

1. Две формы представления информации. Способы представления дискретной информации. Системы счисления, используемые в вычислительной технике: двоичная, 8-я, 10-я, 16-я, двоично-десятичная.

Информация представляет собой понятие довольно емкое, вмещающее в себя весь окружающий нас мир (это вещи, явления, история, литература, искусство и многое другое). Вся информацию можно представить в двух формах Непрерывная и Дискретная

Непрерывные - Физические величины, а точнее их значения, характеризуют объекты и явления. Например, человека могут характеризовать такие физические величины, как масса тела, рост, температура тела, давление и т.д. В качестве явления, например, природы можно рассмотреть ураган, который будет характеризоваться такими физическими величинами, как скорость ветра, температура воздуха, количество выпавших осадков.

Величины, которые могут принимать не все возможные значения, а только вполне конкретные, называют дискретными. Дискретные величины характеризуются тем, что все их значения можно пронумеровать целыми числами. Примерами дискретных величин являются: Геометрические фигуры; Буквы алфавита; Цвета радуги.

Из цифр - числа; Из букв - слова; Из цифр, букв и математических символов - формулы и т.д. Во всех этих объектах заключена информация: В числах - информация о значениях; В словах - информация об именах и свойствах объектов; В формулах - информация о зависимостях между величинами.

2. Представление чисел с фиксированной точкой. Прямой, обратный и дополнительный код. Формирование битовых признаков переноса, переполнения, отрицательного результата, нуля.

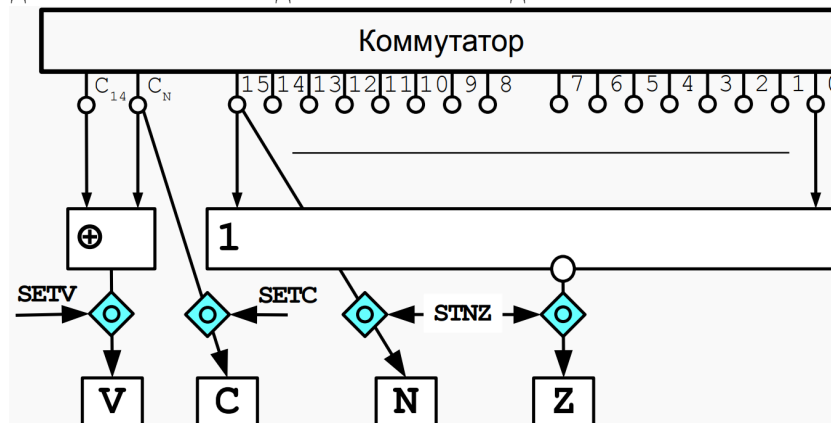
Целочисленный формат (формат с фиксированной точкой) используется для представления в компьютере целых (англ. integer) положительных и отрицательных чисел. Для этого, как правило, используются форматы, кратные байту: 1, 2, 4 байта. В форме с фиксированной запятой числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой (или точки), отделяющей целую часть от дробной. Эта форма проста и привычна для большинства пользователей, но имеет небольшой диапазон представления чисел и поэтому не всегда пригодна при вычислениях.

Прямой код представляет собой одинаковое представление значимой части числа для положительных и отрицательных чисел и отличается только знаковым битом. В прямом коде число 0 имеет два представления «+0» и «-0».

Обратный код для положительных чисел имеет тот же вид, что и прямой код, а для отрицательных чисел образуется из прямого кода положительного числа путем инвертирования всех значащих разрядов прямого кода. В обратном коде число 0 также имеет два представления «+0» и «-0».

Дополнительный код для положительных чисел имеет тот же вид, что и прямой код, а для отрицательных чисел образуется путем прибавления

1 к обратному коду. Добавление 1 к обратному коду числа 0 дает единое представление числа 0 в дополнительном коде.



3. Представление символьных и строковых данных. Принципы построения кодовых таблиц ASCII, КОИ-8, ISO8859-5, Windows-1251, UTF-8, UTF-16.

Базовый тип данных `char` понимается тройко: как байт - минимальная адресуемая единица представления данных в компьютере, как целое со знаком (в диапазоне $-127 \dots +127$) и как символ текста. Этот факт отражает общепринятые стандарты на представление текстовой информации, которые «защиты» как в архитектуре компьютера (клавиатура, экран, принтер), так и в системных программах. Стандартом установлено соответствие между символами и присвоенными им значениями целой переменной (кодами). Строки состоят из последовательности символов и заканчиваются `/0`. Символы могут кодироваться в различных кодировках.

- ASCII - представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов. Изначально разработанная как 7-битная, с широким распространением 8-битного байта ASCII стала восприниматься как половина 8-битной.
- КОИ-8 (код обмена информацией, 8 битов), KOI8 — восьмибитовая ASCII-совместимая кодовая страница, разработанная для кодирования букв кириллических алфавитов.
- ISO Стандарт ISO 8859-5 является международным и поэтому в нем представлены не только символы алфавита русского языка, но и других языков, использующих кириллицу:
 - Белорусский
 - Украинский
 - Болгарский
 - Сербский
 - Македонский

Это обстоятельство является особенно важным при разработке систем передачи, обработки и хранения информации. Кроме этого, кодировка ISO 8859-5 широко используется в основных международных и промышленных стандартах средств обработки, и передачи информации.

- Windows-1251 — набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для русских версий Microsoft Windows до 10-й версии. В прошлом пользовалась довольно большой популярностью. Была создана на базе кодировок, использовавшихся в ранних «самопальных» русификаторах Windows в 1990—1991 гг. совместно представителями «Параграфа», «Диалога» и российского отделения Microsoft.
- UTF - 8 распространённый стандарт кодирования символов, позволяющий более компактно хранить и передавать символы Юникода, используя переменное количество байт (от 1 до 4), и обеспечивающий полную обратную совместимость с 7-битной кодировкой ASCII. Стандарт UTF-8 официально закреплён в документах
- UTF - 16 Оба UTF-8 и UTF-16 являются кодировками переменной длины. Однако в UTF-8 символ может занимать минимум 8 бит, тогда как в UTF-16 длина символа начинается с 16 бит.

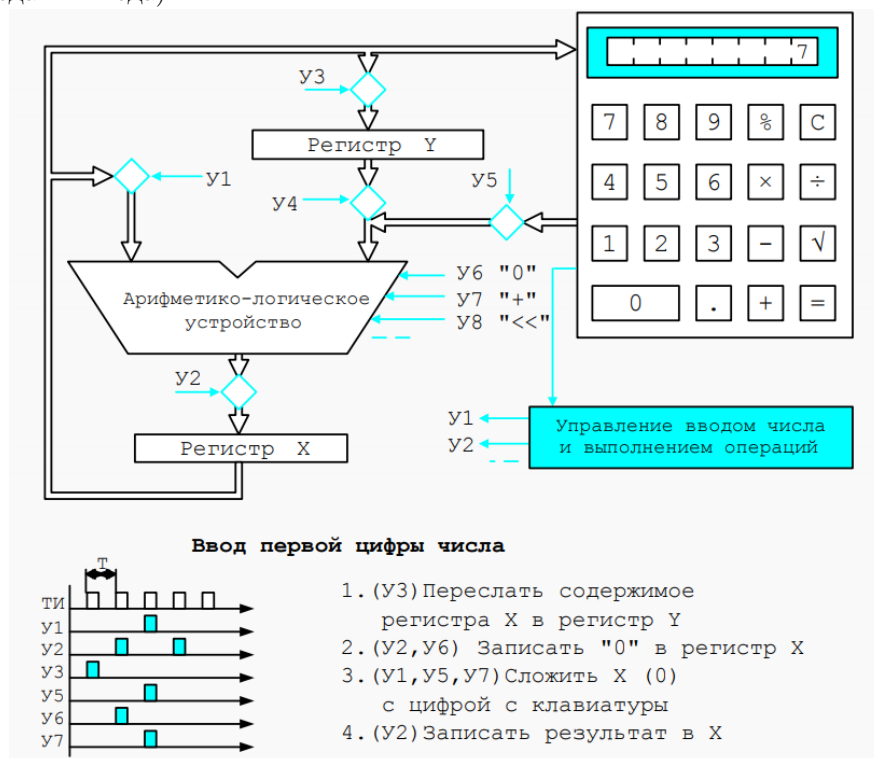
4. Базовые элементы вычислительной техники: ячейки, регистры, шины, вентили, тактовые генераторы, логические схемы, триггеры, регистры, счетчики, сумматоры

- Логические элементы(и или xor)
- Элементы хранения SRAM статическая память DRAM динамическая память
- Триггер – электронная схема, обладающая двумя устойчивыми состояниями. Переход из одного устойчивого состояния в другое происходит скачкообразно под воздействием управляющих сигналов. При этом также скачкообразно изменяется уровень напряжения на выходе триггера. Триггеры служат основой для построения регистров, счетчиков и других элементов, обладающих функцией хранения.
- Регистр – внутреннее запоминающее устройство процессора или внешнего устройства, предназначенное для временного хранения обрабатываемой или управляющей информации. Регистры представляют собой совокупность триггеров, количество которых равняется разрядности регистра, и вспомогательных схем, обеспечивающих выполнение некоторых элементарных операций.
- Сумматор устройство, производящее операцию сложения
- Тактовый генератор предназначен для синхронизации различных процессов. Он вырабатывает электрические импульсы (обычно прямоугольной формы) заданной частоты, которая часто используется как эталонная — считая количество импульсов, можно, например, измерять временные интервалы.

- Ячейка - Определенная ячейка в памяти запоминающего устройства, имеющая свой уникальный индекс в этой памяти
- Шина - соединение, служащее для передачи данных между функциональными блоками компьютера.
- Счётчик - устройство, на выходах которого получается двоичный или двоично-десятичный код, определяемый числом поступивших импульсов.

5. Структура и принцип функционирования ЭВМ. Порядок функционирования простого процессора на примере калькулятора.

ЭВМ состоит из нескольких основных устройств (арифметико-логическое устройство, управляющее устройство, память, внешняя память, устройства ввода и вывода).



6. Операционная система Unix — ядро ОС и файловая система.

Ядро - это самый низкий уровень программного обеспечения, которое взаимодействует с аппаратными средствами компьютера. Оно отвечает за взаимодействие всех приложений, работающих в пространстве пользователя вплоть до физического оборудования. Также позволяет процессам, известным как сервисы получать информацию друг от друга. Основные функции

- Инициализация системы - функция запуска и раскрутки.

- Управление процессами и нитями - функция создания, завершения и отслеживания существующих процессов и нитей ("процессов выполняемых на общей виртуальной памяти).
- Управление памятью - функция отображения практически неограниченной виртуальной памяти процессов в физическую оперативную память компьютера, которая имеет ограниченные размеры. Соответствующий компонент ядра обеспечивает разделяемое использование одних и тех же областей оперативной памяти несколькими процессами с использованием внешней памяти.
- Управление файлами - функция, реализующая абстракцию файловой системы, - иерархии каталогов и файлов.
- Коммуникационные средства - функция, обеспечивающая возможности обмена данными между процессами, выполняющимися внутри одного компьютера

Файловая система

Основана на том, что все есть файлы. У каждого файла есть его уникальный номер. Файлы имеют свои названия и номера, директории тоже файлы, которые хранят другие файлы

В ОС Windows жесткие диски называются латинскими буквами (C:, D:, ...), и каждый из дисков представляет собой корневой каталог с собственным деревом папок. В ОС Linux файловая система представлена единым корневым каталогом, обозначаемым как слэш (/).

Понятие «файл» в Linux имеет несколько другое значение, нежели в Windows. «Файлом» можно назвать обычный файл, содержащий данные, и интерпретируемый программой. Директория также является «файлом», содержащим в себе ссылки на другие директории или файлы с данными. Файлы устройства указывает на драйвер, благодаря которому система взаимодействует с физическими устройствами. Имеются и многие другие типы файлов.

Также стоит отметить чувствительность файловой системы Linux к регистру.

inode хранится метаданные о стандартных файлах, каталогах или других объектах файловой системы, кроме непосредственно данных и имени. фактические данные файла хранятся в блоках данных. Имя файла хранится в файлах специального типа — каталогах. Каталог файловой системы `s5fs` представляет собой таблицу, каждый элемент которой имеет фиксированный размер в 16 байтов: 2 байта хранят номер индексного дескриптора файла, а 14 байтов — его имя. Это накладывает ограничение на число inode, которое не может превышать 65 535.

7. Операционная система Unix — интерпретаторы, стандартные потоки ввода вывода, фильтры.

Интерпретатор команд

Интерпретатор командной строки - это любая программа, которая позволяет вводить команды и затем выполнять эти команды в операционной системе. Это буквально интерпретатор команд.

В отличие от программы, имеющей графический интерфейс, интерпретатор командной строки принимает строки текста с клавиатуры в качестве команд и затем преобразует эти команды в функции, понятные операционной системе.

Интерпретатор командной строки позволяет использовать некоторые из этих команд, при этом не нужно использовать их все сразу, что полезно в системах, у которых нет ресурсов для запуска графической программы.

shell bash, он запускает скрипты, которые мы ему передаем, при запуске он запускается и начинает работать с утилитами, которые нам нужны

Нужен для ввода команд через интерфейс командной строки или последовательного исполнения пакетных командных файлов.

По сути через него мы работаем с системой.

- `stdin(0)` поток ввода данных
- `stdout(1)` поток вывода данных
- `stderr(2)` поток вывода ошибок

мы можем давать различным потокам различные задачи тем самым фильтруя вывод

С помощью команд `grep sort cat` мы можем фильтровать информацию, используя их и их ключи, знак `|` передает всю данные другой команде

- `Cat` - посмотреть содержимое небольшого файла, склеить несколько файлов, фильтровать вывод
- `Grep` нахождение слов в файлах
- `Sort` Это утилита для вывода текстовых строк в определенном порядке. Проще говоря, для сортировки

8. Операционная система Unix — основные команды, права файлов и способы их задания.

`read write execute` - читать писать запускать
Владелец, группа, остальные

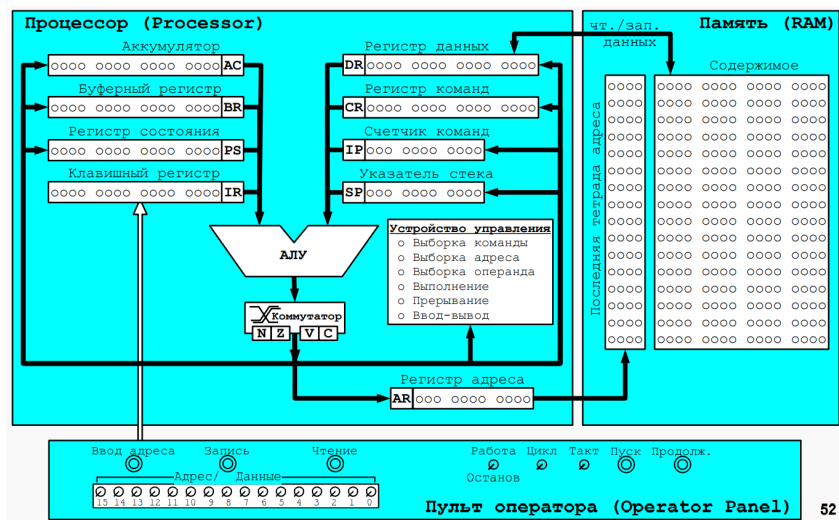
- 0 - никаких прав;
- 1 - только выполнение;
- 2 - только запись;
- 3 - выполнение и запись;
- 4 - только чтение;
- 5 - чтение и выполнение;
- 6 - чтение и запись;
- 7 - чтение запись и выполнение.

- $u+x$ - разрешить выполнение для владельца;
- $ugo+x$ - разрешить выполнение для всех;
- $ug+w$ - разрешить запись для владельца и группы;
- $o-x$ - запретить выполнение для остальных пользователей;
- $ugo+rw$ - разрешить все для всех;

`chmod u+x file`

`chmod 766 file`

9. Состав и структура БЭВМ. Адресные пространства БЭВМ. Система команд БЭВМ, форматы команд. Машинные циклы.



Адресуемая память БЭВМ

В БЭВМ память адресуема, т.е. мы подаем адрес и получаем из него данные. 16 разрядная память. На входе стоит дешифратор адреса 11 разрядный, после него данные или записываются или читаются.

Форматы команд

Команды

Код операции, операционный код, **опкод** — часть машинного языка, называемая инструкцией и определяющая операцию, которая должна быть выполнена.

1. Адресные

- Прямая абсолютная адресация 0 в 11

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				Адрес											
0															

КОП Код операции -> адрес ячейки (часто для изменения аккумулятора)

- Относительная адресация

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				1	Режим		Смещение								
0				1											

7-1 смещение относительно счетчика команд
+127 -128 ячеек от текущей

- Загрузкой операнда

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				1	1	1	1	Число							
0				1	1	1	1								

Используется младшие 8 бит как число для математических операций пример ADD 3 просто прибавляет число к аккумулятору

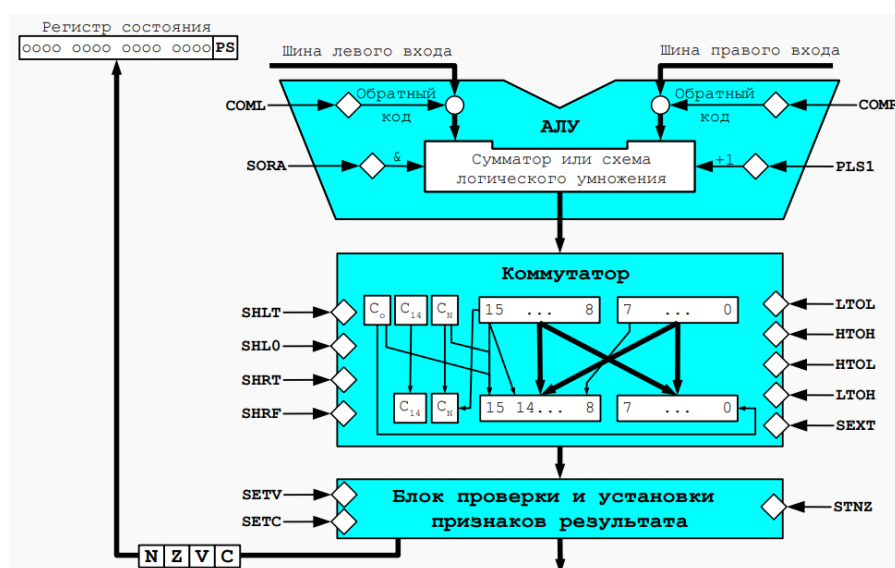
2. Безадресные

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				Расширение КОП											
0															

Всегда начинается с 0000

- Цикл выборки команды
- Цикл выборки адреса
- Цикл выборки операнды
- Цикл исполнения
- Цикл прерывания

10. Организация вычислений в БЭВМ. Сдвиги, арифметические и логические операции. Цикл выборки команды.



Цикл выборки команды

- IP -> BR, AR Содержимое IP через АЛУ записывается в BR и AR
- BR + 1 -> IP, MEM(AR) -> DR
- DR -> CR Содержимое DR через АЛУ записывается в CR

11. Организация массивов данных. Режимы адресации. Цикл выборки адреса и операнда БЭВМ.

Цикл выборки операнда

- DR -> AR Младшие 11 разрядов DR (адрес операнда из команды) пересылаются в AR
- MEM(AR) -> DR Загрузка из памяти по адресу 21 значения в DR

Код				Мнемоника	Описание	Реализация машинных циклов AF, OF
11	10	9	8			
0	M	M	M	ADD 0ADDR ADD \$L	Прямая абсолютная	CR → DR, DR → AR; MEM(AR) → DR
1	0	0	0	ADD (L)	Косвенная относительная	SXT_CR(0..7) → BR, BR + IP → AR, MEM(AR) → DR, DR → AR; MEM(AR) → DR
1	0	0	1		Резерв	
1	0	1	0	ADD (L) +	Косвенная автоинкрементная (постинкремент)	SXT_CR(0..7) → BR, BR + IP → AR, MEM(AR) → DR, DR + 1 → DR, DR → MEM(AR), DR - 1 → DR, DR → AR; MEM(AR) → DR
1	0	1	1	ADD - (L)	Косвенная автодекрементная (предекремент)	SXT_CR(0..7) → BR, BR + IP → AR, MEM(AR) → DR, DR - 1 → DR, DR → MEM(AR), DR → AR; MEM(AR) → DR
1	1	0	0	ADD &N ADD (SP+N)	Косвенная относительная, со смещением (SP)	SXT_CR(0, 7) → BR, BR + SP → DR, DR → AR; MEM(AR) → DR
1	1	0	1		Резерв	
1	1	1	0	ADD L ADD (IP+N)	Прямая относительная	SXT_CR(0..7) → BR, BR + IP → DR, DR → AR; MEM(AR) → DR
1	1	1	1	ADD #N	Прямая загрузка	CR(0, 7) → BR, BR → DR

Где здесь
AF и OF?

23

12. Управление вычислительным процессом в БЭВМ. Команды ветвлений, цикл исполнения команды LOOP.

Команды вычисления

- Логическое умножение AND M
- Логическое или OR M
- Сложение ADD M 4XXX M
- Сложение с переносом ADC M
- Вычитание SUB M
- dec
- inc

Команды ветвления

- Переход, если равенство BEQ D
- Переход, если неравенство BNE D
- Переход, если минус BMI D
- Переход, если плюс BPL D P
- Переход, если ниже/перенос BLO D
- Переход, если выше/нет переноса BHI D
- Переход, если переполнение BVS D
- Переход, если нет переполнения BVC D
- Переход, если меньше BLT D

- Переход, если больше или равно BGE D
- Безусловный переход BR D

Цикл исполнения Loop

DR после м.ц. OF содержит значение операнда

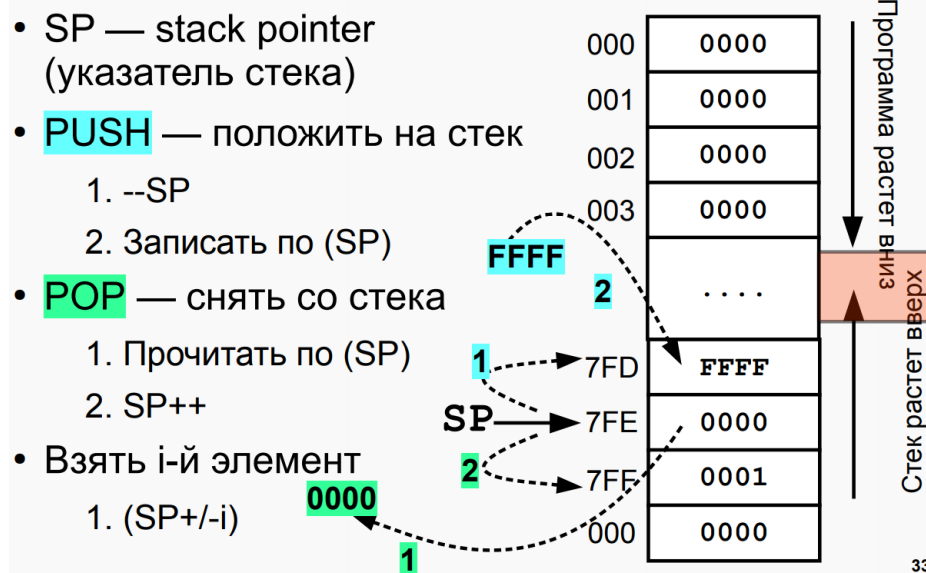
AR адрес операнда

- $\sim 0 + DR \rightarrow DR$; $\overline{0x0} = 0xFFFF = -1$; Вычитание единицы из DR
- $DR \rightarrow MEM(AR)$; Записываем значение операнда в память
- $\sim 0 + DR \rightarrow DR$; Вычитание еще единицы (ЗАЧЕМ?!)
- **if** DR(15) = 0 **then** GOTO INT ; Проверка на положительный (DR-1) и если да, то завершение цикла
- $IP + 1 \rightarrow IP$; Перескок через команду, если DR-1 отрицательное
- **GOTO INT** ; Завершение цикла

13. Подпрограммы в БЭВМ. Цикл исполнения команд перехода и возврата из подпрограммы. Стек, передача параметров. Позиционно-независимый код. Загрузчик и библиотеки.

Вызов подпрограммы call 0xXXX

Выход из неё - ret



CALL - цикл исполнения

- $DR \rightarrow BR$; Адрес перехода записать в BR
- $IP \rightarrow DR$; Подготовить адрес возврата для записи в стек
- $BR \rightarrow IP$; Переход на подпрограмму
- $0 + SP \rightarrow SP, AR$; Уменьшить стек на 1
- $DR \rightarrow MEM(AR)$; Записать адрес возврата
- GOTO INT ; Завершение цикл

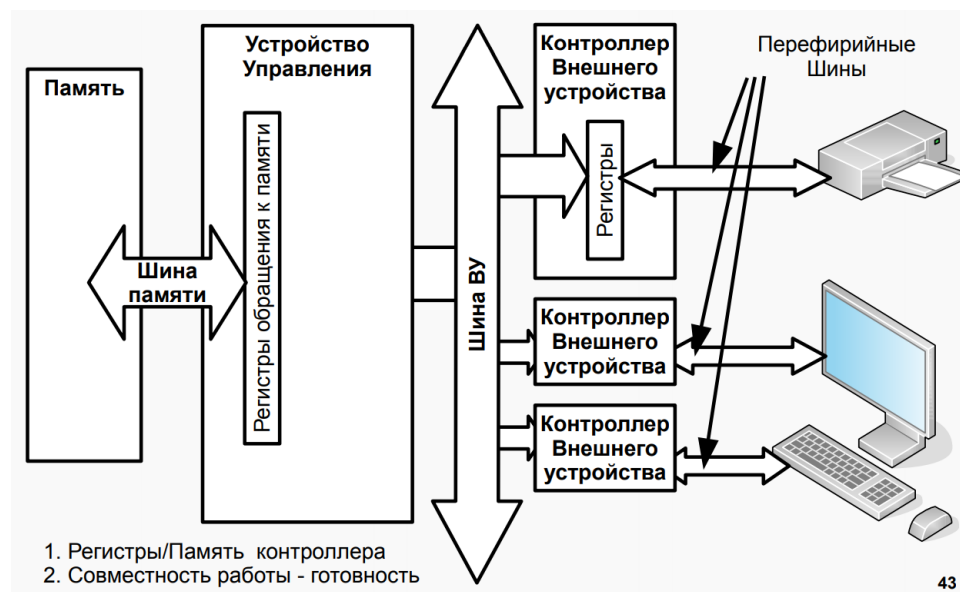
RET - цикл исполнения

- $SP \rightarrow AR$; Вершину стека поместить в AR
- $MEM(AR) \rightarrow DR$; Прочитать адрес возврата
- $DR \rightarrow IP$; Вернуться из подпрограммы
- $SP + 1 \rightarrow SP$; Увеличить стек на 1
- GOTO INT ; Завершение цикла

Любая ОС имеет соответствующую программу или часть ядра

- Загрузка по выбранному ОС адресу (даже в виртуальной памяти)
- Изменение константных частей адресов в программе
- Загрузка базовых значений регистров
- Динамическая загрузка разделяемых библиотек
- Связывание адресов основной программы с вызываемыми библиотеками

14. Организация ввода-вывода в вычислительных системах. Инициация обмена, передача информации и завершение обмена. Драйверы



Ввод-вывод

- Управляемы аппаратурой PDP
- Программно управляемый
 - Инициация обмена – Синхронная – Асинхронная – Управляемая прерываниями
 - Передача данных – Синхронная/Асинхронная
 - Завершение обмена и получение драйвером (программой) результата обмена – Синхронное/асинхронное

Драйверы

- Организуют совместную работу с устройством
- «Знают» о принципах работы устройства, адресах регистров, поддерживаемых режимах работы
- Управляются единообразным программным интерфейсом

15. Организация ввода-вывода в БЭВМ. Устройства ввода-вывода, команды.

Команды ввода-вывода

- Запрет прерываний DI
- Разрешение прерываний EI
- Ввод IN REG REG -> AC
- Вывод OUT REG AC -> REG
- Прерывание INT NUM Програмное прерывание с вектором NUM
- Возврат из прерывания IRET

ВУ в БЭВМ

- ВУ0 - Таймер Устанавливает готовность (и вызывает прерывание) раз в $100 \cdot DR$ миллисекунд
- ВУ1 - Устройство вывода
- ВУ2 - Устройство ввода
- ВУ3 - Устройство ввода-вывода
- ВУ4 - По функционалу похоже на ВУ-3 Адресуется 4-мя регистрами
- ВУ5 - Текстовый принтер Печатает символы из РДВУ в заданной кодировке
- ВУ6 - Бегающая строка Размер матрицы: 32x8
- ВУ7 - 8-ми разрядный 7-сегм. индикатор

- ## 16. Организация асинхронного обмена в БЭВМ. Пример программы. Временные издержки асинхронного обмена.

Сколько циклов команд БЭВМ будет ждать ввода второго символа?

[illegible]

- Совокупность адреса программы обработки прерывания и регистра состояния (PS)

- Необходимо инициализировать перед началом обработки прерывания
 - Хотя бы установить на подпрограмму, которая ничего не делает
 - Ответственность OS и БИОС
- В БЭВМ-NG ячейки с 0x000 по 0x10 – Всего 8 векторов, по два слова на вектор
 - На одном векторе может быть несколько прерываний

18. Организация обмена по прерыванию программы в БЭВМ.

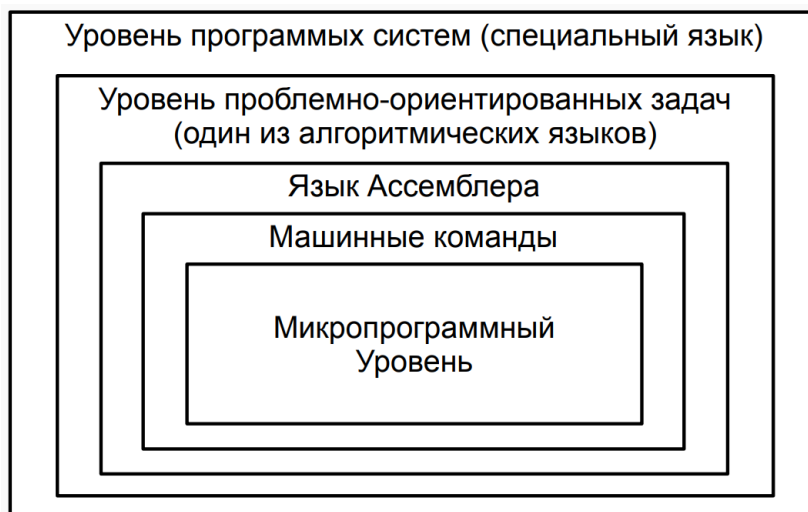
Пример программы. Цикл прерывания.

Цикл прерывания

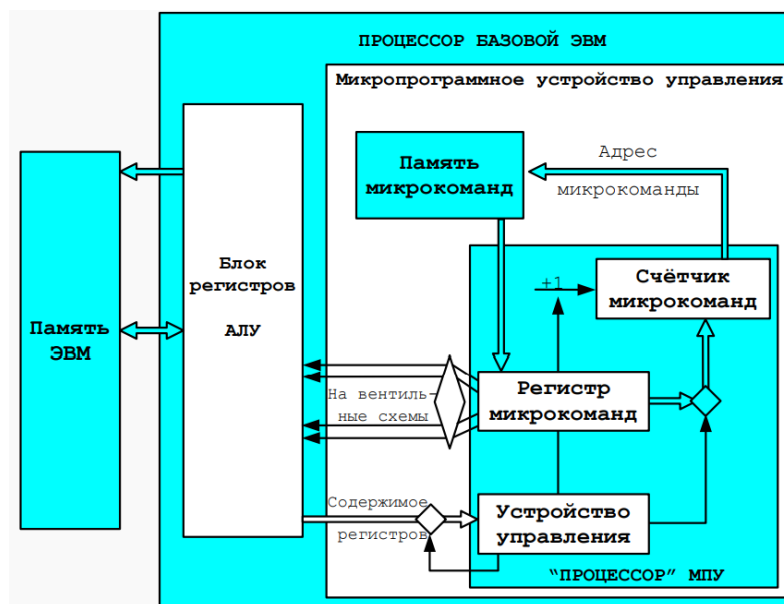
- if $PS(W) = 0$ then GOTO STOP; Проверка тумблера работа-останов, стоп если останов
- if $PS(IRQ) = 0$ then GOTO INFETCH; Если нет прерывания, то на выборку след. команды
- IRQSC ; Сформировать сигнал предоставления прерывания
- IP \rightarrow SP
- PS \rightarrow SP
- $SHL(BR) \rightarrow BR, AR$; Вычисляем адрес ячейки с переходом на подпрограмму обработки прерывания, как номер вектора * 2
- $MEM(AR) \rightarrow DR$; адрес обработчика прерывания записать в DR ...
- $DR \rightarrow IP$; ... а затем в IP
- $LTOI(BR + 1) \rightarrow AR$; ... выбрать адрес следующей ячейки вектора прерывания, ограничивая результат 8-ю разрядами
- $MEM(AR) \rightarrow DR$; содержимое PS обработчика прерывания записать в DR ...
- $DR \rightarrow PS$; ... а затем установить его в регистр

Проба - лабаб

19. Понятие многоуровневой ЭВМ. Понятие и пример программы на разных уровнях.



20. Микропрограммный уровень БЭВМ. Структура МПУ. Форматы микрокоманд.



Микрокоманды

устройством (АЛУ) и областью памяти, и на долгие годы она стала эталоном для создания ВС.

Время шло, и в 70-х годах прошлого века появились полупроводники, в которых можно было создавать сотни микроскопических проводников. Проблема множества контактов была снята, и наступила эра гарвардской архитектуры

- Гарвардская В гарвардской архитектуре принципиально невозможно осуществить операцию записи в память программ, что исключает возможность случайного разрушения управляющей программы в случае ошибки программы при работе с данными или атаки третьих лиц. Кроме того, для работы с памятью программ и с памятью данных организуются отдельные шины обмена данными (системные шины)
- Фон Неймана - Наша БЭВМ В архитектуре фон Неймана применяется однородная память микропроцессора. В эту память могут записываться различные программы. При этом специальная программа-загрузчик работает с ними как с данными. Затем управление может быть передано этим программам и они уже начинают выполнять свой алгоритм. При подобном подходе к управлению микропроцессором удастся достигнуть максимальной гибкости микропроцессорной системы.

В качестве недостатка архитектуры фон Неймана можно назвать возможность непреднамеренного нарушения работоспособности системы (программные ошибки) и преднамеренное уничтожение ее работы (вирусные атаки).

Разница между ними в том, что в Гарвардской Архитектуре память для инструкций и память для данных программы разделены, что позволяет оптимизировать работу

24. Архитектура многопроцессорных ЭВМ. Системный коммутатор. Архитектуры UMA и NUMA

UMA состоит из n процессоров, k модулей памяти и коммуникационной сети, связывающей процессоры и память. Сеть может стать причиной значительной задержки при обращении процессора к памяти. Система, в которой такая задержка одинакова для всех операций доступа к памяти, называется мультипроцессорной системой с однородным доступом к общей памяти (Uniform Memory Access, UMA) или системой с общей памятью. Поскольку процессоры выполняют команды с огромной скоростью, слишком большие задержки на выборку из памяти команд и данных для них не приемлемы. UMA - Мультипроцессорность - UNIFORM MEMORY ACCESS. У нас есть системная шина, в которой происходит основной обмен информацией, если мы используем один процессор, то все ок, но если мы хотим использовать 2 и более то возникли проблемы с синхронизацией. При добавлении процессоров шина перестала быстро работать.

Поняв это, люди придумали коммутатор, который соединяет все процессоры со всеми нужными им устройствами(память, устройства ввода вывода)

NUMA - NON UNIFORM MEMORY ACCESS

На 1 плате собратны 4 процессора их кэши их локальные памяти. Все эти платы соединены коммутатором с общей памятью и устройствами ввода-вывода NUMA каждый процессор имеет доступ не только к собственной локальной памяти, но и к памяти других процессоров сети. Но поскольку при обращении к памяти других процессоров запросы проходят через сеть, они выполняются дольше, чем обращения к локальной памяти. Системы этого типа называются мультипроцессорными системами с неоднородным доступом к памяти

Сокращения выводов БИС добиваются за счет использования двунаправленных шин. Так, вместо шин «Чтение» и «Запись» можно иметь лишь одну шину, по которой в зависимости от значения дополнительного управляющего сигнала данные передаются либо из памяти в процессор, либо наоборот. Полученная таким образом структура называется структура с раздельными шинами. Дальнейшего сокращения выводов БИС микропроцессора с трехшинной структурой (ША, ШД, ШУ) можно добиться за счет объединения адресной шины и шины данных. При этом возникает так называемая мультиплексируемая шина, по которой в одни моменты времени передаются адреса, а в другие – данные.

25. Структура современных процессоров. Окружение процессора. CISC, RISC, VLIW

- Complex Instruction Set Computer – Традиционные процессоры (например Intel), отягощенные совместимостью. 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.

- Reduced Instruction Set Computer – Простой набор инструкций, выполнение инструкции за такт Каждая команда выполняется за 1 машинный цикл

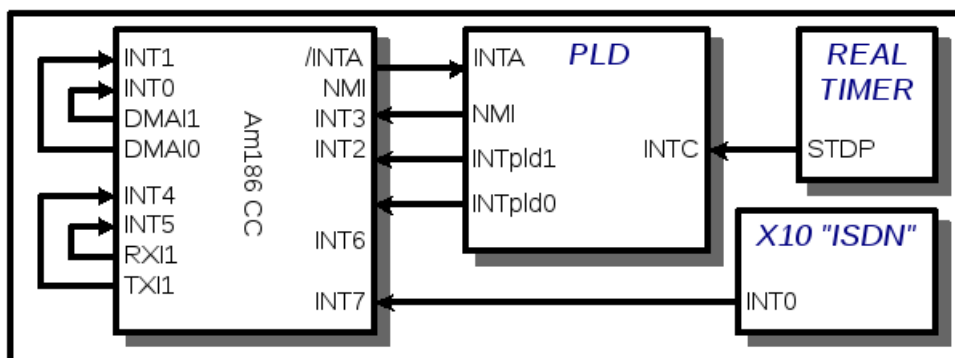
Все команды имеют одинаковую длину и используют минимум адресных форматов

Обращение к памяти только во время записи и чтения

Должно быть обеспечение языков высокого уровня

- Very Long Instructions Word – Несколько инструкций, упакованных в одну команду – Упаковка операций в инструкцию ложится на компилятор

VLIW (Very Long Instruction Word - очень длинная машинная команда) - архитектура процессоров, характеризующаяся возможностью объединения нескольких простых команд в так называемую связку. Входящие в нее команды должны быть независимы друг от друга и выполняться параллельно. Таким образом, из нескольких независимых машинных команд транслятор формирует одно «очень длинное командное слово».



26. Адресуемая память, организация и временные диаграммы. Конструктивные особенности современной памяти

- **SRAM** Статическая память Статическая память или Static Random Access — SRAM или СОЗУ — это полупроводниковая энергозависимая оперативная память с положительной обратной связью, обеспечивающей хранение двоичных или троичных разрядов. Отсутствие конденсаторов в конструкции микросхем исключает потерю заряда и, соответственно не требует поддержки состояния регенерации, что необходимо для работы динамической памяти. Микросхема статической памяти имеет произвольный доступ к хранимой информации и позволяет выбрать для чтения или записи любые биты или байты

Главным достоинством статической памяти является её высокое быстродействие, превышающее показатели динамической памяти. Благодаря минимальному времени доступа, не превышающему 2 нс, микросхемы статической памяти могут работать синхронно с процессорами, обеспечивая высокое быстродействие.

Главным недостатком СОЗУ является то, что её микросхема имеет намного меньшую плотность хранения данных, чем динамическая ОЗУ. Это означает увеличенные габариты, более высокую стоимость и меньшую ёмкость хранения данных.

- **DRAM** Динамическая память Динамическая оперативная память (DRAM – Dynamic Random Access Memory) – энергозависимая память с произвольным доступом, каждая ячейка которой состоит из одного конденсатора и нескольких транзисторов. Конденсатор хранит один бит данных, а транзисторы играют роль ключей, удерживающих заряд в конденсаторе и разрешающих доступ к конденсатору при чтении и записи данных.

Однако транзисторы и конденсатор – неидеальные, и на практике заряд с конденсатора достаточно быстро истекает. Поэтому периодически, несколько десятков раз в секунду, приходится дозаряжать конденсатор. К тому же процесс чтения данных из динамической памяти – деструктивен, то есть при чтении конденсатор разряжается, и необходимо его заново подзаряжать, чтобы не потерять навсегда данные, хранящиеся в ячейке памяти

Главное преимущество этого типа памяти заключается в чрезвычайно плотной упаковке ячеек, что позволяет создавать память большой ёмкости, при этом само устройство занимает очень мало места.

- SRAM статична (нет необходимости в перезаписи данных), в то время как SDRAM динамична (требуется периодическая перезапись данных). Время доступа в SDRAM зависит от тактового сигнала, в то время как в SRAM доступ осуществляется напрямую. Чип SDRAM может содержать несколько гигабит, в то время как чип SRAM (прим.: такого же размера) может содержать только несколько десятков мегабит. Ток потребления SRAM стабильный, в то время как у SDRAM он динамический и возрастает с увеличением циклов перезаписи. SRAM более дорогостоящая, чем SDRAM, т.к. более высокоскоростная.
- Адресуемая асинхронная память Мультиплексор осуществляет передачу данных с входов(много) на выход (один).

На вход приходит номер банка, номер строки n , номер столбца m - >номер банка поступает на дешифратор адреса банка, после выбора банка n поступает на дешифратор адреса строки, а m поступает на мультиплексор. Тем самым мы получаем меньшую схему

- Адресуемая память с фиксацией строк и столбцов
Разбиение адреса на 2 части - адрес строки и адрес столбца. Появилось два сигнала (RAS roll address select) выбор строки (CAS column address select) выбор столбца. Так половина адреса подается вместе с cas а другая половина подается вместе с gas. Они подаются на дешифратор и мультиплексор и они определяют нужную строку и столбец
- Синхронная память sdram использующегося в компьютере и других цифровых устройствах в качестве ОЗУ(оперативной памяти)
Синхронная оперативная память (SDRAM) — это первая технология оперативной памяти со случайным доступом (DRAM) разработанная для синхронизации работы памяти с тактами работы центрального процессора с внешней шиной данных. SDRAM основана на основе стандартной DRAM и работает почти также, как стандартная DRAM, но она имеет несколько отличительных характеристик, которые и делают ее более прогрессивной:
Синхронизированна по тактам с CPU Основана на стандартной DRAM, но значительно быстрее — вплоть до 4 раз
Как она работает - у неё есть тактовый генератор, и она работает по его тактам на определённой частоте. RAS и CAS инвертированы. Сигнал чтения данных и сигнал чтение/запись

27. Память, ориентированная на записи (блочная память).

Организация дисковой памяти и памяти на магнитных лентах

Блочная структура (схема). Адресное пространство памяти разбивается на группы последовательных адресов, и каждая такая группа обеспечивается отдельным банком памяти (рис. 25). Для обращения к ОП используется 9-разрядный адрес, семь младших разрядов которого (A6-A0) поступают

параллельно на все банки памяти и выбирают в каждом из них одну ячейку. Два старших разряда адреса (A8, A7) содержат номер банка. Выбор банка обеспечивается либо с помощью дешифратора номера банка памяти, либо путем мультиплексирования информации (на рис. 25 показаны оба варианта). В функциональном отношении такая ОП может рассматриваться как единое ЗУ, емкость которого равна суммарной емкости составляющих банков, а быстродействие – быстродействию отдельного банка

28. Характеристики запоминающих устройств. Пирамида памяти.

- Месторасположение – процессорные, внутренние, внешние
- Емкость – В метрических (Кило-) и двоичных (Киби-) множителях
- Единица пересылки – Слово, строка кэша, блок на диске
- Метод доступа – Произвольный (адресный), ориентированных на записи (прямой), последовательный, ассоциативный

	Объем	Тд	*	Тип	Управл.
CPU	100-1000 б.	<1нс	1с	Регистр	компилятор
L1 Cache	32-128Кб	1-4нс	2с	Ассоц.	аппаратура
L2-L3 Cache	0.5-32Мб	8-20нс	19с	Ассоц.	аппаратура
Основная память	0.5Гб-4ТБ	60-200нс	50-300с	Адресная	программно
SSD	128Гб-1Тб/drive	25-250мкс	5д	Блочн.	программно
Жесткие диски	0.5Тб-4Тб/drive	5-20мс	4м	Блочн.	программно
Магнитные ленты	1-6Тб/к	1-240с	200л	Последов.	программно

29. Ассоциативная память, Кэш-память. Влияние промахов кэш-памяти на производительность.

Ассоциативное ЗУ, АЗУ (англ. associative memory, content-addressable memory, CAM) — вид памяти, в котором адресация осуществляется на основе содержания данных, а не их местоположения, чем обеспечивается ускорение поиска необходимых записей (Материал из Википедии — свободной энциклопедии). АЗУ полезны и нужны для аппаратного ускорения поиска. Ведь любая задача поиска в конечном результате сводится к нахождению адреса ячейки памяти (для ускорения этого процесса данные сортируют, создают индексы и т.д.).

Кэш-память — это высокоскоростная память произвольного доступа, используемая процессором компьютера для временного хранения информации. Она увеличивает производительность, поскольку хранит наиболее

часто используемые данные и команды «ближе» к процессору, откуда их можно быстрее получить

Cache Miss(промах кэша) случается, когда запрашиваемые данные отсутствуют в кэше и их нужно подгружать из основного источника.

Так, например, у любого современного процессора есть кэш, в котором он хранит недавно прочитанные блоки оперативной памяти и обращение к кэшированным данным происходит существенно быстрее, чем "перекачивание" этих же данных из оперативной памяти на кристалл процессора. Если всё-таки запрашиваются данные, которые в данный момент в кэше отсутствуют или они уже не актуальны, они вместе с другими находящимися к ним близко данными, поступают в кэш процессора.

30. Предназначение и организация виртуальной памяти.

Сегментно-страничная организация. Устройство управления памятью (MMU), буфер трансляции (TLB)

Виртуальная память — метод управления памятью компьютера, позволяющий выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем. Для выполняющейся программы данный метод полностью прозрачен и не требует дополнительных усилий со стороны программиста, однако реализация этого метода требует как аппаратной поддержки, так и поддержки со стороны операционной системы. В системе с виртуальной памятью используемые программами адреса, называемые виртуальными адресами, транслируются в физические адреса в памяти компьютера. Трансляцию виртуальных адресов в физические выполняет аппаратное обеспечение, называемое блоком управления памятью. Для программы основная память выглядит как доступное и непрерывное адресное пространство либо как набор непрерывных сегментов, 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.

Сегментно-страничная организация виртуальной памяти адресное пространство пользователя разбивается на ряд сегментов по усмотрению программиста. Каждый сегмент в свою очередь разбивается на страницы фиксированного размера, равные странице физической памяти. С точки зрения программиста, логический адрес в этом случае состоит из номера сегмента и смещения в нем. С позиции операционной системы смещение в сегменте следует рассматривать как номер страницы определенного сегмента и смещение в ней. С каждым процессом связана одна таблица сегментов и несколько (по одной на сегмент) таблиц страниц. При работе определенного процесса в регистре процессора хранится начальный адрес соответствующей таблицы сегментов. Получив виртуальный адрес, процессор использует его часть, представляющую номер сегмента, в качестве индекса в таблице сегментов для поиска таблицы страниц данного сегмента. После этого часть адреса, представляющая собой номер страницы, используется для поиска номера физической страницы в таблице страниц. Затем часть адреса, представляющая смещение, используется для получения искомого физического адреса путем добавления к начальному адресу физической страницы.

MMU: Трансляция адресов. Поступает на вход виртуальный адрес, и далее он идет в таблицу преобразования присутствует ли эта страница в виртуальной памяти и её физический адрес.

TLB - туда записываются виртуальный адреса, и соответствующие им физические адреса, за счет этого увеличивается скорость

Трансляция адресов Трансляция виртуального адреса – это определение реального (физического) расположения ячейки памяти с данным виртуальным адресом, т. е. преобразование виртуального адреса в физический.

31. Сетевые технологии, Понятие сети ЭВМ, классификация компьютерных сетей. Сообщение и пакет. Модель взаимодействия открытых систем.

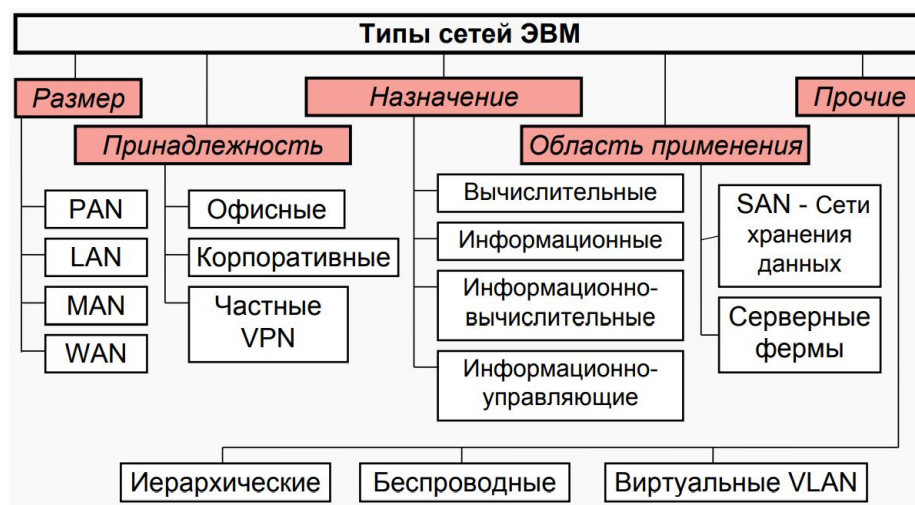
Сеть ЭВМ (вычислительная сеть, компьютерная сеть) = СВТ+СТК.

- Средства вычислительной техники (СВТ): ЭВМ, вычислительные комплексы (ВК) и вычислительные системы (ВС) - реализуют обработку данных.

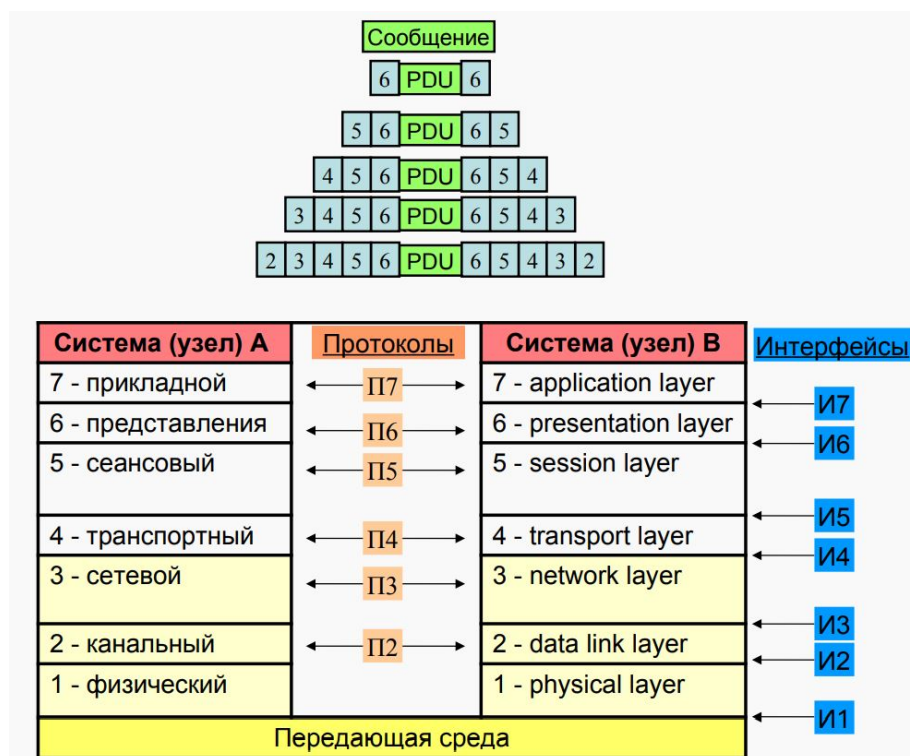
Средства телекоммуникаций (связи) (СТК): совокупность каналов связи и каналообразующей аппаратуры - реализуют передачу данных.

Классификация сетей ЭВМ

Разница между PAN и WAN - расстояние



Сообщение бьется по пакетам по 1500 байт, к нему присоединяется заголовок Адрес назначения, отправителя и концевик - контрольная сумма. На каждом уровне сообщение оборачивается новыми заголовками и концевиками, образуя матрешку.



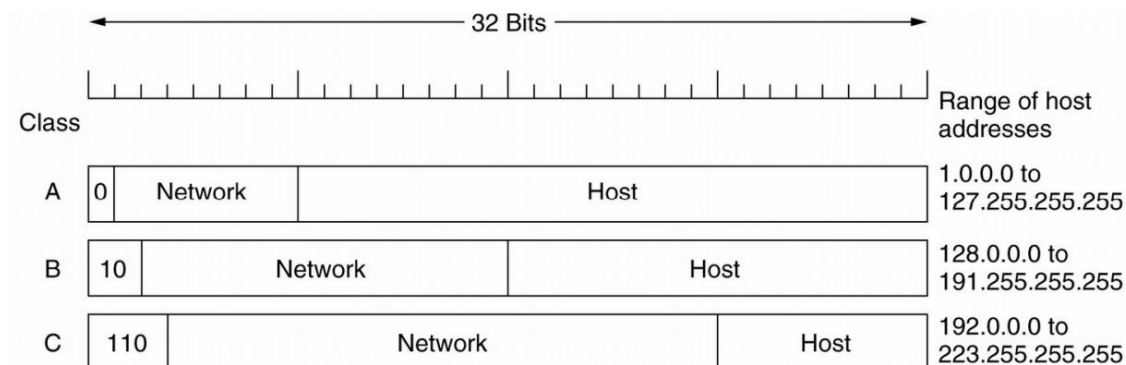
32. Модель TCP/IP: передающая среда, канальный и сетевой уровень. Адресация, передача и маршрутизация пакетов.

Уровни TCP/IP	Протоколы TCP/IP
Прикладной (поток Stream)	TELNET, FTP, SMTP, HTTP
Транспортный (сегмент Segment)	TCP, UDP
Сетевой (дейтаграмма Datagram)	IP
Доступ к среде (кадр Frame)	Ethernet, X.25, ATM, PPP, TokenRing

Канальный уровень ethernet

Представлен пакетами, которые состоят из преамбулы, адреса назначения, адреса источника, тип пакета, ip. Каждый физически mac адрес состоит из 6 байт информации, и выдается каждому wifi адаптеру. Каждый компьютер содержит локальную арт-таблицу, где есть соответствие, между локальными ip адресами и ethernet адресами.

Сетевой уровень IP



Если мы не знаем куда послать пакет, то мы обращаемся к маршрутизатору, он сообщает нам адрес, который записывается в таблицу, и в следующий раз мы найдем этот адрес там.

В начале мы спрашиваем кто знает кто IP ... если его нет в таблице, то маршрутизатор говорит что он знает кто это и отправляет свой MAC адрес компьютеру, далее мы отправляем запрос маршрутизатору с нужным MAC адресом и он уже обеспечивает запрос на нужный адрес.

33. Модель TCP/IP: выделение адресов (DHCP), сервисы имен, транспортный и прикладной уровни.

DHCP - Протокол, который существует в рамках локальных IP сетей, который автоматически выдает IP адреса.

Сетевые протоколы транспортного уровня – TCP и UDP

TCP — Transmission Control Protocol

Делает акцент на взаимном подключении сервиса и лица перед передачей данных, после передачи отправляется сообщение типа OK, данные пришли.

Низкая скорость но целостность

- Надежный, управляет перепосылкой данных
- Организует виртуальное соединение между гнездами (Socket=IP:port) на двух системах
- HTTP, FTP, SSH, SMTP

Используется для точной передачи

UDP

Не устанавливает готовность, за счет этого большая скорость но иногда получают битые пакеты

- Послал сообщение и забыл
- Контроль надежности оставлен разработчику
- Максимальная скорость передачи
- SNMP, TFTP, DHCP, DNS,

Онлайн

Протоколы прикладного уровня TCP/IP работают с прикладным программным обеспечением, запущенном на компьютере. Прикладной уровень не определяет работу самого приложения, но определяет службы, которые ему необходимы. Например, протокол приложения HTTP определяет, как веб-браузеры могут извлекать содержимое веб-страницы с веб-сервера. Проще говоря, прикладной уровень обеспечивает интерфейс между программным обеспечением, работающим на компьютере, и самой сетью.

34. Интерфейсы ввода-вывода. Контроллеры внешних устройств. Уровни стандартизации, сопряжения с системной шиной, циклы обмена. Регистры контроллера.