1) Структурная схема персонального компьютера (на примере Pentium PRO I II II).

2) Программно-аппаратная архитектура IA\_32 процессоров Intel. Общее понятие

3) Определение процессора. Основные характеристики процессора

4) Типы архитектур процессора: RISC- и CISC-процессоры, Гарвардская и Принстонская арх

5) Специализированные процессоры и их функциональные характеристики: математический

6) Классификация запоминающих устройств ЭВМ. Внешняя память - типы, организация,

7) Внутренняя память ROM, RAM. Особенности организации DRAM, SRAM, SDRAM, RDRAM,

8) Основные количественные характеристики ОЗУ. Понятие "бутылочного горлышка"

9) Классификация интерфейса с устройствами ввода-вывода: по пересечению с адресным

10) Интерфейс ввода-вывода IDE (ATA), ATAPI, SATA – назначение, отличительные

11) Интерфейс ввода-вывода SCSI – назначение, отличительные особенности каждой

12) Интерфейс ввода-вывода USB. Назначение и организация.

13) Программная модель архитектуры IA\_32. Режимы работы процессора архитектуры IA\_32

14) Набор регистров процессора архитектуры IA\_32. Назначение основных флагов регистра

15) Организация памяти компьютера архитектуры IA\_32. Сегментированная модель памяти.

16) Формат машинных команд IA\_32. Функциональная классификация машинных команд

17) Синтаксис ассемблера. Структура программы на ассемблере. Директивы сегментации

18) Переменные и директивы их описания. Стандартные типы данных и их размещение в

19) Способы адресации операндов в ассемблере

20) Команды передачи данных. Арифметические команды сложения/вычитания/сравнения.

21) Организация условного/безусловного перехода. Команды условного/безусловного

22) Организация циклов при помощи команды LOOP, организация вложенных циклов,

23) Организация подпрограмм. Команды для организации подпрограмм, примеры

24) Организация системы прерываний. Типы прерываний в компьютерных системах.

25) Понятие параллельных вычислений. Требования к архитектурным принципам построения

26) Классификация вычислительных систем по систематике Флинна и пути достижения

27) Варианты организации систем MIMD класса «мультипроцессоры». Особенности архитектур NUMA (NCC-NUMA, CC-NUMA, COMA). Примеры архитектур (реализации Sun, IBM, Cray и др.)

28) Варианты организации систем MIMD класса «мультикомпьютеры»  (NORMA). Понятия транспьютерной системы, кластерной системы. Особенности построения архитектур МРР

29) Эволюция сетей ЭВМ. Обобщенная (функционально-логическая и физическая) структура 30 Эталонная модель взаимодействия открытых систем (ЭМВОС): основные понятия и

31) Способы организации обмена данными в вычислительной сети. Методы коммутации

32) Способы организации виртуальных каналов и управления потоками данных: протоколы с

33) Методы маршрутизации информации: простая, фиксированная, адаптивная. Протоколы

34) Принципы межсетевого взаимодействия, назначение и типы межсетевых шлюзов.

35) Типы IP-адресов и классы подсетей. Протокол ICMP.

36) Понятие маски. Использование масок для структуризации IP сетей (пример)

37) Классы транспортных протоколов и типы сетевых соединений. Функции транспортной

38) Процедура передачи данных и метод окна в TCP. Адаптивные свойства протокола TCP.

39) Особенности организации модели взаимодействия локальных вычислительных сетей

40) Технологии семейства Ethernet. Иерархия стандартов Ethernet. Метод доступа CSMA/CD.

41) Технологии семейства Ethernet – основные характеристики стандартов серии 10Base-х,

42) Технологии Ethernet – основные характеристики стандартов серии 1000Base-х.

43) Домен коллизий и логическая структуризация сетей семейства Ethernet. Принцип функционирования коммутаторов локальных сетей

**1. Структурная схема персонального компьютера (на примере Pentium PRO I II II). Машинный язык и язык ассемблера**

У компьютера есть органы восприятия информации из внешнего мира это клавиатура, мышь, накопители на магнитных дисках. У компьютера есть органы, «переваривающие» полученную информацию, — это центральный процессор и оперативная память. И наконец, у компьютера есть органы речи, выдающие результаты переработки. Для того чтобы компьютер правильно работал ему нужна информация, в виде строго организованных последовательностей нулей и единиц, комбинации которых и составляют машинный язык. Компьютер понимает только один язык язык машинных команд. Еще в 50-е гг. программисты стали использовать для программирования символический аналог машинного языка, который назвали языком ассемблера. На ассемблере пишут в основном программы, которые должны обеспечить эффективную работу с аппаратной частью компьютера. Иногда на ассемблере пишутся критичные ко времени выполнения или расходованию памяти фрагменты программы. Современные технологии программирования позволяют создавать программы, не написав ни строчки кода. Но в компьютер команды все же поступают на машинном языке. Чтобы эффективно использовать все возможности компьютера, применяют символический аналог машинного языка — язык ассемблера.

**2. Программно-аппаратная архитектура IA\_32 процессоров Intel. Общее понятие об архитектуре ЭВМ. Архитектура и свойства машины фон Неймана.**

**Архитектура ЭВМ** — это абстрактное представление ЭВМ, которое отражает ее структурную, схемотехническую и логическую организацию. **Понятие архитектуры ЭВМ** — иерархическое. Все современные компьютеры обладают некоторыми общими и индивидуальными архитектурными свойствами. На сегодняшний день общие архитектурные свойства большинства современных компьютеров подпадают под понятие фон-неймановской архитектуры. Фон Нейман предложил схему ЭВМ с программой в памяти и двоичной логикой вместо десятичной. **Логически машину фон Неймана составляли пять блоков**: оперативная память, арифметико-логическое устройство (АЛУ) с аккумулятором, блок управления, устройства ввода и вывода. **Свойства и принципы работы машины фон Неймана.**

1. **Линейное пространство памяти**. Для оперативного хранения информации компьютер имеет совокупность ячеек с последовательной нумерацией - оперативной памятью. 2. **Принцип хранимой программы.** код программы и ее данные находятся в одном и том же адресном пространстве оперативной памяти.

3. **Принцип микропрограммирования**. Машинный язык еще не является той конечной субстанцией, которая физически приводит в действие процессы в машине. В состав процессора входит устройство микропрограммного управления, поддерживающее набор действий-сигналов, которые нужно сгенерировать для физического выполнения каждой машинной команды.

4. **Последовательное выполнение программ**. Процессор выбирает из памяти команды строго последовательно.

5. **Отсутствие разницы между данными и командами в памяти**. С точки зрения процессора, нет принципиальной разницы между данными и командами. Данные и машинные команды находятся в одном пространстве памяти в виде последовательности нулей и единиц. 6. **Безразличие к назначению данных.** Машине все равно, какую логическую нагрузку несут обрабатываемые ею данные.

**3 Определение процессора. Основные характеристики процессора - технологический процесс, тактовая частота и производительность, потребляемая мощность**

**Процессор, CPU** — центральное процессорное устройство, «мозг» персонального компьютера, отвечает за обработку информации на основе организации вычислительных процессов согласно набору предустановленных команд.  **Основные характеристики центрального процессора:**

**Тактовая частота** процессора измеряется (МГц, ГГц) - количество тактов (вычислений) в секунду. Как правило, тактовая частота процессора, пропорциональна частоте шины (FSB).

**Частота шины -** происходит обмен данными между процессором и системной шиной материнской платы

**Множитель -** Коэффициент умножения, на основании которого производится расчет конечной тактовой частоты процессора, (FSB)\* (множитель).

**Разрядность** Разрядность (32/64 bit) — максимальное количество бит информации, которые процессор может обрабатывать и передавать одновременно

**Кэш-память L1 L2 L3** — это блок высокоскоростной памяти, к расположен на ядре процессора, в него помещаются данные из оперативной памяти. повышает быстродействие процессора

**Техпроцесс**

Масштаб технологии (мкм), которая определяет размеры полупроводниковых элементов, составляющих основу внутренних цепей процессора. Совершенствование технологии и пропорциональное уменьшение размеров элементов способствуют улучшению характеристик процессоров. Для сравнения, у ядра Willamette, выполненного по техпроцессу 0.18 мкм — 42 миллиона элементов, а у ядра Prescott, техпроцесс 0.09 мкм — 125 миллионов

**4 Типы архитектур процессора: RISC- и CISC-процессоры, Гарвардская и Принстонская архитектура**

**RISC** — компьютер (процессор) с сокращенным набором команд. У RISC-процессоров система команд меньше, команды более простые и выполняются быстрее. Однако программа для RISC-процессоров длиннее.

**CISC**— компьютер (процессор) с полным (сложным) набором команд. У CISC-процессоров система команд больше, команды более сложные и выполняются медленнее, но программа для них короче.

**Гарвардская архитектура** характеризуется раздельной памятью команд и данных. Она была реализована в Гарвардском университете в компьютере Mark I.

Основным признаком **Принстонской архитектуры** является объединенная (смешанная) память команд и данных (что предоставляет широкие возможности по модификации команд). Она ассоциируется с архитектурой фон Неймана

**Процессоры со скрытым параллелизмом называются супескалярными**. - процессор содержит не менее двух конвейеров. Конвейеры работают параллельно. Распараллеливание происходит на аппаратном уровне. Пример суперскалярного процессора — Pentium.

**Подклассом EPIC-процессоров являются процессоры с длинным командным словом**. Команды VLIW-процессора имеют специальные поля, содержащие информацию для каждого из параллельных обрабатывающих устройств процессора, предписывающую им определенные действия.

**По возможности изменения системы команд выделяют два типа процессоров: - процессор** с фиксированной системой команд (с произвольной логикой); **-процессор** с модифицируемой системой команд (с возможностью микропрограммирования). Микропрограммное управление дает гибкость в ущерб быстродействию, а также позволяет легче скорректировать ошибки в выполнении команд.

**5 Специализированные процессоры и их функциональные характеристики: математический, физический, сигнальный, звуковой, графический, сетевой, криптографический процессоры.**

**Специализированные процессоры** часто являются основным элементом соответствующей подсистемы ЭВМ, но могут выступать и в качестве центрального процессора.

**Математический процессор** выполняет операции над числами с плавающей точкой. **Физический процессор** выполняет операции, связанные с различными физическими процессами, например: движением твердых тел и жидкостей.

**Сигнальный процессор** предназначены для выполнения следующих функций цифровой обработки сигналов: фильтрации сигнала; свертки (смешения) двух сигналов; вычисления корреляционных функций сигналов; усиления, ограничения, трансформации сигналов; прямого и обратного преобразования Фурье сигнала.

Для их выполнения они на аппаратном уровне поддерживают базовые операции цифровой обработки сигналов, к которым относятся: умножение с накоплением, модульная адресная арифметика, нормировка результатов арифметических операций. Для выполнения этих операций, в свою очередь, сигнальные процессоры имеют аппаратный умножитель, позволяющий выполнять умножение двух чисел за один такт.

Сигналы существуют самых разных типов. Соответственно сигнальные процессоры могут иметь ту или иную специализацию. **Медийный процессор** предназначены для обработки мультимедийной информации: аудиосигналов, графики — а также для выполнения коммуникационных функций. **Звуковой процессор** также называются звуковыми контроллерами. В составе звуковой подсистемы персонального компьютера они используются вместе со звуковыми кодеками. **Графический процессор** выполняет операции по обработке и построению изображений. **Сетевой процессор** - это процессор, предназначенный для использования в сетевых устройствах. Его основной задачей является обработка пакетов.

**6. Классификация запоминающих устройств ЭВМ. Внешняя память - типы, организация, основные характеристики**

По типу обращения *ЗУ* делятся на устройства, допускающие как чтение, так и запись информации, и постоянные *запоминающие устройства* (ПЗУ), предназначенные только для чтения записанных в них данных. ***ЗУ* первого типа** используются в процессе работы процессора для хранения выполняемых программ, исходных данных, промежуточных и окончательных результатов. В ПЗУ, как правило, хранятся системные программы, необходимые для запуска компьютера в работу, а также константы.

В ***ЗУ с произвольным доступом*** время доступа не зависит от места расположения участка памяти. В ***ЗУ с прямым доступом*** благодаря непрерывному вращению носителя информации возможность обращения к некоторому участку носителя циклически повторяется.

В ***ЗУ с последовательным доступом*** производится последовательный просмотр участков носителя информации, пока нужный участок не займет некоторое нужное положение напротив головок чтения/записи. ***Оперативная память*** - устройство, которое служит для хранения информации, непосредственно используемой в ходе выполнения программы в процессоре.

Для заполнения пробела между РП и ОП по объему и времени обращения используется ***кэш-память***, которая организована как более быстродействующая статическая *ОП* со специальным механизмом записи и считывания информации и нужна для хранения информации, которая чаще используется. Часть *кэш-памяти* располагается на кристалле микропроцессора, а часть - вне его. *Кэш-память* программно недоступна. Для обращения к ней используются аппаратные средства процессора и компьютера.

Внешняя память организуется, как правило, на магнитных и оптических дисках. *Емкость* дисковой памяти достигает десятков гигабайт при времени обращения менее 1 мкс. Магнитные ленты вследствие своего малого *быстродействия* и большой *емкости* используются в настоящее время в основном только как устройства резервного копирования данных, обращение к которым происходит редко, а может быть и никогда. Время обращения для них может достигать нескольких десятков секунд.

**7. Внутренняя память ROM, RAM. Особенности организации DRAM, SRAM, SDRAM, RDRAM, DDR. Поколения DDR и их характеристики. Особенности организации многоканальной памяти DDR – DDR II, DDR III**

ROM (ПЗУ) – это форма постоянного хранения, в то время как RAM (ОЗУ) является временным хранилищем. ROM представляет энергонезависимую память, а RAM - энергозависимую. Память ROM хранит данные без электричества, а RAM требует электричества для хранения. **SRAM** требует постоянного потока энергии для работы. Из-за постоянной мощности SRAM не нужно «обновлять», чтобы помнить сохраненные данные. SRAM это энергозависимая память. Все данные теряются после отключения питания. **DRAM** требует периодического «обновления» мощности для работы. Конденсаторы, которые хранят данные в DRAM, постепенно разряжают энергию. Отсутствие энергии означает, что данные теряются. DRAM также считается энергозависимой памятью. Все данные теряются после отключения питания. **SDRAM** - это классификация DRAM, которая работает синхронно с тактовой частотой процессора. В начале ожидает тактового сигнала, прежде чем ответить на ввод данных. ЦП может параллельно обрабатывать перекрывающиеся инструкции, а именно «конвейерная обработка» - возможность получать новую инструкцию до того, как предыдущая инструкция полностью разрешена. **DRDRAM** позволяет достичь очень больших пиковых скоростей передачи данных. Вся подсистема состоит из следующих компонентов: основной контроллер, канал, разъем для модулей, модуль памяти, генератор дифференциальных синхроимпульсов и сами микросхемы памяти. **Описание DDR1, DDR2 и DDR3.** В DDR и DDR2 данные передаются по обоим срезам. Разница лишь в более быстрой шине на планках DDR2, на которую передача может совершаться одновременно из четырех мест. В результате скорость обработки выросла в два раза. Также DDR2 характеризуется более низким энергопотреблением. На смену DDR2 пришли мощные DDR3. Создателям DDR3 удалось увеличить скорость работы ОП и снизить энергопотребление плат.

**8. Основные количественные характеристики ОЗУ. Понятие "бутылочного горлышка" фон-неймановской архитектуры.** **Способы увеличения производительности памяти.** **Многоуровневая организация оперативной памяти, понятие кэш-памяти.**   
**Тактовая частота** выражается в циклах. Каждая запись и каждое считывание данных — это один цикл. **Тайминг оперативной памяти** - время задержки доступа к ячейкам памяти между операциями чтения/записи. **CAS Latency** – этот показатель говорит, сколько времени уходит на поиск необходимых данных когда ЦП попросит доступ на считывание. **RAS to CAS Delay**- показатель демонстрирует время полного доступа к данным. **Row Precharge Delay** - динамическая память, ее ячейки разряжаются и нуждаются в периодической перезарядке. Поэтомк данные, которые в ней, обновляются. **Activate to Precharge Delay** - минимальное время между активацией строки и подачей команды на предзаряд. **Бутылочным горлышком** фон Неймана является идея о том, что пропускная способность компьютерной системы ограничена из-за относительной способности процессоров по сравнению с максимальными скоростями передачи данных. **Регистр процессора** - блок ячеек памяти, образующий сверхбыструю оперативную память внутри процессора; используется самим процессором и большой частью недоступен программисту. **Кэш-память** - это высокоскоростная память произвольного доступа, используемая процессором для временного хранения информации. Она увеличивает производительность, хранит часто используемые данные и команды «ближе» к процессору. **Кэш первого уровня (L1)** - быстрый уровень кэш-памяти, работает напрямую с ядром процессора, данный уровень обладает наименьшим временем доступа. Он буфер между процессором и кэш-памятью второго уровня. **Кэш второго уровня (L2)** – второй уровень более масштабный, но обладает меньшими «скоростными характеристиками». Буфер между уровнем L1 и L3. **Кэш третьего уровня (L3)** – третий уровень, более медленный, нежели два предыдущих. Он гораздо быстрее, чем ОП. Если два предыдущих уровня разделяются на каждое ядро, то данный уровень является общим для всего процессора.

**9** **Классификация интерфейса с устройствами ввода-вывода: по пересечению с адресным пространством памяти (основным адресным пространством) процессора: по способу синхронизации работы процессора с устройствами ввода-вывода, по количеству линий данных, по направлению передачи, по топологии.**

**Пропускная способность интерфейса и методика ее определения**

**Интерфейс** – это совокупность линий и шин сигналов, электрических схем, а также алгоритмов (протоколов), осуществляющих обмен информацией между устройствами ЭВМ.

Линии, сгруппированные по функциональному признаку или назначению, называют **шинами** интерфейса. Совокупность всех линий образует **магистраль** интерфейса.

**Классификация интерфейсов**

Признаки классификации и типы интерфейса с устройствами ввода-вывода таковы:

**Пересечение с адресным пространством памяти процессора:**

Изолированный ввод-вывод, ввод-вывод по принципу доступа к памяти.

Преимущества изолированного ввод-вывода в том, что он не занимает адресного пространства МП. Недостаток: программная часть ограничивается командами IN и OUT. В основном используется изолированный ввод-вывод.

**По способу синхронизации работы процессора с устройствами ввода-вывода или, что инициирует обмен - процессор или устройство ввода-вывода: синхронный** ввод-вывод по опросу, асинхронный ввод-вывод по прерыванию.

**По количеству линий данных:** Параллельный, последовательный.

**По направлению передачи информации:** односторонний,двусторонний поочередный, двусторонний.

**По топологии:** многоточечные, двухточечные (радиальные): параллельные последовательные.

**10** **Интерфейс ввода-вывода IDE (ATA), ATAPI,** **SATA – назначение, отличительные особенности каждой модификации, пропускная способность**

**Интерфейс** **IDE** – интерфейс устройств со встроенным контроллером. Поддерживает несколько способов обмена. Первый способ производит обмен данными через регистры процессора под его непосредственным управлением. Следствием этого является высокая загрузка процессора при операциях ввода/вывода. Вторым способом является использование режима прямого доступа к памяти, при котором контроллер интерфейса IDE и контроллер прямого доступа к памяти материнской платы пересылают данные между диском и оперативной памятью, не загружая центральный процессор. В целях развития возможностей интерфейса IDE была предложена его расширенная спецификация ЕIDE (синонимы ATA, ATA-2). Она поддерживает накопители емкостью свыше 504 Мбайт, поддерживает несколько накопителей IDE и позволяет подключать к одному контроллеру до четырех устройств, а также поддерживает периферийные устройства, отличные от жестких дисков. Расширение спецификации IDE для поддержки иных типов накопителей с интерфейсом IDE называют также **ATAPI**.

**SATA** (Serial ATA) – последовательный интерфейс обмена данными с накопителями информации. Для подключения используется 8-pin разъем. SATA является развитием параллельного интерфейса ATA (IDE), который после появления SATA был переименован в PATA (Parallel ATA).Стандарт SATA (SATA150) обеспечивал пропускную способность равную 150 МБ/с (1,2 Гбит/с). SATA 2 (SATA300). Стандарт SATA 2 увеличивал пропускную способность в двое, до 300 МБ/с (2,4 Гбит/с), и позволяет работать на частоте 3 ГГц. Стандартны SATA и SATA 2 совместимы между собой, однако для некоторых моделей необходимо вручную устанавливать режимы, переставляя джамперы. SATA 3, хотя про требованию спецификаций правильно называть SATA 6Gb/s. Этот стандарт в двое увеличил скорость передачи данных до 6 Гбит/с (600 МБ/с).

**11 Интерфейс ввода-вывода SCSI (Small Computer System Interface) – назначение, отличительные особенности каждой из модификаций, пропускная способность.** **Особенности и параметры интерфейса Serial Attached SCSI (SAS)**

Основные преимущества интерфейса SCSI заключаются в возможности подключения до 15 устройств на канал и возможности существенно более глубокой оптимизации очередей команд, передаваемых от контроллера подключенным к нему HDD, что выражается в более высокой производительности при работе под нагрузками. Этот интерфейс предоставляет возможность работать с самым разным оборудованим, например: стримеры, накопители на магнитных лентах, магнитооптические накопители, CD-ROM, CD-R и RW, устройства ввода-вывода. Эти устройства подключаются к компьютеру через специальный SCSI адаптер, а операционная система получает к ним доступ через нужные драйверы. отличительной особенностью интерфейса является необходимость использования согласующих заглушек на концах цепочки устройств для устойчивой передачи данных, и уникальные номера устройств на шине. Есть несколько спецификаций SCSI:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Разрядность, бит | Скорость передачи данных, Мбайт/сек | Разъем, количество контактов |
| SCSI-1 | 8 | 5 | 25 или 50 |
| SCSI-2 (Fast SCSI) | 8 | 10 | 50 |
| Wide SCSI | 16 | 20 | 68 или 80 |
| Ultra SCSI (Fast-20) | 8 | 20 | 50 |
| Wide Ultra SCSI или SCSI-3 | 16 | 40 | 68 или 80 |
| Ultra2 SCSI LVD | 16 | 80 | 68 или 80 |
| Ultra3 SCSI/SPI-3 (Ultra160 SCSI) LVD | 16 | 160 | 68 или 80 |
| Ultra3 SCSI/SPI-4 (Ultra320 SCSI) LVD | 16 | 320 | 68 или 80 |

**12 Интерфейс ввода-вывода USB. Назначение и организация. Характеристики пропускной способности USB 1.х, USB 2.0, USB 3.x, USB4, Inter-Chip USB, Wireless USB**

USB шина появилась в 1996 году как попытка решения проблемы множественности интерфейсов. К тому времени персональные компьютеры (ПК) были оснащены большим количеством разнообразных внешних полезных и необходимых интерфейсов, но все они требовали своего специального разъема и, чаще всего, выделенного аппаратного прерывания (IRQ, Interrupt ReQuest).

USB была разработана группой из семи компаний([Compaq](http://h18000.www1.hp.com/country/index.html), Digital Equipment Corp, [IBM PC Co.](http://www.ibm.com/ru/ru/), [Intel](http://www.intel.com/cd/corporate/europe/emea/rus/381096.htm), [Microsoft](http://www.microsoft.com/ru-ru/default.aspx), [NEC](http://www.nec.ru/) и [Northern Telecom](http://www.nortel.com/)). Первая спецификация (версия 1.0) USB была опубликована в начале 1996 года, а осенью 1998 года появилась спецификация 1.1, исправляющая проблемы, обнаруженные в первой редакции. Весной 2000 года была опубликована версия 2.0, в которой предусматривалось 40-кратное повышение пропускной способности шины. Так, спецификация 1.0 и 1.1 обеспечивает работу на скоростях 12 Мбит/с и 1,5 Мбит/с, а спецификация 2.0 – на скорости 480 Мбит/с. При этом предусматривается обратная совместимость USB 2.0 с USB 1.х. Окончательная спецификация USB 3.0 появилась в 2008 году. Созданием USB 3.0 занимались компании [Intel](http://www.intel.com/cd/corporate/europe/emea/rus/381096.htm), [Microsoft](http://www.microsoft.com/ru-ru/default.aspx), [Hewlett-Packard](http://www8.hp.com/ru/ru/home.html), [Texas Instruments](http://www.ti.com/ww/ru/), [NEC](http://www.nec.ru/) и . [NXP Semiconductors](http://www.ru.nxp.com/) В спецификации USB 3.0 разъёмы и кабели обновлённого стандарта физически и функционально совместимы с USB 2.0. В дополнение к четырем линиям USB 2.0 в USB 3.0 добавляется еще четыре линии связи (две витых пары). Новые контакты в разъемах USB 3.0 расположены отдельно от старых на другом контактном ряду. Спецификация USB 3.0 повышает максимальную скорость передачи информации до 4,8 Гбит/с, таким образом, скорость передачи возрастает с 60 Мбайт/с до 600 Мбайт/с и позволяет передать 1 Тб не за 8-10 часов, а за 40 минут-1 час. Версия 3.0 так же может похвастаться увеличенной силой тока с 500 мА до 900 мА, поэтому пользователь может не только подпитывать от одного хаба большее количество устройств, но и сами устройства во многих случаях смогут избавиться от отдельных блоков питания.

**13. Программная модель архитектуры IA\_32. Режимы работы процессора архитектуры IA\_32.**

Программную модель процессора в архитектуре IА-32 составляет следующий набор ресурсов:

**1.** пространство адресуемой памяти до **2.** набор регистров для хранения данных общего назначения; **3.** набор сегментных регистров; **4.** набор регистров состояния и управления; **5.** набор регистров устройства вычислений с плавающей точкой(сопроцессора); **6.** набор регистров целочисленного ММХ-расширения, отображенных на регистры сопроцессора; **7.** набор регистров ММХ-расширения с плавающей **8.** программный стек — специальная информационная структура, работа с которой предусмотрена на уровне машинных команд.

Это основной набор ресурсов. Кроме того, к ресурсам, поддерживаемым архитектурой IА-32, необходимо отнести порты ввода-вывода, счетчики мониторинга производительности. В программную модель данных процессоров входят 8- и 16-разрядные регистры общего назначения, сегментные регистры и адресное пространство памяти размером до 1 Мбайт.

**В рамках архитектуры IА-32 доступны следующие режимы работы процессора.**

**1.** Режим реальных адресов, или просто реальный режим (real mode) **2.** Защищенный режим (protected mode) позволяет максимально реализовать все идеи, заложенные в процессорах архитектуры IА-32 **3.** Режим виртуального процессора предназначен для организации многозадачной работы программ, разработанных для реального режима, совместно с программами защищенного режима. **4.** Режим системного управления (SММ) — это новый режим работы процессора. Он обеспечивает операционную систему перевод компьютера в режим пониженного энергопотребл или выполнения действий по защите системы.

**14. Набор регистров процессора архитектуры IA\_32. Назначение основных флагов регистра состояния.**

В процессор включены два регистра, постоянно содержащие информацию о состоянии как самого процессора, так и программы, команды которой он в данный момент обрабатывает:

**1.** регистр-указатель команд ЕIР/IР;

**2.** регистр флагов ЕFLAGS/FLAGS

**Регистр-указатель команд** ЕIР/IР имеет разрядность 32(16) бита и содержит смещение следующей подлежащей выполнению команды.

**Разрядность регистра флагов** ЕFLAGS/FLAGS равна 32(16) битам. Отдельные биты данного регистра имеют определенное функциональное назначение и называются флагами.

**Флаг переноса** СF: 1 — арифметическая операция произвела перенос из старшего бита результата 0 — переноса не было.

**Флаг четности** РF: 1 - 8 младших разрядов результата содержат четное число единиц; 0 - содержат нечетное число единиц.

**Флаг нуля** ZF. 1 - результат нулевой; 0 — результат ненулевой.

**Флаг знака** SF отражает состояние старшего бита результата (биты 7, 15 или 31 для 8-, 16- или 32-разрядных операндов соответственно);

1 — старший бит результата равен 1; 0 — старший бит результата равен 0.

**Флаг вложенности задачи** NТ используется в защищенном режиме работы процессора для фиксации того факта, что одна задача вложена в другую.

**Флаг возобновления** RF используется при обработке прерываний от регистров отладки.

**15. Организация памяти компьютера архитектуры IA\_32. Сегментированная модель памяти. Формирование физ адреса** Физическая память, к которой процессор имеет доступ по шине адреса, называется **оперативной памятью**. На самом нижнем уровне память компьютера можно рассматривать как массив бито. Но процессору неудобно работать с памятью на уровне битов, поэтому реально ОЗУ организовано как последовательность ячеек — байтов. Каждому байту соответствует свой уникальный адрес (его номер), называемый физическим.

Механизм управления памятью полностью аппаратный.

Процессор аппаратно поддерживает две модели использования оперативной памяти.

**В сегментированной модели** программе выделяются непрерывные области памяти (сегменты), а сама программа может обращаться только к данным, которые находятся в этих сегментах. **Сегментация** — механизм адресации. В основе механизма сегментации лежит понятие сегмента, который представляет собой независимый поддерживаемый на аппаратном уровне блок памяти. Внутри сегмента программа обращается к адресам относительно начала сегмента линейно, то есть начиная с 0 и заканчивая эффективным адресом.

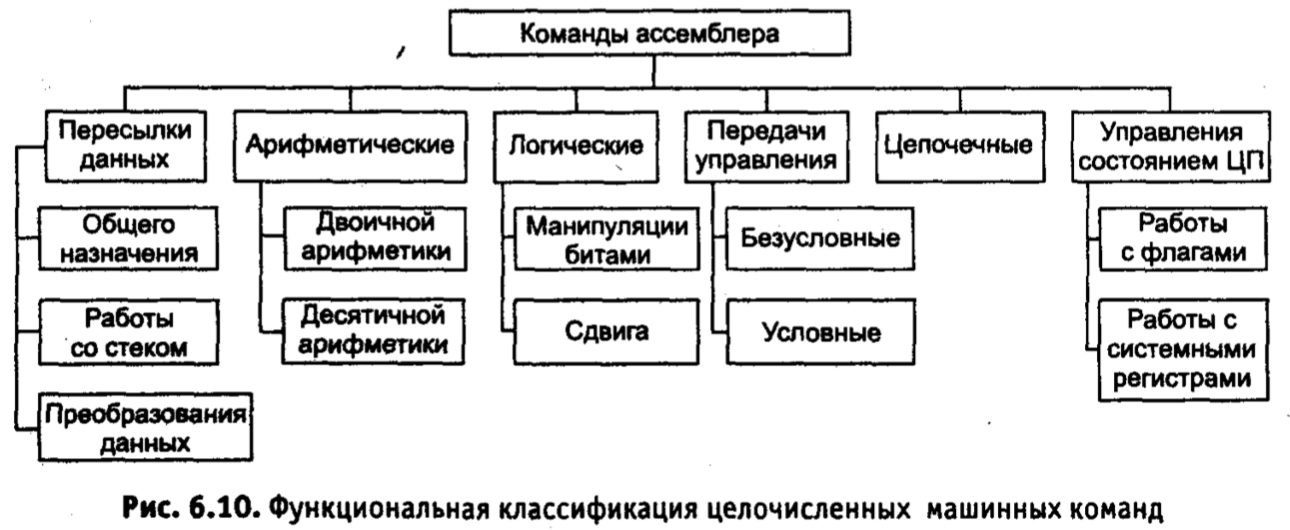
**Физический адрес формируется** следующим образом. В сегментных регистрах хранится селектор, содержащий индекс дескриптора в таблице дескрипторов. Далее происходит обращение к соответствующей таблице дескрипторов и соответствующему дескриптору, который содержит начальный 24- битный адрес сегмента, размер сегмента и права доступа, после чего вычисляется необходимый физический адрес путём сложения адреса сегмента со смещением из 16-разрядного регистра.

Для обращения к конкретному физ. адресу оперативной памяти необходимо опред. адрес начала сегмента и смещение внутри сегмента.

**16. Формат машинных команд IA\_32. Функциональная классификация машинных команд IA\_32.**

**Машинная команда** — закодированное указание процессору на выполнение некоторой операции. Прямого соответствия между структурой команды ассемблера и соответствующей машинной командой нет. Машинная команда всегда однозначна по отношению к производимым ею действиям на уровне аппаратуры. Программирование на уровне машинных команд — это тот минимальный уровень, на котором еще возможно программирование компьютера. Каждая машинная команда состоит из двух частей: операционной части, определяющей, «что делать», и операндной части, определяющей объекты обработки. В машинную команду микропроцессора входят следующие элементы: **1.** поле префиксов — элемент команды уточняет либо модифицирует действие этой команды **2.** поле кода операции, определяющее действие данной команды. **3.** поле операндов; содержит от 0 до 2 элементов.

Возможны только следующие сочетания операндов в команде: регистр - регистр; регистр - память; память - регистр; операнд - регистр; операнд - память;

Набор машинных команд можно структурировать:

**17. Синтаксис ассемблера. Структура программы на ассемблере.**

Предложения, составляющие программу, могут представлять собой синтаксическую конструкцию, соответствующую команде, макрокоманде, директиве или комментарию. Для того, чтобы транслятор ассемблера мог распознать их, они должны формироваться по определенным синтаксическим правилам. **1. имя метки** — идентификатор, значением которого является адрес первого байта того предложения исходного текста, программы, которое он обозначает; **2. имя** — идентификатор, отличающий данную директиву от других одноименных директив. **3. ход операции (КОП) и директива** — это мнемонические обозначения соответствующей машинной команды, макрокоманды или директивы транслятора; **4. операнды —** части команды, макрокоманды директивы обознач. объекты, над которыми производятся действия. Предложения ассемблера формируются из лексем, представляющих собой синтаксически неразделимые последовательности допустимых символов языка, имеющие смысл для транслятора. Лексемами являются: **1. идентификаторы** — последовательности допустимых символов, использующиеся для обозначения таких объектов программы, как коды операций, имена переменных и названия меток. **2. цепочки символов** — последовательности символов, заключенные в одинарные или двойные кавычки; **3. целые числа** в одной из следующих систем счисления: двоичной, десятичной и тд. Программа на ассемблера, может состоять из нескольких частей-**модулей**. В каждом модуле могут быть определены один или несколько сегментов данных, стека и кода. Любая законченная программа на ассемблере должна включать один главный модуль, с которого начинается ее выполнение. Модуль может содержать сегменты кода, данных и стека, объявленные при помощи директив. Перед объявлением сегментов нужно указать модель памяти при помощи директивы .MODEL.

**18. Переменные и директивы их описания. Стандартные типы данных и их размещение в памяти**

Общая структура директивы определения переменной:

[имя] DK операнд-инициализатор [, операнды ] Где К = B|W|D|Q|T|F|P

db — резервирование памяти для данных размером 1 байт.

dw — 2 байта выражение или константа или 2-байтовую строку, заключенная в кавычки.

dd — 4 байта. выражение или константу, относительное или адресное выражение, строку длиной до 4 символов, заключенную в кавычки.

df и dp — 6 байт; выражение или константу аналогично dd строку длиной до 6 байт, заключенную в кавычки.

dq — 8 байт. выражение или константу аналогично dd, относительное или адресное выражение аналогично, константу со знаком константу без знака, строку длиной до 8 байт, заключенную в кавычки.

dt — 10 байт. выражение или константу аналогично dd, адресное выражение строку длиной до 10 байт, заключенную в кавычки;

**1. Байт** — 8 последовательных бит.

**2. Слово** — последовательность из двух байтов, имеющих последовательные адреса.

**3. Двойное слово** — последовательность 4 байтов

**4. Учетверенное слово** — последовательность 8 байт

**5. 128-битный упакованный тип данных**. Этот тип данных появился в микропроцессоре Pentium III.

**6-7. Целый** тип со знаком и без знака **8. Указатель** на память.

**9.** Непрерывный набор байтов до 4 Гбайт.

**10. Типы данных с плавающей точкой**. Сопроцессор имеет несколько собственных типов данных, несовместимых с типами данных целочисленного устройства.

**11. Типы данных ММХ-расширения** (Pentium ММХ/II).

**19. Способы адресации операндов в ассемблере.**

Для указания адреса некоторой ячейки памяти необходимо указать оба элемента адре­са: сегмент и смещение. Способ, которым указывается смещение в машинной или ассемблерной команде, называется ***способом адресации***. Большая часть команд процессора работает с кодами данных (операн­дами). Эффективность методов адресации во многом определяет эф­фективность работы всего процессора в целом.

**Регистровая адресация:** Операнды могут располагаться в любых регистрах общего назначения и сегмент­ных регистрах. Регистровая адресация предполагает, что операнд (входной или выходной) находится во внутреннем регистре процессора.

**Непосредственная адресация:** предполагает, что операнд (вход­ной) находится в памяти непосредственно за кодом команды.

**Прямая (абсолютная) адресация (адресация по смещению):** Если у операнда, располагающегося в памяти, известен адрес, то его можно использовать. Адресация отличается для реального и защищённого режимов.

**Косвенная адресация:** По аналогии с регистровыми и непосредственными операндами адрес операн­да в памяти также можно не указывать, а хранить в любом регистре.

**Косвенно-регистровая (она же косвенная) адресация:** во внутреннем регистре процессора находится не сам операнд, а его адрес в памяти.

**Адресация по базе с индексированием:** В этом методе адресации смещение операнда в памяти вычисляется как сумма чисел, содержащихся в двух регистрах, и смещения, если оно указано. Как и для прямой адресации, вместо непос­редственного указания числа разрешено использовать имя переменной, заданной одной из директив определения данных.

**20. Команды передачи данных. Арифметические команды.** Группа команд передачи данных предназначена для организации обмена между регистрами стека, вершиной стека сопроцессора и ячейками оперативной памяти.

**Команды загрузки числа из области памяти на вершину стека сопроцессора**: FLD,FILD FBLD

**Команды сохранение числа из вершины стека сопроцессора в память:**FST, FSTP FIST, FISTP , FBST, FBSTP

Р- происходит выталкивание вещественного числа из стека после его сохранения в памяти

**Команды обмена:** FXCH

**Команды загрузки** констант на вершину регистрового стека сопроцессора: FLDZ ,FLD1 ,FLDPI , FLDL2T ,FLDL2E

Сложение: двоичное – A, десятичное - AP;

Вычитание: двоичное – S, десятичное SP;

Сравнение: двоичное – C, десятичное – CP).

Умножение: двоичное – M, десятичное - MP; Деление: двоичное – D, десятичное DP

Команды сложения и вычитания двух десятичных чисел, одно из которых расположено по адресу D1+(B1) и имеет длину L1 байтов, а другое — расположено по адресу D2+(B2) и имеет длину L2 байтов. Результат выполнения каждой из команд помещается в поле первого операнда если он помещается , при этом он выравнивается по правой границе.

СР (СРАВНЕНИЕ ДЕСЯТИЧНОЕ) производится вычисление разности двух десятичных чисел, длина каждого из которых указывается отдельно.

При выполнении команды СР производится проверка на правильность подготовки исходных данных. Если хотя бы один из операндов не является правильным упакованным десятичным числом, фиксируется особый случай в данных.

**21.**  **Организация условного/безусловного перехода. Команды условного/безусловного перехода, примеры**

**Безусловный переход :** Команда безусловного перехода имеет следующий синтаксис: JMP <операнд>. Операнд указывает адрес перехода

**Прямой переход :** Если в команде перехода указывается метка команды, на которую надо перейти, то переход называется прямым. **Косвенный переход:** При косвенном переходе в команде перехода указывается не адрес перехода, а регистр или ячейка памяти, где этот адрес находится. **jmp ebx**

**Команды условного перехода** На языке ассемблера с помощью набора команд сравнения и условного перехода вы можете реализовать логическую структуру любой сложности. На языке ассемблер сначала выполняются такие команды, как CMP, AND или SUB, влияющие на флаги состояния процессора. Затем выполняется команда условного перехода, которая анализирует значение нужных флагов, и в случае если они установлены, выполняют переход по указанному адресу. **Jxx <метка>**

Все команды условного перехода можно разделить на три группы. **которые** ставятся после команды сравнения JE JNE JL JG

**которые** ставятся после команд, отличных от команды сравнения, реагируют значение какого-либо флага. JZ ZF JNZ ZF

**проверяющие** не флаги, а значение регистра ECX или CX: JCXZ JECXZ

**Пример реализации конструкции if-then-else на языке ассемблера.**

mov a, eax

cmp eax, b

jne ElsePart

mov c, d

jmp EndOfIf

ElsePart:

inc b;

EndOfIf:

**22. Организация циклов при помощи команды LOOP, примеры.**

Для организации цикла предназначена команда LOOP. У этой команды один операнд — имя метки, на которую осуществляется переход. В качестве счётчика цикла используется регистр CX. Команда LOOP выполняет декремент CX, а затем проверяет его значение. Если содержимое CX не равно нулю, то осуществляется переход на метку, иначе управление переходит к следующей после LOOP команде.

**Пример цикла:** В качестве примера программа, которая будет печатать все буквы английского алфавита. Для вывода символа на экран используется функция DOS 02h

**use16 ;Генерировать 16-битный код**

**org 100h ;Программа начинается с адреса 100h**

**mov ah,02h ;Для вызова функции DOS 02h - вывод символа**

**mov dl,'A' ;Первый выводимый символ**

**mov cx,26 ;Счётчик повторений цикла**

**metka:**

**int 21h ;Обращение к функции DOS**

**inc dl ;Следующий символ**

**loop metka ;Команда цикла**

**mov ah,09h ;Функция DOS 09h - вывод строки**

**mov dx,press ;В DX адрес строки**

**int 21h ;Обращение к функции DOS**

**mov ah,08h ;Функция DOS 08h - ввод символа без эха**

**int 21h ;Обращение к функции DOS**

**mov ax,4C00h ;\**

**int 21h ;/ Завершение программы**

**press:**

**db 13,10,'Press any key...$'**

**23. Организация подпрограмм. Команды для организации подпрограмм, примеры.**

Подпрограммы позволяют сократить объём текста программы, применять модульный принцип построения программ, использовать одни и те же подпрограммы в различных программах, что значительно сокращает время создания программ и уменьшает время отладки. Описание подпрограммы в языке ассемблер имеет следующую структуру:

**<имя процедуры> proc <тип процедуры>**

**операторы тела подпрограммы**

**ret [<выражение>]**

**<имя процедуры> endp**

**<тип процедуры> - определяет тип перехода: near (близкий), far (дальний).**

Вызов подпрограммы выполняется командой call [<тип вызова> ptr] <адрес процедуры> < тип вызова > - near (word) или far (dword). **Пример.** Пусть в сегменте данных описаны переменные:

**FADDR dd ?**

**NADDR dw ?**

**в сегменте кода описаны подпрограммы:**

**FPROC proc far**

**FPROC endp**

**NPROC proc**

**NPROC endp**

**Рассмотрим различные примеры команд вызова:**

**call FPROC ;дальний вызов п/п FPROC**

**call FPROC ;дальний вызов п/п FPROC**

**call FADDR ;дальний вызов п/п, чей адрес в FADDR**

**call NADDR ;ближний вызов п/п, чей адрес в NADDR**

**call dx ; ближний вызов п/п, чей адрес в DX**

**call word ptr [BX] ; косвенный ближний вызов п/п**

**call dword ptr [BX] ; косвенный дальний вызов п/п**

**24. Организация системы прерываний. Команды прерываний, примеры** Прерывание в микропроцессорах — это событие, вызывающее прекращение рабо­ты основной программы и переход к выполнению процедуры, или подпрограммы, предназначенной для его обработки. Совокупность аппаратных средств, команд и программ, обслуживающих прерывания, образуют систему прерываний. Прерыва­ние инициируется специальным сигналом, при появлении которого система преры­ваний изменяет ход выполнения программы. **Можно выделить три вида прерываний:аппаратные** - инициируются внешними устройствами с помощью сигнала запроса на прерывание; **программные** - инициируются специальными; **исключения**- вызванные возникновением особых условий выполне­ния текущей команды**. Прерывания используются**:для обмена информацией между процессором и внешним устройством;в аварийных ситуациях; при переполнении и антипереполнении при выполнении операции с плавающей точкой;

Вызов прерывания осуществляется с помощью директивы INT номер\_прерывания.  
00h – 1Fh – прерывания BIOS  
20h – 3Fh – прерывания DOS  
40h – 5Fh – зарезервировано  
60h – 7Fh – прерывания пользователя  
80h – FFh – прерывания Бейсика

**25. Понятие параллельных вычислений. Требования к архитектурным принципам построения**

В общем плане под *параллельными вычислениями* понимаются процессы обработки данных, в которых одновременно могут выполняться нескольких машинных операций. Достижение параллелизма возможно при выполнимости требований:

**избыточность элементов вычислительной системы - организация избыточности может осуществляться в следующих основных формах:**

*использование специализированных устройств* таких, например, как отдельных процессоров для целочисленной и вещественной арифметики, устройств многоуровневой памяти (регистры, кэш);

*дублирование устройств ЭВМ* путем использования, например, нескольких однотипных обрабатывающих процессоров или нескольких устройств оперативной памяти.

Дополнительной формой обеспечения параллелизма может служить **конвейерная реализация** обрабатывающих устройств, при которой выполнение операций в устройствах представляется в виде исполнения последовательности составляющих операцию подкоманд;

При рассмотрении проблемы организации параллельных вычислений следует различать следующие возможные режимы выполнения независимых частей программы:

*многозадачный режим (режим разделения времени)*, при котором для выполнения процессов используется единственный процессор;

*параллельное выполнение*, когда в один и тот же момент времени может выполняться несколько команд обработки данных;

*распределенные вычисления*; данный термин обычно используют для указания параллельной обработки данных, при которой используется несколько обрабатывающих устройств, достаточно удаленных друг от друга и в которых передача данных по линиям связи приводит к существенным временным задержкам.

**26. Классификация вычислительных систем по систематике Флинна и пути достижения**

По-видимому, самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году М.Флинном.Классификация базируется на понятии *потока*, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. На основе числа потоков команд и потоков данных Флинн выделяет четыре класса архитектур: SISD,MISD,SIMD,MIMD.

**SISD** (single instruction stream / single data stream) - одиночный поток команд и одиночный поток данных. К этому классу относятся, прежде всего, классические последовательные машины, или иначе, машины фон-неймановского типа, например, PDP-11 или VAX 11/780.

**SIMD** (single instruction stream / multiple data stream) - одиночный поток команд и множественный поток данных. В архитектурах подобного рода сохраняется один поток команд, включающий, в отличие от предыдущего класса, векторные команды. Это позволяет выполнять одну арифметическую операцию сразу над многими данными - элементами вектора.

**MISD**(multiple instruction stream / single data stream) - множественный поток команд и одиночный поток данных. Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных.

**MIMD** (multiple instruction stream / multiple data stream) - множественный поток команд и множественный поток данных. Этот класс предполагает, что в вычислительной системе есть несколько устройств обработки команд, объединенных в единый комплекс и работающих каждое со своим потоком команд и данных

**27.** **Варианты организации систем MIMD класса «мультипроцессоры». Особенности архитектур NUMA (NCC-NUMA, CC-NUMA, COMA).**

MIMD (Multiple Instruction, Multiple Data) – системы c множественным потоком команд и множественным потоком данных. К подобному классу относится большинство параллельных многопроцессорных вычислительных систем. Системы, в которых обеспечивается огерентность локальных кэшей разных процессоров - CC-NUMA Основным параметром классификации паралелльных компьютеров является наличие общей (SMP) или распределенной памяти (MPP). Нечто среднее между SMP и MPP представляют собой NUMA-архитектуры, где память физически распределена, но логически общедоступна. Кластерные системы являются более дешевым вариантом MPP. При поддержке команд обработки векторных данных говорят о векторно-конвейерных процессорах, которые, в свою очередь могут объединяться в PVP-системы с использованием общей или распределенной памяти. Все большую популярность приобретают идеи комбинирования различных архитектур в одной системе и построения неоднородных систем. При организациях распределенных вычислений в глобальных сетях (Интернет) говорят о мета-компьютерах, которые, строго говоря, не представляют из себя параллельных архитектур.

**28.** **Варианты организации систем MIMD класса «мультикомпьютеры»  (NORMA). Понятия транспьютерной системы, кластерной системы. Особенности построения архитектур МРР**

Мультикомпьютеры (многопроцессорные системы с распределенной памятью) уже не обеспечивают общего доступа ко всей имеющейся в системах памяти ( no-remote memory access или NORMA). При всей схожести подобной архитектуры с системами с распределенной общей памятью , мультикомпьютеры имеют принципиальное отличие: каждый процессор системы может использовать только свою локальную память, в то время как для доступа к данным, располагаемым на других процессорах, необходимо явно выполнить операции передачи сообщений (message passing operations). Данный подход применяется при построении двух важных типов многопроцессорных вычислительных систем - массивно-параллельных систем (massively parallel processor или MPP) и кластеров (clusters). Среди представителей первого типа систем — IBM RS/6000 SP2, Intel PARAGON, ASCI Red, транспьютерные системы Parsytec и др.; примерами кластеров являются, например, системы AC3 Velocity и NCSA NT Supercluster. Следует отметить чрезвычайно быстрое развитие многопроцессорных вычислительных систем кластерного типа – общая характеристика данного подхода приведена, например, в обзоре . Под кластером обычно понимается множество отдельных компьютеров, объединенных в сеть, для которых при помощи специальных аппаратно-программных средств обеспечивается возможность унифицированного управления ( single system image ), надежного функционирования ( availability ) и эффективного использования ( performance)

**29. Эволюция сетей эвм. Обобщенная структура вычислительной сети и назначение основных компонентов**

Первые вычислительные цепи появились в 60-х годах. По сути, они произвели своего рода техническую революцию, сравнимую с появлением первых ЭВМ, т. к. была предпринята попытка объединить технологию сбора, хранения, передачи и обработки информации на ЭВМ с техникой связи.

В настоящее время в мире зарегистрировано боле 200 глобальных сетей, 54 из которых, созданы в США, 16 – в Японии.

Компьютерные сети бывают локальными и распределёнными.

Локальной называется компьютерная сеть, объединяющая компьютеры, расположенные в одном здании или в соседних зданиях. Если же соединённые компьютеры находятся в разных частях города, а иногда и в разных городах и странах, то такие сети называют распределёнными. Распределённые сети мирового масштаба также называют глобальными сетями. Интернет и является самой известной глобальной компьютерной сетью.

Объединение ЛВС и глобальных сетей позволило получить доступ к мировым информационным ресурсам.

Серверы – служат для преобразования и передачи информации от одной ЭВМ к другой по каналам связи и коммутационным устройствам (host - ЭВМ). К качеству и мощности серверов предъявляются повышенные требования.

По способу передачи информации вычислительные сети делятся на сети коммутации каналов, сети коммутации сообщений, сети коммутации пакетов и интегральные сети.

При коммутации каналов – сообщение между клиентами осуществляется по прямому каналу неизменному в течение всего сеанса.

При коммутации сообщений информация передаётся порциями, называемыми сообщениями.  Каждым сервером осуществляется приём информации, её сборка, проверка, маршрутизация и передача сообщения. Недостатком коммутации сообщений является низкая скорость передачи данных и невозможность проведения диалога между клиентами.

При коммутации пакетов обмен производится короткими пакетами фиксированной структуры.

**30. Эталонная модель взаимодействия открытых систем (эмвос). Основные понятия и определения. Особенности организации. Назначения и функции отдельных уровней эмвос.**

Цель ЭМВОС ­­­­– обеспечение «открытости» сетевых архитектур для обеспечения подключения к конюктурной сети различного оборудования без дополнительной доработки сетевого, программного и аппаратного обеспечения.

Особенностью ЭМВОС явл. разработка единого подхода к организации протоколов и интерфейсов разл уровня, при этом кажд уровню ставится в соотв набор опред ф-ий, связ с реш конкр задачи по организ взаимод открыт систем. ЭМВОС – не определяет средства реализации протоколов, а только специфицирует их. Поэтому функции каждого уровня могут быть реализованы различными аппаратными и программными средствами.

Взаимодействие между 2-мя смежными уровнями осуществляется через стандартно организованный межуровневый интерфейс Т. е. существуют точки доступа, через которые уровни обмениваются информацией. **Уровни:**

**Физический** – выполняет функции физического соединения двух сетевых систем. Определяет правила физического кодирования сигналов, набор управленческих процедур.

**Канальный** –функции обнаружения и исправления ошибок, при передачи бит информации по физическим линиям связи

**Сетевой** – обеспечивает адресацию сетевых устройств, маршрутизацию передаваемых блоков информации , повышение надежности передачи информации.

**Транспортный** – обеспечивает требуемое качество обслуживания пользовательских процессов, увеличение надежности информационного канала.

**Сеансовый** – предназначен для организации сеансов связи, выполняет функции восстановления в случае разрыва сеансовых соединений..

**Представительный** – предназначен для интерпретации пользовательских данных и данных управленческих процедур в единой для взаимодействующих систем форме.

***31. Способы организации обмена данными в вычислительной сети. Методы коммутации информации.:***

*Дейтаграммный способ* – передача осуществляется как совокупность независимых пакетов. Каждый пакет двигается по сети по своему маршруту и пользователю пакеты поступают в произвольном порядке.

**- Достоинства**: простота процесса передачи.

**- Недостатки:** низкая надежность засчет возможности потери пакетов.

Логический канал - передача последовательности связанных в цепочки пакетов, сопровождающихся установкой предварительного соединения и подтверждением приема каждого пакета.

Виртуальный канал – это логический канал с передачей по фиксированному маршруту последовательности связанных в цепочки пакетов.

Коммутация — это процесс соединения различных абонентов коммуникационной сети через транзитные узлы.

Cпособы коммутации:

**Коммутация каналов** образование непрерывного составного физического канала из последовательно соединенных отдельных канальных участков для прямой передачи данных между узлами.

**Достоинства** коммутации каналов: постоянная и известная скорость передачи данных;правильная последовательность прихода данных;

**Недостатки** коммутации каналов:возможен отказ сети в обслуживании запроса на установление соединения; обязательная задержка перед передачей данных из-за фазы установления соединения.

1. коммутация пакетов (packet switching);

Коммутация пакетов - это особый способ коммутации узлов сети, который специально создавался для наилучшей передачи компьютерного трафика (пульсирующего трафика).

1. коммутация сообщений (message switching).

Коммутация сообщений – разбиение информации на сообщения, каждый из которых состоит из заголовка и информации.

**32.Способы организации виртуальных каналов и управления потоками данных:протоколы с остановками окна в процедурах управления потоками данных**

Существует режим работы сети - передача пакетов по виртуальному каналу. В перед тем, как начать передачу данных между двумя конечными узлами, должен быть установлен виртуальный канал, единственный маршрут, соединяющий эти конечные узлы. Виртуальный канал может быть динамическим или постоянным. Динамический виртуальный канал устанавливается при передаче в сеть специального пакета - запроса на установление соединения. Этот пакет проходит через коммутаторы и проклад. виртуальный канал. Нужно пере подключать при

***1)Протокол с остановками и ожиданием:***При этой процедуре одновременно может передаваться только один кадр. После этого передающая сторона ждет подтверждения. Если поступит отрицательное подтверждение или произойдет просрочка времени ожидания ответа, кадр передается повторно.*2)****N-возвращения, или непрерывная передача:*** кадры передаются непрерывно без ожидания подтверждения. при получении отрицательного подтверждения или истечения установленного времени ожидания неподтвержденный кадр и все последующие кадры передаются вновь. ***3)Выборочное повторение:*** В этом случае повторная передача требуется только для кадра, о котором поступило отрицательное подтверждение или для которого истекло установленное время ожидания. **Метода окна** Для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе, без получения на эти кадры ответных квитанций. Обычно кадры при обмене нумеруются циклически, от 1 до W. При отправке кадра с номером 1 источнику разрешается передать еще W-1 кадров до получения квитанции на кадр 1. Если же за это время квитанция на кадр 1 так и не пришла, то процесс передачи приостанавливается, и по истечению некоторого тайм-аута кадр 1 считается утерянным и он передается снова

**33. Методы маршрутизации информацииПротоколы RIP,OSPF**

1)Простая-это маршрут-ция, не изменяющ-ся при измен. тополог. и сост. БСОД: **случайная** – это передача пакета в случ. выбранном направлен., кроме направления, по кот. поступил пакет в узел.

**лавинная** – передача пакета во всех направлен, кроме направл., по кот. он поступил. Хотя б одно из направлен. обеспеч. доставку пакета за мин. время. **по предыдущему опыту** – это передача пакета в направлен., выбираемом на основе анализа потока, проходящего через узел.

2) **фиксированная** – это способ выбора направлен. передачи по табл. маршрут-ции, устанавливающей направлен. передачи для кажд. узла назначен. **однопутевая** – строится на осн. единств. пути между взаимод. абонентами. **многопутевая** – если в табл. маршрут. указыв. неск. возможных путей передачи пакета и вводится правило выбора целесообразности пути.

3) адаптивная – способ выбора направлен. передачи, учитывающий изменен сост БСОД: **распределенная** – основана на использ. инф., получаемой от соседн. узлов сети. Кажд. узел формирует в табл. маршрутов с учетом минимизации задержки передачи. **централизованная**– основана на использ. инф., получаемой от центра маршрутиз. Кажд. узел формирует сообщения о своем состоян, кот. передается центром маршрут-ции. Он на осн. этих сообщен. формир. маршрутные таблицы, кот. рассылаются узлам. **Протокол RIP** является дистанционно-векторным протоколом маршрутизации. В качестве метрики в протоколе RIP используется количество транзитных устройств или переходов (hop count – прыжок пакета) из одной сетевой структуры в другую. Максимальное число таких переходов равно 15. А все сети, число переходов до которых превышает 15, считаются недостижимыми. **OSPF** — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала и использующий для нахождения кратчайшего пути алгоритм Дейкстры. ... Протокол OSPF представляет собой протокол внутреннего шлюза

**34. Принципы межсетевого взаимодействия для протоколов без установления логического соединения. Межсетевые дейтаграммы.**

**Шлюз** – устройство, объединяющее несколько подсетей и выступающее для каждой подсети как ГВМ. В каждой подсети шлюз адресуется по тем же правилам, что и любое другое ГВМ этой подсети. **Различают шлюзы 2 типов:**

1. только считывают сообщения, поступающие из сети, определяют маршрут последней передачи сообщения и передают сообщение в др. сеть, предварительно вложив их в упаковку уже этой сети- трансляторы.

2. осуществляют преобразование одного или нескольких протоколов, используемых в сети, в протоколы др. сети путем замены сообщений, принятых из одной сети др. сообщениями

Все ГВМ обмениваются стандартными пакетами – межсетевыми дейтаграммами (МД). Адресная информация содержится в заголовке МД. ГВМ отправитель подготавливает дейтаграмму которая содержит заголовок и исходное сообщение, затем выбирается шлюз собственной подсети, который будет использован для дальнейшей передачи дейтаграммы через объединенную сеть. ГВМ отправитель посылает дейтаграмму вложенную в пакет подсети к этому шлюзу. Шлюз принимает пакет извлекает из него дейтаграмму, анализирует заголовок IP, определяет адрес следующего шлюза одной из подсетей с которыми он непосредственно соединен.

Назначение IP – передача блоков данных (МД) от источников к адресатам.

**IPv4** использует 32-битную схему адресов, позволяющую использовать от 2 до 32 адресов или чуть более 4 миллиардов адресов. Его основной задачей в основном является передача блоков данных от узла отправки к узлу назначения. IPv4-адреса, в большинстве случаев, записываются способом с 4 десятичными числами от 0 до 255 и разделены точкой. Например: 172.128.1.2.Существует минимальный адрес и максимальный адрес; возможный минимальный адрес - 0.0.0.0.0, а максимальный - 255.255.255.255.255.

**35. Типы IP адресов и классы подсетей.Протокол ICMP.Межсетевые протоколы** Изначально адресация в сетях IP осуществлялась на основе классов: первые биты определяли класс сети, а по классу сети можно было сказать — сколько бит было отведено под номер сети и номер узла. Всего существовало 5 классов:



Типы и виды IP-адресов: Локальный, внутренний, внутрисетевой, «серый» — IP-адрес внутри частной сети провайдера или домашней сети. Внешний, «белый» — это IP-адрес, который входит в диапазон IP-адресов предназначенных для сети Интернет). Динамические -постоянно меняется. Такие IP-адреса меняются при каждом соединением с оборудованием, которое раздает адреса на подключающиеся к нему устройства. Статический - неизменный. Как правило, статические IP-адреса прописываются вручную на сетевой карте (если адрес «серый») или закрепляются за регистрационными данными абонента (в случае с «белым»).

ICMP (ang. Internet Control Message Protocol) - это один из протоколов сетевого уровня в модели ISO/OSI. Его задачей является обслуживание функции контроля правильности работы сети. Устройство пакета ICMP следующее:

• Заголовок в 4 байтов - первый байт определяет тип пакета, второй - код операции, третий и четвёртый представляют собой контрольную сумму.

• Поле данных с долготой зависимой от типа пакета и его функции. В некоторых случаях может быть установленным с уровня инструмента, напр. догадливая команда ping в Виндоуз устанавливает размер данных пакета ECHO\_REQUEST на 32 байта, а версия, встречающаяся в системах Юникс на 56 байтов.

**36. Понятие маски. Использование масок для структуризации ip адресов.**

Маска подсети — битовая маска для определения по IP-адресу адреса подсети и адреса узла (хоста, компьютера, устройства) этой подсети. В отличие от IP-адреса маска подсети не является частью IP-пакета.

Благодаря маске можно узнать, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети.

Для стандартных классов сетей маски имеют следующие значения:

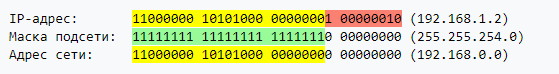
255.0.0.0 - маска для сети класса А,

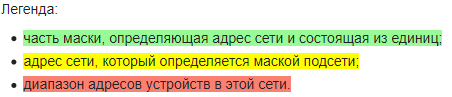
255.255.0.0 - маска для сети класса В,

255.255.255.0 - маска для сети класса С.

В масках, которые использует администратор для увеличения числа сетей, количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты.

Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию поразрядной конъюнкции (побитовое И). Например, в случае более сложной маски:

****

****

**37. Классы транспортных протоколов и типы сетевых соединений. Функции транспортной службы.**

Протоколы 3х верхних уровней ориентированы на обслуживание прикладных процессов и при этом не осуществляют управление информационным обменом процессами. Реализацию этих функций возлагают на транспортные протоколы основное назначение которых является обеспечение надёжной связи между прикладными процессами. **АННР** - протокол обмена данными ГВМ. **ТСР** - протокол управления передачи. **Х.224.Тр** - транспортный протокол. **Х.214.Тр** - транспортная служба.

Соотношение типов сетевых соединений и классов транспортных протоколов. **Тип А** определяет сетевое соединение с приемлемой частотой ошибок и допустимых сбоев о которых сообщается транспортной службе. **Тип В** определяется как соединение с приемлемой частотой ошибок, но с неприемлемой интенсивностью сигнализирующих повреждений. **Класс ТР0** определяет простое транспортное соединения для которого определен минимум функций. Для данного класса надо установить транспортное соединение. При сбоях он не предусматривает восстановление соединений. В класс ТР1 включено восстановления при сбоях. Классы ТР2, ТР3, ТР4 предоставляют дополнительные услуги по повышению надёжности сетевого соединения

TCP— «гарантированный» транспортный механизм с предварительным установлением соединения, предоставляющий приложению надёжный поток данных, дающий уверенность в безошибочности получаемых данных, перезапрашивающий данные в случае потери и устраняющий дублирование данных. TCP позволяет регулировать нагрузку на сеть, а также уменьшать время ожидания данных при передаче на большие расстояния

Функционирование транспортных протоколов вычислительной сети сводится к формированию сообщений;Представлению их в пакетной форме для отправки в СПД;Сборке сообщений из поступающих пакетов;Интерпретации поступающих команд и сообщений-ответов в соответствии с процедурной характеристикой протокола.

**38. Процедура передачи данных и метод окна в TCP. Адаптивные свойства протокола TCP.**

На протокол TCP возложена задача – обеспечить надежную доставку сообщений, используя в качестве основы ненадежный дейтаграммный протокол IP. Установленные на конечных узлах протокольные модули TCP решают задачу обеспечения надежного обмена данными путем установления между собой логических соединений. Благодаря логическому соединению TCP следит, чтобы передаваемые сегменты не были потеряны, не были продублированы и пришли к получателю в том порядке, в котором были отправлены. При установлении логического соединения модули TCP договариваются между собой о параметрах процедуры обмена данными. В протоколе TCP каждая сторона соединения посылает противоположной стороне следующие параметры:

-максимальный размер сегмента, который она готова принять; -максимальный объем.

Создание соединения означает также выделение операционной системой на каждой стороне соединения определенных системных ресурсов: для организации буферов, таймеров, счетчиков. **АЛГОРИТМ СКОЛЬЗЯЩЕГО ОКНА**. частный случай квитирования. При установлении соединения обе стороны договариваются о начальном номере байта, с которого будет вестись отсчет в течение всего данного соединения. У каждой стороны свой начальный номер. Нумерация байтов в пределах сегмента осуществляется так, что первый байт данных сразу вслед за заголовком имеет наименьший номер, а следующие за ним байты имеют следующие порядковые номера. В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в которое помещает число (номер подтверждения), на единицу превышающее максимальный номер байта в полученном сегменте. Квитанция (подтверждение) в протоколе TCP посылается только в случае правильного приема данных, отрицательные квитанции не посылаются. Таким образом, отсутствие квитанции означает либо потерю сегмента, либо прием искаженного сегмента, либо потерю квитанции.

**39.**  **Особенности организации модели взаимодействия для ЛВС. Протоколы и стандарты ЛВС.**

Локальная вычислительная сеть объединяет абонентов, находящихся на небольшом расстоянии друг от друга (в пределах 10-15 км). Обычно такие сети строятся в пределах одного предприятия или организации.

Системы на базе ЛВС решают задачи: хранение данных; обработка данных; организация доступа пользователей к данным; передача данных и результатов их обработки пользователям. Существует множество способов классификации сетей. Основным критерием классификации принято считать способ администрирования.

Чаще всего локальные сети построены на технологиях Ethernet. Для построения простой локальной сети используются маршрутизаторы, коммутаторы, точки беспроводного доступа, беспроводные маршрутизаторы, модемы и сетевые адаптеры. Иногда в локальной сети организуются рабочие группы - формальное объединение нескольких компьютеров в группу с единым названием. **Logical Link Control** — подуровень управления логической связью — по стандарту [IEEE 802](https://ru.wikipedia.org/wiki/IEEE_802) — верхний, осуществляет: управление передачей данных; проверку и правильность передачи информации по соединению.

**Структура кадра:** По своему назначению все кадры уровня LLC подразделяются на: **Информационные кадры** ( (1) предназначены для передачи) информации в процедурах с установлением логического соединения и должны обязательно содержать поле информации. **Управляющие кадры** (1)..команд и ответов в процедурах с установлением логического соединения **Ненумерованные кадры** (1)..ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, с установлением логического соединения — установление и разъединение логического соединения, а также информирование об ошибках.

**40. Технология Ethernet. Иерархия стандартов Ethernet. Основные этапы доступа к среде. Обработка коллизий.**

Ethernet — пакетная технология компьютерных сетей. Иерархия стандартов Ethernet(10Base-5, 10Base-2, 10Base-T, 10Base-F, 100Base-T) **.**Этапы доступа: Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения. Чтобы получить возможность передавать кадр, станция должна убедиться, что разделяемая среда свободна. Признаком незанятости среды является отсутствие на ней несущей частоты. Если среда свободна, то узел имеет право начать передачу кадра.

Узел обнаруживает, что среда свободна, и начинает передавать свой кадр. Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, а затем посылает по кабелю кадр-ответ. Адрес станции-источника содержится в исходном кадре, поэтому станция-получатель знает, кому нужно послать ответ.

Коллизия – это ситуация, когда содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации — методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала. Обработка коллизий может происходить следующим образом. Если ячейка, свободна, то имя записывается в поле имени, а в поле признака устанавливается признак конечного элемента списка. Если строка занята, то новое имя и имя, находящееся в ячейке, сравниваются. При совпадении функции HASH присваивается значение - 1 и ее выполнение завершается.

**41-43 Технология Ethernet – стандарт 10Base5, 10Base2, 10BaseT. Стандарты оптоволоконных технологий. Домен коллизий и логическая структуризация сетей Ethernet.**

Исторически первые сети технологии Ethernet были созданы на коаксиальном кабеле диаметром 0,5 дюйма.

**10Base-5** - коаксиальный кабель диаметром 0,5 дюйма. Имеет волновое сопротивление 50 Ом. Макс длина сегмента - 500 м.

**10Base-2** - коаксиальный кабель диаметром 0,25 дюйма. Имеет волновое сопротивление 50 Ом. Макс длина сегмента - 185 м.

**10Base-T** - кабель на основе неэкранированной витой пары

Число 10 –битовая скорость передачи данных этих стандартов , а слово «Base» - метод передачи на одной базовой частоте 10. **Оптоволоконные стандарты** в качестве основного типа кабеля для 10 мегабитного Ethernet рекомендуют дешевое многомодовое оптическое волокно, обладающее полосой пропускания 500-800МГц при длине кабеля 1 км.

Домен коллизий — это часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части этой сети коллизия возникла. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Домен коллизий соответствует одной разделяемой среде. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий. Узлы, образующие один домен коллизий, работают синхронно, как единая распределенная электронная схема.

**Логическая структуризация** сетей Ethernet - сегментация единой разделяемой среды на несколько и: соединить полученные сегменты сети такими устройствами, как мосты, коммутаторы или маршрутизаторы. Логическая структуризация позволяет решить несколько задач, основные из них: повышение производительности, гибкости, безопасности и управляемости сети.

**42. Технология Fast Ethernet – структура физического уровня. Спецификации 100Base**

Технология Fast Ethernet является эволюционным развитием классической технологии Ethernet. Ее основными достоинствами являются: -Увеличение пропускной способности сегментов сети до 100 Мб/c; -Сохранение метода случайного доступа Ethernet;

-Сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных витой пары и оптоволоконного кабеля.

**Физический уровень включает три элемента:**

-уровень согласования (reconciliation sublayer)

-независимый от среды интерфейс (Media Independent Interface, МП)

-устройство физического уровня (Physical layer device, PHY)

Официальный стандарт 100Base-T (802.3u) установил три различных спецификации для физического уровня для поддержки следующих типов кабельных систем:

-100BASE-TX обеспечивает передачу данных со скоростью до 100 Мбит/с по кабелю, состоящему из двух витых пар 5-й категории.. Длина линии связи ограничена 100 метрами, но по одному стандартному кабелю, имеющему 4 пары, можно организовать два 100-мегабитных канала связи.

-100BASE-T4 обеспечивает передачу данных со скоростью до 100 Мбит/с по кабелю, состоящему из четырёх витых пар 3-й категории.

-100BASE-FX использует волоконно-оптический кабель и обеспечивает связь излучением с длиной волны 1310 нм по двум жилам — для приёма (RX) и для передачи (TX). Длина сегмента сети может достигать 400 метров в полудуплексном режиме и 2 километров в полнодуплексном при использовании многомодового волокна.