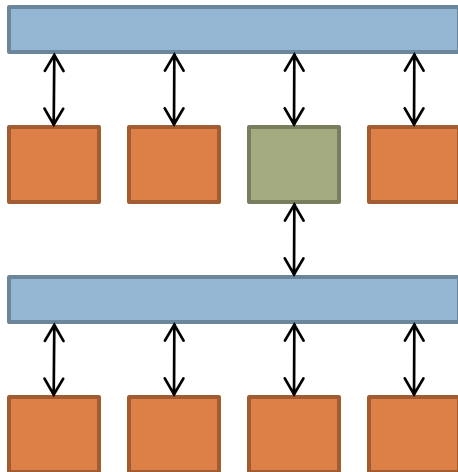
Дерево



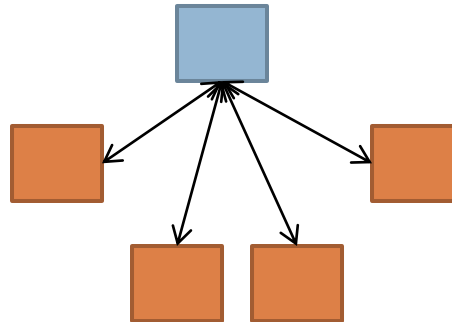
Точка-точка



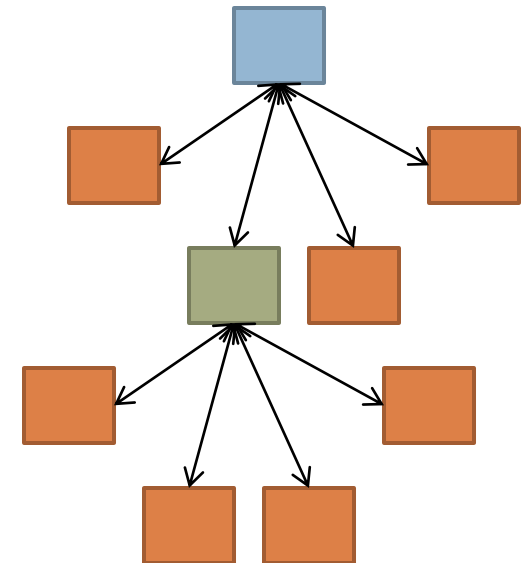
Многоуровневая шина



Звезда



Многоярусная звезда




# Характеристики интерфейсов

Интерфейс предназначен для передачи данных, адресов и управляющих сигналов, поэтому наиважнейшей его характеристикой является скорость передачи, или **пропускная способность**. Она измеряется в байтах в секунду (для параллельных) или в битах в секунду (для последовательных).

Зачастую выделяют «сырую» (raw) пропускную способность, *пиковую* (peak), *теоретическую*, *усредненную* (реально достижимую). Связано это с затратами на кодирование, избыточность (для определения и коррекции ошибок), квитирование, арбитраж, получение доступа к среде передачи, «зазоры», процедуры установление соединения, передачу различной управляющей информации (маркеры, номера пакетов, служебные поля в пакетах и т.д.).

Другие характеристики специфичны для интерфейсов различного типа. Например, количество подключаемых устройств, расстояние подключения, количество каскадов, поддержка функций «горячего» подключения, самонастройки и т.д.



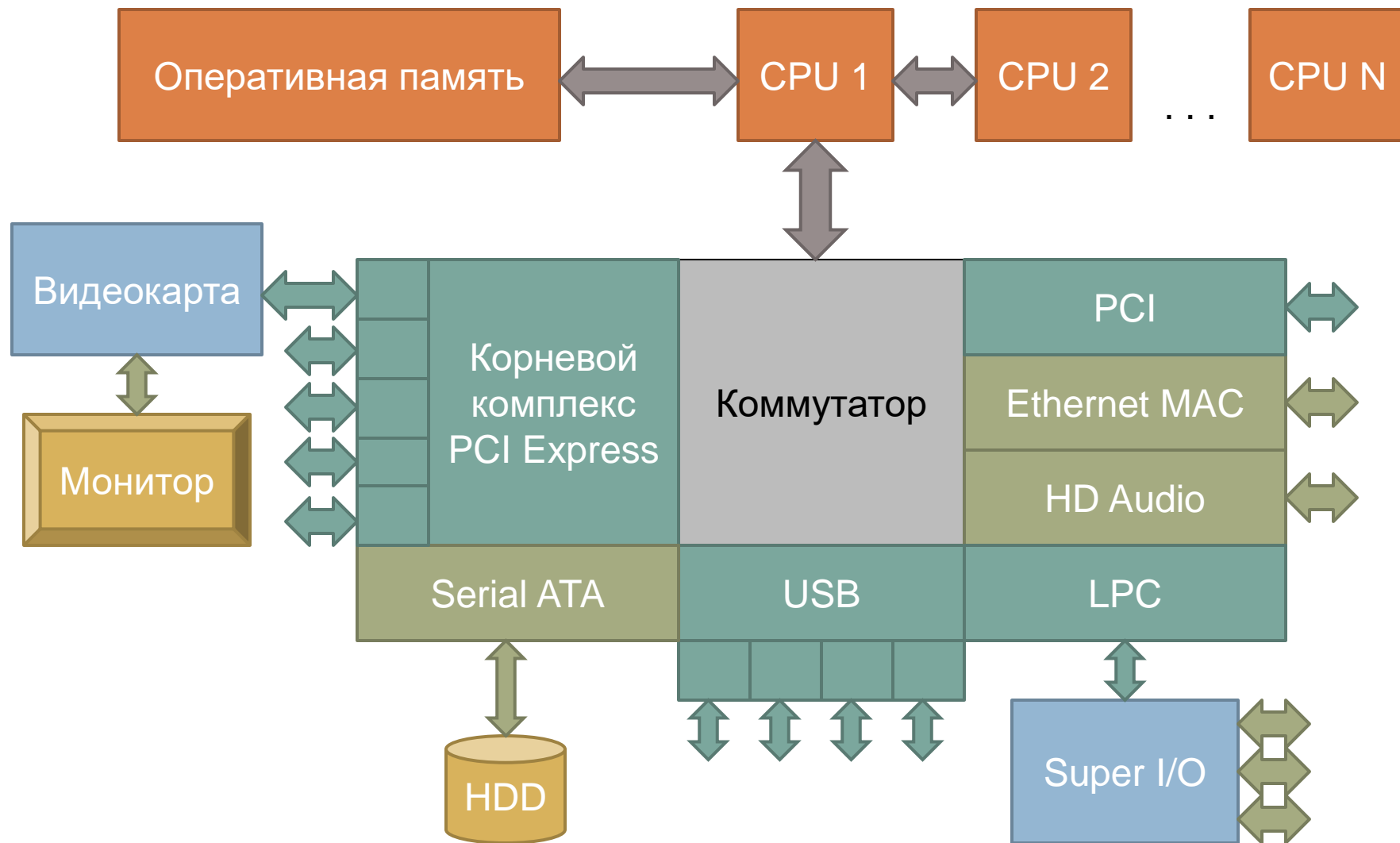


*Пропускная способность канала* - Наибольшая возможная в данном канале скорость передачи информации называется его пропускной способностью. Пропускная способность канала есть скорость передачи информации при использовании «наилучших» (оптимальных) для данного канала источника, кодера и декодера, поэтому она характеризует только канал.

Пропускная способность дискретного (цифрового) канала без помех  $C = \log(m)$  бит/символ где  $m$  — основание кода сигнала, используемого в канале.

*Скорость передачи информации* в дискретном канале без шумов (идеальном канале) равна его пропускной способности, когда символы в канале независимы, а все  $m$  символов алфавита равновероятны (используются одинаково часто).

# Архитектура современного ПК



# Архитектура современного ПК

Персональный компьютер (ПК) семейства IBM PC, будучи созданным в качестве персонального делового инструмента для обработки текста, таблиц, баз данных и деловой графики, стал основой для большого семейства ЭВМ различного класса. Практически все ЭВМ, использующие процессоры архитектуры x86, построены на базе архитектуры IBM PC AT с некоторыми дополнениями (прежде всего в плане поддержки многопроцессорности, управления энергопотреблением, самоконфигурации и т.д.). Данная архитектура применяется также для:

- серверов, в т.ч. многопроцессорных;
- рабочих станций;
- мобильных ПК;
- встраиваемых систем (embedded);
- промышленных систем (industrial );
- смартфонов.

# Архитектура современного ПК

Большинство систем класса ПК, а также «выросших» из этой архитектуры, конструктивно состоят из следующих блоков:

- материнская плата с микросхемами системной логики (чипсетом) и разъемами расширения;
- процессор(ы);
- модули памяти;
- платы расширения;
- внутренние устройства хранения данных;
- устройства питания и охлаждения.

Всю периферийную часть, изначально интегрированную в систему, в общем случае можно разделить на:

- графическая подсистема;
- подсистема хранения данных;
- подсистема интерфейса с пользователем;
- аудио-подсистема;
- подсистема сетевых соединений.

# Архитектура современного ПК

Подключение устройств во внутреннем исполнении (обычно они имеют контроллеры, подключаемые к логической системной шине), обеспечивает универсальная шина ввода-вывода **PCI Express**, а также устаревшая шина **PCI**.

Подключение внешних устройств обеспечивает универсальная шина **USB**.

Для подключения устройств хранения данных, а также аудио- и сетевых кодеков используются специализированные интерфейсы.

Для совместимости с устройствами старого поколения сохранилась шина **LPC** (логически эмулирует ISA) и контроллер устаревших внешних интерфейсов **Super I/O**.

# Заключение

## Основные понятия

Архитектура ЭВМ, ядро ( АЛУ, УУ, ОЗУ) и ПУ

Классификация ПУ

Классификация интерфейсов, характеристика интерфейсов

Архитектура ПК

SIMD

IP

PIO

BIOS

API

DMA

CDS

PC

X86, IBM PC AT

PCI, Serial ATA, USB, LPC, ISA, Ethernet MAC, HD Audio, Super I/O