Архитектура ЭВМ и ВС
Тема 1.6. Внутренняя организация процессора

Преподаватель:
Шершова Л.Н.

Тема 1.6. Внутренняя организация процессора

Занятие 14. 1) Реализация принципов фон Неймана в ЭВМ. Структура процессора. 2) Устройство управления: назначение и упрощенная функциональная схема. 3) Регистры процессора: сущность, назначение, типы. Регистры общего назначения, регистр команд, счетчик команд, регистр флагов. 4) Принципы распараллеливания операций и построения конвейерных структур .

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM. 2) Арифметико-логическое устройство (АЛУ): назначение и классификация. Структура и функционирование АЛУ. 3) Интерфейсная часть процессора: назначение, состав, функционирование. Организация работы и функционирование.

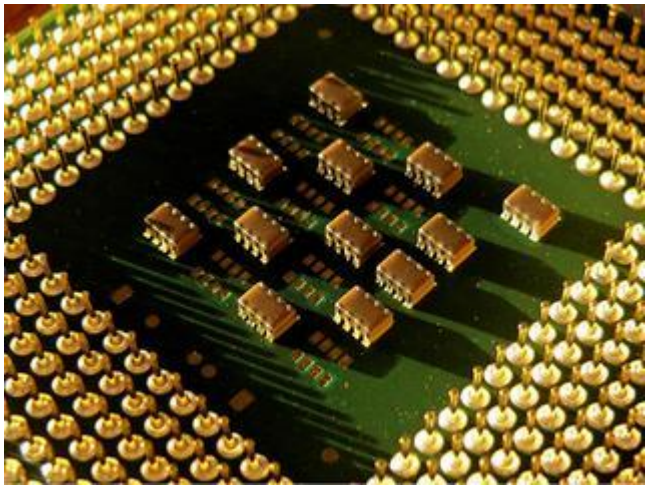
Занятие 16. 1) Построение последовательности машинных операций для реализации простых вычислений. 2) Структура команды процессора.

Занятие 17. 1) Цикл выполнения команды. Понятие рабочего цикла, рабочего такта. 2) Формат машинных команд.

Занятие 14. 1) Реализация принципов фон Неймана в ЭВМ. Структура процессора. 2) Устройство управления: назначение и упрощенная функциональная схема.

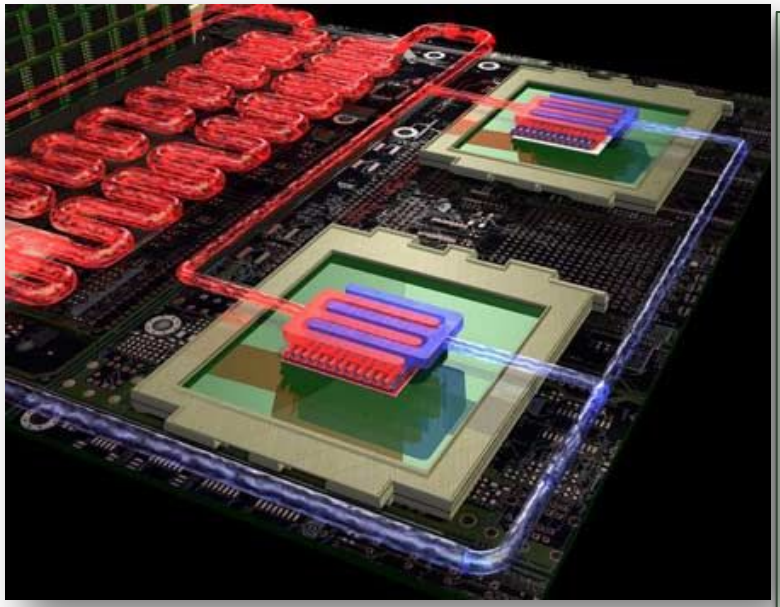
микропроцессор Netronome

- **Центральный процессор** (CPU, от англ. Central Processing Unit) — это основной рабочий компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера. Центральный процессор в общем случае содержит в себе:
  1. **арифметико-логическое устройство;**
  2. **шины данных и шины адресов;**
  3. **регистры;**
  4. **счетчики команд;**
  5. **кэш — очень быструю память малого объема (от 8 до 512 Кбайт);**
  6. **математический сопроцессор чисел с плавающей точкой.**
- **Современные процессоры выполняются в виде микропроцессоров**. Физически микропроцессор представляет собой интегральную схему — тонкую пластинку кристаллического кремния прямоугольной формы площадью всего несколько квадратных миллиметров, на которой размещены схемы, реализующие все функции процессора. Кристалл-пластинка обычно помещается в пластмассовый или керамический плоский корпус и соединяется золотыми проводками с металлическими штырьками, чтобы его можно было присоединить к системной плате компьютера.
- **Микропроцессор Netronome** — серия новых чипов компании Netronome будет выпускаться на предприятиях Intel с использованием самого передового на сегодняшний день 22 нм технологического процесса. Единый чип занимается одновременной обработкой пакетов данных и вопросами безопасности их передачи, виртуализацией ввода/вывода в серверах, устройствах и сетевых элементах в промышленных сетях и сетях общего пользования. Их производительность достигает 40 Гбит/с, позволяя использовать такие процессоры крупным сетевым провайдерам. Появление первых тестовых образцов нового поколения процессоров запланировано на 2013 год.



- **Кремний** – основа микропроцессоров **Микропроцессор - это интегральная схема**, сформированная на маленьком кристалле кремния. Кремний применяется в микросхемах в силу того, что он обладает полупроводниковыми свойствами: его электрическая проводимость больше, чем у диэлектриков, но меньше, чем у металлов. Кремний можно сделать как изолятором, препятствующим движению электрических зарядов, так и проводником - тогда электрические заряды будут свободно проходить через него. Проводимостью полупроводника можно управлять путем введения примесей. Микропроцессор содержит миллионы транзисторов, соединенных между собой тончайшими проводниками из алюминия или меди и используемых для обработки данных. В результате микропроцессор выполняет множество функций.

Занятие 14. 1) Реализация принципов фон Неймана в ЭВМ. Структура процессора. 2) Устройство управления: назначение и упрощенная функциональная схема.



Суперкомпьютер с одним из самых быстрых микропроцессоров. Используется новаторская система водяного охлаждения каждого микропроцессора.

- **Микропроцессор (МП),** или central processing unit (CPU) – функционально - законченное программно-управляемое устройство обработки информации, выполненное в виде одной или нескольких больших (БИС) или сверхбольших (СБИС) интегральных схем .

Функции микропроцессора:

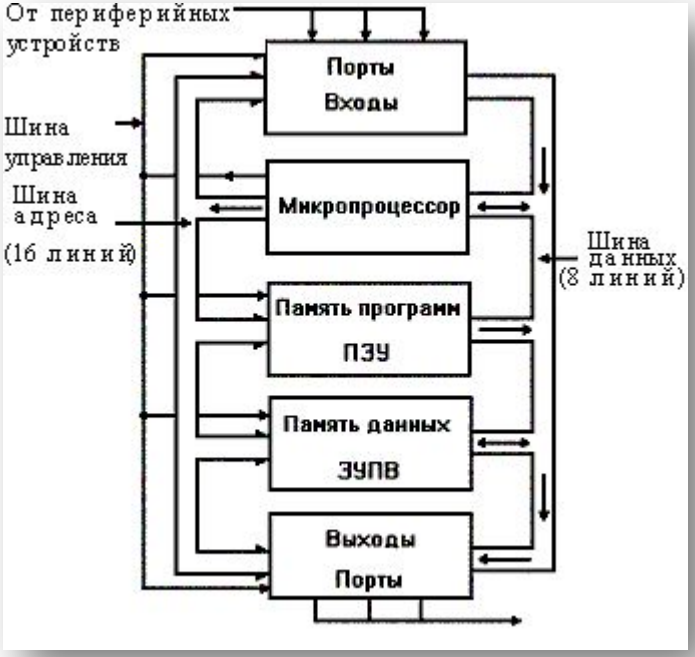
- вычисление адресов операндов или команд;
- выборка и дешифрация команд из оперативной памяти (ОП);
- выборка данных из ОП, регистров микропроцессорной памяти (МПП) и регистров адаптеров внешних устройств (ВУ);
- прием и обработка запросов и команд от адаптеров на обслуживание ВУ;
- выработка управляющих сигналов для всех прочих узлов и блоков ПК;
- переход к следующей команде.

Основные параметры микропроцессоров:

- разрядность (определяется разрядностью внутренних регистров, над которыми одновременно могут выполняться операции);
- адресное пространство (максимальное количество ячеек основной памяти которое может быть непосредственно адресовано микропроцессором);
- рабочая тактовая частота (частота исполнения тактов команд микропроцессором – определяет внутреннее быстродействие МП);
- кэш-память (память внутри ПМ работающая с частотой ядра МП);
- состав инструкций (перечень, вид и тип команд автоматически исполняемых МП);
- конструктив (размеры и разъемные соединения);
- рабочие напряжения (напряжение питания ядра и периферии ПМ)

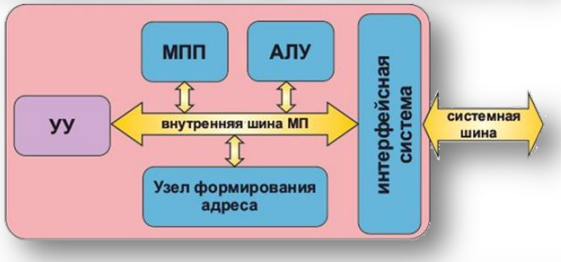
Занятие 14. 1) Реализация принципов фон Неймана в ЭВМ. Структура процессора. 2) Устройство управления: назначение и упрощенная функциональная схема.
Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

- **Первый микропроцессор был выпущен в 1971 году фирмой Intel (США)** – МП I4004. В настоящее время разными фирмами выпускаются разные процессоры и их разбивают **на 4 группы:**
- **МП типа CISC** (Complex Instruction Set Command) – **с полным набором системы команд**. Большинство персональных компьютеров использует CISC микропроцессоры. Большое разнообразие команд упрощает программирование, но с другой стороны усложняет структуру самого микропроцессора и ограничивает быстродействие.
- **МП типа RISC** (Reduced Instruction Set Command) – **с усеченным набором системы команд.** Процессор содержит только набор простых наиболее часто исполняемых команд. Сложные команды реализуются группами простых инструкций. Такая организация помогает упростить структуру ПМ и тем самым повысить быстродействие. Используются в машинах серверах.
- **МП типа VLIW** (Very Length Instruction Word) – **со сверхбольшим командным словом**. Большое машинное слово позволяет выполнять инструкции с одновременно, что увеличивает производительность. Сложность и высокая стоимость позволяет использовать такие архитектуры в больших и супер ЭВМ.
- **МП типа MISC** (Minimum Instruction Set Command) – **с минимальным набором системы команд**. Идея заключается в создании сверхбыстродействующих простых **МП параллельная организация работы** которых позволяет получить ЭВМ с высоким быстродействием. Применяется при создании супер ЭВМ.



Архитектура типового микропроцессора

Упрощенная структура микропроцессора



- **Устройство управления (УУ)** – является функционально наиболее сложным устройством микропроцессора. Оно вырабатывает управляющие сигналы во все блоки МП и компьютера сцелом для исполнения заданной команды.
- **Арифметико-логическое устройство (АЛУ)** – предназначено для выполнения арифметических и логических операций преобразования информации.
- **Микропроцессорная память (МПП)** – сверхбыстродействующая память, выполненная на регистрах и используемая микропроцессорам при непосредственном выполнении команд. Количество регистров МПП составляет несколько десятков.
- **Узел формирования адреса** – блок, отвечающий за формирования адресов для выбора следующих команд или данных.
- **Интерфейсная система** – выводы и схемы сопряжения, предназначенные для эффективной передачи адресов, данных, команд и управляющих сигналов.
- **Внутренняя шина МП** – обеспечивает связь составляющих частей микропроцессора.

- При включении микропроцессора интерфейсная система вводит команду по системной шине **в устройство управления**. В устройстве управления код команды дешифрируется и создается последовательность микрокоманд.
- **Устройство управления** согласно микрокомандам переключает (управляет) блоками процессора и компьютера заставляя их выполнять необходимые действия для исполнения заданной команды. Одновременно с этим формируется адрес для загрузки следующей команды или данных, если это необходимо.
- Такая **организация выполнения команд по одной цепочке имеется в скалярных микропроцессорах.** Начиная с процессоров Pentium, и им совместимые относятся к суперскалярным МП и способны выполнять несколько команд параллельно.
- В действительности структура процессора довольна сложна. В помощь процессору может использоваться кэш-память, математический сопроцессор, графический сопроцессор, сопроцессоры ввода-вывода, разгружающие центральный процессор от несложных операций взаимодействия с устройствами .
- В современных процессорах может находиться несколько ядер работающих параллельно на уровне оперативной памяти.
- Процессоры персональных компьютеров за время развития прошли несколько этапов эволюционирования от 16 разрядных до 32 и 64 битных и выше структур. Для обеспечения их совместимости **МП обеспечивают различные режимы работы: реальный, виртуальный, защищенный, 64 битный режим, режим совместимости**. Так же в самых последних процессорах имеются инструкции всех предыдущих поколений.
- Для уменьшения выделяемого тепла и потребляемой мощности, в последних моделях микропроцессоров введены блоки управления производительностью в зависимости от загруженности.

Занятие 14. 2) Устройство управления: назначение и упрощенная функциональная схема.

- Коды операции команд программы, воспринимаемые управляющей частью микропроцессора, расшифрованные и преобразованные в ней, дают информацию о том, какие операции надо выполнить, где в памяти расположены данные, куда надо направить результат и где расположена следующая за выполняемой команда.
- **Управляющее устройство** имеет достаточно средств для того, чтобы после восприятия и интерпретации информации, получаемой в команде, обеспечить переключение (срабатывание) всех требуемых функциональных частей машины, а также для того, чтобы подвести к ним данные и воспринять полученные результаты. Именно срабатывание, т. е. изменение состояния двоичных логических элементов на противоположное, позволяет посредством коммутации вентилей выполнять элементарные логические и арифметические действия, а также передавать требуемые операнды в функциональные части микроЭВМ.
- **Устройство управления в** строгой последовательности в рамках тактовых и цикловых временных интервалов работы микропроцессора (такт - минимальный рабочий интервал, в течение которого совершается одно элементарное действие; цикл - интервал времени, в течение которого выполняется одна машинная операция) **осуществляет: выборку команды; интерпретацию ее с целью анализа формата, служебных признаков и вычисления адреса операнда (операндов); установление номенклатуры и временной последовательности всех функциональных управляющих сигналов; генерацию управляющих импульсов и передачу их на управляющие шины функциональных частей микроЭВМ и вентили между ними; анализ результата операции и изменение своего состояния так, чтобы определить месторасположение (адрес) следующей команды.**

Занятие 14. 2) Устройство управления: назначение и упрощенная функциональная схема.

Упрощенная функциональная схема устройства управления микропроцессора

- ***Регистр команд,*** в котором хранится код выполняемой команды. Регистр команд расположен в интерфейсной части МП, в блоке регистров команд.
- ***Дешифратор операций*** - логический блок, выбирающий в соответствии с поступающим из регистра команд кодом операции (КОП) один из множества имеющихся у него выходов.
- ***Постоянное запоминающее устройство микропрограмм*** - хранит в своих ячейках управляющие сигналы, необходимые для выполнения в АЛУ операций обработки информации. Импульс, поступающий с соответствующего коду операции выхода дешифратора, считывает из ПЗУ микропрограмм необходимую последовательность управляющих сигналов.
- ***Узел формирования адреса*** (находится в интерфейсной части МП) - устройство, вычисляющее полный (физический) адрес ячейки памяти (регистра) по реквизитам, поступающим из регистра команд и регистров МПП.
- ***Кодовые шины данных, адреса и инструкций (управления)*** - часть внутренней интерфейсной шины микропроцессора

Занятие 14. 3) Регистры процессора: сущность, назначение, типы. Регистры общего назначения, регистр команд, счетчик команд, регистр флагов.

- Регистр процессора — **блок ячеек памяти**, **образующий сверхбыструю оперативную память (СОЗУ) внутри процессора**; используется самим процессором и большой частью недоступен программисту: например, при выборке из памяти очередной команды она помещается в **регистр команд,** к которому программист обратиться не может.
- Имеются также регистры, которые в принципе программно доступны, но обращение к ним осуществляется из программ операционной системы, например, **управляющие регистры и теневые регистры дескрипторов сегментов. Этими регистрами пользуются в основном разработчики операционных систем.**
- Существуют также так называемые **регистры общего назначения (РОН), представляющие собой часть регистров процессора, использующихся без ограничения в арифметических операциях**, но имеющие определенные ограничения, например в строковых. РОН, не характерные для эпохи мейнфреймов типа IBM/370 стали популярными в микропроцессорах архитектуры X86 — i8085, i8086 и последующих.
- **Специальные регистры** содержат данные, необходимые для работы процессора — смещения базовых таблиц, уровни доступа и т. д. Часть специальных регистров принадлежит устройству управления, которое управляет процессором путём генерации последовательности микрокоманд.
- Доступ к значениям, хранящимся в регистрах, как правило, в несколько раз быстрее, чем доступ к ячейкам оперативной памяти (даже если кеш-память содержит нужные данные), но объём оперативной памяти намного превосходит суммарный объём регистров .

Архитектура x86

IP (англ. InstructionPointer) — регистр, обозначающий смещение следующей команды относительно кодового сегмента.

IP — 16-битный (младшая часть EIP)

EIP — 32-битный аналог (младшая часть RIP)

RIP — 64-битный аналог

Сегментные регистры — регистры, указывающие на сегменты.

CS (англ. Code Segment), DS (англ. DataSegment), SS (англ. StackSegment), ES, FS, GS

В реальном режиме работы процессора сегментные регистры содержат адрес начала 64Kb сегмента, смещенный вправо на 4 бита.

В защищенном режиме работы процессора сегментные регистры содержат селектор сегмента памяти, выделенного ОС.

CS — указатель на кодовый сегмент. Связка CS:IP (CS:EIP/CS:RIP — в защищенном/64-битном режиме) указывает на адрес в памяти следующей команды.

Регистры данных — служат для хранения промежуточных вычислений.

RAX, RCX, RDX, RBX, RSP, RBP, RSI, RDI, R8 — R15 — 64-битные

EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI, R8D — R15D — 32-битные (extended AX)

AX (англ. Accumulator), CX (англ. Count Register), DX (англ. Data Register), BX (англ. Base Register), SP (англ. Stack Pointer), BP (англ. Base Pointer), SI (англ. Source Index), DI (англ. Destination Index), R8W — R15W — 16-битные

AH, AL, CH, CL, DH, DL, BH, BL, SPL, BPL, SIL, DIL, R8B — R15B — 8-битные (половинки 16-тибитных регистров)

например, AH — high AX — старшая половинка 8 бит

AL — low AX — младшая половинка 8 бит

где x — 8..15.

Регистры RAX, RCX, RDX, RBX, RSP, RBP, RSI, RDI, Rx, RxD, RxW, RxB, SPL, BPL, SIL, DIL доступны только в 64-битном режиме работы процессора.

Регистры общего назначения, регистр команд, счетчик команд, регистр флагов.

RAX			RCX			RDX			RBX		
	EAX			ECX			EDX			EBX	
	AX			CX			DX			BX	
	AH	AL		CH	CL		DH	DL		BH	BL

RSP			RBP			RSI			RDI			Rx	
	ESP			EBP			ESI			EDI		RxD	
	SP			BP			SI			DI		RxW	
		SPL			BPL			SIL			DIL		RxB

- **Регистр флагов** FLAGS (16 бит) / EFLAGS (32 бита) / RFLAGS (64 бита) — содержит текущее состояние процессора.
- Регистром называется функциональный узел, осуществляющий приём, хранение и передачу информации. Регистры состоят из группы триггеров, обычно D. По типу приёма и выдачи информации различают 2 типа регистров:
  1. С последовательным приёмом и выдачей информации — **сдвиговые регистры.**
  2. С параллельным приёмом и выдачей информации — **параллельные регистры**.
- Сдвиговые регистры представляют собой последовательно соединённую цепочку триггеров. Основной режим работы — сдвиг разрядов кода от одного триггера к другому на каждый импульс тактового сигнала.
- **По назначению регистры различаются на:**
  1. **аккумулятор** — используется для хранения промежуточных результатов арифметических и логических операций и инструкций ввода-вывода;
  2. **флаговые** — хранят признаки результатов арифметических и логических операций;
  3. **общего назначения** — хранят операнды арифметических и логических выражений, индексы и адреса;
  4. **индексные** — хранят индексы исходных и целевых элементов массива;
  5. **указательные** — хранят указатели на специальные области памяти (указатель текущей операции, указатель базы, указатель стека);
  6. **сегментные —** хранят адреса и селекторы сегментов памяти;
  7. **управляющие** — хранят информацию, управляющую состоянием процессора, а также адреса системных таблиц.

Счетчик команд IP

- **IP** (англ. *InstructionPointer*) — регистр, содержащий адрес-смещение следующей команды, подлежащей исполнению, относительно кодового сегмента CS в процессорах семейства x86.
- Регистр IP связан с CS в виде CS:IP, где CS является текущим кодовым сегментом, а IP — текущим смещением относительно этого сегмента.
- Регистр IP является 16-разрядным регистром-указателем. Кроме него, в состав регистров этого типа входят SP (англ. *StackPointer* — указатель стека) и BP (англ. *BasePointer* — базовый указатель).

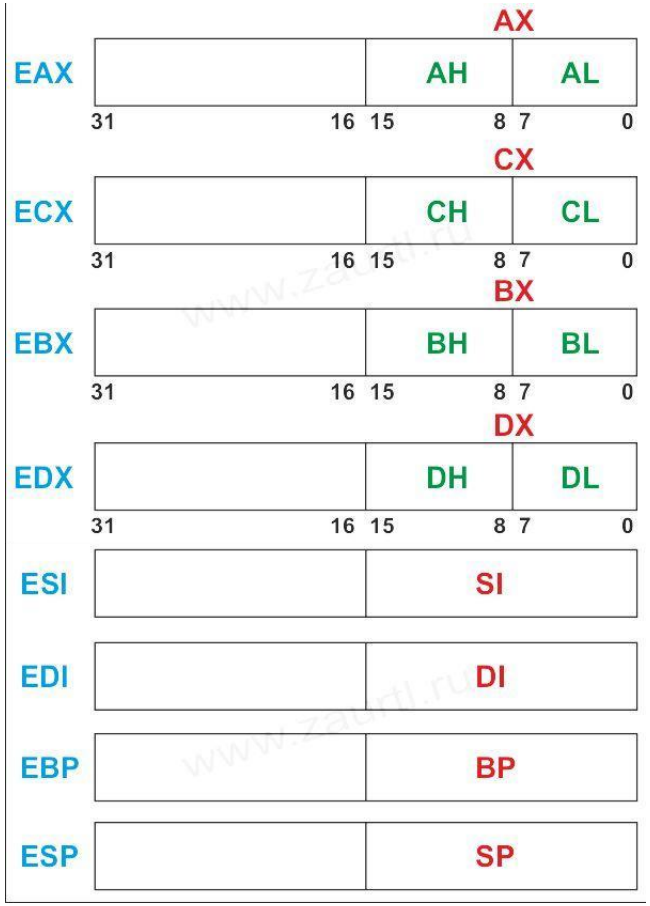
Принцип работы

- Например, CS содержит значение 2CB5[0]H, в регистре IP хранится смещение 123H.
- Адрес следующей инструкции, подлежащей исполнению, вычисляется путем суммирования адреса в CS (сегменте кода) со смещением в регистре IP:
- 2CB50H + 123H = 2CC73H
- Таким образом, адрес следующей инструкции для исполнения равен 2CC73H.
- При выполнении текущей инструкции процессор автоматически изменяет значение в регистре IP, в результате чего регистровая пара CS:IP всегда указывает на следующую подлежащую исполнению инструкцию.
- **EIP**
- Начиная с процессора 80386 была введена 32-разрядная версия регистра-указателя — **EIP** (англ. *InstructionPointer*). В данном случае IP является младшей частью этого регистра (первые 16 разрядов). Принцип работы EIP в целом схож с работой регистра IP. Основная разница состоит в том, что в защищённом режиме, в отличие от реального режима, регистр CS является селектором (селектор указывает не на сам сегмент в памяти, а на его дескриптор сегмента в таблице дескрипторов).
- **RIP**
- В 64-разрядных процессорах используется свой регистр-указатель инструкций — **RIP**.
- Младшей частью этого регистра является регистр EIP.
- На основе RIP в 64-разрядных процессорах введён новый метод адресации RIP-relative. В остальном работа RIP аналогична работе регистра EIP.

Занятие 14. 3) Регистры процессора: сущность, назначение, типы. Регистры общего назначения, регистр команд, счетчик команд, регистр флагов. 4) Принципы распараллеливания операций и построения конвейерных структур .

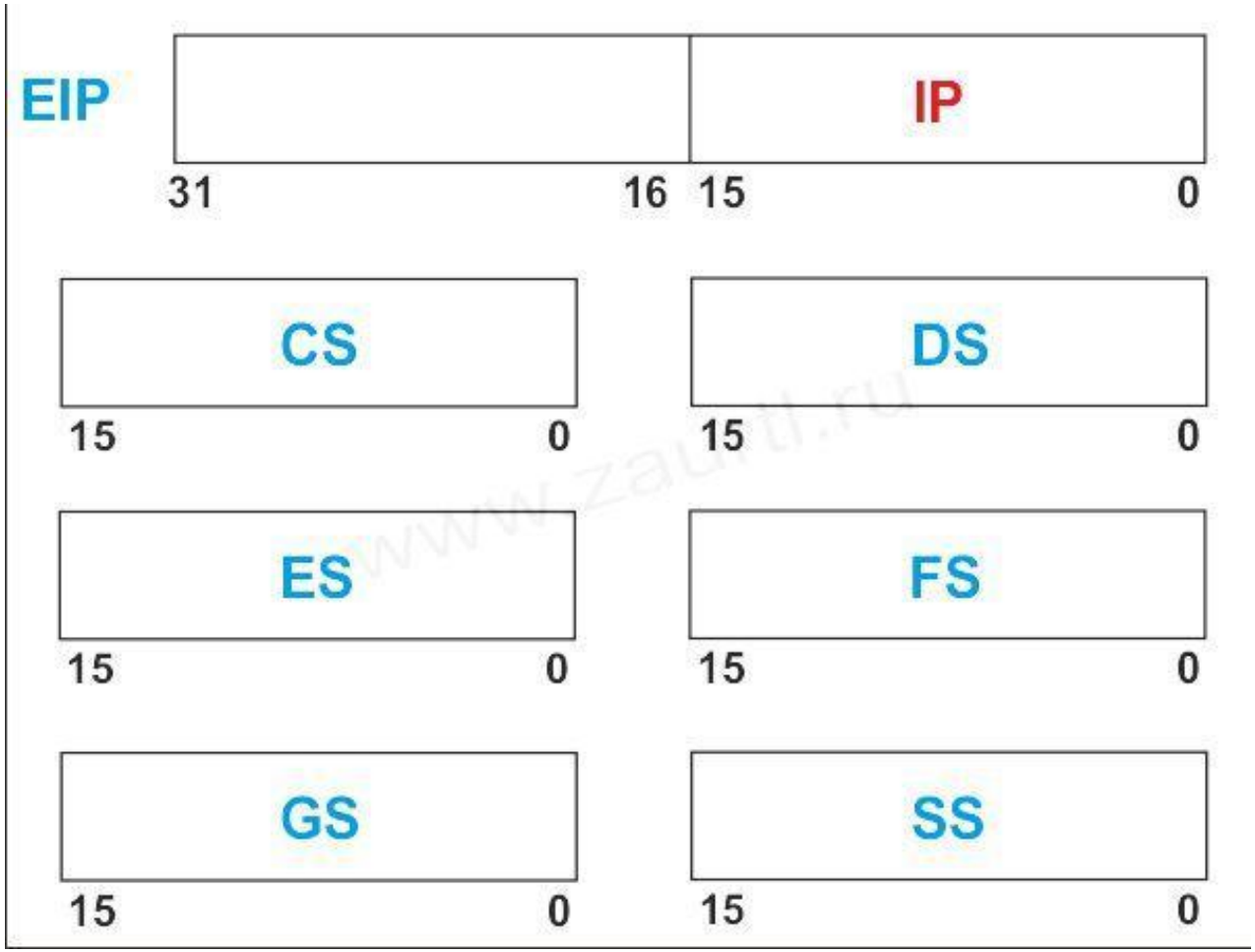
- Чтобы понять работу микропроцессора необходимо знать его организацию с программной стороны, т.е. **математическую модель**. Рассмотрим организацию процессора на примере Intel 8086. За время развития, процессоры Intel испытали ряд изменений структуры, и для обеспечения их программной совместимости были созданы различные режимы работы микропроцессора. Наиболее классическим остается режим реальной адресации памяти или просто реальный режим x86 соответствующий процессору I8086. Эквивалентом реального режима является виртуальный режим – поддержка реального режима в защищенном режиме. Поэтому, **рассмотрим параллельно математическую модель процессора** в режимах x86 и V86.
  Начиная с 32 разрядных, в процессорах Intel содержится **16 основных регистров для пользовательских программ** и еще **11 регистров для работы с числами с плавающей запятой и мультимедийными приложениями.** Помимо них доступны также **регистры управления памятью, регистры управления, отладочные регистры и машинно-специфичные регистры.**

Организация регистров общего назначения показана на следующем рисунке



- **Регистры общего назначения.** 32 битные регистры EAX (аккумулятор), EBX (база), ECX (счетчик), EDX (регистр данных) **могут использоваться без ограничений для любых целей – временного хранения данных, аргументов или результатов различных операций**. Названия этих регистров исходят от того, что некоторые команды используют их специальным образом: так, аккумулятор часто используется для хранения результата действий, выполняемых над двумя операндами, регистр данных в этих случаях получает старшую часть результата, если он не умещается в аккумулятор, регистр-счетчик используется как счетчик в циклах и строковых операциях, а регистр-база при так называемой адресации по базе. Младшие 16 бит каждого из регистров могут использоваться как самостоятельные регистры и имеют имена (AX, BX, CX, DX). Базовый процессор Intel 8086 был 16 разрядным и внутренние регистры были 16 битными (AX, BX, CX, DX) . С появлением 32 разрядных процессоров регистры расширились до 32 бит (EAX, EBX, ECX, EDX). Кроме этого, к частям 16 битных регистров можно обращаться по частям. Например, старшие байты (AH, BH, CH, DH) и младшие байты (AL, BL, CL, DL).
- Другие **четыре регистра общего назначения** – ESI (индекс источника), EDI (индекс приемника), EBP (указатель базы), ESP (указатель стека) – **имеют конкретное назначение.** Если они не используются по назначению, то могут применяться для хранения всевозможных переменных. Регистры ESI и EDI используются в строковых операциях, EBP и ESP используются при работе со стеком. К младшим частям рассмотренных регистров можно обращаться как SI, DI, BP, SP.

- Сегментные регистры и регистр управления. При использовании сегментированной модели памяти, для формирования любого адреса применяется два числа – адрес начала сегмента (сегментный адрес) и смещение (смещение искомого байта относительно начала сегмента). Расположение сегментов в оперативной памяти определяется операционной системой автоматически. В процессорах Intel предусмотрено шесть 16 битных регистров – DS, ES, GS, FS (регистры сегментов данных) и CS, SS (регистры сегмента кода и сегмента стека). Смещение следующей выполняемой команды всегда хранится в специальном регистре EIP (указатель инструкций). Шестнадцатибитная форма этого регистра IP. Все команды передачи управления (условный и безусловный переход, цикл, вызов подпрограммы и т.д.) выполняются с использованием регистров CS и EIP.
- Стек - это специальным образом организованный участок оперативной памяти, используемый для временного хранения переменных, для передачи параметров вызываемым подпрограммам и для хранения адреса возврата при вызове процедур и прерываний. При записи данных в стек, они копятся в виде стопки. Последняя информация всегда остается сверху. Для доступа к нижним частям записанной информации необходимо забрать все сверху расположенные данные. Стек располагается в сегменте памяти, указанном в SS, а текущее смещение вершины стека обозначает содержимое (ESP или SP).


EIP
IP
31
16
15
0
CS
DS
ES
FS
GS
SS
15
0

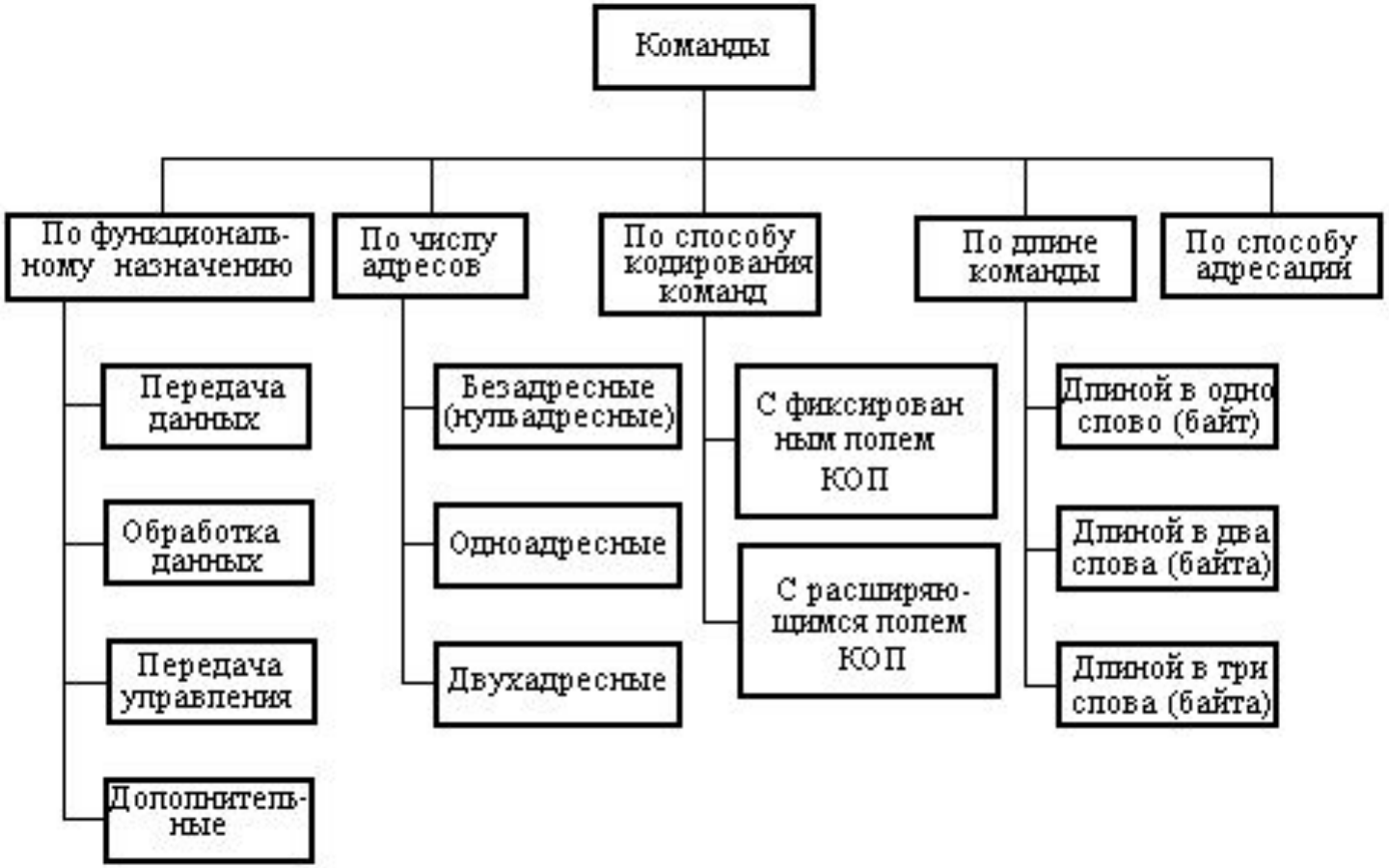
- **Регистр флагов** – **используется для хранения всех признаков после выполнения команд.** 32 битная форма регистра флагов – EFLAGS, 16 аналог – FLAGS. В этом регистре каждый бит является флагом, т.е. устанавливается 1 при определенных условиях или наличие 1 в определенной битовой позиции может повлиять на поведение процессора. Например регистр FLAGS фиксирует следующие признаки:

CF – флаг переноса.
PF – флаг четности.
AF – флаг полупереноса.
ZF – флаг нуля.
SF – флаг знака.
TF – флаг ловушки.
IF – флаг прерываний.
DF – флаг направления.
OF – флаг переполнения.
IOPL, NT – применяются в защищенном режиме.

Мы рассмотрели только **модель реального режима**. **В защищенном режиме работы микропроцессора используется дополнительная группа регистров**, каждый из которых имеет определенное назначение.

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

- **Проектирование системы команд оказывает влияние на структуру ЭВМ.** Оптимальную систему команд иногда определяют как совокупность команд, которая удовлетворяет требованиям проблемно-ориентированных применений таким образом, что избыточность аппаратных и аппаратно-программных средств на реализацию редко используемых команд оказывается минимальной. В различных программах ЭВМ частота появления команд различна; например, по данным фирмы DEC в программах для ЭВМ семейства PDP-11 наиболее часто встречается команда передачи MOV(B), на ее долю приходится приблизительно 32% всех команд в типичных программах. Систему команд следует выбирать таким образом, чтобы затраты на редко используемые команды были минимальными.
- При наличии статистических данных можно разработать (выбрать) ЭВМ с эффективной системой команд. Одним из подходов к достижению данной цели является разработка команд длиной в одно слово и кодирование их таким образом, чтобы разряды таких коротких команд использовать оптимально, что позволит сократить время реализации программы и ее длину.
- Другим подходом к оптимизации системы команд является использование микроинструкций. В этом случае отдельные биты или группы бит команды используются для кодирования нескольких элементарных операций, которые выполняются в одном командном цикле. Эти элементарные операции не требуют обращения к памяти, а последовательность их реализации определяется аппаратной логикой.
- Сокращение времени выполнения программ и емкости памяти достигается за счет увеличения сложности логики управления.
- Важной характеристикой команды является ее формат, определяющий структурные элементы команды, каждый из которых интерпретируется определенные образом при ее выполнении. Среди таких элементов (полей) команды выделяют следующие: код операции, определяющий выполняемое действие; адрес ячейки памяти, регистра процессора, внешнего устройства; режим адресации; операнд при использовании непосредственной адресации; код анализируемых признаков для команд условного перехода.
- Классификация команд по основным признакам представлена на рисунке. Важнейшим структурным элементом формата любой команды является код операции (КОП), определяющей действие, которое должно быть выполнено. Большое число КОП в процессоре очень важно, так как аппаратная реализация команд экономит память и время. Но при выборе ЭВМ необходимо концентрировать внимание на полноте операций с конкретными типами данных, а не только на числе команд, на доступных режимах адресации. Число бит, отводимое под КОП, является функцией полного набора реализуемых команд.



Классификация команд

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

- При использовании фиксированного числа бит под КОП для кодирования всех m команд необходимо в поле КОП выделить двоичных разрядов. Однако, учитывая ограниченную длину слова мини- и микроЭВМ, различное функциональное назначение команд, источники и приемники результатов операций, а также то, что не все команды содержат адресную часть для обращения к памяти и периферийным устройствам, в малых ЭВМ для кодирования команд широко используется принцип кодирования с переменным числом бит под поле КОП для различных групп команд.
- В некоторых командах необходим только один операнд и они называются однооперандными (или одноадресными) командами в отличие от двухоперандных (или двухадресных), в которых требуются два операнда. При наличии двух операндов командой обычно изменяется только один из них. Так как информация берется только из одной ячейки, эту ячейку называются источником; ячейка, содержимое которой изменяется, называется приемником.
- Ниже приведен формат двухадресной (двухоперандной) команды процессоров СМ.

а	15 11	10 6	5 0	б	15 11	10 0
	КОП	Источник	Приемник		КОП	Приемник

Формат команд процессоров СМ:
а) двухадресная команда;
б) одноадресная команда.

Примеры кодирования двухадресных команд в процессорах СМ

КОП	Мнемоника команды	Комментарий
0001 0010 0110 1110	MOV CMP ADD SUB	Передача данных Сравнение Сложение Вычитание
0000 1000	- -	Кодирование группы одноадресных команд

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

- Четырехбитный КОП (биты 15-12) кодирует ряд двухоперандных операций, приведенных в таблице 1. Биты (11-6) и (5-0) для команд данного типа определяют адреса источника и приемника данных. Как видно из таблицы, комбинации 0000 и 1000 поля КОП определяют группы одноадресных команд (рис 1,б). КОП 1 (биты 15-12), соответствующий кодам 0000 и 1000, определяет группу одноадресных команд, а КОП 2 (биты 11-6) кодирует конкретную операцию команд данной группы. Таким образом, команды, использующие один операнд, кодируются 10-битным КОП (биты 15-6).
- Наиболее гибкая команда требует до четырех операндов. Например, команда сложения может указывать адреса слагаемых, адрес результата и адрес следующей команды. Если для задания адреса требуется 16 бит, то четырехоперандная команда займет 8 байт памяти, не учитывая код операции. Следовательно, получится медленнодействующая ЭВМ с огромной памятью. Поэтому в большинстве микроЭВМ любой команде требуется не более двух операндов. Это достигается следующими приемами:
- 1. Адрес следующей команды указывается только в командах переходов; в остальных случаях очередная команда выбирается из ячеек памяти, следующих за выполненной командой.
- 2. Использование ячейки, в которой находится один из операндов, для запоминания результата (например, сумма запоминается в ячейки первого операнда).
- Локализацию и обращение к операндам обеспечивают режимы адресации. При введении нескольких режимов адресации необходимо отвести в команде биты, указывающие режимы адресации для каждого операнда. Если предусмотрено восемь режимов адресации, то для задания каждого из них нужно три бита.
- Почти во всех форматах команд первые биты отводятся для кода операции, но далее форматы команд разных ЭВМ сильно отличаются друг от друга. Остальные биты должны определять операнды или их адреса, и поэтому они используются для комбинации режимов, адресов регистров, адресов памяти, относительных адресов и непосредственных операндов. Обычно длина команды варьируется от 1 до 3 и даже 6 байт.
- **По форматам команд можно судить о возможностях ЭВМ.**

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

Формат и разновидность команд микропроцессора

- Все выполняемые микропроцессором операции кодированы в виде двоичных чисел. В современных процессорах CISC (персональные компьютеры) **число команд (инструкций) достигает несколько сотен**. В общем случае команда имеет следующий **формат:**

Префикс – уточняет или модифицирует действие команды.

Код операции – определяет действие данной команды.

Режим адресации – определяет используемую форму адреса операнда.

Масштаб-индекс-база – расширяет возможности адресации операндов.

Смещение – значение эффективного адреса операнда.

Операнды – указывают данные, над которыми нужно выполнить действие и место, куда надо поместить результат.

Из рисунка видно, что размер команды может меняться от 1 байта до 16 байт.

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

- **Способ указания операндов называется способом адресации**. От разнообразия способов адресации зависит скорость исполнения команд и эффективность программы. В современных процессорах ПК используется несколько десятков способов адресации. Самыми основными из них можно выделить следующие варианты адресации:

- непосредственная адресация, операнд указывается непосредственно в команде в виде числа;
- прямая адресация, операнд указывается в команде регистром или адресом ячейки памяти;
- косвенная адресация, на операнд указывает число в регистре, обозначенном в команде;
- неявная адресация, операнд в команде не указывается, но подразумевается командой по умолчанию.

- **Для удобства программирования** используется язык программирования ассемблер (язык низкого уровня). В этом языке командам сопоставляются некие сокращения слов, значащие действие команды. Программа на ассемблере представляет собой текстовый файл команд, оформленный по определенным правилам. Специальная программа транслятор-компилятор переводит текст с ассемблера в машинные коды, и готовить программу для процессора в виде исполняемого файла. Откомпилированные двоично-кодированные программы практически человеком не читаемы. Но используя программы отладчики можно сделать эти файлы читаемыми и просматривать действия этой программы по шагам.

- **Рассмотрим основные команды микропроцессоров, на примере некоторых базовых команд микропроцессоров ПК (Intel8086) используя язык ассемблер.**

Команды пересылки данных:

MOV <приемник>,<источник> - пересылка данных. Копирование содержимого источника в приемник;

PUSH <источник> - запись в стек. Занесение слова в стек;

POP <приемник> - чтение из стека. Чтение слова из стека.

Стек – это область оперативной памяти организованная по принципу «первым вошел - последним вышел». В команде работы со стеком адрес источника или приемника не указывается, он выбирается автоматически и называется верхушкой стековой памяти. Стековые команды характеризуются быстрой записью и считыванием информации в область оперативной памяти.

Арифметические команды:

ADD <приемник>,<источник> - сложение. Сложение источника с приемником и запись результата в приемник;

SUB <приемник>,<источник> - вычитание. Вычитает из приемника значение источника и помещает результат в приемник;

CMP <приемник>,<источник> - сравнение. Выполняется сравнение вычитанием источника из приемника. Признаки операции формируются, но значения источника и приемника остаются неизменными.

MUL <источник> - умножение без знака. Если источник равен одному байту то источник умножается на содержимое регистра AL и результат помещается в AX. Если же источник равен машинному слову (два байта), то значение источника умножается на содержимое регистра AX и результат помещается в регистровую пару DX:AX;

DIV <источник> - деление без знака. Если источник – байт, то делимое в AX, делитель в <источник>, целая часть деления в AL, остаток в AH. В случае если источник равен двум байтам, в качестве делителя берется содержимое пары регистров DX:AX, результат от деления помещается в AX, остаток в DX;

INC <приемник> - инкрементирование. К содержимому приемника прибавляется единица;

DEC <приемник> - декрементирование. Содержимое приемника уменьшается на единицу.

Логические команды:

OR <приемник>,<источник> - логическое сложение. Выполняет поразрядную дизъюнкцию источника и приемника. Результат помещается в приемник;

AND <приемник>,<источник> - логическое умножение. Выполняет поразрядную конъюнкцию источника и приемника. Результат помещается в приемник;

NOT <приемник> - логическое отрицание. Все биты приемника инвертируются.

Команды условной и безусловной передачи управления:

JMP <адрес> - безусловный переход. Выполняет безусловный переход по указанному адресу. Адресом может быть определенная метка в программе или регистр, косвенно указывающий адрес;

J* <метка> - общий формат условной передачи управления. Имеется довольно много условий перехода. Наиболее часто используемые:

JG – переход, если больше;

JNG – переход, если не больше;

JL – переход, если меньше;

JNL – переход если не меньше;

JE – переход если равно;

JZ – переход если ноль.

Команды управления циклами:

LOOP <метка> - повторять до обнуления содержимого счетчика CX. Перед использованием команды LOOP необходимо в CX занести число повторений. Команда LOOP уменьшает значение регистра CX на единицу и осуществляет переход на метку, если CX не равно нулю.

Команды работы с подпрограммами:

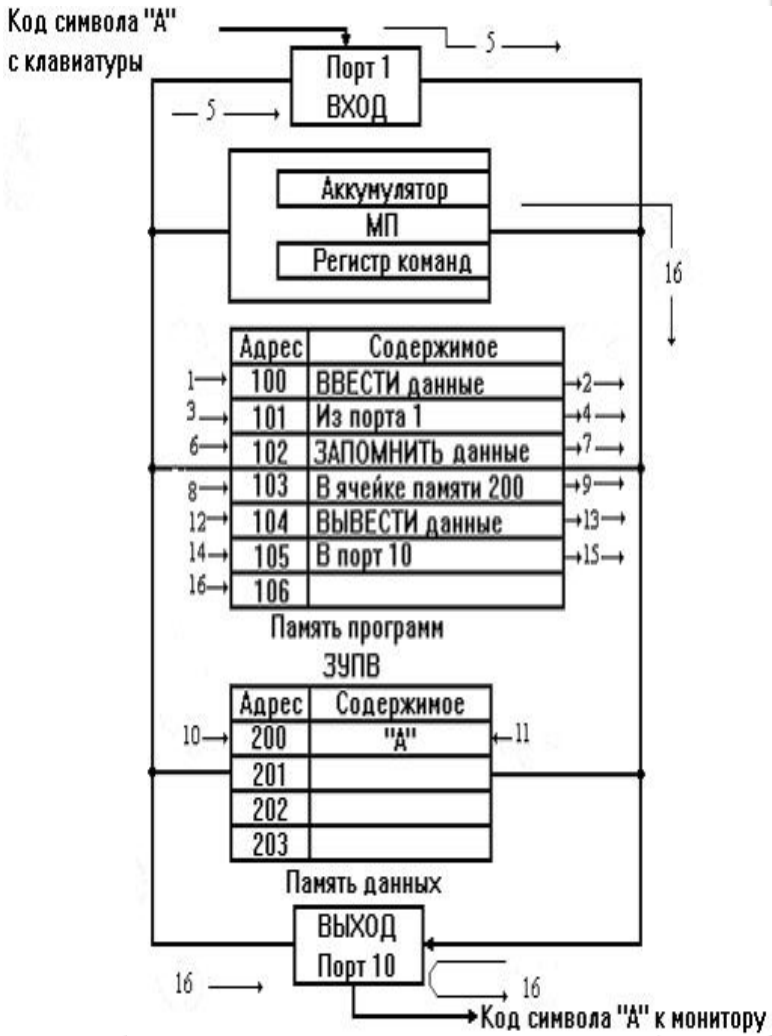
CALL <адрес> - вызов подпрограммы. Безусловная передача управления подпрограмме. Адресом может быть определенная метка в программе или регистр, косвенно указывающий адрес;

RET – возврат из подпрограммы.

Занятие 15. 1) Классификация команд. Системы команд и классы процессоров: CISC, RISC, MISC, VLIM.

- Кроме рассмотренных нами операций, имеются:
  - команды математического сопроцессора;
  - команды обращения к внешним устройствам;
  - команды работы со строковыми переменными;
  - команды обслуживания и вспомогательные команды.

  В 32 разрядных процессорах к базовым командам добавляются 32 разрядные команды, а к 64 разрядным процессорам еще добавляются команды работы с 64 битными данными. Из-за ограниченности программы, на уровне данной дисциплины остальные команды рассматривать не будем.



Ввод символа А с клавиатуры

- В качестве примера, иллюстрирующего работу микроЭВМ, рассмотрим процедуру, для реализации которой нужно выполнить следующую последовательность элементарных операций:

1. Нажать клавишу с буквой "А" на клавиатуре.
2. Поместить букву "А" в память микроЭВМ.
3. Вывести букву "А" на экран дисплея.

- Это типичная процедура ввода-запоминания-вывода, рассмотрение которой дает возможность пояснить принципы использования некоторых устройств, входящих в микроЭВМ.
- На рисунке приведена подробная диаграмма выполнения процедуры ввода-запоминания-вывода. Обратите внимание, что команды уже загружены в первые шесть ячеек памяти. Хранимая программа содержит следующую цепочку команд:

1. Ввести данные из порта ввода 1.
2. Запомнить данные в ячейке памяти 200.
3. Переслать данные в порт вывода 10.

- В данной программе всего три команды, хотя на рисунке может показаться, что в памяти программ записано шесть команд. Это связано с тем, что команда обычно разбивается на части. Первая часть команды 1 в приведенной выше программе - команда ввода данных. Во второй части команды 1 указывается, откуда нужно ввести данные (из порта 1). Первая часть команды, предписывающая конкретное действие, называется кодом операции (КОП), а вторая часть - операндом. Код операции и операнд размещаются в отдельных ячейках памяти программ. На рисунке КОП хранится в ячейке 100, а код операнда - в ячейке 101 (порт 1); последний указывает откуда нужно взять информацию.
- В МП на рисунке выделены еще два новых блока - регистры: аккумулятор и регистр команд.

- Рассмотрим прохождение команд и данных внутри микроЭВМ с помощью занумерованных кружков на диаграмме. Напомним, что **микропроцессор - это центральный узел, управляющий перемещением всех данных и выполнением операций**.

- Итак, при выполнении типичной процедуры ввода-запоминания-вывода в микроЭВМ происходит следующая последовательность действий:
  1. МП выдает адрес 100 на шину адреса. По шине управления поступает сигнал, устанавливающий память программ (конкретную микросхему) в режим считывания.
  2. ЗУ программ пересылает первую команду ("Ввести данные") по шине данных, и МП получает это закодированное сообщение. Команда помещается в регистр команд. МП декодирует (интерпретирует) полученную команду и определяет, что для команды нужен операнд.
  3. МП выдает адрес 101 на ША; ШУ используется для перевода памяти программ в режим считывания.
  4. Из памяти программ на ШД пересылается операнд "Из порта 1". Этот операнд находится в программной памяти в ячейке 101. Код операнда (содержащий адрес порта 1) передается по ШД к МП и направляется в регистр команд. МП теперь декодирует полную команду ("Ввести данные из порта 1").
  5. МП, используя ША и ШУ, связывающие его с устройством ввода, открывает порт 1. Цифровой код буквы "А" передается в аккумулятор внутри МП и запоминается.Важно отметить, что при обработке каждой программной команды МП действует согласно микропроцедуре выборки-декодирования-исполнения.
  6. МП обращается к ячейке 102 по ША. ШУ используется для перевода памяти программ в режим считывания.
  7. Код команды "Запомнить данные" подается на ШД и пересылается в МП, где помещается в регистр команд.
  8. МП дешифрирует эту команду и определяет, что для нее нужен операнд. МП обращается к ячейке памяти 103 и приводит в активное состояние вход считывания микросхем памяти программ.
  9. Из памяти программ на ШД пересылается код сообщения "В ячейке памяти 200". МП воспринимает этот операнд и помещает его в регистр команд. Полная команда "Запомнить данные в ячейке памяти 200" выбрана из памяти программ и декодирована.
  10. Теперь начинается процесс выполнения команды. МП пересылает адрес 200 на ША и активизирует вход записи, относящийся к памяти данных.
  11. МП направляет хранящуюся в аккумуляторе информацию в память данных. Код буквы "А" передается по ШД и записывается в ячейку 200 этой памяти. Выполнена вторая команда. Процесс запоминания не разрушает содержимого аккумулятора. В нем по-прежнему находится код буквы "А".
  12. МП обращается к ячейке памяти 104 для выбора очередной команды и переводит память программ в режим считывания.
  13. Код команды вывода данных пересылается по ШД к МП, который помещает ее в регистр команд, дешифрирует и определяет, что нужен операнд.
  14. МП выдает адрес 105 на ША и устанавливает память программ в режим считывания.
  15. Из памяти программ по ШД к МП поступает код операнда "В порт 10", который далее помещается в регистр команд.
  16. МП дешифрирует полную команду "Вывести данные в порт 10". С помощью ША и ШУ, связывающих его с устройством вывода, МП открывает порт 10, пересылает код буквы "А" (все еще находящийся в аккумуляторе) по ШД. Буква "А" выводится через порт 10 на экран дисплея.

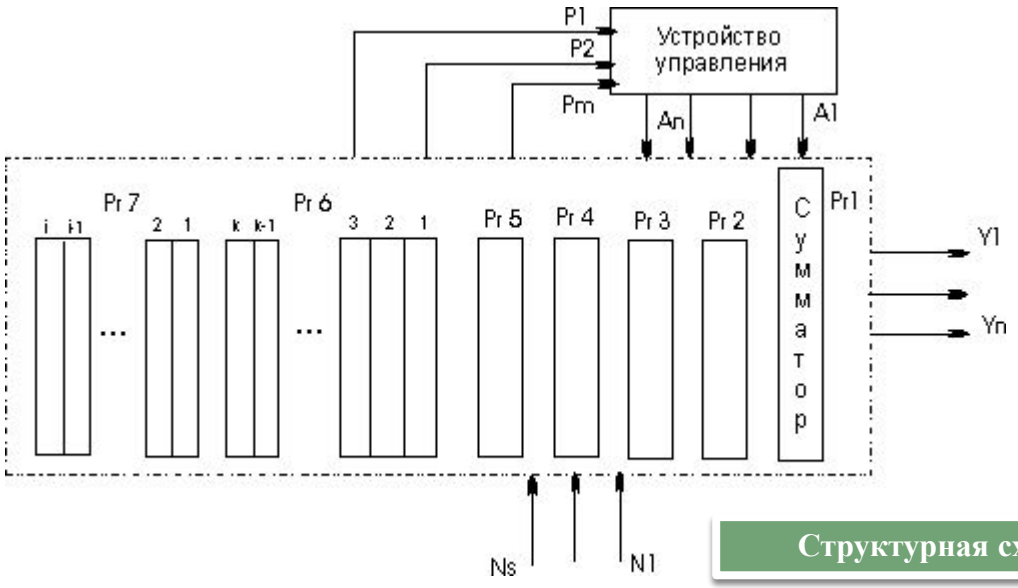
- **В большинстве микропроцессорных систем (МПС) передача информации осуществляется способом, аналогичным рассмотренному выше. Наиболее существенные различия возможны в блоках ввода и вывода информации.**

- **Подчеркнем еще раз, что именно микропроцессор является ядром системы и осуществляет управление всеми операциями. Его работа представляет последовательную реализацию микропроцедур выборки-дешифрации-исполнения. Однако фактическая последовательность операций в МПС определяется командами, записанными в памяти программ**.

Занятие 15. 2) Арифметико-логическое устройство (АЛУ): назначение и классификация. Структура и функционирование АЛУ.

- **Арифметико-логическое устройство (АЛУ)** - центральная часть процессора, выполняющая арифметические и логические операции.
- **АЛУ реализует важную часть процесса обработки данных.** Она заключается в выполнении набора простых операций. Операции АЛУ подразделяются на три основные категории: арифметические, логические и операции над битами. Арифметической операцией называют процедуру обработки данных, аргументы и результат которой являются числами (сложение, вычитание, умножение, деление,...). Логической операцией именуют процедуру, осуществляющую построение сложного высказывания (операции И, ИЛИ, НЕ,...). Операции над битами обычно подразумевают сдвиги.
- Разработчик компьютера ENIAC, Джон фон Нейман, был первым создателем АЛУ. В 1945 году он опубликовал первые научные работы по новому компьютеру, названному EDVAC (Electronic Discrete Variable Computer). Годом позже он работал со своими коллегами над разработкой компьютера для Принстонского института новейших исследований (IAS). Архитектура этого компьютера позже стала прототипом архитектур большинства последующих компьютеров. В своих работах фон Нейман указывал устройства, которые, как он считал, должны присутствовать в компьютерах. Среди этих устройств присутствовало и АЛУ. Фон Нейман отмечал, что АЛУ необходимо для компьютера, поскольку оно гарантирует, что компьютер будет способен выполнять базовые математические операции включая сложение, вычитание, умножение и деление.
- **АЛУ состоит из регистров, сумматора с соответствующими логическими схемами и элемента управления выполняемым процессом. Устройство работает в соответствии с сообщаемыми ему именами (кодами) операций, которые при пересылке данных нужно выполнить над переменными, помещаемыми в регистры.**
- Арифметико-логическое устройство функционально можно разделить на две части :
  1. микропрограммное устройство (устройство управления), задающее последовательность микрокоманд (команд);
  2. операционное устройство (АЛУ), в котором реализуется заданная последовательность микрокоманд (команд).

Занятие 15. 2) Арифметико-логическое устройство (АЛУ): назначение и классификация. Структура и функционирование АЛУ.



Структурная схема АЛУ

- **Структурная схема АЛУ** и его связь с другими блоками машины показаны на рисунке. В состав АЛУ входят регистры Рг1 - Рг7, в которых обрабатывается информация , поступающая из оперативной или пассивной памяти N1, N2, ...NS; логические схемы, реализующие обработку слов по микрокомандам, поступающим из устройства управления.
- Закон переработки информации задает микропрограмма , которая записывается в виде последовательности микрокоманд A1,A2, ..., An-1,An. При этом различают два вида микрокоманд: внешние, то есть такие микрокоманды, которые поступают в АЛУ от внешних источников и вызывают в нем те или иные преобразования информации (на рис. 1 микрокоманды A1,A2,..., An), и внутренние, которые генерируются в АЛУ и воздействуют на микропрограммное устройство, изменяя естественный порядок следования микрокоманд. Например, АЛУ может генерировать признаки в зависимости от результата вычислений: признак переполнения, признак отрицательного числа, признак равенства 0 всех разрядов числа др. На рис. 1 эти микрокоманды обозначены p1, p2,..., pm.

Занятие 15. 2) Арифметико-логическое устройство (АЛУ): назначение и классификация. Структура и функционирование АЛУ.

- Результаты вычислений из АЛУ передаются по кодовым шинам записи y1, y2, ...,ys, в ОЗУ. Функции регистров, входящих в АЛУ:
- Рг1 - сумматор (или сумматоры) - основной регистр АЛУ, в котором образуется результат вычислений;
- Рг2, Рг3 - регистры слагаемых, сомножителей, делимого или делителя (в зависимости от выполняемой операции);
- Рг4 - адресный регистр (или адресные регистры), предназначен для запоминания (иногда и формирования) адреса операндов и результата;
- Ргб - k индексных регистров, содержимое которых используется для формирования адресов;
- Рг7 - i вспомогательных регистров, которые по желанию программиста могут быть аккумуляторами, индексными регистрами или использоваться для запоминания промежуточных результатов.
- Часть операционных регистров является программно-доступной, то есть они могут быть адресованы в команде для выполнения операций с их содержимым. К ним относятся : сумматор, индексные регистры, некоторые вспомогательные регистры.
- Остальные регистры программно-недоступные, так как они не могут быть адресованы в программе. Операционные устройства можно классифицировать по виду обрабатываемой информации, по способу обработки информации и логической структуре.
- АЛУ может оперировать четырьмя типами информационных объектов: булевскими (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В АЛУ выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования "операция/ режим адресации" базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

Занятие 15. 2) Арифметико-логическое устройство (АЛУ): назначение и классификация. Структура и функционирование АЛУ.

Классификация АЛУ

- *По способу действия над операндами* АЛУ делятся на последовательные и параллельные. В последовательных АЛУ операнды представляются в последовательном коде, а операции производятся последовательно во времени над их отдельными разрядами. В параллельных АЛУ операнды представляются параллельным кодом и операции совершаются параллельно во времени над всеми разрядами операндов.
- *По способу представления чисел* различают АЛУ:
  - для чисел с фиксированной точкой;
  - для чисел с плавающей точкой;
  - для десятичных чисел.
- *По характеру использования элементов и узлов* АЛУ делятся на блочные и многофункциональные. В блочном АЛУ операции над числами с фиксированной и плавающей точкой, десятичными числами и алфавитно-цифровыми полями выполняются в отдельных блоках, при этом повышается скорость работы, так как блоки могут параллельно выполнять соответствующие операции, но значительно возрастают затраты оборудования. В многофункциональных АЛУ операции для всех форм представления чисел выполняются одними и теми же схемами, которые коммутируются нужным образом в зависимости от требуемого режима работы.
- *По своим функциям* АЛУ является операционным блоком, выполняющим микрооперации, обеспечивающие приём из других устройств (например, памяти) операндов, их преобразование и выдачу результатов преобразования в другие устройства. Арифметическо-логическое устройство управляется управляющим блоком, генерирующим управляющие сигналы, инициирующие выполнение в АЛУ определённых микроопераций. Генерируемая управляющим блоком последовательность сигналов определяется кодом операции команды и оповещающими сигналами.

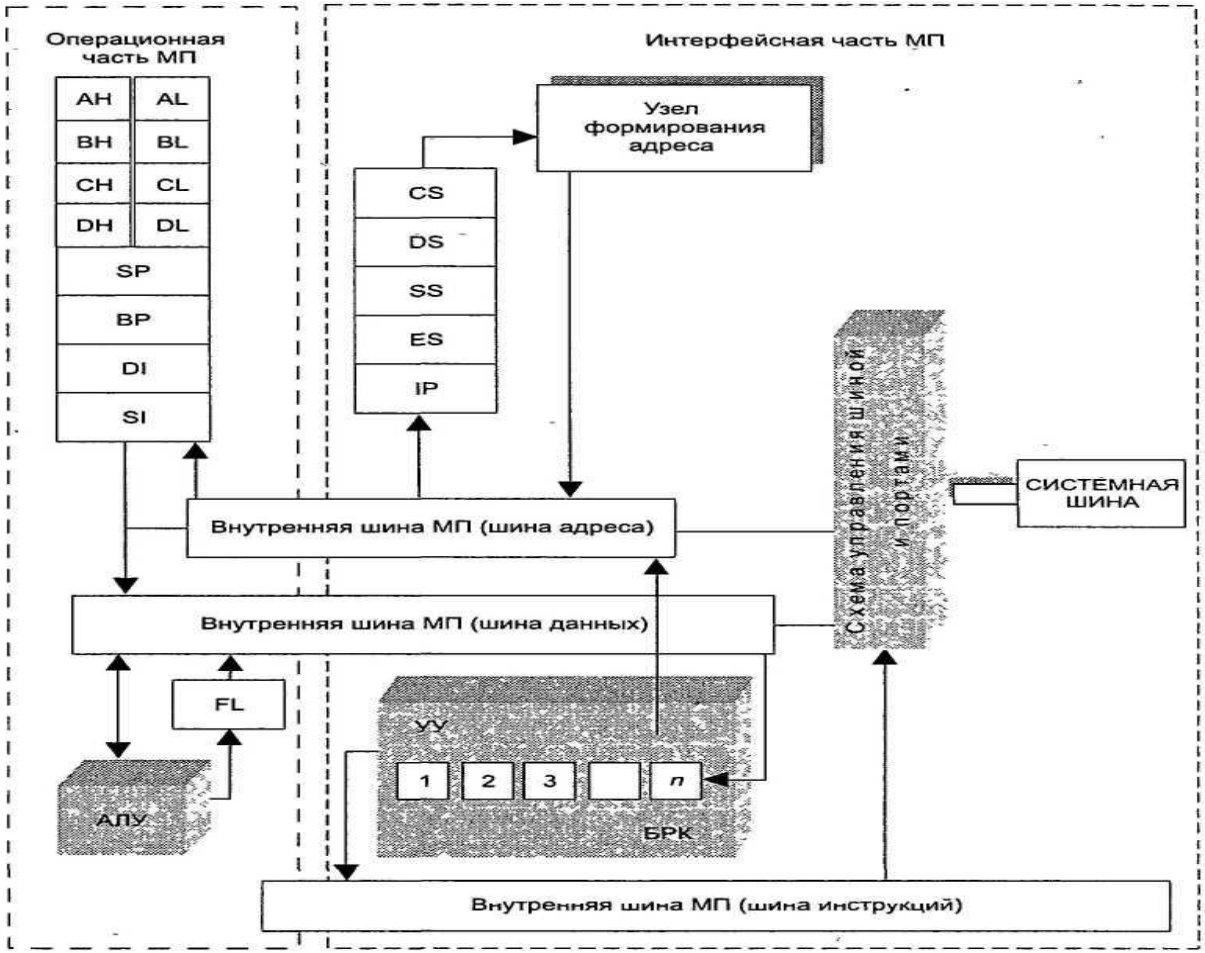
Операции в АЛУ

- Выполняемые в АЛУ операции можно разделить на следующие группы:
- операции двоичной арифметики для чисел с фиксированной точкой;
- операции двоичной (или шестнадцатеричной) арифметики для чисел с плавающей точкой;
- операции десятичной арифметики;
- операции индексной арифметики (при модификации адресов команд);
- операции специальной арифметики;
- операции над логическими кодами (логические операции);
- операции над алфавитно-цифровыми полями.
- Современные ЭВМ общего назначения обычно реализуют операции всех приведённых выше групп, а малые и микроЭВМ, микропроцессоры и специализированные ЭВМ часто не имеют аппаратуры арифметики чисел с плавающей точкой, десятичной арифметики и операций над алфавитно-цифровыми полями. В этом случае эти операции выполняются специальными подпрограммами. К арифметическим операциям относятся сложение, вычитание, вычитание модулей («короткие операции») и умножение и деление («длинные операции»). Группу логических операций составляют операции дизъюнкция (логическое ИЛИ) и конъюнкция (логическое И) над многоразрядными двоичными словами, сравнение кодов на равенство. Специальные арифметические операции включают в себя нормализацию, арифметический сдвиг (сдвигаются только цифровые разряды, знаковый разряд остаётся на месте), логический сдвиг (знаковый разряд сдвигается вместе с цифровыми разрядами). Обширна группа операций редактирования алфавитно-цифровой информации

Занятие 15. 3) Интерфейсная часть процессора: назначение, состав, функционирование. Организация работы и функционирование.

- **Интерфейсная часть МП** предназначена для связи и согласования МП с системной шиной ПК, а также для приема, предварительного анализа команд выполняемой программы и формирования полных адресов операндов и команд.
- **Интерфейсная часть включает в свой состав** адресные регистры МПП, узел формирования адреса, блок регистров команд, являющийся буфером команд в МП, внутреннюю интерфейсную шину МП и схемы управления шиной и портами ввода-вывода.
- ***Порты ввода-вывода*** - это пункты системного интерфейса ПК, через которые МП обменивается информацией с другими устройствами. Всего портов у МП может быть 65536. Каждый порт имеет адрес - номер порта, соответствующий адресу ячейки памяти, являющейся частью устройства ввода-вывода, использующего этот порт, а не частью основной памяти компьютера.
- **Порт устройства содержит аппаратуру сопряжения и два регистра памяти** - для обмена данными и обмена управляющей информацией. Некоторые внешние устройства используют и основную память для хранения больших объемов информации, подлежащей обмену. Многие стандартные устройства (НЖМД, НГМД, клавиатура, принтер, сопроцессор и др.) имеют постоянно закрепленные за ними порты ввода-вывода.
- ***Схема управления шиной и портами*** выполняет следующие функции:
  1. формирование адреса порта и управляющей информации для него (переключение порта на прием или передачу и др.);
  2. прием управляющей информации от порта, информации о готовности порта и его состоянии;
  3. организацию сквозного канала в системном интерфейсе для передачи данных между портом устройства ввода-вывода и МП.
- **Схема управления шиной и портами** использует для связи с портами кодовые шины инструкций, адреса и данных системной шины: при доступе к порту МП посылает сигнал по КШИ, который оповещает все устройства ввода-вывода, что адрес на КША является адресом порта, а затем посылает и сам адрес порта. То устройство, адрес порта которого совпадает, дает ответ о готовности, после чего по КШД осуществляется обмен данными.

Занятие 15. 3) Интерфейсная часть процессора: назначение, состав, функционирование. Организация работы и функционирование.



Упрощенная структурная схема микропроцессора

Контрольные вопросы

1. **Назначение центрального процессора как устройство управления?**
2. **Почему кремний применяется в микросхемах?**
3. **Назначение и функции микропроцессора.**
4. **Основные параметры микропроцессора.**
5. **Когда был выпущен первый микропроцессор?**
6. **Назовите и охарактеризуйте классы микропроцессоров.**
7. **Приведите упрощенную схему структуры микропроцессора (с объяснением каждого узла схемы).**
8. **Как выполняются команды в скалярном микропроцессоре?**
9. **Назовите режимы работы микропроцессора, для чего они существуют?**
10. **Назначение устройства управления микропроцессора, что оно осуществляет?**
11. **Приведите упрощенную функциональную схему устройства управления микропроцессора (с объяснением каждого узла схемы).**

Контрольные вопросы

12. **Регистры процессора – это...?**
13. **Для чего предназначены регистры общего назначения?**
14. **Что содержат специальные регистры, регистры данных, сегментные регистры?**
15. **Для чего используется регистр флагов?**
16. **Команда – это...?**
17. **Приведите классификацию команд микропроцессора.**
18. **Назначение, состав, классификация и структурная схема арифметико-логического устройства (АЛУ).**
19. **Для чего предназначена интерфейсная часть микропроцессора и из чего она состоит?**