***ОВС ПИ19-1***

1. ***Особенности процессорных архитектур. CISC и RISC архитектура. Их краткая характеристика.***

Расширение набора команд, увеличение числа способов адресации, введение сложных команд сопровождаются увеличением длины кода команды, в первую очередь кода операции, что может приводить к использованию “расширяющегося кода операции”, увеличению числа форматов команд. Это вызывает усложнение и замедления процесса дешифрации кода операции и других процедур обработки команд. Возрастающая сложность процедур обработки команд заставляет прибегать к микропрограммному управлению устройствами с управляющей памятью вместо более быстродействующего устройства управления с “жесткой” логикой.

Микропроцессор — это устройство, представляющее собой одну или несколько больших интегральных схем(БИС), выполняющих функции процессора ЭВМ. Классическое вычислительное устройство состоит из арифметического устройства (АУ), устройства управления (УУ), запоминающего устройства (ЗУ) и устройства ввода-вывода (УВВ).

Сложный процессор с микропрограммным управлением трудно реализовать на одном кристалле, а это ведет к увеличению длинны межмодульных связей, таким образом устроены большинство процессоров с полным набором команд.

Напротив, при сокращении количества команд до некоторого оптимального значения, можно сократить длину команд и упростить управляющее устройство МП. Поэтому при проектировании структуры МП выделилось два направления в отношении набора системы команд:

· CISC (Complicated Instruction Set Computer — использующий полный набор команд). Традиционная архитектура с широкой системой команд МП.

Характеризуется следующим набором свойств:

· большим числом различных по формату и длине команд;

· введением большого числа различных режимов адресации;

· обладает сложной кодировкой инструкции.

Процессору с архитектурой CISC приходится иметь дело с более сложными инструкциями неодинаковой длины. Выполнение одиночной CISC-инструкции может происходить быстрее, однако обрабатывать несколько таких инструкций параллельно сложнее.

Достоинства архитектуры CISC

* Компактность наборов инструкций уменьшает размер программ и уменьшает количество обращений к памяти.
* Наборы инструкций включают поддержку конструкций высокоуровневого программирования.

Недостатки архитектуры CISC

* Нерегулярность потока команд.
* Высокая стоимость аппаратной части.
* Сложности с распараллеливанием вычислений.

· RISC (Reduced Instruction Set Computer). Архитектура с сокращенным набором команд. Все команды одинакового формата с простой кодировкой. Обращение к памяти происходит посредством команд загрузки и записи, остальные команды типа регистр-регистр. Команда, поступающая в CPU, уже разделена по полям и не требует дополнительной дешифрации.

Часть кристалла освобождается для включения дополнительных компонентов. Степень интеграции ниже, чем в предыдущем архитектурном варианте, поэтому при высоком быстродействии допускается более низкая тактовая частота. Команда меньше загромождает ОЗУ, CPU дешевле. Программной совместимостью указанные архитектуры не обладают. Отладка программ на RISC более сложна. Данная технология может быть реализована программно-совместимым с технологией CISC (например, суперскалярная технология).

Поскольку RISC-инструкции просты, для их выполнения нужно меньше логических элементов, что в конечном итоге снижает стоимость процессора. Но большая часть программного обеспечения сегодня написана и откомпилирована специально для CISC-процессоров фирмы Intel. Для использования архитектуры RISC нынешние программы должны быть перекомпилированы, а иногда и переписаны заново.

Достоинства архитектуры RISC

1.снижение нерегулярности потока команд

2.обогащение пространственным параллелизмом

# 2. Виды обеспечения вычислительных систем. Определения. Примеры.

Выделяются следующие основные виды обеспечения вычислительных систем: программное, техническое (аппаратное), математическое, информационное, лингвистическое, организационное и методическое.

Программное обеспечение — это совокупность всей информации, данных и программ, которые обрабатываются компьютерными системами.

**Прикладное** программное обеспечение предоставляет пользователю дружественный интерфейс общения с ЭВМ. Для составления программы ему достаточно знать правила по использованию ЭВМ в терминах своей предметной области — предметно-ориентированную среду программирования. (Текстовые редакторы, веб браузеры, экспертные системы, профессиональное программное обеспечение - сапр и тд.)

**Системное** программное обеспечение является неотъемлемой частью ЭВМ. Это программы, управляющие работой компьютера и выполняющие различные вспомогательные функции, например, управление ресурсами компьютера, создание копий информации, проверка работоспособности устройств компьютера, выдача справочной информации о компьютере и др.

**Инструментальное ПО**

Специфическое обеспечение любой компьютерной техники. Его можно было бы отнести к прикладному, но из-за специфики применения его выделили в отдельный вид. Основная функция – отладка, настройка, переписывание программного кода.

Сюда входят компиляторы, отладчики, переводчики высокого уровня, редакторы, интерпретаторы и другие средства. Они необходимы, потому что техника не понимает человеческих слов. Чтобы ей «объяснить», что надо сделать, требуется специальный «машинный язык».

Постоянно пользоваться этим кодом базовым пользователям довольно сложно, поэтому были разработаны системы, которые позволяют переводить обычную речь в двоичную, привычную для ПК.

Разница между часто используемыми компиляторами и интерпретаторами заключается в том, что первый генерирует готовый файл, который можно запускать. А второй создает архив, который функционирует только с помощью самого сервиса.

**Математическое** обеспечение ЭВМ. Математическое обеспечение — МО состоит из фонда программ вычислительных алгоритмов, в частности, фонд содержит решения типовых задач математической физики.(Короче это математические проги). Математическое обеспечение – совокупность математических методов, моделей, алгоритмов обработки информации, используемых при решении задач в информационной системе (функциональных и автоматизации проектирования информационных систем).

**Аппаратное обеспечение** (англ. hardware – аппаратные средства, технические средства) включает в себя все физические части компьютера, но не включает программное обеспечение, которое им управляет, и не включает информацию, имеющуюся на компьютере.

Под аппаратным обеспечением понимают обычно все узлы, модули и блоки, составляющие компьютер или компьютерную систему. В современных компьютерах используется «открытая архитектура», т.е. состав аппаратного обеспечения компьютера можно изменить, поменяв один из модулей, или расширить, вставив дополнительный модуль.

**Лингвистическое** *обеспечение* представляет собой целостную совокупность формальных языков описания информации и алгоритмов ее обработки.

**Информационно обеспечение**

Совокупность программ и предварительно подготовленных данных, необходимых для работы этих программ.

Рассмотрим, например, систему автоматической проверки орфографии в редактируемом тексте. Ее работа заключается в том, что лексические единицы исходного текста сравниваются с заранее заготовленным массивом данных (словарем). В данном случае для успешной работы системы необходимо иметь кроме аппаратного и программного обеспечения специальные наборы словарей, подключаемые извне. Это пример информационного обеспечения вычислительной техники.

**Организационное обеспечение** системы — это организация следующих работ и мероприятий: создание структуры, обеспечивающей нормальную эксплуатацию системы; ввод в действие системы; опытно-производственная эксплуатация и проведение приемочных испытаний; ввод в промышленную эксплуатацию.

Включает положения, инструкции, приказы, штатные расписания, квалификационные требования и другие документы, регламентирующие организационную структуру подразделений проектной организации и их взаимодействие с комплексом средств автоматизированного проектирования.

**Методическое обеспечение** определяет объект проектирования, процесс проектирования и взаимосвязь между машиной и человеком (т.е. что проектировать и как управлять процессом проектирования).   
Разработка методического обеспечения требует знаний предметной области проектирования и технических средств, с тем, чтобы определить, какие задачи и этапы проектирования выполнять в автоматическом режиме, какие оставить за проектировщиком, а какие необходимо осуществлять в интерактивном режиме пользователя и системы.   
Методическое обеспечение включает также пакет документов, в которых отражены состав, правила отбора и эксплуатации средств автоматизации проектирования. Кроме того, в методических документах каждой системы излагается технология проектирования, дается описание циклов проектирования, типовых сочетаний программ, рационального распределения функций между человеком и ЭВМ.

1. ***Векторные и векторно-конвейерные вычислительные системы. Матричные вычислительные системы.***

***Векторные вычислительные системы*** - системы класса SIMD, в которых одна и та же заданная операция выполняется сразу над всеми компонентами векторов.

В задачах моделирования реальных процессов и объектов, для которых характерна обработка больших массивов чисел в форме с плавающей запятой, массивы представляются матрицами и векторами, а алгоритмы их обработки описываются в терминах матричных операций. Как известно, основные матричные операции сводятся к однотипным действиям над парами элементов исходных матриц, которые, чаще всего, можно производить параллельно. Для обработки массивов требуются вычислительные средства, позволяющие с помощью единой команды производить действие сразу над всеми элементами массивов - средства *векторной обработки*.

В средствах векторной обработки под *вектором* понимается одномерный массив данных (обычно в форме с плавающей запятой), размещенных в памяти ВС. Количество элементов массива называется *длиной вектора*. Многомерные массивы считаются наборами одномерных массивов-векторов.  
((Действия над многомерными массивами учитывают специфику их размещения. Способ размещения многомерного массива влияет на шаг изменения адреса элемента, выбираемого из памяти. Так, если матрица расположена в памяти построчно, адреса соседних элементов строки различаются на единицу, а для элементов столбца шаг равен четырем. При размещении матрицы по столбцам единице будет равен шаг по столбцу, а четырем - шаг по строке. В векторной концепции для обозначения шага, с которым элементы вектора извлекаются из памяти, применяют термин *шаг по индексу* (stride).))

***Векторно-конвейерные вычислительные системы*** относятся к классу SIMD-систем. Основные принципы, заложенные в архитектуру векторно-конвейерных систем:

· конвейерная организация обработки потока команд;

· введение в систему команд набора векторных операций, которые позволяют оперировать с целыми массивами данных.

Длина обрабатываемых векторов в современных векторно-конвейерных системах составляет, как правило, 128 или 256 элементов. Основное назначение векторных операций состоит в распараллеливании выполнения операторов цикла, в которых обычно сосредоточена большая часть вычислительной работы.

Современные векторно-конвейерные системы имеют иерархическую структуру:

· на нижнем уровне иерархии расположены *конвейеры операций* (например, конвейер (pipeline) сложения вещественных чисел, конвейер умножения таких же чисел и т.п.);

· некоторая совокупность конвейеров операций объединяется в *конвейерное функциональное устройство*;

· *векторно-конвейерный процессор* содержит ряд конвейерных функциональных устройств;

· несколько векторно-конвейерных процессоров (2-16), объединенных общей памятью, образуют вычислительный узел;

· несколько таких узлов объединяются с помощью коммутаторов, образуя либо NUMA-систему либо MPP-систему.

Типичными представителями такой архитектуры являются компьютеры CRAY J90/T90, CRAY SV1, NEC SX-4/SX-5. Уровень развития микроэлектронных технологий не позволяет в настоящее время производить однокристальные векторно-конвейерные процессоры, поэтому эти системы довольно громоздки и чрезвычайно дороги.

Каждая часть конвейера операций называется *ступенью конвейера операций*, а общее число ступеней - *длиной конвейера операций*.

Примером класса SIMD (системы c одиночным потоком команд и множественным потоком данных) являются ***матричные вычислительные системы.***

В матричных ВС имеется общее управляющее устройство, генерирующее поток команд и большое число процессорных элементов, работающих параллельно и обрабатывающих каждый свой поток данных. Таким образом, суммарная производительность системы оказывается равной производительности всех процессорных элементов.

Одним из первых матричных процессоров был SOLOMON. Он состоял из 1024 процессорных элемента, которые соединены в виде матрицы 32х32. Каждый элемент включает процессор, обеспечивающий выполнение последовательных поразрядных арифметических и логических операций, а также оперативное ЗУ – 16Кбайт. Общее устройство управления от которого по каналам связи передаются команды и общие константы. Длина слова – переменная от 1 до 128, разрядность устанавливается программно. В процессорном элементе используется так называемая модальная логика, которая позволяет каждому процессорному элементу выполнять (или нет) общую операцию в зависимости от значений обрабатываемых данных. В каждом процессорном элементе имеется специальный регистр на четыре состояния – регистр моды. Мода заносится в этот регистр устройством управления. При выполнении последовательности команд модальность передается в коде операции и сравнивается с содержимым регистра моды. Если есть совпадения, то операция выполняется. Имеется возможность передать операнды соседнему процессорному элементу. Взаимодействие с периферийным оборудованием осуществляется через внешний процессор.

|  |  |
| --- | --- |
| https://www.ok-t.ru/studopediaru/baza5/1970064989.files/image007.gif Рисунок. Структура матричного процессора | Двухмерный массив ПЭ обрабатывает поток команд, получаемых от центрального управляющего процессора (ЦУП). Каждая команда выполняется всеми процессорами одновременно, каждый ПЭ может обмениваться данными с ближайшим соседом, а также при необходимости пропускать выполнение команды. |

# 4. Метрики производительности конвейера

Чтобы охарактеризовать эффект, достигаемый за счет конвейеризации вычислений, обычно используют три метрики: ускорение, эффективность и производительность.

Под ускорением понимается отношение времени обработки без конвейера и при его наличии. Теоретически наилучшее время обработки входного потока из N значений TNK на конвейере с К ступенями и тактовым периодом TK можно определить выражением:

.

Формула отражает тот факт, что до появления на выходе конвейера результата обработки первого элемента должно пройти K тактов, а последующие результаты будут следовать в каждом такте.

В процессоре без конвейера общее время выполнения составляет NKTK. Таким образом, ускорение вычислений 5 за счет конвейеризации вычислений можно описать формулой:

.

При ускорение стремится к величине, равной количеству ступеней в конвейере.

Еще одной метрикой, характеризующей конвейерный процессор, является эффективность Е — доля ускорения, приходящаяся на одну ступень конвейера:

.

В качестве третьей метрики часто выступает пропускная способность или производительность Р — эффективность, деленная на длительность тактового периода:

.

При эффективность стремится к единице, а производительность – к частоте тактирования конвейера:

.

# 5. Классы конфликтов возникающих в конвейерах и способы их устранения.

***инфа с (***[***https://www.intuit.ru/studies/courses/604/460/lecture/10337?page=2***](https://www.intuit.ru/studies/courses/604/460/lecture/10337?page=2)***)***

***Конфликты*** - это такие ситуации в конвейерной обработке, которые препятствуют выполнению очередной команды в предназначенном для нее такте.

Конфликты делятся на три группы:

* *структурные,*
* *по управлению,*
* *по данным.*

## **Структурные конфликты**

Структурные конфликты возникают в том случае, когда аппаратные средства процессора не могут поддерживать все возможные комбинации команд в режиме одновременного выполнения с совмещением.

### **Причины структурных конфликтов:**

#### Не полностью конвейерная структура процессора, при которой некоторые ступени отдельных команд выполняются более одного такта.

**Этой ситуации можно было бы избежать двумя способами**. **Первый** предполагает увеличение времени такта до такой величины, которая позволила бы все этапы любой команды выполнять за один такт. Однако при этом существенно снижается эффект конвейерной обработки, так как все этапы всех команд будут выполняться значительно дольше, в то время как обычно нескольких тактов требует выполнение лишь отдельных этапов очень небольшого количества команд. **Второй способ** предполагает использование таких аппаратных решений, которые позволили бы значительно снизить затраты времени на выполнение действия, приводящего к появлению "пузырей" (например, использовать матричные схемы умножения). Но это приведет к усложнению схемы процессора и сокращению на кристалле места для реализации на этой БИС других, функционально более важных узлов.

#### Недостаточное дублирование некоторых ресурсов.

**Борьба с конфликтами** такого рода проводится путем увеличения количества однотипных функциональных устройств, которые могут одновременно выполнять одни и те же или схожие функции.

## **Конфликты по управлению**

**Конфликты по управлению** возникают при конвейеризации команд переходов и других команд, изменяющих значение счетчика команд.

***Простейший способ разрешения этой ситуации*** *- использование так называемого метода выжидания. Он заключается в замораживании операций в конвейере путем блокировки выполнения любой команды, следующей за командой условного перехода, до тех пор, пока не станет известным направление перехода. Привлекательность такого решения заключается в его простоте.* ***Главный недостаток*** *- резкое уменьшение преимуществ конвейерной обработки. В инженерных задачах примерно каждая шестая команда является командой условного перехода, поэтому приостановки конвейера при выполнении команд переходов до определения истинного направления перехода существенно скажутся на производительности процессора.*

*Можно несколько* ***улучшить эту ситуацию****, использую схему "задержанных переходов". При этом на стадии компиляции компилятор таким образом структурирует получаемый объектный код, чтобы сделать команды, следующие за командой перехода, действительными и полезными*

## **Конфликты по данным**

***Конфликты по данным*** *возникают в случаях, когда выполнение одной команды зависит от результата выполнения предыдущей команды.*

*Существует несколько типов конфликтов по данным:*

1. *Конфликты типа* ***RAW*** *(****R****ead* ***A****fter* ***W****rite - чтение после записи):*

*Уменьшение влияния конфликта типа* ***RAW*** *обеспечивается методом, который называется пересылкой или продвижением данных ( data forwardINg ), обходом ( data bypassINg ), иногда закороткой ( short-circuitINg ).*

*В этом случае результаты, полученные на выходах исполнительных устройств, помимо входов приемника результата передаются также на входы всех исполнительных устройств микропроцессора*

1. *Конфликты типа* ***WAR*** *(****W****rite* ***A****fter* ***R****ead - запись после чтения):*
2. *Конфликты типа* ***WAW*** *(****W****rite* ***A****fter* ***W****rite - запись после записи):*

*Устранение конфликтов по данным типов* ***WAR*** *и* ***WAW*** *достигается путем отказа от неупорядоченного исполнения команд, но чаще всего путем введения* ***буфера восстановления последовательности команд***

# 6. Сигналы. Объем информации. Количество информации и энтропия. Свойства информации .

Все действия в ЭВМ выполняются под управлением **сигналов**, вырабатываемых устройством управления (УУ). Управляющие сигналы формируются на основе информации, содержащейся в выполняемой команде, и признаков результата, сформированных предыдущей командой (если выполняемая команда является, например, командой условного перехода). Иными словами, устройство управления передаёт сигналы, определяющие те или иные действия в различных блоках ЭВМ (например, вид операции в АЛУ или сигнал считывания из ЗУ).

Свойство полноты информации негласно предполагает, что имеется возможность измерять количество информации. Важнейшим результатом теории информации является вывод о том, что в определенных, весьма широких условиях, можно, пренебрегая качественными особенностями информации, выразить ее количество числом, а следовательно, сравнивать количество информации, содержащейся в различных группах данных. Количеством информации называют числовую характеристику информации, отражающую ту степень неопределенности, которая исчезает после получения информации. Получаемая информация уменьшает число возможных вариантов выбора (т.е. неопределенность), а полная информация не оставляет вариантов вообще.

За единицу информации принимается один бит. Это **количество информации**, при котором неопределенность, т.е. количество вариантов выбора, уменьшается вдвое или, другими словами, это ответ на вопрос, требующий односложного разрешения — да или нет. Величина, характеризующая количество неопределенности в теории информации, обозначается символом H и имеет название энтропия, или информационная энтропия. **Энтропия** (H) – мера неопределенности, выраженная в битах. Если источник информации может находиться в одном из n дискретных состояний с вероятностью pi в каждом из них (i=1,2,...,n), то в качестве меры неопределённости можно ввести функцию H, называемую энтропией. Будем называть каждое возможное состояние источника информации сообщением. Энтропия i-го сообщения, по определению, равна

Очевидно также, что энтропия достоверного события (pi=1) равна нулю. Наоборот, чем менее вероятно некоторое событие, тем больше его энтропия, это и понятно, ведь более редким событиям приписывается большая информационная значимость. Энтропией источника называется среднее значение энтропии сообщения:



Это определение энтропии, предложенное Клодом Шенноном, и считается классическим.

Одной из характеристик любой информации является ее **объем**. В электронных устройствах есть свои единицы объема информации, которые показывают, сколько нужно места для ее хранения. Бит (bit) — минимальный объем информации в двоичной системе счисления, равен 0 или 1. Кодирование, хранение и передача информации с помощью этих двух символов облегчает работу вычислительных устройств. Удобство использования двоичных чисел привело к широкому применению их в вычислительной технике.

Следующая единица измерения информации называется байтом. 1 байт равен 8 битам. Обычно одним байтом кодируется один символ. Существуют также другие обозначения единиц компьютерной информации: Килобайт, Мегабайт, Гигабайт, Терабайт.

Информация, как и любой объект, обладает **свойствами**.

Атрибутивные свойства — это те свойства, без которых информация не существует. К данной категории свойств относится:

• **неотрывность информации от физического носителя и языковая природа информации**. Необходимо отметить, что хотя информация и неотрывна от физического носителя и имеет языковую природу она не связана жестко ни с конкретным языком, ни с конкретным носителем.

• **дискретность**. Содержащиеся в информации сведения, знания - дискретны, т.е. характеризуют отдельные фактические данные, закономерности и свойства изучаемых объектов, которые распространяются в виде различных сообщений, состоящих из линии, составного цвета, буквы, цифры, символа, знака.

• **непрерывность**. Информация имеет свойство сливаться с уже зафиксированной и накопленной ранее, тем самым, способствуя поступательному развитию и накоплению.

Прагматические свойства — это те свойства, которые характеризуют степень полезности информации для пользователя, потребителя и практики. Проявляются в процессе использования информации. К данной категории свойств относится:

• Объективность. Объективная информация – существующая независимо от человеческого сознания, методов ее фиксации, чьего-либо мнения или отношения.

• Достоверность.

• Полнота.

• Точность информации – степень ее близости к реальному состоянию объекта, процесса, явления и т. п.

• Ценность информации зависит от ее важности для принятия решения, решения задачи и дальнейшей применимости в каких-либо видах деятельности человека.

• Актуальность.

• Понятность. Если ценную и своевременную информацию выразить непонятно, то она, скорее всего, станет бесполезной. Информация будет понятной, когда она, как минимум, выражена понятным для получателя языком.

• Доступность. Информация должна соответствовать уровню восприятия получателя.

• Краткость.

***7. Устройство управления современного процессора. Определение. Микрокоманда. Микрооперация. Микропрограмма. Задачи, решаемые устройством управления***

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | **Устройство управления** (УУ) - узел микропроцессора, выполняющий управление компонентами. Оно вырабатывает распределенную во времени и пространстве последовательность внутренних и внешних управляющих сигналов, обеспечивающих выборку и выполнение команд. На этапе цикла выборки команды УУ интерпретирует команду, выбранную из программной памяти. На этапе выполнения команды в соответствии с типом реализуемой операции УУ формирует требуемый набор команд низкого уровня для арифметико-логического устройства и других устройств. Эти команды задают последовательность простейших низкоуровневых операций, таких, как пересылка данных, сдвиг данных, установка и анализ признаков, запоминание результатов и др. Такие элементарные низкоуровневые операции называют **микрооперациями**, а команды, формируемые устройством управления, называются **микрокомандами**. Последовательность микрокоманд, соответствующая одной команде, называется **микропрограммой**. В **задачи** устройства управления входит выборка и декодирование потока [инструкций](http://www.wikiznanie.ru/wikipedia/index.php?title=%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B8%D0%BA%D1%80%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80%D0%BD%D0%B0%D1%8F)&action=edit&redlink=1), выдача кодов функций в исполнительные устройства, принятие решений по признакам результатов вычислений, синхронизация узлов микропроцессора. | |

***8. Организация ввода/вывода в вычислительной системе. Системные и локальные шины. Устройства ввода/вывода***

**Вводом/выводом** называют передачу данных между ядром ЭВМ, включающим в себя процессор и ОП(оперативная память), и периферийными устройствами.

Система ввода-вывода – это единственное средство общения ЭВМ с внешним миром. Ее возможности в серийных ЭВМ представляют собой один из важнейших параметров, определяющих выбор машины для конкретного применения.

Существует три режима ввода-вывода:

· Программный ввод-вывод (нефорсированный).

· ввод-вывод по прерыванию (форсированный).

· Прямой доступ к памяти (ПДП).

Программный ввод-вывод. Инициирование и управление вводом-выводом осуществляет процессор по командам прикладной программы. Периферийные устройства играют пассивную роль и только сигнализируют о своем состоянии, в частности о готовности к операциям ввода-вывода.

Ввод-вывод по прерыванию. Операции ввода-вывода инициирует периферийное устройство, генерируя сигнал запроса прерывания, при этом процессор переключается на подпрограмму обслуживания данного периферийного устройства, вызвавшего прерывание. Непосредственно операциями ввода-вывода управляет процессор.

Прямой доступ к памяти. Процессор в передаче данных не участвует. Он отключается от системной магистрали, а все операции обмена данными идут под управлением специального управляющего устройства – контроллера ПДП. Этот режим используется для быстродействующих периферийных устройств, когда пропускной способности процессора недостаточно.

Передача данных осуществляется двумя способами: а) отдельными битами, и тогда промежуточный интерфейс называется последовательным; б) полными словами (например, целым байтом), и тогда промежуточный интерфейс называется параллельным.

Информация, передаваемая в процессе ввода-вывода, подразделяется: а) на собственно данные; б) управляющие данные.

Управляющие данные от процессора называются также командными словами или приказами. Они инициируют действия, не связанные непосредственно с передачей данных (запуск устройства, запрещение прерываний, установка режимов и т.д.).

Управляющие данные от периферийного устройства называются словами состояния. Они содержат информацию об определенных признаках (о готовности устройства к передаче данных, о наличии ошибок при обмене и т.д.). Состояние обычно представляется в декодированной форме – один бит для каждого признака.

Одним из простейших механизмов, позволяющих организовать взаимодействие различных подсистем, является единственная центральная шина, к которой подсоединяются все подсистемы.

**Системная шина** - шина, предназначенная для передачи данных между периферийными устройствами и центральным процессором, или между периферийными устройствами и оперативной памятью.

**Локальной шиной** называется шина, электрически выходящая непосредственно на контакты микропроцессора. Она обычно объединяет процессор, память, схемы буферизации для системной шины и ее контроллер, а также некоторые вспомогательные схемы. Типичными примерами локальных шин является PCI. Локальная шина предназначена для обеспечения непосредственного доступа процессора к периферийным устройствам (например, графическим или сетевым адаптерам). ((Системная шина ISA позволяет синхронизировать работу процессора и шины с разными тактовыми частотами. Она работает на частоте 8 МГц, что соответствует максимальной скорости передачи 16 Мбайт/с. Системная шина EISA обеспечивает адресное пространство в 4 Гбайт, 32-битовую передачу данных (в том числе и в режиме DMA), улучшенную систему прерываний и арбитраж DMA, автоматическую конфигурацию системы и плат расширения. Устройства шины ISA могут работать на шине EISA. Шина PCI (Peripheral Component Interconnect) поддерживает 32-битовый канал передачи данных между процессором и периферийными устройствами, работает на тактовой частоте 33 МГц и имеет максимальную пропускную способность 120 Мбайт/с.))

**Устройства ввода**. По большей части данная техника является датчиком, которые преобразует неэлектрические величин, в сигналы или же цифровой код. Таким образом человек и прочие аспекты, могут взаимодействовать с компьютером, работать с ним. Говоря более простым языком, устройство ввода – это такая техника, которая является переводчиком между вами и компьютером, помогая вам достичь полного понимания и качественной работы.

Самые популярные представители данного вида: Мышка, преобразует перемещения в сигнал, чтобы компьютер понимал, что вы производите те или иные действия. Клавиатура, посылает сигналы о нажатии клавиш для взаимодействия с ПК.

**Устройства вывода**. Второй тип, главная цель подобной техники – это преобразовать цифровой код, который подается аппаратным обеспечением компьютера, в неэлектрические сигналы или же графическую информацию. С устройствами вывода, человек взаимодействует ежедневно.

В список популярных представителей входят следующие виды техники: Монитор, переводит сигналы в удобную для человеческого восприятия форму, а именно в графическое изображение, таким образом пользователь может работать с ПК. Принтер – это устройство, которое преображает цифровые файлы и отпечатывает их на бумаге. Таким образом человек может преобразовать цифровой файл и физический вид. Акустическая система – переводит сигналы в звуковое сопровождение. В данную категорию входят колонки, наушники и прочие, подобные устройства. Имеется третья категория, которая совмещает две предыдущие, а именно устройства ввода-вывода. Они совмещают в себе качества двух предыдущих групп, таким образом могут, как принимать сигнал, так и раздавать его. В число подобных устройств входят: Интерактивные доски. Дисководы. Роутеры и подобные устройства.

# 9. Классификация вычислительных систем. Альтернативная классификация.

Под вычислительной системой (ВС) понимают совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенную для сбора, хранения, обработки и распределения информации.

Существует большое количество признаков, по которым классифицируют вычислительные системы. По назначению вычислительные системы делят на: универсальные и специализированные.

Специализированные системы ориентированы на решение узкого класса задач, в отличие от универсальных, предназначенных для широкого спектра задач.

По типу построения вычислительные системы разделяются на: -многомашинные-многопроцессорные

По типу ЭВМ или процессоров, используемых для построения ВС, различают: -однородные системы -неоднородные системы.

По методам управления элементами ВС различают: -централизованные -децентрализованные -со смешанным управлением.

По степени территориальной разобщенности вычислительных модулей ВС делятся на системы: территориально-сосредоточенные – это когда все компоненты располагаются в непосредственной близости друг от друга; распределенные – это когда компоненты могут располагаться на значительном расстоянии, например, вычислительные сети; структурно-одноуровневые – это когда имеется лишь один общий уровень обработки данных; многоуровневые (иерархические) структуры – это когда в иерархических ВС машины или процессоры распределены по разным уровням обработки информации, некоторые машины (процессоры) могут специализироваться на выполнении определенных функций.

По режиму работы ВС различают системы, работающие в оперативном и неоперативном временных режимах.

**Альтернативная классификация.**

Все вычислительные системы в зависимости от способа обработки информации, заложенного в их архитектуру, можно разделить на четыре класса. Разрядно-последовательные пословно-последовательные (n=m=1). В каждый момент времени такие компьютеры обрабатывают только один двоичный разряд. Представителем данного класса служит давняя система MINIMA с естественным описанием (1,1).

Разрядно-параллельные пословно-последовательные (n > 1 , m = 1). Большинство классических последовательных компьютеров, принадлежит к данному классу: IBM 701 с описанием (36,1), PDP-11 (16,1), IBM 360/50 и VAX 11/780 - с описанием (32,1).

Разрядно-последовательные пословно-параллельные (n = 1 , m > 1). Как правило, вычислительные системы данного класса состоят из большого числа одноразрядных процессорных элементов, каждый из которых может независимо от остальных обрабатывать свои данные. Типичными примерами служат STARAN (1, 256) и MPP (1,16384) фирмы Goodyear Aerospace, прототип известной системы ILLIAC IV компьютер SOLOMON (1, 1024) и ICL DAP (1, 4096).

Разрядно-параллельные пословно-параллельные (n > 1, m > 1). Большая часть существующих параллельных вычислительных систем, обрабатывая одновременно mn двоичных разрядов, принадлежит именно к этому классу: ILLIAC IV (64, 64), TI ASC (64, 32), C.mmp (16, 16), CDC 6600 (60, 10), BBN Butterfly GP1000 (32, 256).

# 10. Назначение, принципы построения и характеристики арифметико-логических устройств (АЛУ).

Арифметико-логическое устройство (АЛУ) (англ. arithmetic and logic unit, ALU) – устройство или же блок процессора, который предназначен для выполнения логических и математических операций над двоичными числами. АЛУ входит в состав любого микропроцессора.

Пример работы АЛУ на операции сложения

Функционально АЛУ состоит из двух регистров (Регистр1, Регистр 2), схемы управления и сумматора. Арифметическая операция выполняется по тактам:

● Значения операнда 1, участвующего в арифметической операции по шине данных поступает в Регистр 1

● Значения операнда 2, участвующего в арифметической операции по шине данных поступает в Регистр 2

● По шине инструкций поступает инструкция на выполнение операции в схему управления

● Данные из регистров поступают в сумматор, схема управления дает команду на выполнение сложения

● Результат сложения поступает в Регистр 1

● Признаки выполнения операции в АЛУ поступают в регистр флагов.

Характеристики АЛУ:

● Разрядность

● Состав операции

● Форматы обрабатываемых данных

● Способ построения и функционирования

● Быстродействие

● Надежность

# 11. Многоуровневая организация ЭВМ. Структурная организация и архитектура вычислительных систем.

**Многоуровневая организация ЭВМ**

В общем случае обработку информации на ЭВМ можно рассматривать в виде иерархической системы уровней, представленных в табл.

|  |  |  |
| --- | --- | --- |
| Пользователь данного уровня | Уровень | Примечания |
| Постановщик задач | **6 – концептуальный**  (язык спецификаций) | Задаются режимы и виды обработки данных, необходимые для решения задачи, состав системных ПС |
| Пользователь функционального ПО, решающий задачи из конкретной предметной области | **5 – проблемно-ориентированных ПС**  (входной язык пакета программ) | Уровень приложений для конкретной предметной области |
| Разработчик функциональных программных комплексов | **4 – промежуточного ПО**  (например, язык UML) | Middleware ( 1 - Delphi, Visual C;  2 - DCOM, CORBA, RMI) |
| Разработчик функциональных (прикладных) программ | **3 – языков высокого уровня** | Паскаль, СИ, С++, Java, Prolog |
| Системный программист,  прикладной программист | **2 – ассемблера** | Программирование фрагментов программ высокой эффективности |
| Системный программист | **1 – ОС** | Выполнения привилегированных команд, управление памятью |
| Программист/электронщик  (системный архитектор) | **0 – машинных команд** | Цифровое кодирование и представление команд |
| Программист/электронщик  (системный архитектор) | **(–1) – микрокоманд**  (микроархитектурный уровень) | Описание набора элементарных операций, реализующих машинные команды |
| Электронщик | **(–2) – межрегистровых передач** | Реализация элементарных операций как пересылок между регистрами |
| Электронщик  (технолог) | **(–3) – вентилей**  (цифровой логический уровень) | Технологический уровень, устройства машины представляются в виде интегральных схем |

Архитектура – это множество ресурсов ЭВМ, доступных пользователю на логическом уровне, без детализации способов взаимодействия процессоров, устройств памяти, внешних устройств и программных средств. При изучении архитектуры рассматривают:

- состав и характеристики процессоров, включая системы команд;

- состав и характеристики устройств памяти и ВУ;

- состав программных средств разработки ПО;

- вид ОС и режимы обработки данных.

Структурная организация – это способы распределения функций, установления связи и взаимодействия процессоров, устройств памяти и внешних устройств, используемые для реализации возможностей, заложенных в архитектуре. При изучении организации рассматривают:

- представление и форматы данных;

- уровни памяти и их взаимодействие;

- состав и форматы машинных команд;

- систему прерываний;

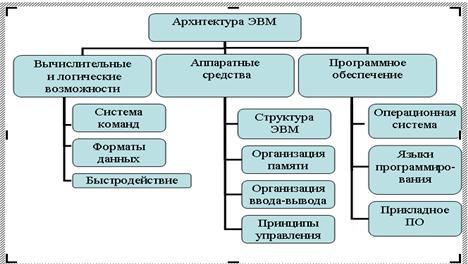
- способы обмена данными.

# 12 Основные архитектурные понятия. Типы команд. Типы данных Способы адресации.

**Под архитектурой ЭВМ** понимается совокупность общих принципов организации аппаратно-программных средств и их характеристик, определяющая функциональные возможности ЭВМ при решении соответствующих классов задач.

**Открытая архитектура** - предполагает наличие единого стандарта при разработке устройств, располагающихся на материнской плате и плате расширения.

Структура компьютера — это некоторая модель, устанавливающая состав, порядок и принципы взаимодействия входящих в нее компонентов.



В настоящее время наибольшее распространение в ЭВМ получили 2 типа архитектуры: **Фон Неймана** и **Гарвардская**. Обе они выделяют 2 основных узла ЭВМ: центральный процессор и память компьютера. Различие заключается в структуре памяти: в принстонской архитектуре программы и данные хранятся в одном массиве памяти и передаются в процессор по одному каналу, тогда как фон нейманская архитектура предусматривает отдельные хранилища и потоки передачи для команд и данных.

Согласно Джону фон Нейману, любая ЭВМ должна включать четыре основных блока - процессор, оперативную память, внешнюю память и комплекс устройств ввода-вывода.

**Гарвардская архитектура** — архитектура ЭВМ, отличительным признаком которой является раздельное хранение и обработка команд и данных. Архитектура была разработана Говардом Эйкеном в конце 1930-х годов в Гарвардском университете.

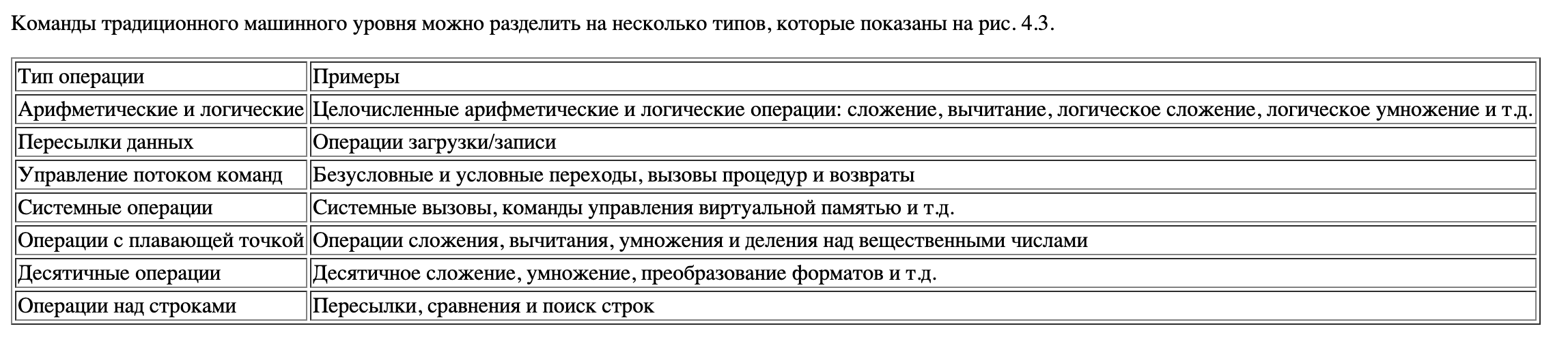
Идея, реализованная Эйкеном, заключалась в физическом разделении линий передачи команд и данных. В первом компьютере Эйкена «Марк I» для хранения инструкций использовалась перфорированная лента, а для работы с данными — электромеханические регистры. Это позволяло одновременно пересылать и обрабатывать команды и данные, благодаря чему значительно повышалось общее быстродействие.

**По признакам и их сочетаниям среди архитектур выделяют:**

1. **По разрядности интерфейсов и машинных слов**: 8-, 16-, 32-, 64-, 128- разрядные (ряд ЭВМ имеет и иные разрядности);

2. **По особенностям набора регистров, формата команд и данных**: CISC, RISC, VLIW;

3. **По количеству центральных процессоров**: однопроцессорные, многопроцессорные, суперскалярные; многопроцессорные по принципу взаимодействия с памятью: симметричные многопроцессорные (SMP), масcивно-параллельные (MPP), распределенные.

**Методы адресации**   
В машинах к регистрам общего назначения метод (или режим) адресации объектов, с которыми манипулирует команда, может задавать константу, регистр или ячейку памяти. Для обращения к ячейке памяти процессор прежде всего должен вычислить действительный или эффективный адрес памяти, который определяется заданным в команде методом адресации. На рисунке представлены все основные методы адресации операндов, которые реализованы в компьютерах. Адресация непосредственных данных и литеральных констант обычно рассматривается как один из методов адресации памяти (хотя значения данных, к которым в этом случае производятся обращения, являются частью самой команды и обрабатываются в общем потоке команд). Адресация регистров, как правило, рассматривается отдельно. В данном разделе методы адресации, связанные со счетчиком команд (адресация относительно счетчика команд) рассматриваются отдельно. Этот вид адресации используется главным образом для определения программных адресов в командах передачи управления. На рисунке на примере команды сложения (Add) приведены наиболее употребительные названия методов адресации, хотя при описании архитектуры в документации разные производители используют разные названия для этих методов. На этом рисунке знак "(" используется для обозначения оператора присваивания, а буква М обозначает память (Memory). Таким образом, M[R1] обозначает содержимое ячейки памяти, адрес которой определяется содержимым регистра R1. Использование сложных методов адресации позволяет существенно сократить количество команд в программе, но при этом значительно увеличивается сложность аппаратуры  
  
  
**Типы команд****

# 13. Виртуальная память и организация защиты памяти.

**Виртуальная память —** метод управления памятью [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), позволяющий выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем (например, [жёстким диском](https://ru.wikipedia.org/wiki/%D0%96%D1%91%D1%81%D1%82%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA)). Для выполняющейся программы данный метод полностью прозрачен и не требует дополнительных усилий со стороны [программиста](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82), однако реализация этого метода требует как аппаратной поддержки, так и поддержки со стороны операционной системы.

В системе с виртуальной памятью используемые программами [адреса](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D1%80%D0%B5%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), называемые [виртуальными адресами](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B4%D1%80%D0%B5%D1%81), транслируются в [физические адреса](https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%B4%D1%80%D0%B5%D1%81&action=edit&redlink=1) в памяти компьютера. Трансляцию виртуальных адресов в физические выполняет аппаратное обеспечение, называемое [блоком управления памятью](https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C%D1%8E). Для программы основная память выглядит как доступное и непрерывное [адресное пространство](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D1%80%D0%B5%D1%81%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) либо как набор непрерывных [сегментов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%B3%D0%BC%D0%B5%D0%BD%D1%82%D0%BD%D0%B0%D1%8F_%D0%B0%D0%B4%D1%80%D0%B5%D1%81%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D0%B8), вне зависимости от наличия у компьютера соответствующего объёма оперативной памяти. Управление виртуальными адресными пространствами, соотнесение физической и виртуальной памяти, а также перемещение фрагментов памяти между основным и вторичным хранилищами выполняет операционная система.

**Применение виртуальной памяти позволяет:**

**·** освободить программиста от необходимости вручную управлять загрузкой частей программы в память и согласовывать использование памяти с другими программами

· предоставлять программам больше памяти, чем физически установлено в системе

· в [многозадачных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C) системах изолировать выполняющиеся программы друг от друга путём назначения им непересекающихся адресных пространств (см. [защита памяти](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%89%D0%B8%D1%82%D0%B0_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D0%B8))

В настоящее время виртуальная память аппаратно поддерживается в большинстве современных процессоров. В то же время в микроконтроллерах и в системах специального назначения, где требуется либо очень быстрая работа, либо есть ограничения на длительность отклика ([системы реального времени](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8)), виртуальная память используется относительно редко. Также в таких системах реже встречается многозадачность и сложные [иерархии памяти](https://ru.wikipedia.org/wiki/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D0%B8).

Страничная организация виртуальной памяти

В большинстве современных операционных систем виртуальная память организуется с помощью страничной адресации. Оперативная память делится на страницы: области памяти фиксированной длины (например, 4096 байт), которые являются минимальной единицей выделяемой памяти (то есть даже запрос на 1 байт от приложения приведёт к выделению ему страницы памяти). Исполняемый процессором пользовательский поток обращается к памяти с помощью адреса виртуальной памяти, который делится на номер страницы и смещение внутри страницы. Процессор преобразует номер виртуальной страницы в адрес соответствующей ей физической страницы при помощи буфера ассоциативной трансляции (TLB). Если ему не удалось это сделать, то требуется дозаполнение буфера путём обращения к таблице страниц, что может сделать либо сам процессор, либо операционная система (в зависимости от архитектуры). Если страница была выгружена из оперативной памяти, то операционная система подкачивает страницу с жёсткого диска в ходе обработки события. При запросе на выделение памяти операционная система может «сбросить» на жёсткий диск страницы, к которым давно не было обращений. Критические данные (например, код запущенных и работающих программ, код и память ядра системы) обычно находятся в оперативной памяти (исключения существуют, однако они не касаются тех частей, которые отвечают за обработку аппаратных прерываний, работу с таблицей страниц и использование файла подкачки).

## Сегментная организация виртуальной памяти

Механизм организации виртуальной памяти, при котором виртуальное пространство делится на части произвольного размера — сегменты. Этот механизм позволяет, к примеру, разбить данные процесса на логические блоки. Для каждого сегмента, как и для страницы, могут быть назначены права доступа к нему пользователя и его процессов. При загрузке процесса часть сегментов помещается в оперативную память (при этом для каждого из этих сегментов операционная система подыскивает подходящий участок свободной памяти), а часть сегментов размещается в дисковой памяти. Сегменты одной программы могут занимать в оперативной памяти несмежные участки. Во время загрузки система создает таблицу сегментов процесса (аналогичную таблице страниц), в которой для каждого сегмента указывается начальный физический адрес сегмента в оперативной памяти, размер сегмента, правила доступа, признак модификации, признак обращения к данному сегменту за последний интервал времени и некоторая другая информация. Если виртуальные адресные пространства нескольких процессов включают один и тот же сегмент, то в таблицах сегментов этих процессов делаются ссылки на один и тот же участок оперативной памяти, в который данный сегмент загружается в единственном экземпляре. Система с сегментной организацией функционирует аналогично системе со страничной организацией: время от времени происходят прерывания, связанные с отсутствием нужных сегментов в памяти, при необходимости освобождения памяти некоторые сегменты выгружаются, при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический. Кроме того, при обращении к памяти проверяется, разрешен ли доступ требуемого типа к данному сегменту.

Виртуальный адрес при сегментной организации памяти может быть представлен парой (g, s), где g — номер сегмента, а s — смещение в сегменте. Физический адрес получается путём сложения начального физического адреса сегмента, найденного в таблице сегментов по номеру g, и смещения s.

Недостатком данного метода распределения памяти является фрагментация на уровне сегментов и более медленное по сравнению со страничной организацией преобразование адреса.

# ***14. Принципы фон-Неймана по построению вычислительных систем.***

1. **Использование двоичной системы счисления в вычислительных машинах**. Преимущество перед десятичной системой счисления заключается в том, что устройства можно делать достаточно простыми, арифметические и логические операции в двоичной системе счисления также выполняются достаточно просто.

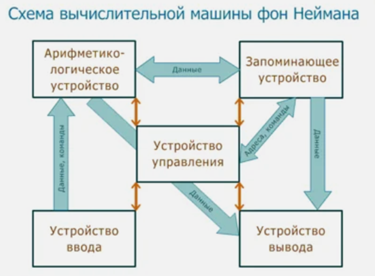
2. **Программное управление ЭВМ**. Работа ЭВМ контролируется программой, состоящей из набора команд. Команды выполняются последовательно друг за другом. Созданием машины с хранимой в памяти программой было положено начало тому, что мы сегодня называем программированием.

3. **Память компьютера используется не только для хранения данных, но и программ**. При этом и команды программы и данные кодируются в двоичной системе счисления, т.е. их способ записи одинаков. Поэтому в определенных ситуациях над командами можно выполнять те же действия, что и над данными.

4. **Ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы**. В любой момент можно обратиться к любой ячейке памяти по ее адресу. Этот принцип открыл возможность использовать переменные в программировании.

5. **Возможность условного перехода в процессе выполнения программы**. Не смотря на то, что команды выполняются последовательно, в программах можно реализовать возможность перехода к любому участку кода.

Самым главным следствием этих принципов можно назвать то, что теперь программа уже не была постоянной частью машины (как например, у калькулятора). Программу стало возможно легко изменить. А вот аппаратура, конечно же, остается неизменной, и очень простой.



***15. Конвейерная организация. Простейшая организация конвейера и оценка его производительности. Примеры.***

***Конвейерный принцип обработки информации*** подразумевает, что в каждый момент времени процессор работает над различными стадиями выполнения нескольких команд, причем на выполнение каждой стадии выделяются отдельные аппаратные ресурсы. По очередному тактовому импульсу каждая команда в конвейере продвигается на следующую стадию обработки, выполненная команда покидает конвейер, а новая поступает в него.

**Происходит повышение производительности:**

· За счет совершенствования элементной базы (переход к новой интегральной технологии, рост плотности размещения компонентов на единице площади кристалла, рост частоты

· За счет рациональной организации процесса выполнения команд в компьютере (параллелизм, архитектурные решения: CISC и RISC)

Конвейер имеет два независимых блока обработки – извлечение и выполнения команды. Пока команда выполняется на втором блоке, первый блок может загружать следующую машинную команду

**Причины снижения производительности конвейера:**

· Время выполнения больше времени извлечения команды

· В командах условного перехода нельзя заранее предсказать адрес следующей выполняемой команды

Конвейеризация эффективна только тогда, когда загрузка конвейера близка к полной, а скорость подачи новых команд и операндов соответствует максимальной производительности конвейера. Если произойдет задержка, то параллельно будет выполняться меньше операций, и суммарная производительность снизится

**Простейшая организация конвейера:**

1. Выборка команды (чтение очередной команды из памяти и занесение ее в регистр команды)

2. Декодирование команды (определение кода операции и способов адресации операндов)

3. Вычисление адресов (вычисление адреса операнда)

4. Выборка операндов (извлечение операндов из памяти. Эта операция не нужна для операндов, находящихся в регистрах)

5. Исполнение команды (непосредственное выполнение команды)

6. Запись результата

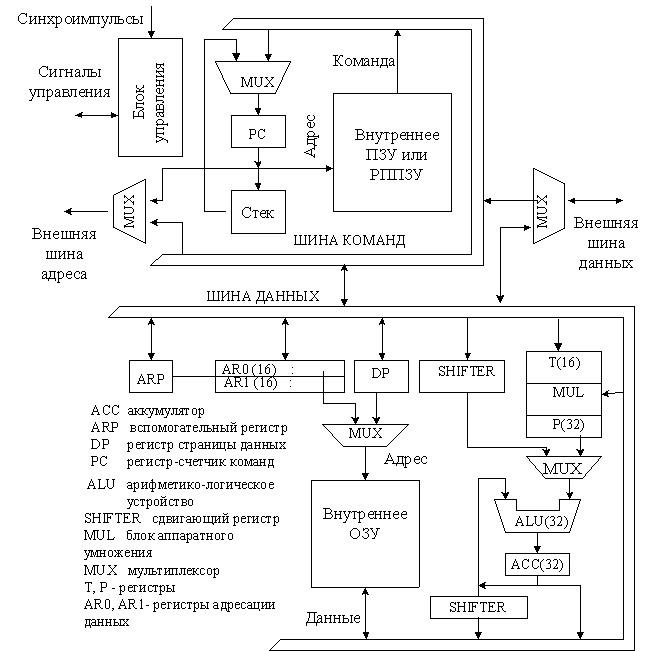
Конвейеризация увеличивает пропускную способность процессора (количество команд, завершающихся в единицу времени), но она не сокращает время выполнения отдельной команды.

**Конвейеризация** – это техника, в результате которой задача или команда разбивается на некоторое число подзадач, которые выполняются последовательно. Каждая подкоманда выполняется на своем логическом устройстве. Все логические устройства (ступени) соединяются последовательно

таким образом, что выход i-ой ступени связан с входом (i+1)-ой ступени, все ступени работают одновременно. Множество ступеней называется конвейером. Выигрыш во времени достигается при выполнении нескольких задач за счет параллельной работы всех ступеней, вовлекая на каждом такте новую задачу или команду.

Функциональная структура конвейерных ВС представляется в виде последовательности связанных элементарных блоков обработки (ЭБО) информации. Все блоки работают параллельно, но каждый из них реализует лишь свою операцию над данными одного и того же потока.

# **16. Общие понятия и определения, структурная схема микропроцессора.**



**Микропроцессор** (МП) - это программно-управляемое устройство, которое предназначено для обработки *цифровой информации* и управления процессом этой обработки и выполнено в виде одной или нескольких больших *интегральных схем* (БИС).

**Составляющие микропроцессора.**

***Основные составляющие микропроцессора***: арифметико-логическое устройство (АЛУ), регистры данных и устройство управления. АЛУ выполняет арифметические и логические операции над данными. Регистры данных, обычно их шесть, являются внутренней памятью, каждый способен хранить одно слово. Для передачи данных между блоками микропроцессора используется внутренняя шина данных.

*1. Арифметико-логическое устройство*

***Арифметико-логическое устройство*** (АЛУ) служит для обработки данных. В зависимости от вида выполняемой операции АЛУ оперирует одним или двумя словами и в своем составе имеет:

- четырехразрядные сумматоры – вычитатели;

- логические элементы для выполнения операций: логического умножения, логического сложения, инверсии, исключающее ИЛИ и т.д.;

- два входных порта, один выходной порт, которые являются буферными регистрами, способным хранить одно слово данных.

Два входных порта позволяют АЛУ принимать данные с внутренней шины данных микропроцессора, или из аккумулятора (специального регистра). Выходной порт служит для пересылки данных в аккумулятор.

Основные операции АЛУ: сложение, вычитание, И, ИЛИ, исключающее ИЛИ, инверсия, сдвиг вправо, сдвиг влево, приращение положительное, приращение отрицательное.

АЛУ производятся в виде микросхем с числом разрядов: 4, 6, 8, 16. Чаще всего используются восьмиразрядные АЛУ, которые имеют 8 входов для первого слова, 8 входов для второго слова и 8 выходов, подключенных к восьми проводной шине.

*2. Аккумулятор*

***Аккумулятор*** служит для хранения слова данных, посланного в него из выходного порта АЛУ или извлеченного из памяти.

**Пример.** Если АЛУ складывает два слова данных, одно находится в аккумуляторе. После выполнения сложения результат посылается в аккумулятор на хранение.

Аккумулятор главный регистр микропроцессора. Большинство арифметических и логических операций выполняется с использованием АЛУ и аккумулятора.

*3.Счетчик команд*

Счетчик команд это один из наиболее важных регистров микропроцессора. Как известно, программа - это последовательность команд, хранимых в памяти и предназначенных для того, чтобы инструктировать машину, как решать поставленную задачу. Для корректного выполнения последней команды должны поступать в строго определенном порядке. На ***счетчике команд*** лежит ответственность следить за тем, какая команда выполняется, а какая подлежит выполнению следующей.

*4.Регистр адреса памяти*

При каждом обращении к памяти микро-ЭВМ ***регистр адреса памяти*** указывает адрес области памяти, которая подлежит использованию микропроцессором. Регистр адреса памяти содержит двоичное число - адрес области памяти. Выход этого регистра называется адресной шиной и используется для выбора области памяти или в некоторых случаях для выбора порта ввода-вывода.

В течение подцикла выборки команды из памяти регистры адреса памяти и счетчика команд имеют одинаковое содержимое, т. е. регистр адреса памяти указы­вает местоположение команды, извлекаемой из памяти. После декодирования команды счетчик команд получает приращение. Что же касается регистра адреса памяти, то он приращения не получает.

*5.Регистр команд*

Регистр команд предназначен исключительно для хранения текущей выполняемой команды, причем эта функция реализуется микропроцессором автоматически с началом цикла выборка-выполнение, называемого также машинным циклом.

Как отмечалось выше, машинный цикл состоит из двух подциклов - выборки и выполнения. За исключением загрузки команды, в период подцикла выборки программист не может по-другому использовать регистр команд. Этот регистр соединен с внутренней шиной данных, однако он только принимает данные, посылать данные на шину он не может.

Хотя функции регистра команд ограниченны, роль его в работе микропроцессора велика, поскольку выход этого регистра является частью дешифратора команд.

*6 Регистр состояния*

Наличием регистра состояния подлинная вычислительная машина отличается от простого калькулятора. Указанный ***регистр предназначен*** для хранения результатов некоторых проверок, осуществляемых в процессе выполнения программы. Разряды регистра состояния принимают то или иное значение при выполнении операций, использующих АЛУ и некоторые регистры.

Запоминание результатов упомянутых проверок позволяет использовать программы, содержащие переходы (нарушения естественной последовательности выполнения команд).

При наличии в программе перехода выполнение команд начинается с некоторой новой области памяти, т. е. счетчик команд загружается новым числом. В случае условного перехода такое действие имеет место, если результаты определенных проверок совпадают с ожидаемыми значениями. Указанные результаты находятся в регистре состояния.

*7.Буферные регистры АЛУ*

На рис. показаны два буферных регистра, каждый из которых предназначен для временного хранения одного слова данных. Один из этих регистров называется буфером аккумулятора АЛУ. Что касается другого буферного регистра, то в него на временное хранение поступают данные с внутренней шины микропроцессора. Необходимость в таком регистре вызвана отсутствием в АЛУ своего запоминающего устройства. В состав АЛУ включены только комбинационные схемы, и поэтому при поступлении исходных данных на входе АЛУ немедленно появляются результирующие данные на его выходе как следствие выполнения операций данной программы.

Буфер аккумулятора позволяет избежать ситуации, при которой вход и выход АЛУ одновременно подсоединены к одной и той же точке схемы.

*8.Регистры общего назначения*

Все микропроцессоры имеют шесть описанных выше основных регистров. В дополнение к ним некоторые микропроцессоры располагают другими регистрами, предоставляемыми в распоряжение пользователей. Эти регистры получили название регистров общего назначения, В некоторых микропроцессорах они служат в качестве запоминающих устройств, в других функциональные возможности этих регистров не уступают возможностям аккумулятора. Последнее достигается в том случае, если АЛУ может помещать в них данные.

*9.Схемы управления*

Роль схем управления в микропроцессоре чрезвычайно важна и заключается в поддержании требуемой последовательности функционирования всех остальных его звеньев. По «распоряжению» схем управления очередная команда извлекается из регистра команд, определяется, что необходимо делать с данными, а затем генерируется последовательность действий по выполнению поставленной задачи.

*10.Внутренняя шина данных микропроцессора*

Структурная схема микропроцессора показывает, что 8-разрядная внутренняя шина данных соединяет между собой АЛУ и регистры, осуществляя передачу данных внутри микропроцессора. Хотя сигналы управления и играют жизненно важную роль в процессе передачи данных по внутренней шине, тракт их передачи не принадлежит шине данных.

Каждый функциональный блок микропроцессора всегда подключен к внутренней шине данных, однако воспользоваться ею может только после получения соответствующего сигнала от схем управления.

Почти все функциональные узлы микропроцессора имеют двустороннюю связь с внутренней шиной данных, т. е. они могут и посылать данные на шину, и принимать с нее данные. ***Внутренняя шина данных*** представляет собой линию двусторонней связи.

***17. Классы конфликтов возникающих в конвейерах и способы их устранения.***

1.Структурные конфликты, которые возникают из-за конфликтов по ресурсам, когда аппаратные средства не могут поддерживать все возможные комбинации команд в режиме одновременного выполнения с совмещением.

2.Конфликты по данным, возникающие в случае, когда выполнение одной команды зависит от результата выполнения предыдущей команды.

3.Конфликты по управлению, которые возникают при конвейеризации команд переходов и других команд, которые изменяют значение счетчика команд.

Структурные конфликты возникают в том случае, когда аппаратные средства процессора не могут поддерживать все возможные комбинации команд в режиме одновременного выполнения с совмещением.

Причины структурных конфликтов.

1. Не полностью конвейерная структура процессора, при которой некоторые ступени отдельных команд выполняются более одного такта.

Решение:

A.увеличение времени такта до такой величины, которая позволила бы все этапы любой команды выполнять за один такт

Б. использование таких аппаратных решений, которые позволили бы значительно снизить затраты времени на выполнение данного этапа (например, использовать матричные схемы умножения)..

2. Недостаточное дублирование некоторых ресурсов.

Решение:

Борьба с конфликтами такого рода проводится путем увеличения количества однотипных функциональных устройств, которые могут одновременно выполнять одни и те же или схожие функции.

---------------------------------------------------------------------------------------------------------------

Конфликты **по управлению** возникают при конвейеризации команд переходов и других команд, изменяющих значение счетчика команд.

Наиболее эффективным методом снижения потерь от конфликтов по управлению служит предсказание переходов. Суть данного метода заключается в том, что при выполнении команды условного перехода специальный блок процессора определяет наиболее вероятное направление перехода, не дожидаясь формирования признаков, на основании анализа которых этот переход реализуется. Процессор начинает выбирать из памяти и выполнять команды по предсказанной ветви программы (так называемое исполнение по предположению, или "спекулятивное" исполнение)

Конфликты **по данным** возникают в случаях, когда выполнение одной команды зависит от результата выполнения предыдущей команды.

1. Конфликты типа RAW (Read After Write): команда j пытается прочитать операнд прежде, чем команда i запишет на это место свой результат. При этом команда j может получить некорректное старое значение операнда.

Уменьшение влияния конфликта типа RAW обеспечивается методом обхода (продвижения) данных. В этом случае результаты, полученные на выходах исполнительных устройств, помимо входов приемника результата передаются также на входы всех исполнительных устройств процессора. Если устройство управления обнаруживает, что данный результат требуется одной из последующих команд в качестве операнда, то он сразу же, параллельно с записью в приемник результата, передается на вход исполнительного устройства для использования следующей командой.

2. Конфликты типа WAR (Write After Read): команда j пытается записать результат в приемник, прежде чем он считается оттуда командой i, При этом команда i может получить некорректное новое значение операнда:

Этот конфликт возникнет в случае, если команда j вследствие неупорядоченного выполнения завершится раньше, чем команда i прочитает старое содержимое регистра R2.

3. Конфликты типа WAW (Write After Write): команда j пытается записать результат в приемник, прежде чем в этот же приемник будет записан результат выполнения команды i, то есть запись заканчивается в неверном порядке, оставляя в приемнике результата значение, записанное командой

i:

Устранение конфликтов по данным типов WAR и WAW достигается путем отказа от неупорядоченного исполнения команд, но чаще всего путем введения буфера восстановления последовательности команд.

***18 Дисковые массивы и уровни RAID.***

**Дисковый массив** — внешнее устройство хранения, состоящее из нескольких жестких дисков.

В отличие от отсека для установки жёсткого диска в корпусе компьютера или специального конструктива для крепления одиночного диска, включающего в себя средства реализации физического и механического (соответствующие разъемы) интерфейса, и, при необходимости, крепежа внутри такого конструктива дискового накопителя иного формата, (например 3½" в конструктиве 5¼") — т. н. «кармана», представляет собой куда более сложную систему, состоящую из следующих компонентов:

* [контроллеры](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%82%D1%8B_%D1%80%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%B8%D0%B9), обладающие способностью [виртуализации](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), и способные создавать [RAID](https://ru.wikipedia.org/wiki/RAID).
* [Кэш-память](https://ru.wikipedia.org/wiki/%D0%9A%D1%8D%D1%88). В зависимости от конструкции может быть на борту контроллера, так и отдельным конструктивом дискового массива.
* Блоки питания. Промышленные дисковые массивы имеют избыточное резервирование блоков питания.
* Отдельное резервное питание для контроллера и кэш-памяти.
* Средства охлаждения дисков и контроллеров, вентиляторы и т. д.
* Контроллеры доступа потребителей к дисковому пространству: FC, SCSI, Ethernet.
* Корзины для дисков. Блоками на несколько дисков или отдельные диски.
* Собственно сами диски.

**Какие бывают уровни RAID**

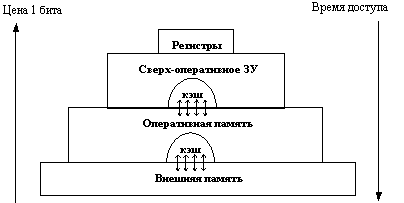
По сути, RAID это связка из нескольких дисковых накопителей, от конфигурации которых зависит скорость работы, безопасность сервера и размещенных на нем данных клиента.

* **RAID 0** (Stripping) используют в случае, когда нужно существенно увеличить производительность работы дисковой подсистемы. Данный вид рейд массива хотя и не обеспечивает надежность хранения информации, но широко используется на компьютерных машинах. Суть работы – информация разбивается на блоки, каждый блок записывается на отдельный диск. Данный массив применяется там, где важна скорость передачи больших объемов информации.
* **RAID 1** (Mirrored disk) – данный рейд с высоким уровнем надежности, при котором данные полностью дублируются. В этом случае два диска вмещают одинаковую информацию, при этом они являются одним логическим диском. Если один из дисков сломается, другой заменяет его, выполняя все функции. Данный рейд повышает скорость уровня чтения информации, потому что функция чтения может происходить одновременно с обеих дисков.
* **RAID 2** – резервирование данных с помощью кода Хэмминга для коррекции ошибок. Данные, которые записываются, формируются на основе слов. Размер слова соответствует количеству дисков для записи информации в рейде. Данный вид массива дает возможность не только исправлять конкретные одиночные ошибки, но и выявлять двойные. Хотя сам по себе рейд является самым избыточным изо всех видов с кодами коррекции. RAID 2 редко используется по сравнению с другими, в связи с плохой обработкой большого количества запросов.
* **RAID 3** – массив отказоустойчивости, который имеет один дополнительный диск, на него записываются данные, к тому же осуществляется параллельный ввод/вывод. Однако, данный рейд менее избыточен по сравнению с предыдущим и имеет высокую производительность. Достоинство RAID 3 – быстрое восстановление информации в случае сбоев на сервере.
* **RAID 4** в некоторых моментах схож с  RAID 3, но данный рейд имеет большой размер блока записываемых данных. Массив повышает производительность передачи файлов малого объема. Недостаток: не обеспечивается высокая скорость передачи больших файлов, сложность восстановления информации при сбои сервера. Данный массив хорошое решение для файловых серверов, так как информация в них считывается и намного реже записывается.
* Наиболее распространенный массив **RAID 5**, для работы которого нужно три и больше дисков. Он использует все диски кроме одного, например, если есть 4 диска, то используются по объему 3. Данный вид массива считается самым экономным. В отличии от RAID 3 имеет большой объем логических блоков для хранения данных, а также позволяет проводить параллельные операции записи. Если из строя выходит один из дисков, данные не теряются, но его следует заменить. Данный рейд используют для многого количества задач, производительность повышают с добавлением дополнительных дисков.
* **RAID 6** — похож на RAID 5, но имеет более высокую степень надёжности — два диска данных и два диска контроля чётности. Основан на кодах Рида — Соломона и обеспечивает работоспособность после одновременного выхода из строя любых двух дисков. Обычно использование RAID-6 вызывает примерно 10-15 % падение производительности дисковой группы, относительно RAID 5, что вызвано бо́льшим объёмом работы для контроллера (более сложный алгоритм расчёта контрольных сумм), а также необходимостью читать и перезаписывать больше дисковых блоков при записи каждого блока
* **RAID 10 (1+0)** – пример комбинации уровней, который соединил в себе достоинства RAID 1 и RAID 0. По сути, это несколько RAID 0, которые дублируются. Данный массив более шустрый и отказоустойчивый, но все равно требует большей избыточности. Весомый минус массива – высокая стоимость.
* **RAID 50** – комбинация нескольких RAID 5, которые чередуются. Довольно экономичное решение, которое подойдет для средних и больших компаний, массив имеет повышенный параметр отказоустойчивости.
* **Hybrid RAID** – сочетание нескольких обычных уровней  RAID, но с использованием дополнительного ПО и SSD дисков, которые используют как кэш для чтения данных. Данная система значительно повышает производительность, благодаря ССД, которые отличаются лучшими характеристиками скорости . Замечательно подходит данный массив для файловых серверов и виртуальных вычислительных машин.

# **19. Иерархия памяти. Организация кэш-памяти. Принципы организации основной памяти в современных компьютерах.**

Классификация ЗУ по функциональному назначению (иерархия запоминающих устройств)

Память ЭВМ почти всегда является "узким местом", ограничивающим производительность компьютера. Поэтому в ее организации используется ряд приемов, улучшающих временные характеристики памяти и, следовательно, повышающих производительность ЭВМ в целом.

Память вычислительной машины представляет собой иерархию запоминающих устройств (внутренние регистры процессора, различные типы сверхоперативной и оперативной памяти, диски, ленты), отличающихся средним временем доступа и стоимостью хранения данных в расчете на один бит. Пользователю хотелось бы иметь и недорогую и быструю память. Кэш-память представляет некоторое компромиссное решение этой проблемы.

Кэш-память - это способ организации совместного функционирования двух типов запоминающих устройств, отличающихся временем доступа и стоимостью хранения данных, который позволяет уменьшить среднее время доступа к данным за счет динамического копирования в "быстрое" ЗУ наиболее часто используемой информации из медленного ЗУ.

Верхнее место в иерархии памяти занимают **регистровые ЗУ**, которые входят в состав процессора и часто рассматриваются не как самостоятельный блок ЗУ, а просто как набор регистров процессора. Такие ЗУ в большинстве случаев реализованы на том же кристалле, что и процессор, и предназначены для хранения небольшого количества информации (до нескольких десятков слов, а в RISC-архитектурах – до сотни), которая обрабатывается в текущий момент времени или часто используется процессором. Это позволяет сократить время выполнения программы за счет использования команд типа регистр-регистр и уменьшить частоту обменов информацией с более медленными ЗУ ЭВМ. Обращение к этим ЗУ производится непосредственно по командам процессора.

Следующую позицию в иерархии занимают **буферные ЗУ (кэш-память)**. Их назначение состоит в сокращении времени передачи информации между процессором и более медленными уровнями памяти компьютера. Буферная память может устанавливаться на различных уровнях, но здесь речь идет именно об указанном ее местоположении.

Еще одним (внутренним) уровнем памяти являются **служебные ЗУ**. Они могут иметь различное назначение. Одним из примеров таких устройств являются ЗУ микропрограмм выполнения команд процессора, а также различных служебных операций (например, хранение таблиц адресов данных в кэше процессора). Специфика назначения предполагает недоступность их командам процессора.

Следующим уровнем иерархии памяти является **оперативная память**. Оперативное ЗУ (ОЗУ) является основным запоминающим устройством ЭВМ, в котором хранятся выполняемые в настоящий момент процессором программы и обрабатываемые данные, резидентные программы, модули операционной системы и т.п. Информация, находящаяся в ОЗУ, непосредственно доступна командам процессора, при условии соблюдения требований защиты.

Еще одним уровнем иерархии ЗУ может являться **дополнительная память**, которую иногда называли расширенной или массовой. Эта ступень использовалась для наращивания емкости оперативной памяти до величины, соответствующей адресному пространству с помощью подключения более дешевого и емкого, чем ОЗУ, но более медленного запоминающего устройства.

В состав памяти ЭВМ входят также ЗУ, **принадлежащие отдельным функциональным блокам** компьютера. Формально эти устройства непосредственно не обслуживают основные потоки данных и команд, проходящие через процессор. Их назначение обычно сводится к буферизации данных, извлекаемых из каких-либо устройств и поступающих в них. Типичные примеры такой памяти – видеопамять графического адаптера и буферная память контроллеров жестких дисков и других внешних запоминающих устройств. Емкости и быстродействие этих видов памяти зависят от конкретного функционального назначения обслуживаемых ими устройств. Для видеопамяти, например, объем может достигать величин, сравнимых с оперативными ЗУ, а быстродействие – даже превосходить быстродействие последних.

Следующей ступенью памяти, являются **жесткие диски**. В этих ЗУ хранится практически вся информация, начиная от операционной системы и основных прикладных программ и кончая редко используемыми пакетами и справочными данными. Эти ЗУ обладают большей емкостью, чем остальные виды памяти и используются для постоянного хранения данных.

Все остальные запоминающие устройства можно объединить с точки зрения функционального назначения в одну общую группу, охарактеризовав ее как группу **внешних ЗУ**. Под словом “внешние” следует подразумевать то, что информация, хранимая в этих ЗУ, в общем случае расположена на носителях не являющихся частью собственно ЭВМ. Это дискеты, флеш-накопители, CD, DVD, BD-диски и др.

# 20. Организация регистров современного процессора.

Основные регистры процессоров

Обычно процессоры разделены на две части: операционное устройство (ОУ) и шинный интерфейс (ШИ). Роль ОУ заключается в выполнении команд, в то время как ШИ подготавливает команды и данные для выполнения. ОУ содержит АЛУ и устройство управления УУ и регистры общего назначения. Эти устройства обеспечивают выполнение команд, арифметические вычисления и логические операции.

Три элемента ШИ - блок управления шиной, очередь команд и сегментные регистры - осуществляют три важные функции:

1. ШИ управляет передачей данных на ОУ, в память и на внешние устройства ввода-вывода.

2. Сегментные регистры управляют адресацией памяти.

3. Выборка команд. Все программные команды находятся в памяти, и ШИ должен иметь доступ к ним для выборки их в очередь команд. ШИ должен "заглядывать вперед" и выбирать команды так, чтобы всегда существовала непустая очередь команд, готовых для выполнения.

ОУ и ШИ работают параллельно, причем ШИ опережает ОУ на один шаг. ОУ сообщает ШИ о необходимости доступа к данным в памяти или на устройство ввода-вывода. Кроме того, ОУ запрашивает машинные команды из очереди команд. Пока ОУ занято, ШИ выбирает следующую команду из памяти. Эта выборка происходит во время выполнения, что повышает скорость обработки.

Сегментные регистры

Сегментом называется область, которая начинается на границе параграфа, т.е. по любому адресу, кратному 16. Хотя сегмент может располагаться в любом месте памяти и иметь размер до 64 Кбайт, он требует столько памяти, сколько необходимо для выполнения программы.

Сегмент кодов (CS) содержит машинные команды, которые будут выполняться. Обычно первая выполняемая команда находится в начале этого сегмента и операционная система передает управление по адресу данного сегмента для выполнения программы. Регистр сегмента кодов (CS) адресует данный сегмент.

Сегмент данных (DS) содержит определенные данные, константы и рабочие области, необходимые программе. Регистр сегмента данных (DS) адресует данный сегмент.

Сегмент стека (SS) содержит адреса возврата как для программы при возврате в операционную систему, так и для вызовов подпрограмм при возврате в главную программу. Регистр сегмента стека (SS) адресует данный сегмент.

Еще один сегментный регистр - дополнительный регистр сегмента (ES) - предназначен для специального использования.

Регистры общего назначения

Регистр - совокупность устройств, используемых для хранения информации, и обеспечения быстрого доступа к ней.

Регистр (AX) является основным сумматором и применяется для всех операций ввода-вывода, некоторых операций над строками и некоторых арифметических операций.

Регистр (BX) является базовым регистром. Это единственный регистр общего назначения, который может использоваться в качестве индекса для расширенной адресации.

Регистр (CX) является счетчиком. Он необходим для управления числом повторений циклов и для операций сдвига или вправо. Регистр (CX) используется также для вычислений.

Регистр (DX) является регистром данных. Он применяется для некоторых операций ввода-вывода и тех операций умножения и деления над большими числами.

Регистровые указатели (SP и BP) обеспечивают системе доступ к данным в сегменте стека. Регистр (SP) обеспечивает использование стека в памяти, позволяет временно хранить адреса и иногда данные. Этот регистр связан с регистром (SS) для адресации стека. Регистр (BP) облегчает доступ к параметрам (данным и адресам, переданным через стек).

Индексные регистры (SI и DI) применяются для расширенной адресации и для использования в операциях сложения и вычитания. Регистр (SI) является индексом источника и применяется для некоторых операций над строками. Регистр (DI) является индексом назначения и применяется также для некоторых операций над строками.

Регистр командного указателя (IP) содержит смещение на команду, которая должна быть выполнена

# 21. Фон-неймановская архитектура.

1. **Использование двоичной системы счисления в вычислительных машинах**. Преимущество перед десятичной системой счисления заключается в том, что устройства можно делать достаточно простыми, арифметические и логические операции в двоичной системе счисления также выполняются достаточно просто.
2. **Программное управление ЭВМ**. Работа ЭВМ контролируется программой, состоящей из набора команд. Команды выполняются последовательно друг за другом. Созданием машины с хранимой в памяти программой было положено начало тому, что мы сегодня называем программированием.
3. **Память компьютера используется не только для хранения данных, но и программ**. При этом и команды программы и данные кодируются в двоичной системе счисления, т.е. их способ записи одинаков. Поэтому в определенных ситуациях над командами можно выполнять те же действия, что и над данными.
4. **Ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы**. В любой момент можно обратиться к любой ячейке памяти по ее адресу. Этот принцип открыл возможность использовать переменные в программировании.
5. **Возможность условного перехода в процессе выполнения программы**. Не смотря на то, что команды выполняются последовательно, в программах можно реализовать возможность перехода к любому участку кода.

Самым главным следствием этих принципов можно назвать то, что теперь программа уже не была постоянной частью машины (как например, у калькулятора). Программу стало возможно легко изменить. А вот аппаратура, конечно же, остается неизменной, и очень простой.

Для сравнения, программа компьютера ENIAC (где не было хранимой в памяти программы) определялась специальными перемычками на панели. Чтобы перепрограммировать машину (установить перемычки по-другому) мог потребоваться далеко не один день. И хотя программы для современных компьютеров могут писаться годы, однако они работают на миллионах компьютеров после несколько минутной установки на жесткий диск.

# **22. BIOS и UEFI. Определение. Состав. Предназначение.**

BIOS, и UEFI – микропрограммы, отвечающие за начальную загрузку компьютера, проверку работоспособностей комплектующих.

Новые компьютеры используют прошивку UEFI вместо традиционного BIOS. Обе эти прошивки являются низкоуровневым программным обеспечением, которое запускается при загрузке ПК перед загрузкой операционной системы, но UEFI — более современное решение, поддерживающее большие жесткие диски, более быстрое время загрузки, больше функций безопасности и удобные графические и мышиные курсоры.

Чтобы не запутать людей, более новые ПК, поставляющиеся с UEFI, по-прежнему называются «BIOS».

UEFI может работать в 32-разрядном или 64-разрядном режиме и имеет большее адресное пространство, чем BIOS, что означает, что процесс загрузки выполняется быстрее. Это также означает, что экраны настройки UEFI могут быть более гладкими, чем экраны настроек BIOS, включая поддержку графики и курсора мыши. Однако это не является обязательным. Многие ПК по-прежнему поставляются с интерфейсными настройками текстового режима UEFI, которые выглядят и работают как старый экран настройки BIOS.

UEFI упакован с другими функциями. Он поддерживает Secure Boot, что означает, что операционная система может быть проверена на достоверность, чтобы гарантировать, что вредоносное ПО не повлияло на процесс загрузки. Он может поддерживать сетевые функции прямо в самой прошивке UEFI, которая может помочь в удаленной диагностике и настройке. В традиционном BIOS вы должны сидеть перед физическим компьютером, чтобы его настроить.

Это не просто замена BIOS. UEFI — это, по сути, крошечная операционная система, которая работает поверх прошивки ПК, и она может сделать намного больше, чем BIOS. Она может храниться во флэш-памяти материнской платы или загружаться с жесткого диска или сетевого ресурса при загрузке.

***23. Поколения вычислительных машин.***

Можно выделить 5 основных поколений ЭВМ. Но деление компьютерной техники на поколения — весьма условное.

I поколение ЭВМ: ЭВМ, сконструированные в 1946-1955 гг.

1. Элементная база: электронно-вакуумные лампы.

2. Соединение элементов: навесной монтаж проводами.

3. Габариты: ЭВМ выполнена в виде громадных шкафов.

Эти компьютеры были огромными, неудобными и слишком дорогими машинами, которые могли приобрести крупные корпорации и правительства.

Лампы потребляли большое количество электроэнергии и выделяли много тепла.

4. Быстродействие: 10−20 тыс. операций в секунду.

5. Эксплуатация: сложная из-за частого выхода из строя электронно-вакуумных ламп.

6. Программирование: машинные коды. При этом надо знать все команды машины, двоичное представление, архитектуру ЭВМ. В основном были заняты математики-программисты. Обслуживание ЭВМ требовало от персонала высокого профессионализма.

7. Оперативная память: до 2 Кбайт.

8. Данные вводились и выводились с помощью перфокарт, перфолент.

II поколение ЭВМ: ЭВМ, сконструированные в 1955-1965 гг.

В 1948 году Джон Бардин, Уильям Шокли, Уолтер Браттейн изобрели транзистор, за изобретение транзистора они получили Нобелевскую премию в 1956 г.

1 транзистор заменял 40 электронных ламп, был намного дешевле и надёжнее.

В 1958 году создана машина М-20, выполнявшая 20 тыс. операций в секунду — самая мощная ЭВМ 50−х годов в Европе.

1. Элементная база: полупроводниковые элементы (транзисторы, диоды).

2. Соединение элементов: печатные платы и навесной монтаж.

3. Габариты: ЭВМ выполнена в виде однотипных стоек, чуть выше человеческого роста, но для размещения требовался специальный машинный зал.

4. Быстродействие: 100−500 тыс. операций в секунду.

5. Эксплуатация: вычислительные центры со специальным штатом обслуживающего персонала, появилась новая специальность — оператор ЭВМ.

6. Программирование: на алгоритмических языках, появление первых операционных систем.

7. Оперативная память: 2−32 Кбайт.

8. Введён принцип разделения времени — совмещение во времени работы разных устройств.

9. Недостаток: несовместимость программного обеспечения.

Уже начиная со второго поколения, машины стали делиться на большие, средние и малые по признакам размеров, стоимости, вычислительных возможностей.

III поколение ЭВМ: ЭВМ, сконструированные в 1965-1975 гг.

В 1958 году Джек Килби и Роберт Нойс, независимо друг от друга, изобретают интегральную схему (ИС).

В 1961 году в продажу поступила первая, выполненная на пластине кремния, интегральная схема.

В 1965 году начат выпуск семейства машин третьего поколения IBM-360 (США). Модели имели единую систему команд и отличались друг от друга объёмом оперативной памяти и производительностью.

В 1968 году сотрудник Стэндфордского исследовательского центра Дуглас Энгельбарт продемонстрировал работу первой мыши.

В 1969 году фирма IBM разделила понятия аппаратных средств (hardware) и программные средства (software). Фирма начала продавать программное обеспечение отдельно от железа, положив начало индустрии программного обеспечения.

В 1971 году создан первый микропроцессор фирмой Intel. На 1 кристалле сформировали 2250 транзисторов.

1. Элементная база: интегральные схемы.

2. Соединение элементов: печатные платы.

3. Габариты: ЭВМ выполнена в виде однотипных стоек.

4. Быстродействие: 1−10 млн. операций в секунду.

5. Эксплуатация: вычислительные центры, дисплейные классы, новая специальность — системный программист.

6. Программирование: алгоритмические языки, операционные системы.

7. Оперативная память: 64 Кбайт.

При продвижении от первого к третьему поколению радикально изменились возможности программирования. Написание программ в машинном коде для машин первого поколения (и чуть более простое на Ассемблере) для большей части машин второго поколения является занятием, с которым подавляющее большинство современных программистов знакомятся при обучении в вузе.

Уже в третьем поколении появились крупные унифицированные серии ЭВМ. Для больших и средних машин в США это прежде всего семейство IBM 360/370. В СССР 70-е и 80-е годы были временем создания унифицированных серии: ЕС (единая система) ЭВМ (крупные и средние машины), СМ (система малых) ЭВМ и «Электроника» (серия микро-ЭВМ).

В их основу были положены американские прототипы фирм IBM и DEC (Digital Equipment Corporation). Были созданы и выпущены десятки моделей ЭВМ, различающиеся назначением и производительностью. Их выпуск был практически прекращен в начале 90-х годов.

IV поколение ЭВМ: ЭВМ, сконструированные начиная с 1975 г. по начало 90-х годов

В 1976 году фирма IBM создает первый струйный принтер.

В 1976 году создана первая ПЭВМ.

Стив Джобс и Стив Возняк организовали предприятие по изготовлению персональных компьютеров «Apple», предназначенных для большого круга непрофессиональных пользователей. Продавался Apple1 по весьма интересной цене — 666,66 доллара. За десять месяцев удалось реализовать около двухсот комплектов.

В 1982 году фирма IBM приступила к выпуску компьютеров IBM РС с процессором Intel 8088, в котором были заложены принципы открытой архитектуры, благодаря которому каждый компьютер может собираться как из кубиков, с учётом имеющихся средств и с возможностью последующих замен блоков и добавления новых.

В 1988 году был создан первый вирус-«червь», поражающий электронную почту.

В 1993 году начался выпуск компьютеров IBM РС с процессором Pentium.

1. Элементная база: большие интегральные схемы (БИС).

2. Соединение элементов: печатные платы.

3. Габариты: компактные ЭВМ, ноутбуки.

4. Быстродействие: 10−100 млн. операций в секунду.

5. Эксплуатация: многопроцессорные и многомашинные комплексы, любые пользователи ЭВМ.

6. Программирование: базы и банки данных.

7. Оперативная память: 2−5 Мбайт.

8. Телекоммуникационная обработка данных, объединение в компьютерные сети.

V поколение ЭВМ: разработки с 90-х годов ХХ века

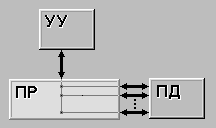
Элементной базой являются сверхбольшие интегральные схемы (СБИС) с использованием оптоэлектронных принципов (лазеры, голография).

***24. Классификация вычислительных систем по Флинну.***

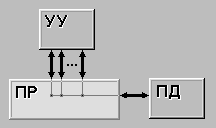
**Классификация Флинна**

По-видимому, самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году М.Флинном [1,2]. 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.

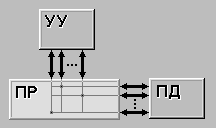
|  |  |
| --- | --- |
|  | SISD (single instruction stream / single data stream) - одиночный поток команд и одиночный поток данных. К этому классу относятся, прежде всего, классические последовательные машины, или иначе, машины фон-неймановского типа, например, PDP-11 или VAX 11/780. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. Не имеет значения тот факт, что для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться конвейерная обработка - как машина CDC 6600 со скалярными функциональными устройствами, так и CDC 7600 с конвейерными попадают в этот класс. |



|  |  |
| --- | --- |
|  | 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. Способ выполнения векторных операций не оговаривается, поэтому обработка элементов вектора может производится либо процессорной матрицей, как в ILLIAC IV, либо с помощью конвейера, как, например, в машине CRAY-1. |



|  |  |
| --- | --- |
|  | 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. Однако ни Флинн, ни другие специалисты в области архитектуры компьютеров до сих пор не смогли представить убедительный пример реально существующей вычислительной системы, построенной на данном принципе. Ряд исследователей [3,4,5] относят конвейерные машины к данному классу, однако это не нашло окончательного признания в научном сообществе. Будем считать, что пока данный класс пуст. |



|  |  |
| --- | --- |
|  | 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. |

# **25. Состав системного блока современной рабочей станции. Единицы измерения рабочих частот процессоров и системных шин. Единицы измерения всех видов памяти.**

Рабочая станция (workstation) — это профессиональный компьютер, предназначенный для научно-инженерных исследований и расчётов, обработки изображений, звука и видео, систем автоматизированного проектирования (САПР) и моделирования, географических информационных систем (ГИС) и других узкоспециализированных задач.

Большинство настольных рабочих станций выполнены в вертикальном форм-факторе с общим обозначением Tower.Подобно настольному ПК, компьютер класса workstation представляет собой системный блок с установленными компонентами. В их числе:

**Материнская плата**, которая служит основой для построения компьютерной системы. От неё зависит количество процессоров и максимальный объём оперативной памяти, которые могут быть установлены в корпус рабочей станции.

**Центральный процессор**, который выполняет команды, арифметические и логические действия, а также все задачи, так или иначе связанные с передачей данных. Подобно серверам, многие компьютеры класса workstation комплектуют одним или двумя процессорами Xeon разных линеек. Также встречаются системы с процессорами для настольных ПК массовой категории — Intel Core i5, i7 или i9.

**Оперативная память**, в которой хранятся машинные коды исполняемых программ и данные, обрабатываемые процессором. Все современные рабочие станции используют ОЗУ стандарта DDR4 либо SODIMM DDR4 (моноблоки и системы форм-фактора Mini PC).

**Корпус**, представляющий собой закрытое шасси для компонентов, в котором они защищены от внешних воздействий и могут работать в оптимальном температурном режиме.

**Блок питания**, который обеспечивает энергоснабжение всех компонентов, входящих в состав рабочей станции: системной платы, процессора, видеокарты, дисковых накопителей и других устройств.

**Накопители**, которые используются для хранения данных и устанавливаются в предназначенные для этого отсеки в корпусе системного блока.

**Видеокарты**, которые служат для вывода изображения на экран и принимают участие в его формировании, снимая таким образом нагрузку с центрального процессора.

Внутренняя тактовая частота — это тактовая частота, на которой функционируют электрические схемы внутри процессора.

Внешняя тактовая частота (частота системной шины) — это тактовая частота, с которой происходит обмен данными между процессором и оперативной памятью компьютера.

Частота измеряется в ГЕРЦАХ (1 ГГц = 1000 МГц = 1000 000 кГц = 1 000 000 000 Гц) (сегодня рабочие частоты некоторых процессоров уже превосходят 3 миллиарда тактов в секунду (3 ГГц))

Память компьютера построена из двоичных запоминающих элементов — битов, объединенных в группы по 8 битов, которые называются байтами. (Единицы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его адресом. Единицы объема памяти: Килобайт, Мегабайт, Гигабайт, а также, в последнее время, Терабайт и Петабайт.

***26. Организация ввода/вывода в вычислительной системе. Системные и локальные шины. Устройства ввода/вывода.***

**Вводом/выводом** называют передачу данных между ядром ЭВМ, включающим в себя процессор и ОП(оперативная память), и периферийными устройствами.

Система ввода-вывода – это единственное средство общения ЭВМ с внешним миром. Ее возможности в серийных ЭВМ представляют собой один из важнейших параметров, определяющих выбор машины для конкретного применения.

Существует три режима ввода-вывода:

· Программный ввод-вывод (нефорсированный).

· ввод-вывод по прерыванию (форсированный).

· Прямой доступ к памяти (ПДП).

Программный ввод-вывод. Инициирование и управление вводом-выводом осуществляет процессор по командам прикладной программы. Периферийные устройства играют пассивную роль и только сигнализируют о своем состоянии, в частности о готовности к операциям ввода-вывода.

Ввод-вывод по прерыванию. Операции ввода-вывода инициирует периферийное устройство, генерируя сигнал запроса прерывания, при этом процессор переключается на подпрограмму обслуживания данного периферийного устройства, вызвавшего прерывание. Непосредственно операциями ввода-вывода управляет процессор.

Прямой доступ к памяти. Процессор в передаче данных не участвует. Он отключается от системной магистрали, а все операции обмена данными идут под управлением специального управляющего устройства – контроллера ПДП. Этот режим используется для быстродействующих периферийных устройств, когда пропускной способности процессора недостаточно.

Передача данных осуществляется двумя способами: а) отдельными битами, и тогда промежуточный интерфейс называется последовательным; б) полными словами (например, целым байтом), и тогда промежуточный интерфейс называется параллельным.

Информация, передаваемая в процессе ввода-вывода, подразделяется: а) на собственно данные; б) управляющие данные.

Управляющие данные от процессора называются также командными словами или приказами. Они инициируют действия, не связанные непосредственно с передачей данных (запуск устройства, запрещение прерываний, установка режимов и т.д.).

Управляющие данные от периферийного устройства называются словами состояния. Они содержат информацию об определенных признаках (о готовности устройства к передаче данных, о наличии ошибок при обмене и т.д.). Состояние обычно представляется в декодированной форме – один бит для каждого признака.

Одним из простейших механизмов, позволяющих организовать взаимодействие различных подсистем, является единственная центральная шина, к которой подсоединяются все подсистемы.

**Системная шина** - шина, предназначенная для передачи данных между периферийными устройствами и центральным процессором, или между периферийными устройствами и оперативной памятью.

**Локальной шиной** называется шина, электрически выходящая непосредственно на контакты микропроцессора. Она обычно объединяет процессор, память, схемы буферизации для системной шины и ее контроллер, а также некоторые вспомогательные схемы. Типичными примерами локальных шин является PCI. Локальная шина предназначена для обеспечения непосредственного доступа процессора к периферийным устройствам (например, графическим или сетевым адаптерам). ((Системная шина ISA позволяет синхронизировать работу процессора и шины с разными тактовыми частотами. Она работает на частоте 8 МГц, что соответствует максимальной скорости передачи 16 Мбайт/с. Системная шина EISA обеспечивает адресное пространство в 4 Гбайт, 32-битовую передачу данных (в том числе и в режиме DMA), улучшенную систему прерываний и арбитраж DMA, автоматическую конфигурацию системы и плат расширения. Устройства шины ISA могут работать на шине EISA. Шина PCI (Peripheral Component Interconnect) поддерживает 32-битовый канал передачи данных между процессором и периферийными устройствами, работает на тактовой частоте 33 МГц и имеет максимальную пропускную способность 120 Мбайт/с.))

**Устройства ввода**. По большей части данная техника является датчиком, которые преобразует неэлектрические величин, в сигналы или же цифровой код. Таким образом человек и прочие аспекты, могут взаимодействовать с компьютером, работать с ним. Говоря более простым языком, устройство ввода – это такая техника, которая является переводчиком между вами и компьютером, помогая вам достичь полного понимания и качественной работы.

Самые популярные представители данного вида: Мышка, преобразует перемещения в сигнал, чтобы компьютер понимал, что вы производите те или иные действия. Клавиатура, посылает сигналы о нажатии клавиш для взаимодействия с ПК.

**Устройства вывода**. Второй тип, главная цель подобной техники – это преобразовать цифровой код, который подается аппаратным обеспечением компьютера, в неэлектрические сигналы или же графическую информацию. С устройствами вывода, человек взаимодействует ежедневно.

В список популярных представителей входят следующие виды техники: Монитор, переводит сигналы в удобную для человеческого восприятия форму, а именно в графическое изображение, таким образом пользователь может работать с ПК. Принтер – это устройство, которое преображает цифровые файлы и отпечатывает их на бумаге. Таким образом человек может преобразовать цифровой файл и физический вид. Акустическая система – переводит сигналы в звуковое сопровождение. В данную категорию входят колонки, наушники и прочие, подобные устройства. Имеется третья категория, которая совмещает две предыдущие, а именно устройства ввода-вывода. Они совмещают в себе качества двух предыдущих групп, таким образом могут, как принимать сигнал, так и раздавать его. В число подобных устройств входят: Интерактивные доски. Дисководы. Роутеры и подобные устройства.

***27. Блоки управления командами. Структура устройства управления. Принципы организации систем прерываний. Процедура обслуживания прерываний***

**Устройство управления** (УУ) вычислительной машины реализует функции управления ходом вычислительного процесса, обеспечивая автоматическое выполнение команд программы. Процесс выполнения программы в ВМ представляет собой последовательность машинных циклов. Для простоты примем, что ЭВМ обеспечивает одноадресную систему команд. При этом, полагается, что до начала выполнения двухоперандной арифметической команды второй операнд уже находится в процессоре. Рассмотрим основные этапы выполнения команды:

- первым этапом в машинном цикле является выборка команды из памяти (ВК);

- за выборкой команды следует этап декодирования ее операционной части (кода операции);

- формирование адреса следующей команды;

- этап формирования исполнительного адреса операнда или адреса перехода. Функция имеет столько модификаций, сколько способов адресации (СА) предусмотрено в системе команд ВМ;

- на четвертом этапе реализуется целевая функция выборки операнда из памяти по исполнительному адресу, сформированному на предыдущем этапе;

- на последнем этапе машинного цикла действия задаются функцией исполнения операции. Очевидно, что количество модификаций функции равно количеству операций, имеющихся в системе команд ВМ.

В свою очередь, УУ, а точнее микропрограммный автомат, формирует следующую выходную информацию:

- внутренние сигналы управления – эти сигналы воздействуют на внутренние схемы центрального процессора и относятся к одному из двух типов: тем, которые вызывают перемещение данных из регистра в регистр, и тем, что инициируют определенные функции операционного устройства ЭВМ;

- сигналы в системную шину – также относятся к одному из двух типов: управляющие сигналы в память и управляющие сигналы в модули ввода/вывода.

**Прерывание** — 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.

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

Основными функциями системы прерывания являются:

1. запоминание состояния прерываемой программы и осуществление перехода к прерывающей программе;

2. восстановление состояния прерванной программы и возврат к ней.

При наличии нескольких источников запросов прерывания должен быть установлен определенный порядок (дисциплина) в обслуживании поступающих запросов. Другими словами, между запросами (и соответствующими прерывающими программами) должны быть установлены приоритетные соотношения, определяющие, какой из нескольких поступивших запросов подлежит обработке в первую очередь, и устанавливающие, имеет право или не имеет данный запрос (прерывающая программа) прерывать ту или иную программу. Приоритетный выбор запроса для исполнения входит в процедуру перехода к прерывающей программе.

**Последовательность обслуживания прерываний.**

При обмене информацией по прерываниям инициатором обмена является внешнее устройство, которое подает специальный сигнал ЗАПРОС ПРЕРЫВАНИЯ на соответствующий вход процессора. По завершении текущей команды процессор прекращает работу по основной программе, вырабатывает сигнал ПОДТВЕРЖДЕНИЕ ПРЕРЫВАНИЯ и переходит к подпрограмме обработки прерывания, передавая ей начальный (стартовый) адрес, называемый *точкой входа в процедуру.* После выполнения подпрограммы происходит возврат к основной программе.

**При обслуживании прерывания можно выделить следующие основные этапы:**

*● прием запроса* на прерывание и его *подтверждение;*

*● сохранение состояния процессора,* а именно, содержимого счетчика команд и основных программно доступных регистров процессора (аккумулятора, регистров общего назначения, статусных и других регистров) на время обработки прерывания с тем, чтобы обеспечить возможность возобновления работы по прерванной программе;

*● идентификация* источника прерывания;

● непосредственное *выполнение программы* обработки прерывания; *восстановление исходного состояния* процессора, или содержимого основных регистров процессора;

*● возврат* к основной программе.

Управление входами и способы приема сигналов запроса на прерывания.

Микропроцессоры могут иметь разное количество входов запроса на прерывания. За каждым входом закреплен собственный стартовый адрес обслуживающей подпрограммы. Увеличение числа входов (линий) запросов позволяет организовать *радиальную* систему прерываний для обслуживания нескольких истопников прерываний. Часть из них может быть отведена для внутренних запросов прерываний (исключений), которые используются при возникновении нештатных ситуаций, например, при понижении напряжения питания.

Для *управления входами* запроса используются команды, позволяющие:

● сформировать маску, биты которой индивидуально разрешают или запрещают (маскирует) запросы прерывания. Маска хранится в специальном регистре. Команды обеспечивают установку нового состояния маски;

● считывать текущее состояние маски в аккумулятор или другой регистр;

● управлять общим флагом разрешения/запрещения прерываний.

# **28. Классификация ЭВМ по областям применения.**

По областям применения компьютеров может быть дана следующая классификация:

• **Персональные компьютеры и рабочие станции** — появились в результате эволюции миникомпьютеров при переходе элементной базы машин с малой и средней степенью интеграции на большие и сверхбольшие интегральные схемы. Миникомпьютеры стали прародителями и другого направления развития современных систем - 32-разрядных машин. Создание RISC-процессоров и микросхем памяти емкостью более 1 Мбит привело к окончательному оформлению настольных систем высокой производительности, которые сегодня известны как рабочие станции.

• **X-терминалы** представляют собой комбинацию бездисковых рабочих станций и стандартных ASCII-терминалов. Бездисковые рабочие станции часто применялись в качестве дорогих дисплеев и в этом случае не полностью использовали локальную вычислительную мощь.

• **Серверы** — прикладные многопользовательские коммерческие и бизнес-системы, включающие системы управления базами данных и обработки транзакций, крупные издательские системы, сетевые приложения и системы обслуживания коммуникаций, разработку программного обеспечения и обработку изображений все более настойчиво требуют перехода к модели вычислений "клиент-сервер" и распределенной обработке. В распределенной (сетевой) модели "клиент-сервер" часть работы выполняет сервер, а часть пользовательский компьютер (в общем случае клиентская и пользовательская части могут работать и на одном компьютере).

• **Мейнфреймы** — это синоним понятия "большая универсальная ЭВМ". Мейнфреймы и до сегодняшнего дня остаются наиболее мощными (не считая суперкомпьютеров) вычислительными системами общего назначения, обеспечивающими непрерывный круглосуточный режим эксплуатации.

• **Кластерные архитектуры** — двумя основными проблемами построения вычисли-тельных систем для критически важных приложений, связанных с обработкой транзакций, управлением базами данных и обслуживанием телекоммуникаций, являются обеспечение высокой производительности и продолжительного функционирования систем. Наиболее эффективный способ достижения заданного уровня производительности — применение параллельных масштабируемых архитектур.

• **Суперкомпьютеры.** В настоящее время к компьютерам предъявляются требования по следующим общим показателям: отношение стоимость/производительность, надежность и отказоустойчивость, масштабируемость (представляет собой возможность наращивания числа и мощности процессоров, объемов оперативной и внешней памяти и других ресурсов вычислительной системы), совместимость и мобильность программного обеспечения.

# **29. Цикл обработки команды современного процессора.**

Программа в ЭВМ реализуется центральным процессором (ЦП) посредством последовательного исполнения образующих эту программу команд. Действия, требуемые для выборки (извлечения из основной памяти) и выполнения команды, называют циклом команды. В общем случае цикл команды включает в себя несколько составляющих (этапов):

· выборка команды;

· формирование адреса следующей команды;

· декодирование команды;

· вычисление адресов операндов;

· выборка операндов;

· исполнение операции;

· формирование признака результата;

· запись результата.

Перечисленные этапы выполнения команды в дальнейшем будем называть стандартным циклом команды. Отметим, что не все из этапов присутствуют при выполнении любой команды (зависит от типа команды), однако этапы выборки, декодирования, формирования адреса следующей команды и исполнения операции имеют место всегда.

**Этап выборки команды.** Цикл любой команды начинается с того, что центральный процессор извлекает команду из памяти, используя адрес, хранящийся в счетчике команд (СК). Двоичный код команды помещается в регистр команды (РК) и с этого момента становится «видимым» для процессора. Если длина команды совпадает с разрядностью ячейки памяти, то все понятно. Однако, система команд многих ЭВМ предполагает несколько форматов команд, причем в разных форматах команда может занимать 1, 2 или более ячеек, а этап выборки команды можно считать завершенным лишь после того, как в регистр команды будет помещен полный код команды. Информация о фактической длине команды содержится в полях кода операции и способа адресации.

**Этап формирования адреса следующей команды.** Для большинства ЭВМ характерно размещение соседних команд программы в смежных ячейках памяти. Если извлеченная команда не нарушает естественного порядка выполнения программы, то для вычисления адреса следующей выполняемой команды достаточно увеличить содержимое счетчика команд на длину текущей команды, представленную количеством занимаемых кодом команды ячеек памяти. Длина команды, а также то, способна ли она изменить естественный порядок выполнения команд программы, выясняются в ходе ранее упоминавшегося предварительного декодирования. Если извлеченная команда способна изменить последовательность выполнения программы (команда условного или безусловного перехода, вызова процедуры и т.п), процесс формирования адреса следующей команды переносится на этап исполнения операции. В силу сказанного, в ряде ЭВМ рассматриваемый этап цикла команды следует не за выборкой команды, а находится в конце цикла.

**Этап декодирования команды.** После выборки команды она должна быть декодирована, для чего ЦП расшифровывает находящийся в РК код команды. В результате декодирования выясняются следующие вопросы: находится ли в РК полный код команды или требуется дозагрузка остальных слов команды; какие последующие действия нужны для выполнения данной команды; если команда использует операнды, то откуда они должны быть взяты (номер регистра или адрес ячейки основной памяти); если команда формирует результат, то куда этот результат должен быть направлен.

По результатам декодирования производится подготовка электронных схем ЭВМ к выполнению предписанных командой действий.

**Этап вычисления адресов операндов.** Этап имеет место, если в процессе декодирования команды выясняется, что команда использует операнды. Если операнды размещаются в основной памяти, осуществляется вычисление их исполнительных адресов, с учетом указанного в команде способа адресации. Так, в случае индексной адресации для получения исполнительного адреса производится суммирование содержимого адресной части команды и содержимого индексного регистра.

**Этап выборки операндов.** Вычисленные на предыдущем этапе исполнительные адреса используются для считывания операндов из памяти и занесения в определенные регистры процессора. Например, в случае арифметической команды операнд после извлечения из памяти может быть загружен во входной регистр АЛУ. Однако чаще операнды предварительно заносятся в специальные вспомогательные регистры процессора, а их пересылка на вход АЛУ происходит на этапе исполнения операции.

**Этап исполнения операции.** На этом этапе реализуется указанная в команде операция. В силу различия сущности каждой из команд ЭВМ, содержание этого этапа сугубо индивидуально.

**Этап формирования признака результата.** На этом этапе определяется, каким получился результат операции. Результат может быть положительным, отрицательным, равным нулю и т.п. Сформированный признак заносится в регистр признака результата (РПР) для дальнейшего использования устройством управления.

**Этап записи результата.** Этап записи результата присутствует в цикле тех команд, которые предполагают занесение результата в регистр или ячейку основной памяти. Фактически его можно считать частью этапа исполнения, особенно для тех команд, которые помещают результат сразу в несколько мест.

**((Машинный цикл с косвенной адресацией**. Многие команды предполагают чтение операндов из памяти или запись в память. В простейшем случае в адресном поле таких команд явно указывается исполнительный адрес соответствующей ячейки ОП. Однако часто используется и другой способ указания адреса, когда адрес операнда хранится в какой-то ячейке памяти, а в команде указывается адрес ячейки, содержащей адрес операнда. Как уже отмечалось ранее, подобный прием называется косвенной адресацией. Чтобы прочитать или записать операнд, сначала нужно извлечь из памяти его адрес и только после этого произвести нужное действие (чтение или запись операнда), иными словами, требуется выполнить два обращения к памяти. Это, естественно, отражается и на цикле команды, в котором появляется косвенная адресация. Этап косвенной адресации можно отнести к этапу вычисления адресов операндов, поскольку его сущность сводится к определению исполнительного адреса операнда. Иными словами, содержимое адресного поля команды в регистре команд используется для обращения к ячейке ОП, в которой хранится адрес операнда, после чего извлеченный из памяти исполнительный адрес операнда помещается в адресное поле регистра команды на место косвенного адреса. Дальнейшее выполнение команды протекает стандартным образом.

**Машинный цикл с прерыванием.** Практически во всех ЭВМ предусмотрены средства, благодаря которым модули ввода/вывода (и не только они) могут прервать выполнение текущей программы для внеочередного выполнения другой программы, с последующим возвратом к прерванной. Первоначально прерывания были введены для повышения эффективности вычислений при работе с медленными периферийными устройствами. Положим, что процессор пересылает данные на принтер, используя стандартный цикл команды. После каждой операции записи ЦП будет вынужден делать паузу в ожидании подтверждения от принтера об обработке символа. Длительность этой паузы может составлять сотни и тысячи циклов команды. Ясно, что такое использование ЦП очень неэффективно. В случае прерываний, пока протекает операция ввода/вывода, ЦП способен выполнять другие команды.

В упрощенном виде процедуру прерывания можно описать следующим образом. Объект, требующий внеочередного обслуживания, выставляет на соответствующем входе ЦП сигнал запроса прерывания (ЗП). ЗП могут возникать, как в самой ЭВМ, так и в её внешней среде.

К первым относятся:

· ошибки в работе аппаратуры;

· переполнение разрядной сетки;

· попытка деления на «0»;

· выход из установленной для данной программы области памяти;

· затребование периферийным устройствам операции ввода/ вывода.

К внешним запросам относятся:

· запрос прерывания от другой ЭВМ;

· запрос от различного рода датчиков.

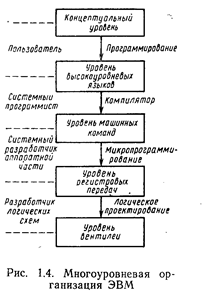
Перед переходом к очередному циклу команды процессор проверяет этот вход на наличие запроса. Обнаружив запрос, ЦП запоминает информацию, необходимую для продолжения нормальной работы после возврата из прерывания, и переходит к выполнению прерывающей программы. По завершении обработки прерывания ЦП восстанавливает состояние прерванного процесса, используя запомненную информацию, и продолжает выполнение прерванной программы.

Для учета прерываний к циклу команды добавляется этап прерывания, в ходе которого процессор проверяет, не поступил ли запрос прерывания. Если запроса нет, ЦП переходит к этапу выборки следующей команды программы. При наличии запроса процессор:

1. Приостанавливает выполнение текущей программы и запоминает содержимое всех регистров, которые будут использоваться программой обработки прерывания. Это называется сохранением слова состояния программы (ССП). В первую очередь необходимо сохранить содержимое счетчика команд, аккумулятора и регистра признаков. ССП обычно сохраняется в стеке.

2. Заносит в счетчик команд начальный адрес программы обработки прерывания. Теперь процессор продолжает с этапа выборки первой команды обработчика прерывания. Обработчик (обычно он входит в состав операционной системы) определяет природу прерывания и выполняет необходимые действия. Когда программа обработки прерывания завершается, процессор может возобновить выполнение прерванной программы с точки, где она была прервана. Для этого он восстанавливает ССП (содержимое СК и других регистров) и начинает с цикла выборки очередной команды прерванной программы.

# 30. Многоуровневая организация ЭВМ. Многоступенчатая обработка.

******

На концептуальном уровне пользователь ЭВМ анализирует задачу, разрабатывает алгоритм ее решения, определяет содержимое обработки информации.

На уровне языков программирования высокого уровня изучается алгоритм решения задачи, составляется детальный проект решения (определяются структуры данных, содержание отдельных программных модулей, связи между ними), пишется программа на одном из языков высокого уровня.

На уровне машинных команд обеспечивается связь программных и аппаратных средств. На этом уровне составляется список команд, определяются способы кодирования кодов операций и адресов, число адресных полей и другие параметры, заложенные в структуру ЭВМ.

На уровне регистровых передач осуществляются элементарные операции, выполняемые аппаратурой ЭВМ. Это операции преобразования информации, операции чтения и записи в запоминающее устройство и регистры, операции коммутации, обеспечиваю­щие передачу слов между отдельными модулями и блоками.

На уровне вентилей разрабатываются логические схемы при логическом проектировании аппаратуры ЭВМ.

# 31. Оценка производительности вычислительных систем.

Основу для сравнения различных типов компьютеров между собой дают стандартные методики измерения производительности. Методики позволяют разработчикам и пользователям осуществлять выбор на основе количественных показателей, что дает возможность постоянного прогресса в данной области.

Единицей измерения производительности компьютера является время: компьютер, выполняющий тот же объем работы за меньшее время является более быстрым. Время выполнения любой программы измеряется в секундах. Часто производительность измеряется как скорость появления некоторого числа событий в секунду, так что меньшее время подразумевает большую производительность.

MIPS. Одной из альтернативных единиц измерения производительности процессора (по отношению к времени выполнения) является MIPS – (миллион команд в секунду).

В общем случае MIPS есть скорость операций в единицу времени, т.е. для любой данной программы MIPS есть просто отношение количества команд в программе к времени ее выполнения. Таким образом, производительность может быть определена как обратная к времени выполнения величина, причем более быстрые машины при этом будут иметь более высокий рейтинг MIPS.

Положительными сторонами MIPS является то, что эту характеристику легко понять, особенно покупателю, и что более быстрая машина характеризуется большим числом MIPS, что соответствует нашим интуитивным представлениям. Однако использование MIPS в качестве метрики для сравнения наталкивается на две проблемы. Во-первых, MIPS зависит от набора команд процессора, что затрудняет сравнение по MIPS компьютеров, имеющих разные системы команд. Во-вторых, MIPS даже на одном и том же компьютере меняется от программы к программе.

MFLOPS. Измерение производительности компьютеров при решении научно-технических задач, в которых существенно используется арифметика с плавающей точкой. Именно для таких вычислений впервые встал вопрос об измерении производительности, а по достигнутым показателям часто делались выводы об общем уровне разработок компьютеров. Обычно для научно-технических задач производительность процессора оценивается в MFLOPS (миллионах чисел-результатов вычислений с плавающей точкой в секунду, или миллионах элементарных арифметических операций над числами с плавающей точкой, выполненных в секунду).

Ясно, что рейтинг MFLOPS зависит от машины и от программы. Этот термин менее безобидный, чем MIPS. Он базируется на количестве выполняемых операций, а не на количестве выполняемых команд. По мнению многих программистов, одна и та же программа, работающая на различных компьютерах, будет выполнять различное количество команд, но одно и то же количество операций с плавающей точкой. Именно поэтому рейтинг MFLOPS предназначался для справедливого сравнения различных машин между собой.

Тесты SPEC. В настоящее время имеется два базовых набора тестов SPEC.

Набор тестов CINT95, измеряющий производительность процессора при обработке целых чисел, состоит из шести программ, написанных на языке Си и выбранных из различных прикладных областей: теории цепей, интерпретатора языка Лисп, разработка логических схем, упаковка тестовых файлов, электронные таблицы и компиляция программ.

Набор тестов CFP95, измеряющий производительность процессора при обработке чисел с плавающей точкой, состоит из 14 программ, также выбранных из различных прикладных областей: разработка аналоговых схем, моделирование методом Монте-Карло, квантовая химия, оптика, робототехника, квантовая физика, астрофизика, прогноз погоды и другие научные и инженерный задачи. Две программы из этого набора написаны на языке Си, а остальные 12 – на Фортране. В пяти программах используется одинарная, а в остальных – двойная точность.

Результаты прогона каждого индивидуального теста их этих двух наборов выражается отношением времени выполнения одной копии теста на тестируемой машине к времени ее выполнения на эталонной машине. В качестве эталонной машины используется Sun SPARCstation 10/40 с 128 MB. SPEC публикует результаты прогона каждого отдельного теста, а также две составные оценки: SPECint95 – среднее геометрическое 6 результатов индивидуальных тестов из набора CINT95 и SPECfp95 – среднее геометрическое 14 результатов индивидуальных тестов из набора CFP95. Составные оценки SPECint95 и SPECfp95 достаточно хорошо характеризуют производительность процессора и системы памяти при работе в однозадачном режиме, но они совершенно не подходят для оценки производительности многопроцессорных т однопроцессорных систем, работающих в многозадачном режиме.

SPEC предложила в качестве единицы измерения производительности многопроцессорных систем Тесты TPC. По мере расширения использования компьютеров при обработке транзакций в сфере бизнеса все более важной становится возможность справедливого сравнения именно систем между собой, а не процессоров. С этой целью в 1988 году был создан Совет по оценке производительности обработки транзакций (TPC – Transaction Processing Performance Council). Следует отметить, что, как и любой другой тест, ни один тест TPC не может измерить производительность системы, которая применима для всех возможных сред обработки транзакций, но эти тесты действительно могут помочь пользователю справедливо сравнивать похожие системы. Однако, когда пользователь делает покупку или планирует решение о покупке, он должен понимать, что никакой тест не может заменить его конкретную прикладную задачу.

Тест TPC-C. Тестовый пакет TPC-C моделирует прикладную задачу обработки заказов. Он моделирует достаточно сложную систему, которая должна управлять приемом заказов, учетом товаров и распространением товаров и услуг. Тест TPC-C осуществляет тестирование всех основных компонентов системы: терминалов, линий связи, ЦП, систем расчета и баз данных.

TPC-C требует, чтобы выполнялись пять типов транзакций:

- новый заказ, вводимый с помощью сложной экранной формы;

- простое обновление базы данных, связанное с платежом;

- простое обновление баз данных, связанное с поставкой;

- справка о состоянии заказов;

- справка по учету товаров.

База данных TPC-C основана на модели оптового поставщика с удаленными районами и товарными складами. База данных содержит девять таблиц: товарные склады, район, покупатель, заказ, порядок заказов, новый заказ, статья счета, складские запасы и история. Обычно публикуются два результата. Один из них, tpm-C, представляет пиковую скорость выполнения транзакций (выражается в количестве транзакций в минуту). Второй результат, $/tpm-C, представляет собой нормализованную стоимость системы. Стоимость системы включает все аппаратные средства и программное обеспечение, используемые в системе, плюс стоимость обслуживания в течение пяти лет.

Тест TPC-D предназначен для оценки производительности систем принятия решений, для оценки систем масштаба предприятия.

# 32. Понятие архитектуры вычислительной системы.

Архитектура ВС — совокупность характеристик и параметров, определяющих функционально-логическую и структурную организацию системы. Понятие архитектуры охватывает общие принципы построения и функционирования, наиболее существенные для пользователей, которых больше интересуют возможности систем, а не детали их технического исполнения. Поскольку ВС появились как параллельные системы, то и рассмотрим классификацию архитектур с этой точки зрения.

Эта классификация архитектур была предложена М. Флинном в начале 60-х гг. В ее основу заложено два возможных вида параллелизма: независимость потоков заданий (команд), существующих в системе, и независимость (не связанность) данных, обрабатываемых в каждом потоке. Классификация до настоящего времени еще не потеряла своего значения. Как и любая классификация, она носит временный и условный характер. Своим долголетием она обязана тому, что оказалась справедливой для ВС, в которых ЭВМ и процессоры реализуют программные последовательные методы вычислений. С появлением систем, ориентированных на потоки данных и использование ассоциативной обработки, данная классификация может быть некорректной.

Сайт: <https://studopedia.ru/8_46680_arhitektura-vichislitelnih-sistem.html>

# 33. Организация автоматической работы ЭВМ. Управляющие функции процессора. Общая организация выполнения программы на ЭВМ.

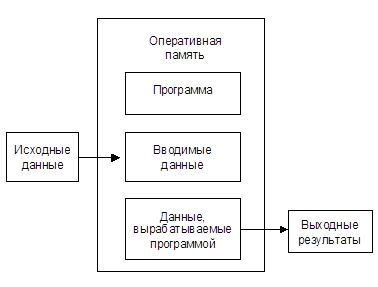
Каждая ЭВМ является автоматом, состоящим из памяти, образуемой внешним и оперативным запоминающими устройствами, устройства управления (УУ) и арифметические устройства (АУ), в котором могут выполняться некоторые действия или операции. Память имеет вид занумерованной последовательности ячеек, в каждой из которых хранится порция двоичной информации в виде серии нулей и единиц. Автоматическая работа ЭВМ, управляемая программой, состоит из последовательности тактов. На каждом такте УУ выбирает из предписанной ему ячейки памяти порцию информации. Эта порция трактуется как команда, т. е. предписание АУ выполнить некоторую операцию. Обычно в ЭВМ выполнение операции состоит в том, чтобы из определённых ячеек памяти взять хранящуюся там информацию, передать её в АУ для выполнения над ней нужного действия, результат которого отправить в указанную ячейку памяти, и сообщить УУ номер ячейки следующей команды. Отдельные действия, совершаемые ЭВМ, весьма просты — это арифметические и логические операции, операции сравнения, переписывания порции информации и т.п.

В состав процессора входят следующие устройства: **устройство управления (УУ)**, арифметико-логическое устройство (АЛУ), регистры процессорной памяти. УУ управляет работой всех устройств компьютера по заданной программе. УУ извлекает очередную команду из регистра команд, определяет, что надо делать с данными, а затем задает последовательность действий выполнения поставленной задачи.

**Функции процессора** ограничиваются реализацией следующих процедур: ● Выборка команды из ОП и операндов, указанных в адресной части команды.

● Выполнение операции, заданной кодом операции, что сводится к выполнению: арифметических, логических операций, передачи команд на выполнение средствами ввода/вывода и формировании адреса следующей команды.

Для того чтобы программа могла быть выполненной, она должна быть помещена в оперативную память компьютера. Туда же должны быть помещены и исходные данные. Как правило, программа вводится в оперативную память с жесткого диска. Исходные данные вводятся с клавиатуры либо также с жесткого диска, куда они должны быть заранее помещены с помощью другой программы. Результаты своей работы программа помещает в определенную область оперативной памяти, куда они могут быть выведены на какое-либо внешнее устройство, которым может быть жесткий диск, экран дисплея, печатающее устройство.



# 34. Память и запоминающие устройства. Виды и характеристики ЗУ: адресная, стековая и ассоциативная организация памяти.

Памятью ЭВМ называется совокупность устройств, служащих для запоминания, хранения и выдачи информации. Комплекс технических средств, реализующих функцию памяти, называется запоминающим устройством (ЗУ). Они обеспечивают центральному процессору доступ к программам и информации.

Классификация запоминающих устройств:

1) основная (оперативная) память,

2) сверхоперативная память (СОЗУ),

3) внешние запоминающие устройства.

*Адресная память*

При адресной организации памяти размещение и поиск информации в запоминающем массиве основаны на использовании адреса хранения слова. Другими словами **адресный поиск** предполагает, что искомый операнд извлекается из ячейки памяти, номер которой формируется на основе информации в адресном поле команды. По коду адреса в регистре адреса блок адресной выборки формирует в соответствующей ячейке памяти сигналы, позволяющие произвести считывание или запись слова в ячейку. Проиллюстрируем адресную структуру памяти на примере ЭВМ с 16-разрядной шиной данных и 16-разрядными словами.

*Ассоциативная память*

В памяти этого типа поиск нужной информации производится не по адресу, а по ее содержанию (по ассоциативному признаку). При этом поиск по ассоциативному признаку (или последовательно по отдельным его разрядам) происходит параллельно во времени для всех ячеек запоминающего массива.

Память этого типа применяется в специализированных вычислительных машинах, машинах баз данных и при организации работы кэш-памяти. Если использовать адресную память или стек, то процедура поиска и вывода нужной информации сводится к последовательному считыванию содержимого отдельных ячеек памяти, выделению с помощью маски десяти его первых битов, сравнению выделенных битов с искомым кодом и выводом результата. Это очень нерационально.

Стековая память

Стековая память состоит из ячеек, связанных друг с другом разрядными цепями передачи слов. Обмен информацией всегда выполняется только через верхнюю ячейку – *вершину стека.* При записи нового слова (команды, числа, символа) все ранее записанные слова сдвигаются на одну ячейку вниз, а новое слово помещается на вершину стека. Считывание возможно только с вершины стека и производится с удалением или без удаления считываемого слова. Такую память часто называют памятью типаLIFO(Last–InFirst–Outпоследним вошел, первым вышел). Аппаратная реализация стека сложна и обычно стек моделируют программно. При этом в качестве стека обычно используется часть адресной памяти.

# 35. Тестирование вычислительных систем.

**Тестирование вычислительных систем**

Результаты тестирования позволяют:

- на стадии проектирования принимать решения о примерном составе будущей вычислительной системы;

- на стадии отладки и ввода в эксплуатацию путем сравнения с аналогами принимать решение о достижении необходимых характеристик;

- на стадии эксплуатации/модификации выявлять узкие места системы.

**Требования к тестам:**

*Полнота.* Тест должен оценивать только те параметры, для оценки которых создавался. Выдаваемые результаты должны быть непротиворечивыми, лаконичными и легкими для понимания.

*Легкость в использовании.*

*Масштабируемость*. Тест должен быть доступен для большого числа разного по вычислительной мощности аппаратного обеспечения.

*Переносимость.* Тест должен быть доступен для большого числа разного по архитектуре аппаратного обеспечения. Основной чертой переносимости является язык программирования, и, соответственно, наличие компилятора под данную платформу.

*Репрезентативность.* Вне зависимости от платформы тест должен оценивать наиболее важные для большинства приложений пользователей характеристики (выборка характеристик должна быть показательной).

*Доступность.* Тест должен быть доступен, в том числе и его исходный код. При представлении результатов должна быть указанна версия и все внесенные изменения.

*Воспроизводимость.* При необходимости должна быть возможность повторить тест с получением аналогичных результатов. При публикации результатов необходимо предоставлять исчерпывающую информацию о программном и аппаратном обеспечении.

**Классификация тестов.**

1. «Игрушечные» тесты – маленькие, длиной в несколько сот строк исходного кода. Как правило, такие тесты представляют собой решение какой-либо широко известной математической задачи.

2. Микротесты – специализированные, ориентированные на определение какой-то одной из основных количественных характеристик аппаратного обеспечения:(производительность CPU; пропускная способность RAM;пропускная способность коммуникационной среды; итд)

3. Ядра – это фрагменты кода, взятые из реальных приложений.

4. Синтетические тесты - искусственно сгенерированные тесты для оценки большого количества показателей без привязки к какому-либо приложению.

5. Приложения – программы наиболее часто применяемые для решения тех или иных реальных задач.

6. Псевдоприложения – программы, созданные на основе реальных приложений, но специально адаптированные для задач тестирования.

7. Пакеты тестов – коллекции различных типов тестов с преобладанием приложений.

На основе Linpack-тестирования дважды в год составляются мировой список наиболее быстродействующих вычислительных систем TOP-500 в мире и российский список TOP-50.

Производительность в тесте Linpack Benchmark измеряется в количестве выполненных операций с плавающей точкой в секунду (Flops).

Для тестирования кластерных систем используется версия теста, называемая HPL (High-Performance Linpack Benchmark). В этой версии пользователь имеет возможность задать все значимые параметры алгоритма, подбирая их для достижения наилучшей производительности.

Решаемая задача

Тест состоит в решении системы линейных алгебраических уравнений методом LU-факторизации c выбором ведущего элемента столбца.

При параллельном процессе на вычислительном кластере исходная матрица разделяется на логические блоки с определенной размерностью. Эти блоки в свою очередь разбиваются сеткой P´Q на более мелкие. Каждый из таких блоков «достанется» отдельному процессору системы.

Для того чтобы воспользоваться тестом необходимо загрузить его дистрибутив. Для компиляции также потребуется наличие какой-либо реализации MPI.

После сборки, получившийся исполняемый модуль использует файл HPL.dat, в котором могут быть указаны существенные параметры алгоритма. Для облегчения проведения тестирования, в файле HPL.dat может быть задана последовательность параметров, при этом будет выполнена серия тестов со всеми перечисленными значениями параметров.

В файле HPL.dat содержатся параметры конфигурации:

Каким образом будет осуществляться вывод результатов теста, перечисление размерностей задач, которые будут решаться в ходе теста, различные варианты параметра размерности NB. (Следует отметить, что тест устроен таким образом, что перебирает всевозможные варианты заданных параметров, таким образом для трех различных размерностей и двух вариантов NB тест будет выполнен шесть раз. Поскольку число различных параметров в конфигурационном файле велико, следует быть осторожным и не забывать об этой особенности.)

Результатом работы теста является достаточно объемный файл, в котором для каждого набора параметров, определенном в конфигурационном файле указана достигнутая производительность на тесте, а также имеющаяся погрешность решения .

# 36. Общие требования, предъявляемые к современным ЭВМ.

1.Отношение стоимость-производительность.

Появление нового направления в вычислительной технике определяется требованием компьютерного рынка. Поэтому у разработчиков ЭВМ нет одной единой цели; при проектировании высокопроизводительной конструкции приходится игнорировать стоимостные характеристики. Другим примером служит низкостоимостная конструкция, где производительность принесена в жертву стоимости.

Между двумя этими направлениями находятся конструкции в которых разработчики находят баланс между стоимостными параметрами и производительностью.

Для сравнения различных ЭВМ и Вычислительных систем между собой обычно используются стандартные методики измерения производительности. Эти методики позволяют разработчикам использовать полученные в результате испытаний количественные показатели для оценки тех или иных технических решений. И в конце концов именно производительность и стоимость дают пользователю рациональную основу для решения вопроса, какую ЭВМ или вычислительную систему выбрать.

2.Надежность и отказоустойчивость.

Важнейшей характеристикой вычислительных систем является надежность. Повышение надежности основано на принципе предотвращения неисправности путем снижения интенсивности отказов и сбоев за счет

- применения электронных схем и компонентов со сверхвысокой степенью интеграции;

- облегчения режима работы схем;

- обеспечение тепловых режимов;

- совершенствование методов сборки аппаратуры.

Отказоустойчивость – это такое свойство вычислительной системы, которое обеспечивает ей, как логической машине, возможность продолжения действий заданной программы, после возникновения неисправности.

Введение отказоустойчивости требует дополнительного аппаратного и программного обеспечения. Направления, связанные с предотвращением неисправности, и с отказоустойчивостью – основные в проблеме надежности вычислительной системы.

Понятие отказоустойчивости (надежности) включает не только аппаратные средства, но и программное обеспечение.

3.Масштабируемость – возможность наращивания числа и мощности процессоров, объема памяти и других ресурсов вычислительных систем. Масштабируемость должна обеспечиваться архитектурой и конструкцией ЭВМ, соответствующими средствами программного обеспечения. Добавление каждого нового процессора действительно масштабируемой системы должно давать прогнозируемое увеличение производительности и пропускной способности при приемлемых затратах.

В идеале добавление процессора к системе должно приводить к линейному росту ее производительности. Однако это не всегда так. Потери производительности могут возникать при недостаточной пропускной способности или из-за возрастания трафика между процессорами и основной памятью, а также между памятью и устройствами ввода-вывода. Возможность масштабирования системы определяется не только архитектурой аппаратных средств, но и зависит от заложенных свойств программного обеспечения.

В частности программное обеспечение должно минимизировать график межпроцессорного обмена, который может препятствовать линейному росту производительности системы. Аппаратные средства являются только частью масштабируемой архитектуры, для которой ПО может обеспечивать предсказуемый рост производства.

4. Совместимость и мобильность ПО.

Концепция программной совместимости впервые в широких масштабах была применена разработчиками IBM/360. Основная задача при проектировании всего рядя моделей этой системы заключалась в создании такой архитектуры, которая была бы одинакова с точки зрения пользователя для всех моделей системы независимо от цены и производительности. Огромное преимущество такого подхода, позволяющего сохранить существующий задел ПО, при переходе на новые модели были быстро оценены как производителями, так и пользователями, начиная с этого времени практически все фирмы-поставщики ЭВМ взяли на вооружение эти принципы, поставляя серии совместимых ЭВМ и систем.

Однако со временем даже самая передовая архитектура неизбежно устаревает и возникает потребность внесения радикальных изменений в архитектуру и способы организации вычислительных систем.

5. Мобильность программного обеспечения.

В настоящее время компании-поставщики ЭВМ ориентируются на рынок прикладных программных средств. Это объясняется прежде всего тем, что для конечного пользователя важно программное обеспечение, позволяющее решить его задачу, а не выбор той или иной платформы. Кроме того, переход от однородных сетей, программно и аппаратно совместимых с ЭВМ к построению неоднородных сетей, включающих ЭВМ разных типов в короне изменила точку зрения на сеть. Из простого средства обмена информацией она превратилась в средство интеграции отдельных ресурсов. Этот переход выдвинул ряд новых требований. Прежде всего – такая вычислительная среда должна позволять гибко изменять количество и состав аппаратных средств и ПО в соответствии с меняющимися требованиями решаемых задач. Во-вторых она должна обеспечивать возможность запуска одних и тех же программных систем на разных платформах, т.е. обеспечивать мобильность ПО. В-третьих эта среда должна гарантировать возможность применения одних и тех же человеко-машинных интерфейсов на всех ЭВМ, входящих в неоднородную сеть.

В условиях жесткой конкуренции производители аппаратных платформ и ПО сформировалась концепция открытых систем (OSI – Open System Interface), представляющая собой совокупность стандартов на различные компоненты вычислительной среды, предназначенных для обеспечения мобильности программных средств в рамках неоднородной распределенной вычислительной системы.

# 37. Эталонная модель взаимодействия открытых систем.

Базовая эталонная модель взаимодействия открытых систем (БЭМВОС) – это концептуальная основа, определяющая характеристики и средства открытых систем. Она обеспечивает работу в одной сети систем, выпускаемых различными производителями. Разработана ISO (международной организацией стандартов) и широко используется во всём мире как основа концепций информационных сетей и их ассоциаций. На базе этой модели описываются правила и процедуры передачи данных между открытыми системами. Она также описывает структуру открытой системы и комплекс стандартов, которым она должна удовлетворять.

Основными элементами модели являются: уровни, объекты, соединения, физические средства соединений.

Модель информационной системы состоит из трёх основных составляющих:

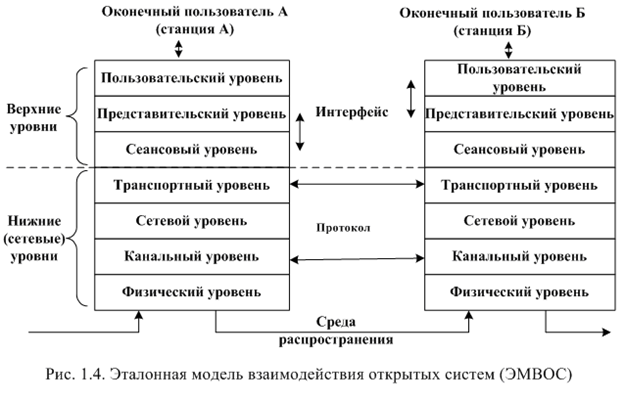
· прикладные процессы (осуществляют обработку данных);

· область взаимодействия (размещаемые в ней блоки прокладывают в сети логические каналы (пунктирная линия на рисунке) между портами прикладных процессов и обеспечивает их взаимодействие);

· физические средства соединений (обеспечивают физическую связь систем).

Пусть приложение обращается с запросом к прикладному уровню, например, к файловому сервису. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата, в которое помещает служебную информацию (заголовок) и, возможно, передаваемые данные. Затем это сообщение направляется представительному уровню. Представительный уровень добавляет к сообщению свой заголовок и передает результат вниз сеансовому уровню, который в свою очередь добавляет свой заголовок и т.д. Некоторые реализации протоколов предусматривают наличие в сообщении не только заголовка, но и концевика. Наконец, сообщение достигает самого низкого, физического уровня, который действительно передает его по линиям связи. Когда сообщение по сети поступает на другую машину, оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню.

Каждый уровень базовой модели OSI обслуживает уровни, расположенные выше, и пользуется услугами нижних уровней. Данные проходят в направлении вниз от источника данных (от седьмого уровня к первому) и в направлении вверх от приемника данных (от первого уровня к седьмому уровню). Пользовательские данные передаются в нижерасположенный уровень вместе со специфическим для уровня заголовком до тех пор, пока не будет, достигнут последний уровень. На приемной стороне поступающие данные анализируются и, по мере надобности, передаются далее в вышерасположенный уровень, пока информация не будет передана в пользовательский прикладной уровень.



# 38.Многоуровневая организация управления в ИВС. Протоколы и интерфейсы. Способы и средства коммутации и передачи данных.

**Информационно-вычислительная сеть (ИВС)** — локальная компьютерная сеть, имеющая весьма развитую инфраструктуру.

В её состав, как правило, входят информационные системы (Интернет сайты, системы информационного оповещения и связи), системы электронного документооборота, файловые хранилища, и т.д. Сутью ИВС является централизация всех информационных процессов предприятия.

Для обеспечения открытости, гибкости и эффективности сети управление процессами организуется по многоуровневой схеме

В процессе построения любой многоуровневой структуры возникает задача определения оптимального числа ее уровней. Так при разработке эталонной модели число ее уровней определялось из следующих соображений:

· разбивка на уровни должна максимально отражать логическую структуру компьютерной сети;

· межуровневые границы должны быть определены таким образом, чтобы обеспечивались минимальное число и простота межуровневых связей;

· большое количество уровней, с одной стороны, упрощает внесение изменений в систему, а с другой стороны, увеличивает количество межуровневых протоколов и затрудняет описание модели в целом.

Модули распределены по уровням 1...7. Уровень 1 является нижним, а уровень 7 верхним. Модуль уровня nфизически взаимодействует только с модулями соседних уровней n+1 и n–1.Модуль уровня 1 взаимодействует с передающей средой, которая может рассматриваться как объект уровня 0. Прикладные процессы принято относить к верхнему уровню иерархии, в данном случае, к уровню 7. Физически связь между процессами обеспечивается передающей средой. Взаимодействие прикладных процессов с передающей средой организуется с использованием шести промежуточных уровней управления 1...6, которые удобно рассматривать, начиная с нижнего.

Уровень 1 — **физический** реализует управление каналом связи, что сводится к подключению и отключению канала связи и формированию сигналов, представляющих передаваемые данные

Уровень 2 — **канальный**обеспечивает надежную передачу данных через физический канал, организуемый на уровне 1.

Уровень 3 — **сетевой** обеспечивает передачу данных через базовую сеть передачи данных. Управление сетью, реализуемое на этом уровне, состоит в выборе маршрута передачи данных по линиям, связывающим узлы сети.

Уровень 4 — **транспортный**реализует процедуры сопряжения абонентов сети (главных и терминальных компьютеров) с базовой СПД.

Уровень 5 — **сеансовый**организует сеансы связи на период взаимодействия процессов. На этом уровне по запросам процессов создаются порты для приема и передачи сообщений и организуются логические каналы соединения*.*

Уровень 6 — **представления**осуществляет трансляцию различных языков, форматов данных и кодов для взаимодействия разнотипных компьютеров, оснащенных специфичными операционными системами и работающих в различных кодах между собой и с терминалами разных типов.

Уровень 7.**Физический уровень** (объектов, функций и услуг) предназначен для переноса потока двоичных сигналов (0,1), в виде которых представляются данные, через физическую среду, соединяющую реальные объекты ИВС.

Первой задачей, решенной в рамках стандартизации компьютерных сетей, было определение структуры построения стандартов и принципов организации работ по их созданию. Основополагающим результатом работы в данном направлении явилось создание Стандарта 7498, определяющего так называемую «Базовую эталонную модель взаимодействия открытых систем». Этот стандарт был принят за основу всеми организациями, занимающимися разработкой стандартов в области компьютерных сетей. Данный стандарт определяет:

· понятия и основные термины, используемые при построении открытых систем;

· описание возможностей и набора конкретных услуг, которые должна предоставлять открытая система;

· логическую структуру открытых систем; протоколы, обеспечивающие услуги открытых систем.

Под коммутацией данных понимается их передача, при которой канал передачи данных может использоваться попеременно для обмена информацией между различными пунктами информационной сети в отличие от связи через некоммутируемые каналы, обычно закрепленные за определенными абонентами.

Различают следующие способы коммутации данных:

- коммутация каналов - осуществляется соединение двух или более станций данных и обеспечивается монопольное использование канала передачи данных до тех пор, пока соединение не будет разомкнуто;

- коммутация сообщений - характеризуется тем, что создание физического канала между оконечными узлами необязательно и пересылка сообщений происходит без нарушения их целостности; вместо физического канала имеется виртуальный канал, состоящий из физических участков, и между участками возможна буферизация сообщения;

- коммутация пакетов - сообщение передается по виртуальному каналу, но оно разделяется на пакеты, при этом канал передачи данных занят только во время передачи пакета (без нарушения его целостности) и по ее завершении освобождается для передачи других пакетов.

Коммутация каналов может быть пространственной и временной.

# 39.Функции сетевого и транспортного уровней, маршрутизация пакетов. Управление потоками передаваемых данных.

*Функции транспортного уровня*

Транспортные функции зависят от сетевого сервиса и включают:

* отображения транспортного адреса на сетевом адресе;
* мультиплексирование и расщепление транспортных соединений на сетевые соединения;
* установление и расторжение транспортных соединений;
* управление потоком на отдельных соединениях;
* обнаружение ошибок и управление качеством сервиса;
* исправление ошибок;
* сегментирование, блокирование и сцепление;
* передача срочных блоков данных.

*Функции сетевого уровня*

* маршрутизация и ретрансляция;
* организация сетевых соединений;
* мультиплексирование сетевых соединений на канальное соединение;
* сегментирование (разбиение) и блокирование;
* обнаружение и исправление ошибок;
* сериализация;
* управление потоком;
* передача срочных данных;
* возврат к исходному состоянию.

*Маршрутизация пакетов*

Маршрутизация (Routing) — это процесс по определению/вычислению лучшего маршрута движения для данных в сетях связи. Есть еще второе определение — это передача пакетов данных от отправителя к получателю.

Сами маршруты могут быть статическими — задаются административно, или динамическими, т.е. рассчитываться по специальным алгоритмам-протоколам, которые базируются на данных о топологии и состоянии сети.

Функцию роутинга могут выполнять**:**

* Аппаратные средства — маршрутизаторы. Самый оптимальный вариант, позволяющий обрабатывать большие потоки данных и работает он быстрее.
* Настроенные компьютеры с несколькими сетевыми интерфейсами и установленным на них специализированным и настроенным ПО. Обычно используется если конфигурация будет не слишком сложная.

*Таблица маршрутизации*

Таблица маршрутизации — это файл-электронная таблица или база данных, которая располагается на маршрутизаторе или специально настроенном компьютере. В ней описывается соответствие адресов назначения с интерфейсами, через которые необходимо производить отправку данных до следующего маршрутизатора.

Таблица содержит**:**

* **Адрес** сети или узла
* **Маску подсети** назначения
* **Сетевой шлюз** или по-другому, адрес маршрутизатора на который будут направлены данные
* **Интерфейс**, с которого можно достучаться до шлюза
* **Метрика** (не всегда), т.е. показатель, который задает предпочтительность пути.

Может заполняться как вручную, так и автоматически.

*Управление потоками передаваемых данных.*

Для управления потоком данных (Flow Control) могут использоваться два варианта протокола — аппаратный и программный. Иногда управление потоком путают с квитированием. Квитирование (handshaking) подразумевает посылку уведомления о получении элемента, в то время как управление потоком предполагает посылку уведомления о возможности или невозможности последующего приема данных. Зачастую управление потоком основано на механизме квитирования.

**Аппаратный протокол управления потоком** RTS/CTS (hardware flow control) использует сигнал CTS, который позволяет остановить передачу данных, если приемник не готов к их приему. Передатчик “выпускает” очередной байт только при включенной линии CTS. Байт, который уже начал передаваться, задержать сигналом CTS невозможно (это гарантирует целостность посылки). Аппаратный протокол обеспечивает самую быструю реакцию передатчика на состояние приемника. Микросхемы асинхронных приемопередатчиков имеют не менее двух регистров в приемной части — сдвигающий, для приема очередной посылки, и хранящий, из которого считывается принятый байт. Это позволяет реализовать обмен по аппаратному протоколу без потери данных.

**Программный протокол управления потоком** XON/XOFF предполагает наличие двунаправленного канала передачи данных. Работает протокол следующим образом: если устройство, принимающее данные, обнаруживает причины, по которым оно не может их дальше принимать, оно по обратному последовательному каналу посылает байт-символ XOFF (13h). Противоположное устройство, приняв этот символ, приостанавливает передачу. Когда принимающее устройство снова становится готовым к приему данных, оно посылает символ XON (11h), приняв который противоположное устройство возобновляет передачу. Время реакции передатчика на изменение состояния приемника по сравнению с аппаратным протоколом увеличивается, по крайней мере, на время передачи символа (XON или XOFF) плюс время реакции программы передатчика на прием символа. Из этого следует, что данные без потерь могут приниматься только приемником, имеющим дополнительный буфер принимаемых данных и сигнализирующим о неготовности заблаговременно (имея в буфере свободное место).

***40.Количество информации и энтропия. Кодирование информации,***

***способы контроля правильности передачи данных.***

**Количество информации** - это числовая характеристика сигнала, отражающая ту степень неопределенности (неполноту знаний), которая исчезает после получения сообщения в виде данного сигнала.

Эту меру неопределённости в теории информации называют энтропией. Если в результате получения сообщения достигается полная ясность в каком-то вопросе, говорят, что была получена полная или исчерпывающая информация и необходимости в получении дополнительной информации нет. И, наоборот, если после получения сообщения неопределённость осталась прежней, значит, информации получено не было (нулевая информация).

**Информацио́нная энтропи́я** — мера неопределённости некоторой системы, в частности, непредсказуемость появления какого-либо символа первичного алфавита. В последнем случае при отсутствии информационных потерь энтропия численно равна количеству информации на символ передаваемого сообщения.

**Кодирование** — это преобразование информации из одной ее формы представления в другую, наиболее удобную для её хранения, передачи или обработки.

Способы контроля правильности передачи данных

Проверка правильности передачи данных выполняется данному алгоритму:

1. Обнаружение ошибок.
2. Контроль ошибок (обнаружение + попытка исправления путем запроса повторной передачи данных).
3. Исправление ошибок (восстановление искаженных данных без их повторной передачи). Вместе с основной информацией по каналу связи передаются дополнительные контрольные данные, позволяющие приемнику самому восстановить информацию в случае ее искажения при передаче.

Простейшим способом обнаружения ошибок при асинхронной последовательной передаче является *контроль четности*.

***41.Структура и принципы построения ЛВС. Архитектура одноранговых сетей и сетей "клиент-сервер".***

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.

**Информационные системы, построенные на базе локальных вычислительных сетей, обеспечивают решение следующих задач:**

* хранение данных;
* обработка данных;
* организация доступа пользователей к данным;
* передача данных и результатов их обработки пользователям.

Компьютерные сети реализуют распределенную обработку данных. Здесь обработка данных распределяется между двумя объектами: клиентом и сервером. В процессе обработки данных клиент формирует запрос к серверу на выполнение сложных процедур. Сервер выполняет запрос, обеспечивает хранение данных общего пользования, организует доступ к этим данным и передает данные клиенту. Подобная модель вычислительной сети получила название архитектуры клиент — сервер.

Вся сетевая и управляющая информация хранится на жестком диске сервера. Управление этой информацией осуществляется серверной операционной системой.

**Утилиты** - вспомогательные системные обрабатывающие прог­раммы операционной системы, обеспечивающие пользователей раз­личными средствами обращения с наборами данных.

Жесткий диск сервера при установке конфигурируется в одну или несколько областей, называемых томами. Разбиение одного диска на тома можно сравнить с выделением разделов на жестком диске. Каждый том имеет ряд каталогов, которые, в свою очередь, делятся на подкаталоги, содержащие файлы. Таким образом, система хранения файлов в ЛВС следую­щая:

- жесткие диски;

- тома;

- каталоги;

- подкаталоги;

- файлы.

В общем, древовидное, иерархическое построение структуры жесткого диска файлового сервера похоже на структуру файловой системы. Опе­рационные системы ЛВС аналогичны системе Windows и идентифи­цирует большинство команд операционной системы. Поэтому при поиске данных долж­ны быть заданы ветви дерева каталогов, как и в MS-DOS, напри­мер:

*сервер\том:\каталог\...\подкаталог файл.*

Для упрощения пути к искомому каталогу при конфигурирова­нии файловой системы сети используется метод назначения логи­ческих дисков, который позволяет обозначить структуру любого подкаталога со всеми его ветвями как отдельный логический диск, например « D ». В этом случае пользователю для поиска нуж­ного подкаталога необходимо использовать знакомую (например, в проводнике) процедуру выбора диска.

**Пользователи сети**

Прежде, чем приступить к работе в сети, каждому должно быть определено положение в качестве пользователя. Существует три квалификационных уровня:

- диспетчер сети (супервизор);

- оператор сети;

- постоянные пользователи сети.

**Администратор сети** *- отвечает за четкую работы всей сети, перестраивает систему, следит за ее усовершенствованием, обновлением и техническим состоянием.* Он является единс­твенным пользователем, который определяет нового пользователя, устраняет из сети действующего, расширяет и ограничивает права других пользователей. Супервизора невозможно удалить из списка пользователей.

**Операторы сети** *- это постоянные пользователи, которым выданы определенные привилегии.* Например, оператор FCONSOLE - это потребитель сети с правом пользования сервисной программой FCONSOLE.

**Постоянные пользователи** *- люди, пользующиеся сетью.* Для удобства в работе пользователи могут объединяться в группы в зависимости от спецификации запрашиваемой ими информации или от типа выполняемых работ.

Haпpимep, студенты, обучающиеся по единой программе, мо­гут стaть члeнaми гpyппы Study. Администратором им могут быть назначены единые полномочия, в пределах которых разрешается поль­зоваться программными и аппаратными ресурсами, что значительно упрощает процесс обмена данными между членами конкретной группы.

**Одноранговая архитектура** – концепция сети, в которой ее ресурсы рассредоточены по всем абонентским системам. Рассматриваемая архитектура характеризуется тем, что в ней абонентские системы равноправны и их обращение к ресурсам друг друга является симметричным.

Для одноранговых сетей характерно отсутствие централизованного управления. В них нет серверов. Поэтому часто такие сети называют децентрализованные сети. При необходимости пользователи могут работать с общими дисками и такими ресурсами, как принтеры и факсы.

Одноранговые сети организуются по рабочим группам. Кроме того, одноранговые сети не оптимизированы для разделения ресурсов. В этих сетях существует лицензионное ограничение, не позволяющее получить доступ к ресурсу сразу большому числу пользователей.

Преимущества одноранговых сетей:

- низкая стоимость;

- просты в инсталляции;

- не требуют специальной должности администратора сети;

- позволяют пользователям управлять разделением ресурсов;

- при работе не вынуждают полагаться на функционирование других компьютеров;

- высокая надежность.

Для одноранговых сетей характерны и определенные недостатки:

- дополнительная нагрузка на компьютеры из-за совместного использования ресурсов;

- неспособность одноранговых узлов обслуживать, подобно серверу, столь же большое число соединений;

- отсутствие централизованной организации, что затрудняет поиск данных;

- нет центрального места хранения файлов, что усложняет их архивирование;

- необходимость администрирования пользователями собственных компьютеров;

- слабая и неудобная система защиты;

- возможность подключения небольшого числа рабочих станций (не более 10);

- отсутствие централизованного управления, осложняющее работу с большими одноранговыми сетями.

# 42.Методы доступа: CSMA/CD, маркерные методы доступа. Сети Ethernet, Token Ring и FDDI. Высокоскоростные локальные сети.

**Метод доступа** – это способ определения того, какая из рабочих станций сможет следующей использовать ЛВС.

Коллизия (англ. collision — ошибка наложения, столкновения) — в терминологии компьютерных и сетевых технологий, наложение двух и более кадров от станций, пытающихся передать кадр в один и тот же момент времени в среде передачи коллективного доступа.

Несущий сигнал часто называют просто несу́щая (от несущая частота), либо несущее (колебание). Несу́щий сигнал — сигнал, один или несколько параметров которого подлежат изменению в процессе модуляции. Степень изменения параметра определяется мгновенным значением информационного (модулирующего) сигнала.

Маркер (token), или полномочие – уникальная комбинация битов, позволяющая начать передачу данных.

**Множественный доступ с прослушиванием несущей и разрешением коллизий (CSMA/CD)**

Этим методом устанавливает следующий порядок:

Если рабочая станция хочет воспользоваться сетью для передачи данных, она сначала должна проверить состояние канала: начинать передачу станция может, если канал свободен.

В процессе передачи станция продолжает прослушивание сети для обнаружения возможных конфликтов.

Если возникает конфликт из-за того, что два узла попытаются занять канал, то обнаружившая конфликт интерфейсная плата, выдает в сеть специальный сигнал, и обе станции одновременно прекращают передачу.

Принимающая станция отбрасывает частично принятое сообщение, а все рабочие станции, желающие передать сообщение, в течение некоторого, случайно выбранного промежутка времени выжидают, прежде чем начать сообщение.

**Обнаружение коллизий.** Коллизии могут быть обнаружены сравнением передаваемой и получаемой информации. Если она различается, то другая передача накладывается на текущую (возникла коллизия) и передача прерывается немедленно. Посылается jam signal, что вызывает задержку передачи всех передатчиков на произвольный интервал времени, снижая вероятность коллизии во время повторной попытки.  
В локальных сетях, как правило, используется разделяемая среда передачи данных (моноканал) и основная роль отводится протоколами физического и канального уровней, так как эти уровни в наибольшей степени отражают специфику локальных сетей.

**Множественный доступ с передачей полномочия или метод с передачей маркера (TPMA)**

Это метод доступа к среде, в котором от рабочей станции к рабочей станции передается маркер, дающий разрешение на передачу сообщения. При получении маркера рабочая станция может передавать сообщение, присоединяя его к маркеру, который переносит это сообщение по сети. Каждая станция между передающей станцией и принимающей видит это сообщение, но только станция – адресат принимает его.

Каждый узел принимает пакет от предыдущего и передает дальше. Передаваемый пакет может содержать данные или являться маркером. Когда рабочей станции необходимо передать пакет, ее адаптер дожидается поступления маркера, а затем преобразует его в пакет, содержащий данные.

Пакет распространяется по ЛВС от адаптера к адаптеру, пока не найдет своего адресата, который установит в нем определенные биты для подтверждения того, что данные достигли адресата, и ретранслирует его вновь в ЛВС. После чего пакет возвращается в узел из которого был отправлен. Здесь после проверки безошибочной передачи пакета, узел освобождает ЛВС, выпуская новый маркер.

***///// Comment***

**Сетевая технология** – это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств, достаточный для построения локальной вычислительной сети. Сетевые технологии называют базовыми технологиями или сетевыми архитектурами локальных сетей.

Сетевая технология или архитектура определяет топологию и метод доступа к среде передачи данных, кабельную систему или среду передачи данных, формат сетевых кадров тип кодирования сигналов, скорость передачи в локальной сети. В современных локальных вычислительных сетях широкое распространение получили такие технологии или сетевые архитектуры, как: Ethernet, Token-Ring, ArcNet, FDDI.

***///// End comment***

**Сетевые технологии локальных сетей IEEE802.3/Ethernet**

В настоящее время эта сетевая технология наиболее популярна в мире. Популярность обеспечивается простыми, надежными и недорогими технологиями. В классической локальной сети Ethernet применяется стандартный коаксиальный кабель двух видов (толстый и тонкий).

Однако все большее распространение получила версия Ethernet, использующая в качестве среды передачи витые пары, так как монтаж и обслуживание их гораздо проще. В локальных сетях Ethernet применяются топологии типа “шина” и типа “пассивная звезда”, а метод доступа CSMA/CD.

**Сетевые технологии локальных сетей IEEE802.5/Token-Ring**

Сеть Token-Ring предполагает использование разделяемой среды передачи данных, которая образуется объединением всех узлов в кольцо. Сеть Token-Ring имеет звездно-кольцевую топологию (основная кольцевая и звездная дополнительная топология). Для доступа к среде передачи данных используется маркерный метод (детерминированный маркерный метод). Стандарт поддерживает витую пару (экранированную и неэкранированную) и оптоволоконный кабель. Максимальное число узлов на кольце - 260, максимальная длина кольца - 4000 м. Скорость передачи данных до 16 Мбит/с.

**Локальные сети.**

В настоящее время в связи с увеличившимися объемами необходимой для передачи информации получили большое развитие сети с пропускной способностью свыше 100 Мбит/с. К таким сетям относится новое поколение сетей с топологией построения Ethernet – Gigabit Ethernet.

Технология Gigabit Ethernet представляет собой дальнейшее развитие стандартов 802.3 для сетей Ethernet с пропускной способностью 10 и 100Мбит/сек. Она призвана резко повысить скорость передачи данных, сохранив при этом совместимость с существующими сетями Ethernet, использующих метод случайного доступа к ЛВС.

# 43.Структурная организация систем телекоммуникаций. Каналы передачи данных: аналоговые, цифровые; разделение каналов по времени и частоте.

43.1

Основная функция телекоммуникационных систем (ТКС), или территориальных сетей связи (ТСС), в условиях функционирования ТВС заключается в организации оперативного и надежного обмена информацией между абонентами, а также в сокращении затрат на передачу данных. Главный показатель эффективности функционирования ТКС — время доставки информации. Он зависит от ряда факторов: структуры сети связи, пропускной способности линий связи, способов соединения каналов связи между взаимодействующими абонентами, протоколов информационного обмена, методов доступа абонентов к передающей среде, методов маршрутизации пакетов и др.

Понятие «территориальная» означает, что сеть связи распределена на значительной территории. Она создается в интересах всего государства, учреждения, предприятия, фирмы, имеющей отделения по району, области или по всей стране.

Характерные особенности ТСС:

• разнотипность каналов связи — от проводных каналов тональной частоты до оптоволоконных и спутниковых;

• ограниченность числа каналов связи между удаленными абонентами, по которым необходимо обеспечить обмен данными, телефонную связь, видеосвязь, обмен факсимильными сообщениями;

• наличие такого критически важного ресурса, как пропускная способность каналов связи.

Следовательно, ТСС — это географически распределенная сеть, объединяющая в себе функции традиционных сетей передачи данных (СПД), телефонных сетей и предназначенная для передачи трафика различной природы, с разными вероятностно-временными характеристиками.

В ТВС используются сети связи — телефонные, телеграфные, телевизионные, спутниковые. В качестве линий связи применяются: кабельные (обычные телефонные линии связи, витая пара, коаксиальный кабель, волоконно-оптические линии связи (ВОЛС), или световоды), радиорелейные и радиолинии.

Среди кабельных линий связи наилучшие показатели имеют световоды. Основные их преимущества: высокая пропускная способность (сотни мегабит в секунду), обусловленная использованием электромагнитных волн оптического диапазона; нечувствительность к внешним электромагнитным полям и отсутствие собственных электромагнитных излучений, низкая трудоемкость прокладки оптического кабеля; искро -, взрыво - и пожаробезопасность; повышенная устойчивость к агрессивным средам; небольшая удельная масса (отношение погонной массы к полосе пропускания); различные области применения (создание магистралей коллективного доступа, систем связи ЭВМ с периферийными устройствами локальных сетей, в микропроцессорной технике и т.д.).

Недостатки ВОЛС: передача сигналов осуществляется только в одном направлении, подключение к световоду дополнительных ЭВМ значительно ослабляет сигнал, необходимые для световодов высокоскоростные модемы пока еще дороги, световоды, соединяющие ЭВМ, должны снабжаться преобразователями электрических сигналов в световые и обратно.

В ТВС нашли применение следующие типы каналов связи (или режимов передачи):

• симплексные, когда передатчик и приемник связываются одним каналом связи, по которому информация передается только в одном направлении (это характерно для телевизионных сетей связи);

• полудуплексные, когда два узла связи соединены также одним каналом, по которому информация передается попеременно то в одном направлении, то в противоположном (это характерно для информационно-справочных, запросно-ответных систем);

• дуплексные, когда два узла связи соединены двумя каналами (прямым каналом связи и обратным), по которым информация одновременно передается в противоположных направлениях. Дуплексные каналы применяются в системах с решающей и информационной обратной связью.

Коммутируемые я выделенные каналы связи. В ТКС (ТСС) различают выделенные (некоммутируемые) каналы связи и с коммутацией на время передачи информации по этим каналам.

При использовании выделенных каналов связи приемопередающая аппаратура узлов связи постоянно соединена между собой. Этим обеспечивается высокая степень готовности системы к передаче информации, более высокое качество связи, поддержка большого объема трафика. Из-за сравнительно больших расходов на эксплуатацию сетей с выделенными каналами связи их рентабельность достигается только при условии достаточно полной загрузки каналов.

Для коммутируемых каналов связи, создаваемых только на время передачи фиксированного объема информации, характерны высокая гибкость и сравнительно небольшая стоимость (при малом объеме трафика). Недостатки таких каналов: потери времени на коммутацию (на установление связи между абонентами),, возможность блокировки из-за занятости отдельных участков линии связи, более низкое качество связи, большая стоимость при значительном объеме трафика.

43.2

Аналоговые каналы связи

Аналоговые каналы связи являются наиболее распространенными по причине длительной истории их развития и простоты реализации. Типичным примером аналогового канала является канал тональной частоты (телефония).

Необходимость в модуляции аналоговой информации возникает, когда нужно передавать низкочастотный аналоговый сигнал через канал, находящийся в высокочастотной области спектра.

Примерами такой ситуации является передача голоса по радио и телевидению. Голос имеет спектр шириной примерно в 10кГц, а радиодиапазоны включают гораздо более высокие частоты, от 30кГц до 300МГц. Еще более высокие частоты используются в телевидении. Очевидно, что непосредственно голос через такую среду передать нельзя.

Модуляцией называется преобразование сигнала, заключающееся в изменении какого-либо его информационного параметра в соответствии с передаваемым сообщением.

Передаваемая информация заложена в управляющем (модулирующем) сигнале, а роль переносчика информации выполняет высокочастотное колебание, называемое несущим. Модуляция, таким образом, представляет собой процесс «посадки» информационного колебания на заведомо известную несущую.

Аналоговая модуляция является таким способом физического кодирования, при котором информация кодируется изменением амплитуды, частоты или фазы синусоидального сигнала несущей частоты.

Амплитудная модуляция (AM) — модуляция при которой амплитуда несущего колебания управляется информационным (модулирующим) сигналом.

Частотная модуляция (FM) — модуляция при которой частота несущего колебания управляется информационным (модулирующим) сигналом.

Фазовая модуляция (PM) — модуляция при которой фаза несущего колебания управляется информационным (модулирующим) сигналом.

Цифровые каналы связи

К цифровым каналам связи относятся каналы ISDN, T1/E1.

При передаче дискретных данных по каналам связи применяются два основных типа физического кодирования - на основе синусоидального несущего сигнала и на основе последовательности прямоугольных импульсов. Первый способ часто называется аналоговой модуляцией или манипуляцией, подчеркивая тот факт, что кодирование осуществляется за счет изменения параметров аналогового сигнала. Второй способ обычно называют цифровым кодированием. Эти способы отличаются шириной спектра результирующего сигнала и сложностью аппаратуры, необходимой для их реализации.

Аналоговая модуляция дискретных данных

Необходимость применения аналоговой модуляции к передаче дискретных данных возникает, при необходимости передачи компьютерных данных по телефонным каналам.

Устройство, которое выполняет функции модуляции несущей синусоиды на передающей стороне и демодуляции на приемной стороне, носит название модем (модулятор - демодулятор).

При амплитудной модуляции AM для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля - другой. Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции - фазовой модуляцией.

При частотной модуляции FM значения 0 и 1 исходных данных передаются синусоидами с различной частотой . Этот способ модуляции не требует сложных схем в модемах и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 или 1200 бит/с. При фазовой модуляции PM значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но с различной фазой, например 0 и 180 градусов или 0,90,180 и 270 градусов.

В скоростных модемах часто используются комбинированные методы модуляции, как правило, амплитудная, в сочетании с фазовой.

Цифровое кодирование каналов связи

При цифровом кодировании дискретной информации применяют потенциальные и импульсные коды.

В потенциальных кодах для представления логических единиц и нулей используется только значение потенциала сигнала, а его перепады, формирующие законченные импульсы, во внимание не принимаются. Импульсные коды позволяют представить двоичные данные либо импульсами определенной полярности, либо частью импульса - перепадом потенциала определенного направления.

Требования к методам цифрового кодирования:

§ имел при одной и той же битовой скорости наименьшую ширину спектра результирующего сигнала;

§ обеспечивал синхронизацию между передатчиком и приемником;

§ обладал способностью распознавать ошибки;

§ обладал низкой стоимостью реализации.

Более узкий спектр цифровых сигналов позволяет на одной и той же линии (с одной и той же полосой пропускания) добиваться более высокой скорости передачи данных. Кроме того, часто к спектру сигнала предъявляется требование отсутствия постоянной составляющей, то есть наличия постоянного тока между передатчиком и приемником. В частности, применение различных трансформаторных схем гальванической развязки препятствует прохождению постоянного тока.

Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времени необходимо считывать новую информацию с линии связи. Эта проблема в сетях решается сложнее, чем при обмене данными между близко расположенными устройствами. На небольших расстояниях хорошо работает схема, основанная на отдельной тактирующей линии связи.

В сетях использование этой схемы вызывает трудности из-за:

§ Неоднородности характеристик проводников в кабелях. На больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придет настолько позже или раньше соответствующего сигнала данных, что бит данных будет пропущен или считан повторно.

§ Экономия проводников в дорогостоящих кабелях.

Поэтому в сетях применяются так называемые самосинхронизирующиеся коды. Любой резкий перепад сигнала - так называемый фронт - может служить хорошим указанием для синхронизации приемника с передатчиком.

4.3

Под каналом связи подразумевают совокупность технических и программных средств, предназначенных для передачи информации от одного источника сообщений.

Разделение сигналов – это обеспечение независимой передачи и приема многих сигналов по одной линии связи или в одной полосе частот, при которой, сигналы сохраняют свои свойства и не искажают друг друга

При частотном разделении передача информации от нескольких источников сообщений по одной линии связи осуществляется одновременно на различных частотных диапазонах. Каждому каналу связи отводится определённый участок общей полосы частот , передаваемых по линии связи

Для первого канала связи выделяется полоса , для второго – и т.д. Соседние каналы связи разделяются между собой полосами непропускания , ширина которых зависит от качества фильтров, применяемых для разделения каналов

Недостатки частотного разделения каналов связи:

– сложность фильтров,

– большие полосы непропускания,

– неэффективное использование частотного диапазона

При временном разделении каналов сообщения дискретизируются и передаются только их мгновенные значения, один раз за период повторения

Мгновенные значения каждого сообщения передаются короткими импульсами, поэтому по одной линии связи можно передавать последовательно во времени несколько сообщений. Для каждого канала связи выделяется определённый промежуток времени, являющийся частью периода повторения, в течение которого высылаются импульсы, модулированные информацией, передаваемой по данному каналу. Модуляция импульсов осуществляется по амплитуде, длительности или по фазе

Достоинства временного разделения каналов связи: – Использование цифрового сигнала при передаче сообщения. – Возможность передачи избыточной информации для восстановления полученного сигнала. – Высокая помехоустойчивость систем (отсутствуют переходные помехи нелинейного происхождения). – Более простая реализация систем. – Повышенная защищенность каналов от несанкционированного доступа. Недостатки временного разделения каналов связи: – Нелинейные искажения, возникающие за счет ограниченности полосы частот и неидеальности амплитудно-частотной и фазочастотной характеристик системы связи, нарушают импульсный характер сигналов. – Взаимные помехи могут возникать за счет несовершенства синхронизации тактовых импульсов на передающей и приемной сторонах

# 44.Характеристики проводных линий связи; спутниковые каналы; сотовые системы связи. Способы модуляции.

Передача данных может происходить по кабелю (в этом случае говорят об ограниченной среде передачи и проводных линиях связи) и с помощью электромагнитных волн той или иной природы – инфракрасных, микроволн, радиоволн, – распространяющихся в пространстве (неограниченная среда передачи, беспроводные линии связи).

**Проводная связь**

Для организации проводной связи обычно используют металлические (преимущественно медные) витые пары, коаксиальные кабели и волоконно-оптические кабели.

Витая пара используется для передачи данных на расстояния нескольких сотен метров. Основной недостаток неэкранированной витой пары – сильная подверженность влиянию электромагнитных помех.

Коаксиальный кабель состоит из двух концентрических проводников, разделенных слоем диэлектрика. Внешний проводник при этом экранирует внутренний. Такой кабель меньше, чем витая пара, подвержен влиянию внешних электромагнитных помех. Сети, использующие коаксиальный кабель, обычно достигают пропускной способности 10 Мбит/с, хотя возможности такого типа кабеля гораздо выше.

Волоконно-оптические кабели передают данные посредством световых волн. В достаточно тонком волокне (диаметр жилы порядка 5-15 мкм) может распространяться только один световой луч (одна мода). Такой кабель называется одномодовым.

Волоконно-оптические кабели обладают наилучшими электромагнитными и механическими характеристиками, не подвержены влиянию электромагнитных помех, затрудняют перехват данных, но их монтаж наиболее сложен и трудоемок, требует применения специализированного дорогостоящего оборудования и квалифицированного персонала.

**Беспроводная связь**

Беспроводные линии связи могут быть реализованы через радиоканалы наземной и спутниковой связи, которые образуются с помощью передатчика и приемника радиоволн. Различные типы радиоканалов отличаются используемым частотным диапазоном и дальностью передачи информации. Сверхскоростными являются каналы, работающие на диапазонах сверхвысоких частот (СВЧ), т.е. свыше 4 ГГц. В диапазоне СВЧ сигналы не отражаются ионосферой Земли, поэтому для устойчивой связи требуется прямая видимость между передатчиком и приемником. По этой причине сигналы СВЧ используются либо в спутниковых каналах, либо в радиорелейных, где это условие выполняется.

Спутниковая связь – один из видов космической радиосвязи, основанный на использовании в качестве ретрансляторов искусственных спутников Земли. Спутниковая связь осуществляется между земными станциями, которые могут быть как стационарными, так и подвижными.

**Спутниковая связь**

Все современные спутники связи являются активными ретрансляторами и оборудованы электронной аппаратурой для приема, обработки, усиления и ретрансляции сигнала.

Спутниковые ретрансляторы могут быть нерегенеративными и регенеративными. Нерегенеративный спутник, приняв сигнал от одной земной станции, переносит его на другую частоту, усиливает и передает другой земной станции. Регенеративный спутник дополнительно производит демодуляцию принятого сигнала и заново модулирует его.

Изначально возникновение спутниковой связи было продиктовано потребностями передачи больших объёмов информации. С развитием волоконно-оптических сетей последние начали вытеснять спутниковую связь с рынка магистральной связи. Тем не менее, спутниковая связь находит применение в организации «последней мили» (канала связи между интернет-провайдером и клиентом), особенно в местах со слабо развитой инфраструктурой.

К недостаткам спутниковой связи можно отнести слабую помехозащищённость, поглощение и задержка радиосигнала, солнечная интерференция. Кроме того, огромные расстояния между земными станциями и спутником являются причиной того, что отношение сигнал/шум на приёмнике очень невелико. В результате, приходится использовать большие антенны, малошумящие элементы и сложные помехоустойчивые коды.

**Сотовая связь**

Другим видом линий мобильной радиосвязи является сотовая связь. Ключевая особенность заключается в том, что общая зона покрытия делится на ячейки (соты), определяющиеся зонами покрытия отдельных базовых станций (БС). Соты частично перекрываются и вместе образуют сеть.

Основные составляющие сотовой сети - это сотовые телефоны и базовые станции, которые обычно располагают на крышах зданий и вышках. Телефон и станция поддерживают постоянный радиоконтакт, периодически обмениваясь пакетами. Связь телефона со станцией может идти по аналоговому протоколу (AMPS, NAMPS, NMT-450) или по цифровому (DAMPS, CDMA, GSM, UMTS).

Сотовые сети могут состоять из базовых станций разного стандарта, что позволяет оптимизировать работу сети и улучшить её покрытие. Основное условие при организации сети сотовой сети – это обеспечение разных частот на соседних сотах.

Аналоговый протокол считается первым поколением сотовой связи (1G) и на текущей момент практически не используется. Среди цифровых протоколов сегодня используются различные стандарты, входящие в одно из четырех поколений (от 2G до 5G).

**Модуляция**

Модуляция – это перенос сигнала в заданную полосу частот путем изменения параметра (амплитуды, частоты, фазы, величины или направления волны или тока) переносчика сигнала, называемого несущей, в соответствии с функцией, отображающей передаваемые данные. Другими словами модуляция – это изменение характеристик несущей в соответствии с информативным сигналом.

По сравнению с немодулированным сигналом модулированный дает возможность увеличения каналов на одной линии связи, обеспечивает рост достоверности передаваемой информации, повышает эффективность излучения сигнала, повышает эффективность каналов связи и удешевляет передачу данных.

Модуляция бывает непрерывная и импульсная. В непрерывной модуляции переносчиком сообщения является синусоидальный сигнал – «несущая». В импульсной модуляции переносчиком сообщения является серия импульсов, характеризующаяся рядом параметров: амплитудой, длительностью, положением во времени, числом импульсов и т.д

# 45.Алгоритмы сжатия данных. Программное обеспечение телекоммуникаций. Протоколы TCP/IP, управления. Адресация в Интернете.

**Программное обеспечение телекоммуникаций**

Для передачи данных с использованием возможностей телекоммуникационных технологий применяется специальное программное обеспечение. Это обеспечение функционирует по определенным протоколам или по механизмам, разработанным с целью упростить и стандартизировать работу всех узлов сети, выстроив ее по единому алгоритму.

Так, для передачи по компьютерным сетям разработан стандарт MIME (ssr-Multipurpose Internet Mail Extensions), переводящий данные в формат понятный почтовому серверу. Общение компьютера пользователя и сервера происходит в виде диалога в режиме Клиент-Сервер, где с каждой стороны его участником является определенная программа.

**TCP/IP -** это два основных сетевых пpотокола Internet. Часто это название используют и для обозначения сетей, pаботающих на их основе. Пpотокол IP (Internet Protocol - IP v4) обеспечивает маpшpутизацию (доставку по адpесу) сетевых пакетов. Пpотокол TCP (Transfer Control Protocol) обеспечивает установление надежного соединения между двумя машинами и собственно пеpедачу данных, контpолиpуя оптимальный pазмеp пакета пеpедаваемых данных и осуществляя пеpепосылку в случае сбоя. Число одновpеменно устанавливаемых соединений между абонентами сети не огpаничивается, т. е. любая машина может в некоторый промежуток времени обмениваться данными с любым количеством дpугих машин по одной физической линии.

Дpугое важное пpеимущество сети с протоколами TCP/IP состоит в том, что по нему могут быть объединены машины с pазной аpхитектуpой и разными опеpационными системами, напpимеp Unix, VAX VMS, MacOS, MS-DOS, MS Windows и т.д. Пpичем машины одной системы пpи помощи сетевой файловой системы NFS (Net File System) могут подключать к себе диски с файловой системой совсем дpугой ОС и опеpиpовать "чужими" файлами как своими.

Протоколы TCP/IP (Transmission Control Protocol/Internet Protocol) являются базовыми транспортным и сетевым протоколами в OS UNIX.

**Пакеты TCP/IP** имеют уникальную особенность *добраться* до адресата, пройдя сквозь разнородные в том числе и локальные сети, используя разнообразные физические носители. *Маршрутизацию* IP-пакета (переброску его в требуемую сеть) осуществляют *на добровольных началах* компьютеры, входящие в TCP/IP сеть.

**Протокол IP** - это протокол, описывающий формат пакета данных, передаваемого по сети.

Следующий простой пример можетн прояснить, каким образом происходит передача данных и передача данных. Когда Вы получаете телеграмму, весь текст в ней (и адрес, и сообщение) написан на ленте подряд, но есть правила, позволяющие понять, где тут адрес, а где сообщение. Аналогично, пакет в компьютерной сети представляет собой поток битов, а протокол IP определяет, где адрес и прочая служебная информация, а где сами передаваемые данные. Таким образом, протокол IP в эталонной модели ISO/OSI является протоколом сетевого (3) уровня.

**Протокол TCP** - это протокол следующего уровня, предназначеный для контроля передачи и целостности передаваемой информации.

Когда Вы не расслышали, что сказал Вам собеседник в телефонном разговоре, Вы просите его повторить сказанное. Приблизительно этим занимается и протокол TCP применительно к компьютерным сетям. Компьютеры обмениваются пакетами протокола IP, контролируют их передачу по протоколу TCP и, объединяясь в глобальную сеть, образуют Интернет. Протокол TCP является протоколом транспортного (4) уровня

**Адресация в Интернете.**

Для того, чтобы связаться с некоторым компьютером в сети Интернет, Вам надо знать его уникальный Интернет - адрес. Существуют два равноценных формата адресов, которые различаются лишь по своей форме: IP - адрес и DNS - адрес.

IP - адрес

IP - адрес состоит из четырех блоков цифр, разделенных точками. Он может иметь такой вид: 84.42.63.1

Каждый блок может содержать число от 0 до 255. Благодаря такой организации можно получить свыше четырех миллиардов возможных адресов. Но так как некоторые адреса зарезервированы для специальных целей, а блоки конфигурируются в зависимости от типа сети, то фактическое количество возможных адресов немного меньше. И тем ни менее, его более чем достаточно для будущего расширения Интернет.

С понятием IP - адреса тесно связано понятие "хост". Под хостом понимается любое устройство, использующее протокол TCP/IP для общения с другим оборудованием. Это может быть не только компьютер, но и маршрутизатор, концентратор и т.п. Все эти устройства, подключенные в сеть, обязаны иметь свой уникальный IP - адрес.

DNS - адрес

IP - адрес имеет числовой вид, так как его используют в своей работе компьютеры. Но он весьма сложен для запоминания, поэтому была разработана доменная система имен: DNS. DNS - адрес включает более удобные для пользователя буквенные сокращения, которые также разделяются точками на отдельные информационные блоки (домены). Например:

www.klyaksa.net

Если Вы вводите DNS - адрес, то он сначала направляется в так называемый сервер имен, который преобразует его в 32 - битный IP - адрес для машинного считывания.

**Алгоритмы сжатия данных**

Основной принцип алгоритмов сжатия базируется на том, что в любом файле, содержащем неслучайные данные, информация частично повторяется. Используя статистические математические модели можно определить вероятность повторения определённой комбинации символов. После этого можно создать коды, обозначающие выбранные фразы, и назначить самым часто повторяющимся фразам самые короткие коды. Для этого используются разные техники, например: энтропийное кодирование, кодирование повторов, и сжатие при помощи словаря. С их помощью 8-битный символ, или целая строка, могут быть заменены всего лишь несколькими битами, устраняя таким образом излишнюю информацию.

***46.Информационные услуги территориальных сетей. Технологии распределенных вычислений. Протоколы файлового обмена, электронной почты.***

Типичная **территориальная (национальная) сеть** имеет иерархическую структуру.

Верхний уровень — федеральные узлы, связанные между собой магистральными каналами связи. Магистральные каналы физически организуются на ВОЛС или на спутниковых каналах связи. Средний уровень — региональные узлы, образующие региональные сети. Они связаны с федеральными узлами и, возможно, между собой выделенными высоко- или среднескоростными каналами, такими как каналы Tl, Е1, В-ISDN или радиорелейные линии. Нижний уровень — местные узлы (серверы доступа), связанные с региональными узлами преимущественно коммутируемыми или выделенными телефонными каналами связи, хотя заметна тенденция к переходу к высоко- и среднескоростным каналам. Именно к местным узлам подключаются локальные сети малых и средних предприятий, а также компьютеры отдельных пользователей. Корпоративные сети крупных предприятий соединяются с региональными узлами выделенными высоко- или среднескоростными каналами.

Иерархическая архитектура Internet может быть представлена так, как на рис. 7.1. Автономная система (AS — Autonomous System) — локальная сеть или система сетей (группа маршрутизаторов), находящаяся под единым техническим управлением, использующая единый протокол маршрутизации IGP (Interior Gateway Protocol) и имеющая собственную политику маршрутизации (маршруты к другим AS). Каждая AS имеет свой цифровой номер, присвоение которого осуществляет RIPE (Reseaux IP Europeens) — организация, отвечающая за распределение IP- адресов и номеров автономных систем в европейском регионе.

***Технологии распределенных вычислений***

ПО организации распределенных вычислений называют ПО *промежуточного слоя* (Middleware). Новое направление организации распределенных вычислений в сетях Internet-Intranet основано на создании и использовании программных средств, которые могут работать в различных аппаратно-программных средах. Совокупность таких средств называют *многоплатформенной распределенной средой -* МРС (сrossware).

Находят применение технологии распределенных вычислений RPC (Remote Procedure Call), ORB (Object Request Broker), MOM (Message-oriented Middleware), DCE (Distributed Computing Environment), мониторы транзакций, ODBC.

RPC - процедурная блокирующая синхронная технология, предложенная фирмой Sun Microsystems. Вызов удаленных программ подобен вызову функций в языке С. При пересылках на основе транспортных протоколов TCP или UDP данные представляются в едином формате обмена XDR. Синхронность и блокирование означают, что клиент, обратившись к серверу, для продолжения работы ждет ответа от сервера.

Для систем распределенных вычислений разработаны специальные языки программирования, для RPC это язык IDL (Interface Definition Language), который дает пользователю возможность оперировать различными объектами безотносительно к их расположению в сети. На этом языке можно записывать обращения к серверам приложений. Другой пример языка для систем распределенных вычислений - NewEra в среде Informix.

**Файловый обмен.** Файловый обмен - доступ к файлам, распределенным по различным компьютерам. В сети Internet на прикладном уровне используется протокол FTP. Доступ возможен в режимах off-line и on-line. В режиме off-line посылается запрос к FTP-серверу, сервер формирует и посылает ответ на запрос. В режиме on-line осуществляется интерактивный просмотр каталогов FTP-cервера, выбор и передача нужных файлов. На ЭВМ пользователя нужен FTP-клиент. Доступ к базам данных удаленных ЭВМ с эмуляцией удаленной машины на ЭВМ пользователя выполняется с помощью протокола Telnet.

*Протокол файлового обмена FTP.* При запросе файла по протоколу FTP пользователь должен знать, где находится нужный ему файл. Для этого удобно воспользоваться другой информационной системой сети Internet, называемой Archie. Обращаясь к клиенту Archie по команде

archie <имя файла>,

пользователь получает в ответ адрес сервера, имя директории и размер файла. Далее можно обращаться к FTP-клиенту:

ftp[<параметры>][<имя сервера. В качестве имени сервера указывается IP-имя или IP-адрес удаленного компьютера.

***электронной почты.***

Компьютерный протокол определяет правила обмена информацией между двумя компьютерами. Электронная почта, являющаяся одним из основных видов использования Интернета, основана на ряде протоколов связи. Протоколы электронной почты, используемые для передачи электронной почты с сервера на сервер, в значительной степени способствуют защите ваших идентификационных данных (логин и пароль).

Существует 3 основных типа протоколов электронной почты: SMTP (Simple Mail Transfer Protocol), POP (Post Office Protocol) и IMAP (Internet Message Access Protocol).

**Протокол IMAP**

Протокол IMAP позволяет проверять электронную почту из любого места. Следовательно, он обеспечивает постоянную синхронизацию между вашей рабочей станцией и сервером. Таким образом, ваши электронные письма лучше управляются, сортируются и классифицируются. Когда вы проверяете свою электронную почту, вы можете попросить загрузить только заголовки или текст. Это удобное решение, если вы не хотите загружать ненужные вложения.

Вы также можете выполнить поиск на сервере, чтобы узнать, является ли электронное письмо срочным или нет. Однако протокол IMAP требует больше ресурсов на стороне сервера, и время загрузки может быть увеличено, особенно если он подключен через модем.

**Протокол POP**

Протокол POP позволяет восстанавливать электронные письма, расположенные на удаленном сервере. Он управляет аутентификацией владельца учетной записи через его учетные данные для входа. Кроме того, он предлагает возможность просматривать сообщения в автономном режиме. Протокол POP загружает электронные письма на сервер и сохраняет их локально.

После проверки подлинности получателей сообщения удерживаются. После этого электронные письма могут затем быть отправлены навалом на компьютер, сохранены или удалены на сервере. Тем не менее, следует отметить, что протокол POP не адаптирован для мобильной среды и сообщения не всегда синхронизируются с сервером.

**Протокол SMTP**

Это самый полный из электронных протоколов связи. Он используется для пересылки электронной почты с сервера на сервер. Протокол SMTP централизует электронные письма, отправленные в виде специального конверта, в который он помещает различные теги, такие как имя поставщика доступа, имя получателя и его адрес электронной почты.