**1. Общая характеристика и классификация информационных систем и инструментальных средств ИС.**

***Информационная система*** ― это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи информации и принятия управленческих решений.

***В информационной системе происходят следующие процессы:***

а) ввод информации из внешних и внутренних источников;

б) обработка входящей информации;

в) хранение информации для последующего ее использования;

г) вывод информации в удобном для пользователя виде;

д) выработка управляющих воздействий, т.е. представление информации, переработанной в данной организации, для корректировки входящей информации.



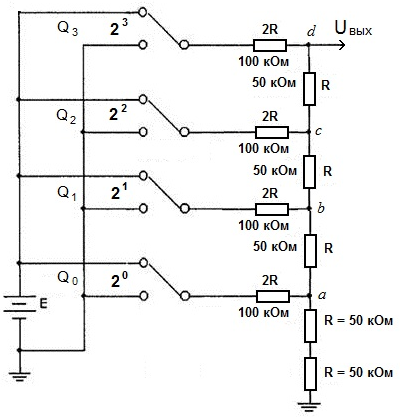
Типовой архитектурой информационной системы является клиент-серверная структура. Основными аппаратными компонентами такой системы являются серверный и клиентский компьютеры, устройства ввода и отображения информации.

**2. Способы и инструментальные средства преобразования цифровых сигналов в аналоговые. ЦАП на основе суммирования токов с весовыми резисторами и резисторными матрицами типа R-2R.**

***Цифро-аналоговый преобразователь*** – это устройство для преобразования цифрового кода в аналоговый сигнал по величине, пропорциональной значению кода. ЦАП применяются для связи цифровых управляющих систем с устройствами, которые управляются уровнем аналогового сигнала. Также ЦАП является составной частью во многих структурах аналого-цифровых устройств и преобразователей.

***Основной характеристикой ЦАП*** является функция (характеристика) преобразования. Она связывает изменение цифрового кода с изменением напряжения или тока. Функция преобразования ЦАП выражается следующим образом .

***ЦАП можно реализовать с помощью*** суммирования токов, суммирования напряжений и деления напряжений. В первом и втором случае в соответствии со значениями разрядов входного кода, суммируются сигналы генераторов токов и источников Э.Д.С. Последний способ представляет собой управляемый кодом делитель напряжения. Два последних способа не нашли широкого распространения в связи с практическими трудностями их реализации.



В конце резистивной цепочки находятся два резистора с сопротивлением 2R. Эти резисторы одним концом соединены друг с другом, другие концы присоединены к корпусу схемы, то есть резисторы соединены параллельно. В результате их общее сопротивление равно R. При соединении резистора R и параллельного соединения двух резисторов 2R образуется делитель напряжения с коэффициентом деления 2. В результате напряжение на его выходе будет в два раза меньше напряжения на его входе.

***Цифро-аналоговым преобразователи на основе R-2R матрицы обладают рядом*** ***преимуществ***, по сравнению с ЦАП с весовыми сопротивлениями. Первое заключается в том, что используются всего два номинала сопротивлений резисторов: R и 2R. Второе – нагрузка источника опорного напряжения (Rвх) не изменяется при любом числе последовательно включенных элементов матрицы и остается равной 2R. Третье – на выходе каждого элемента матрицы получается вдвое меньшее напряжение, чем на его входе.

**3. Способы преобразования аналоговых величин в цифровые. Дискретизация и квантование сигналов.**

***Аналого-цифровое преобразование*** заключается в преобразовании аналогового (непрерывного) процесса (напряжения или тока, сопротивления, емкости, температуры, давления и пр.) в последовательность двоичных чисел ― цифровой код.

***АЦП можно разделить на две группы***: АЦП мгновенных значений напряжения и АЦП средних значений.

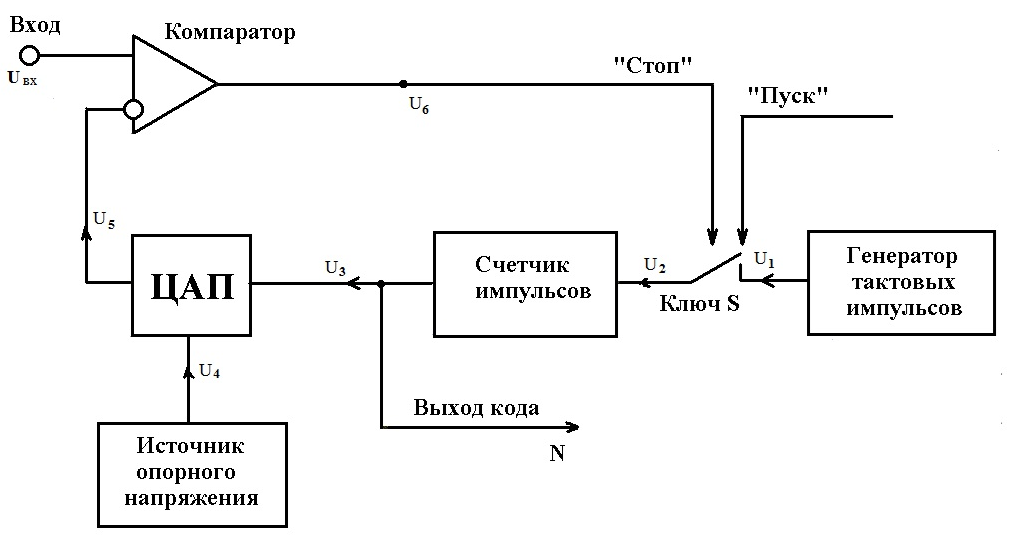
В процессе преобразования напряжения в цифровой код осуществляются ***три независимые операции***: дискретизация по времени; квантование по уровню; кодирование.

В ***процессе дискретизации*** непрерывный сигнал преобразуется в последовательность импульсов, амплитуда которых равна мгновенным значениям сигнала в моменты дискретизации, т.е. осуществляется амплитудно-импульсная модуляция.

Затем ***отсчеты квантуются*** по уровню, которым ставится в соответствие числовой эквивалент, представляющий собой двоичное отображение номера уровня квантования.

**4. АЦП последовательного счета.**

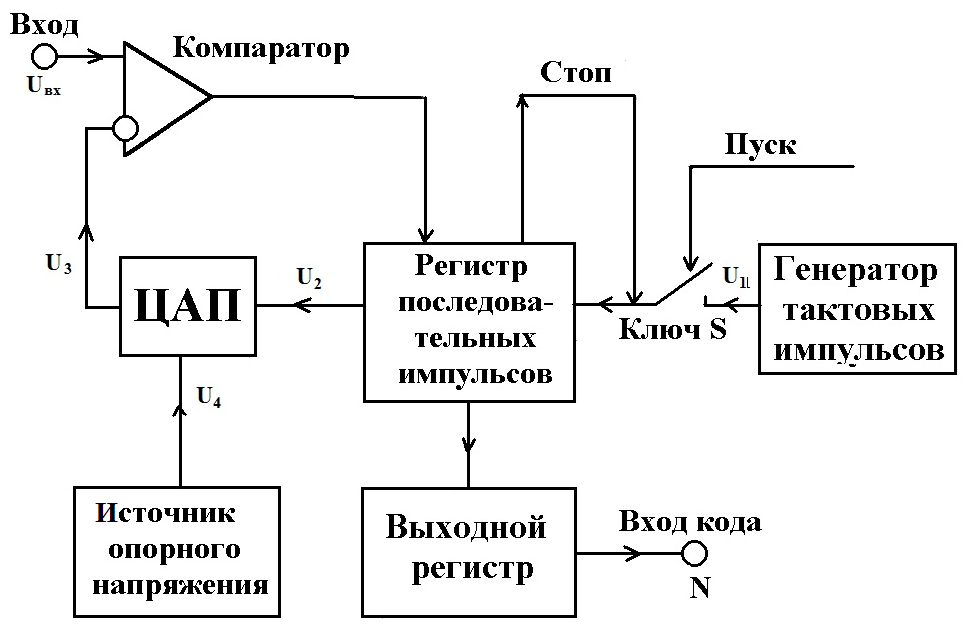
Структурная схема АЦП последовательного счета содержит компара­тор, при помощи которого выполняется сравнение входного на­пряжения с напряжением обратной связи. На прямой вход компа­ратора поступает входной сигнал Uвх, а на инвертирующий ― на­пряжение обратной связи. Работа преобразователя начинается с приходом импульса «ПУСК» от схемы управления, который за­мыкает ключ S. Через замкнутый ключ S импульсы U1 от генера­тора тактовых импульсов поступают на счетчик, выходы которого соединены со входами цифро-аналогового преобразователя. В ре­зультате последовательного увеличения выходного кода счетчика N происходит последовательно-ступенчатое увеличение выходно­го напряжения U5ЦАП.



Когда выходное напряжение ЦАП срав­няется со входным напряжением, произойдет переключение компаратора, и по его выходному сигналу «СТОП» разомкнется ключ S. В результате импульсы от генератора перестанут поступать на вход счетчика. Выходной код, соответствующий равенству Uвх = Us, снимается с выходного регистра счетчика.

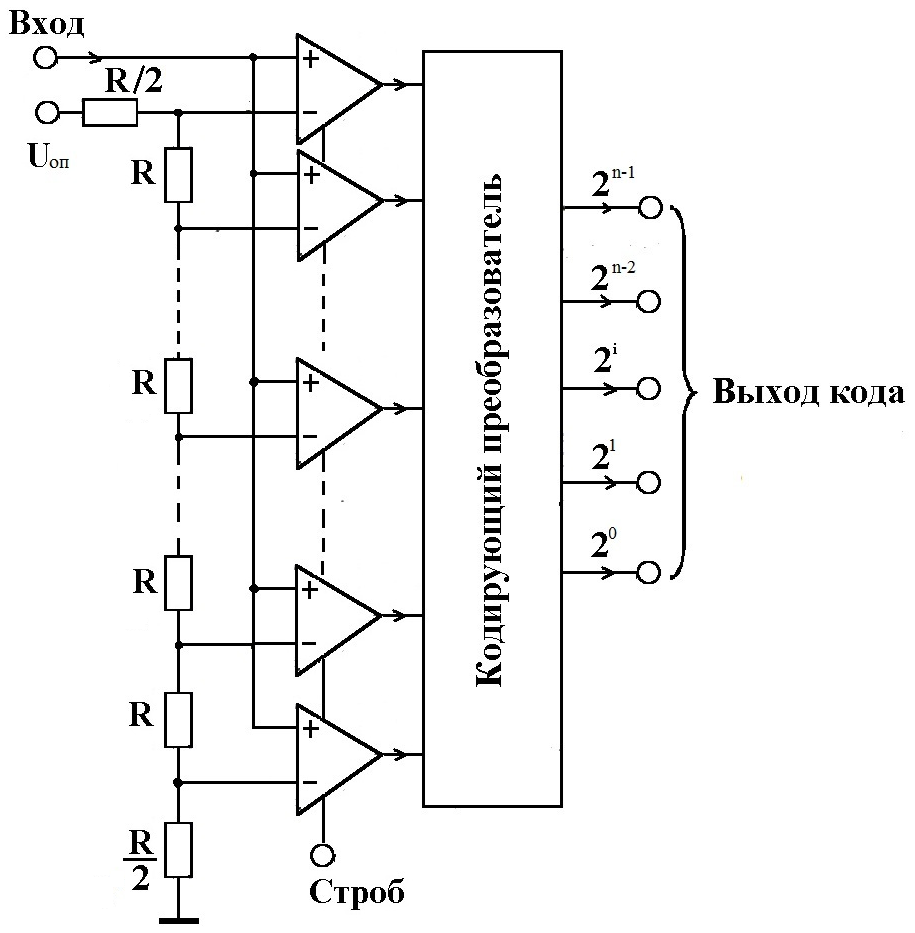
**5. АЦП поразрядным уравновешиванием и параллельного типов.**

Структурная схема ***АЦП последовательного приближения*** от­личается от структурной схемы последовательного счета тем, что вместо счет­чика импульсов включен регистр последовательных приближений (РПП). В основе РПП лежит принцип дихотомии, т.е. последовательного сравне­ния преобразуемого напря­жения Uвх с соответствующими долями возможного макси­мального его значения Um: ½, ¼, 1/8 и т.д. Это позволяет для *n*-раз­рядного АЦП выполнить весь процесс преобразова­ния за *n* последовательных шагов (итераций) вместо (2*n*-1) при ис­пользовании последовательного счета и получить существенный выигрыш по быстродействию.



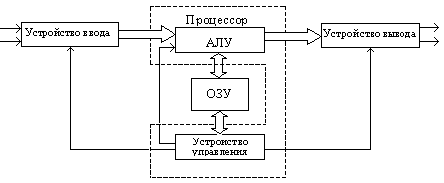
В ***параллельном преобразователе*** используется массив компараторов, каждый из которых сравнивает входное напряжение с индивидуальным опорным напряжением. Такое опорное напряжение для каждого компаратора формируется на встроенном прецизионном резистивном делителе. Значения опорных напряжений начинаются со значения, равного половине младшего значащего разряда, и увеличиваются при переходе к каждому следующему компаратору с шагом, равным Uоп /2N. В результате для 3-х разрядного АЦП требуется 23-1 или семь компараторов. А для 8-разрядного параллельного АЦП нужно уже 255 компараторов.

На выходах компараторов формируется квантованный сигнал, представленный в унитарном коде. Для преобразования унитарного кода в двоичный или двоично-десятичный используется коди­рующий преобразователь. При ра­боте в двоичном коде все резисто­ры имеют одинаковые сопротив­ления R. Время преобразования такого преобразователя составля­ет один такт, т.е. Тпр= Т.



**6. Обобщенная структурная схема компьютера, виды архитектур и форматы машинных слов**

***Компьютером*** называется цифровая программно управляемая система, содержащая взаимосвязанные между собой процессор (П), запоминающее устройство (ЗУ), устройства ввода и вывода и программное обеспечение, предназначенная для арифметической и логической обработки и отображения данных. Процессор в свою очередь состоит из арифметико-логического устройства (АЛУ) и устройства управления (УУ).



***Неймановская архитектура*** компьютера основывается на следующих принципах:

1. *Принцип однородности памяти***.** Программы и данные хранятся в одной и той же памяти. Поэтому ЭВМ не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

2. *Принцип адресуемости памяти***.** Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к хранящимся в них значениям можно было бы впоследствии обращаться или менять их в процессе выполнения программы с использованием присвоенных имен.

3. *Принцип последовательного программного управления***.** Предполагает, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

4. *Принцип жесткости архитектуры***.** Неизменяемость в процессе работы компьютера его структуры и списка команд.

Для ***гарвардской архитектуры*** компьютера характерны следующие признаки:

1. Хранилище команд (инструкций) и хранилище данных представляют собой разные физические устройства, в связи с чем машина гарвардской архитектуры имеет различные адресные пространства для команд и данных.

2. Канал команд и канал данных также физически разделены.

В компьютере с использованием гарвардской архитектуры процессор может читать инструкции и выполнять доступ к памяти данных в одно и то же самое время. Благодаря этому компьютер с гарвардской архитектурой работает быстрее. Недостатком гарвардской архитектуры является более высокая сложность аппаратной реализации компьютера.

**7. Архитектура 8-разрядного однокристального микропроцессора**

***Микропроцессор*** – это устройство для выполнения программно управляемых арифметических и логических операций, содержащее АЛУ, устройство управления, регистры и шины ввода-вывода информации.

***Основная последовательность*** операций микро ЭВМ следующая:

1) передача адреса очередной команды из МП в оперативную память;

2) считывание из памяти и декодирование команды в МП;

3) выполнение команды в МП;

4) формирование адреса следующей команды.

***Аккумулятор*** – это регистр, в котором содержится операнд, подлежащий обработке в АЛУ. Его можно считать основным рабочим регистром, в котором записываются данные из памяти и результаты операций, заносимых обратно в память или в устройство ввода-вывода.

***Счетчик команд***. Команды, образующие программу, хранятся в памяти программы в определенной последовательности. В счетчике команды содержится адрес выполняемой в текущий момент команды.

***Стек*** – это набор регистров или ячеек оперативной памяти, в котором данные или адреса выбираются "сверху" по принципу: "первый – поступивший последним". Стек обычно используется в МП для хранения адресов возврата при обращении к подпрограммам, а также для запоминания состояния внутренних регистров при обработке прерываний.

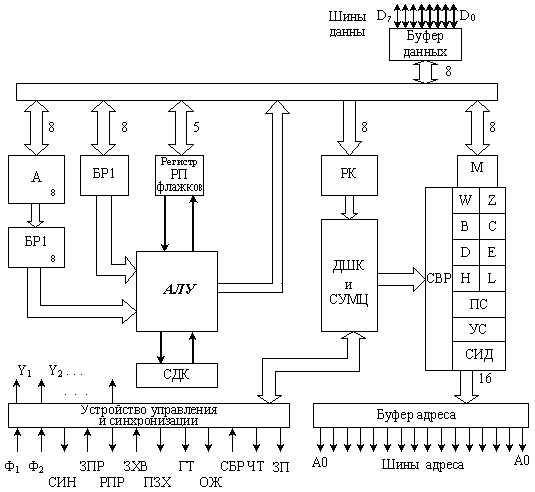
***Регистр команд***. Команда, принимаемая МП из памяти, хранится в регистре команд. Длина команды зависит от типа МП. Простые МП имеют 8-разрядный код операции. Таким образом, всего может быть образовано 2^8=256 различных команд.

***Дешифратор команд*** предназначен для определения операции, которую должен выполнить МП. Сигналы, образующиеся на выходах ДШ, управляют передачей информации между отдельными блоками и задают функции, выполняемые этими блоками.

***Регистры общего назначения*** – сверхоперативная память. Эти регистры применяются в качестве временного запоминающего устройства для различной информации (адресов и данных), которую можно извлечь просто и с большей скоростью, чем из ОЗУ.

***Арифметико-логическое устройство*** выполняет следующие операции: сложение с переносом, вычитание с переносом, сдвиг влево и вправо, счет в прямом и обратном направлении, логическое умножение и сложение, сравнение кодов.

***Схемы синхронизации и управления*** – совместно с ДШК называют устройством управления. Устройство управления расшифровывает поступающую команду и в соответствии с ней вырабатывает необходимую последовательность сигналов, управляющих работой всех остальных блоков МП.



***МП содержит*** шесть 8-разрядных регистров общего назначения РОН (B,C,D,E,H и L) с мультиплексором регистров М, восьмиразрядный аккумулятор A, четыре 8-разрядных буферных регистра БР1, БР2, W и Z, а также 5-разрядный регистр признаков РП. В состав МП входят также регистр команд РК, дешифратор команд ДШК, схема управления машинным циклом СУМЦ, схема десятичной коррекции СДК, схема выборки регистров СВР, программный счетчик ПС, указатель стека УС, схема инкрементации-декрементации СИД, устройство управления, а также буферные регистры данных и адреса.

МП Intel 8080 размещен в корпусе с 48 выводами, но используются только 40 из них. ***Внешние выводы имеют следующее назначение***:

D0 – D7 - двунаправленная шина данных;

A0 – A15 – шина адреса, допускающая подключение памяти объемом 216 = 16Кбайт;

СИН (SYNC) – выход синхроимпульса, вырабатываемого МП в начале каждого машинного цикла;

СБР (RESET) – "Сброс", вход установки микропроцессора в начальное состояние;

ЧТ (DBIN) – "Чтение", выход, сигнал на котором обозначает, что МП принимает информацию с шины данных;

ЗП (WR) – "Запись" – выход, сигнал на котором обозначает, что МП выдает информацию на шину данных;

ГТ (READY) – "Готовность" – вход сигнала готовности от устройства памяти или ввода;

ОЖ (WAIT) – "Ожидание" – выход, признак нахождения МП в состоянии ожидания;

ЗХВ (HOLD) – вход, сигнал на котором заставляет МП войти в режим ЗАХВАТ, когда адресная шина и шина данных МП переходят в состояние высокого сопротивления. Это позволяет внешнему устройству получить управление обеими шинами, например, для прямого доступа в память;

ПЗХВ (HLDA) – выход, сигнал на котором информирует о нахождении МП в режиме ЗАХВАТ;

ЗАПР (INT) – вход запросов на прерывание;

РПР (INTE) – выход сигнала, указывающего на разрешение прерывания;

Ф1, Ф2 – входы для подачи тактовых сигналов.

**8. Функционирование и временные диаграммы микропроцессора**

Формат команды МП580ВМ80 содержит от одного до трех байт. Время, затрачиваемое на извлечение 1 байта информации или выполнение команды, определяемой одним машинным словом, называют машинным циклом. Каждая команда требует для выборки и выполнения от одного до пяти машинных циклов.

Выполнение каждой команды в МП происходит в строгой последовательности, определяемой кодом команды, и синхронизируется сигналами Ф1 и Ф2 тактового генератора. Период синхросигналов Ф1 или Ф2 называется машинным тактом. Любой машинный цикл включает от трех до пяти тактов. Каждый такт длится в течение одного периода синхросигнала.

Цикл М1 – это всегда цикл выборки команды, он длится от 4 до 5 тактов. Циклы М2, М3, М4, М5 обычно состоит из 3-х тактов каждый.

Для микропроцессора 580ВМ80 существует ***десять различных типов циклов***:

1) М1 – извлечение кода команды;

2) М2 – чтение данных из памяти;

3) М3 – запись данных в память;

4) М4 – извлечение из стека;

5) М5 – запись данных в стек;

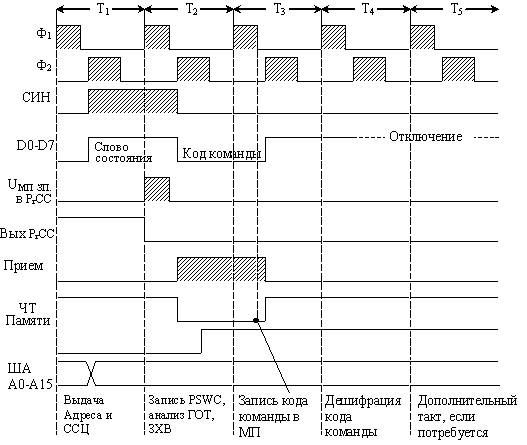
6) М6 – ввод данных из внешних устройств;

7) М7 – запись данных во внешние устройства;

8) М8 – цикл обслуживания прерывания;

9) М9 – останов;

10) М10 – обслуживание прерывания при работе МП в режиме "Останов".



В течение ***такта Т1*** содержимое программного счетчика выдается на адресную шину, а на выходах СИН вырабатывается высокий потенциал. На шину данных подается 8-разрядный код, характеризующий выполняемый цикл. На первом такте каждого машинного цикла МП указывает тип выполняемого цикла с помощью 8-разрядного слова состояния цикла, выдаваемого на шины данных. Слово состояния выдается на шины данных лишь во время импульса СИНХР (такты Т1 и Т2), а используется на протяжении всего машинного цикла. Поэтому его необходимо записывать в специальный регистр слова состояния РгСС. Запись его осуществляется в момент совпадения сигналов СИНХР и Ф1 на втором такте.

За Т1 всегда следует ***такт Т2***, в течение которого проверяется наличие сигналов подтверждения ГТ и ЗАХВАТ, а также проверяется не находится ли МП в состоянии останова HALT. Если на входе READY имеется сигнал готовности, то МП переходит к такту Т3, в противном случае – в состояние ОЖИДАНИЕ и находится в нем до тех пор, пока не появится сигнал готовности. Во время Т2 слово состояния цикла записывается в регистр состояния. Передним фронтом Ф2 заканчивается формирование сигнала СИН, и вырабатывается единичный сигнал Прием, позволяющий поступить байту на вход МП через ШФ. В этом же такте Т2 из сигнала Прием и D7 PSWC формируется сигнал Чт Память, позволяющий поступать данным из памяти на ШД микропроцессора.

В ***такте Т3*** во время заднего фронта Ф1 производится запись кода команды во внутренний регистр кода команды. Положительным фронтом Ф2 оканчивается сигнал ПРИЕМ на выходе МП и сигнал ЧТ Память. Импульс на выходе Прием формируется в машинных циклах: чтение команды, ЧТ данных из памяти, прерывания, чтение из стека или внешнего устройства. На основании декодирования команды ДШК схема управления формируют сигналы управления и синхронизации для внутренних пересылок данных, а также соответствующие дешифрируемой команде машинные циклы.

На последующих ***тактах Т4 и Т5*** ДШК расшифровывает код команды, определяет количество байтов в команде, формирует команды на внутренние пересылки данных и подготавливает МП к выполнению следующих машинных циклов.

**9. Система команд 8-разрядного микропроцессора**

***Все команды делятся на 5 групп:***

1. Команды пересылки данных. Производят пересылку данных между регистрами или между памятью и регистрами.

2. Арифметические команды. Предназначена для выполнения сложения, вычитания, увеличения или уменьшения содержимого реестров или ячеек памяти.

3. Логические команды. Выполняет логические операции «И», «ИЛИ», исключающее ИЛИ, сравнение и сдвиг.

4. Команды переходов условных и безусловных, вызова подпрограмм и возвращение в подпрограммы.

5. Команды ввода/вывода, управления и работы со стеком.

***По длине команды делятся на:***

1. Однобайтные

2. Двухбайтные

3. Трехбайтные

***Режим адресации памяти*** – это процедура или схема преобразования адресной информации об операнде в его исполнительный адрес. В микропроцессоре КР580ВМ80А используется ***пять методов адресации***:

1. *Прямая* – в команде задается адрес ячейки памяти, где расположен операнд; он указывается во втором и в третьем байтах команды. К этой группе также относятся команды, в которых задается адрес порта ввода/вывода:

STA 8020H – требует четырех обращений к памяти;

IN 05H – требует двух обращений к памяти.

2. *Прямая регистровая* – в команде задается адрес регистра или пары регистров, где находится 8- или 16-битный операнд:

MOV A, B – требует одного обращения к памяти;

3. *Непосредственная* – операнд содержится в самой команде:

MVI A, 08H – требует двух обращений к памяти;

LXI M, 8020H – требует трех обращений к памяти.

4. *Косвенная* – адрес М ячейки памяти, где расположен операнд, определяется содержимым парного регистра, явно или неявно указанного в команде:

MOV A, M – пересылка в A из ячейки памяти, на которую указывает HL;

LDAX B – загрузка A из ячейки памяти, на которую указывает пара BC.

5. *Неявная* – адрес операнда не указывается в явном виде, а определяется кодом операции:

ADD B; A ← A+B, аккумулятор не задается в явном виде.

Положение полей в микропроцессоре КР580ВМ80А переменное, и в зависимости от команды, ***назначение поля может иметь следующее значение***:

Byte 1 – содержит код операции, длину команды, адреса регистров;

Byte 2 – содержит адрес порта ввода/ вывода, 8-разрядный операнд или младшую часть 16-разрядного операнда;

Byte 3 – содержит старшую часть 16-разрядного операнда.

**10. Программирование арифметических и логических операций**

***Арифметические операции:***

1. ADD D – сложение содержимого аккумулятора с содержимым регистра

2. ADI d8 – A←d8 + (A)

3. SUB B – вычесть B из А

4. INR C – инкрементировать С

5. DCR C – декрементировать С

6. DAD B – сложить BC с HL

7. RRC и RLC – сдвиг А вправо и влево

***Логические операции:***

1. ANA B – логическое И B с A

2. ANI d8 – логическое И между данными и A

3. CMP D – сравнить A с D

4. CPI d8 – сравнить A с данными, заданными в команде

5. ORA C – логическая операция А ИЛИ С

6. ORI d8 – логическая операция А ИЛИ d8

7. XRA B – исключающее ИЛИ B с A

8. XRI d8 – исключающее ИЛИ d8 с A

**11. Программирование операций ввода/вывода данных**

***Команды ввода/вывода:***

1. OUT pp – записать A в порт pp

2. IN pp – записать данные из порта pp в A

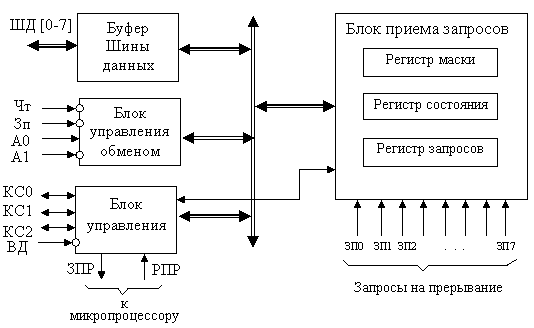
**12. Устройство и функционирование шинных формирователей и буферных регистров**

***Шинные формирователи*** включаются между источником информации и шиной. Они усиливают сигналы по мощности при работе на шину, отключают источник информации от шины, когда он не участвует в обмене, формируют при необходимости требуемые уровни сигналов логической 1 или 0. Двунаправленные ШФ позволяют в зависимости от сигнала управления передавать сигналы в шину или, напротив, принимать их с шины и передавать приемнику данных.

***Буферные регистры***. Буферные регистры служат для подключения к магистрали внешнего устройства. В отличие от шинных формирователей, которые только усиливают сигналы, буферные регистры способны хранить данные. Благодаря этому они могут выполнять функцию портов. Буферные каскады с тремя состояниями на выходах регистра обеспечивают портам возможность отключения от магистрали под действием управляющих сигналов, а также необходимую нагрузочную способность.

**13. Устройство и функционирования контроллера прерываний**

***Программируемый контроллер прерываний*** отвечает за приём запросов прерываний от различных устройств, их хранение в ожидании обработки, выделение наиболее приоритетного из одновременно присутствующих запросов и выдачу его вектора в процессор, когда последний пожелает обработать прерывание.



Запросы на прерывание от внешних устройств подаются на входы ЗП0 – ЗП7 и запоминаются в регистре запросов. В регистре состояния содержатся все запросы на прерывания, обслуживаемые в данный момент. Регистр маски содержит единицы в разрядах, соответствующих маскируемым в настоящий момент входам запросов. Установка в единицу того или иного разряда регистра маски блокирует передачу запроса на прерывание.

***Программирование контроллера*** осуществляется двумя типами команд: ***командами инициализации*** и ***командами управления режимом***. Ввод команд для контроллера прерываний осуществляется микропроцессором, как правило, командой OUT. Однако в системе может быть организовано обращение к контроллеру как к ячейкам ЗУ. ***Команды инициализации*** подаются перед началом работы контроллера. Эти команды задают стартовые адреса подпрограмм обслуживания прерываний, расстояниями между соседними стартовыми адресами и указывают, если необходимо, на наличие других контроллеров в системе. ***Команды управления режимом*** служат для оперативного изменения режимов обслуживания прерываний и могут подаваться в любое время в процессе работы контроллера.

При поступлении запроса на прерывание по одному из входов ЗП0-ЗП7 он фиксируется в регистре запросов, а на выводе ЗПР формируется сигнал запроса на прерывание, который поступает на соответствующий вход микропроцессора. Если запросы на прерывание не запрещены, то процессор завершает текущий цикл, сохраняет в стеке состояние программного счетчика и выдает контроллеру сигнал разрешения прерывания РПР. По этому сигналу контроллер выставляет на шину данных код команды вызова подпрограммы CALL, который поступает в регистр команд процессора. После дешифрации этой команды процессор последовательно выдает еще два сигнала РПР. По первому из них контроллер выдает младший байт адреса подпрограммы обслуживания прерываний, а по второму – старший. Эти байты заносятся в программный счетчик и процессор начинает выполнять команду, начиная из адреса (вектора), поступившего от контроллера прерываний. Подпрограмма обслуживания прерываний должна заканчиваться командой возврата RET, по которой два байта из стека загружаются в программный счетчик и процессор продолжает выполнять прерванную программу.

**14. Устройство и функционирование контроллера прямого доступа в память**

***Контроллер прямого доступа к памяти*** служит для организации обмена данными между внешними устройствами и памятью, без участия центрального процессора.

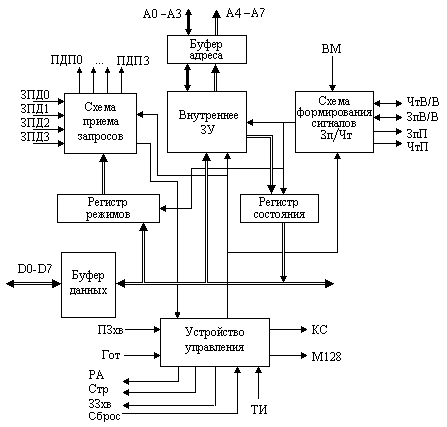
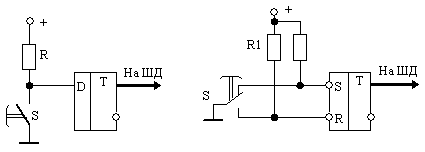


Схема приема запросов предназначена для приема и формирования ответных сигналов запроса в соответствии с приоритетом внешних устройств. Внутреннее запоминающее устройство служит для хранения начального адреса и числа циклов ПД для каждого канала. Для этого в нем расположены для каждого канала шестнадцатиразрядный регистр адреса РгА и регистр циклов РгЦ. РгА загружается адресом первой ячейки памяти, к которой будет обращение. Младшие 14 разрядов РгЦ указывают число циклов минус 1 до конца счета. Разряды 14 и 15 РгЦ указывают на вид обмена 00 – контроль, 01 – запись в ЗУ, 10 – чтение ЗУ, 11 – запрещенная комбинация. Схема формирователя сигналов Зп/Чт обеспечивает обмен информацией между микропроцессором и БИС контроллер с одной стороны, и БИС и ОЗУ и прерывающее устройство – с другой. Регистр установки режимов РгР хранит информацию о режимах работы БИС, к которым относятся «Автозагрузка», «Конец счета-стоп», «Удлиненная запись», «Обычная запись», «Циклический сдвиг приоритета» и «Фиксированный приоритет». Регистр состояния каналов РгС указывает номер канала, который достиг конца счета.

**15. Подключение клавиатуры к микропроцессору**

***Клавиатура компьютера*** представляет собой набор механических контактов, контактов на основе токопроводящей резины, емкостных датчиков или датчиков на основе эффекта Холла. Состояние кнопки фиксируется в триггере, выход которого присоединяется к одной из линий шины данных микропроцессора.

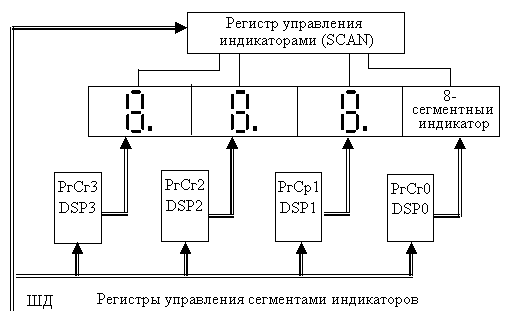


Из схемы видно, что в отпущенном состоянии на входе D–триггера присутствует уровень логической 1 и триггер принимает состояние «1». При нажатии кнопки триггер принимает нулевое состоянии. Недостатком такой схемы является наличие эффекта «дребезга» контакта за счет того, что в момент нажатия ключа подвижный пружинистый контакт несколько раз отскакивает от неподвижного, прежде чем установиться в устойчивое состояние. В результате «дребезга» наблюдается расщепление единичного потенциала в момент смены состояния и возможно неверное фиксирование состояния кнопки. Дребезг контакта может быть устранен программно путем считывания состояния триггера с некоторой задержкой после замыкания контакта.

Для определения номера нажатой клавиши необходимо нулевой потенциал подать только на одну из горизонтальных линий матрицы (путем записи единицы в соответствующий триггер Увыв), а на остальных должна быть «1» и определять ячейку регистра УВв, которая приняла состояние «1». Затем по очереди перемещать «0» в Увыв (сканировать УВыв) и определять разряд Увв с нулевым значением.

**16. Подключение устройств статической индикации к микропроцессору**

***Восьмисегментный дисплей*** представляет собой микросхему с 8-мю светодиодами, выполненными в виде прямоугольных сегментов и расположенных в пространстве в виде цифры 8 с точкой.



При ***статическом способе*** выводы сегментов каждого из индикаторов подключаются к своему регистру. Для управления разрешением высвечивания символа на индикаторе используется отдельный регистр, причем общий электрод каждого из индикаторов подсоединяется к соответствующему выходу этого регистра. Программа управления выводом информации на дисплей состоит из операции выдачи кода символа на соответствующий индикатор (регистр DSP) и вывода разрешающего сигнала на этот индикатор (Регистр SСAN).

Одноименные сегменты каждой ячейки индикатора связаны общим проводом и соединены с соответствующим разрядом регистра сегментов РгСг. Выходы анодов каждого из индикаторов подключены к регистру сканирования РгСк. Наличие уровня логической единицы в соответствующем разряде регистра сканирования РгСк приводит к высвечиванию символа в соответствующем индикаторе дисплея при наличии информации на шине данных. Вариант программы включения сегментов второй ячейки с помощью кода, задаваемого со входного регистра имеет вид:

MVI A, 04 ;поместить в А число 00000100

OUT SCAN ;вывести число на РгСк и включить цифру 2

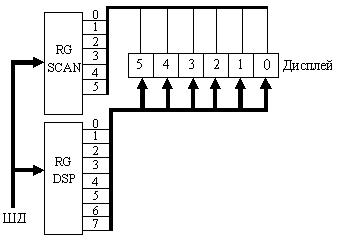
M1:

IN 20 ;ввести данные в А из входного регистра

OUT DSP2 ;записать их в регистр сегментов РгСг дисплея

JMP M1 ;продолжить с метки М1

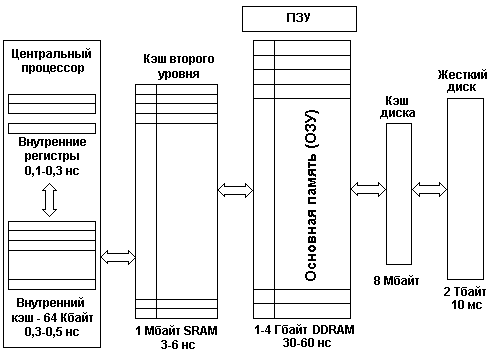
**17. Подключение устройств динамической индикации к микропроцессору**



В этом режиме вывод информации на каждый индикатор дисплея выводится микро-ЭВМ последовательно. Сначала в RGDSP выводится код отображаемого символа, а в регистр RGSCAN – разрешающий потенциал для высвечивания только одного индикатора (например, 5-го).

Цифра или символ на индикаторе высвечивается некоторый промежуток времени, задаваемый подпрограммой задержки. Затем индикатор гасится, выставляется в RGDSP код, который должен быть отображен 4-м индикатором, и подается управляющий сигнал в RGSCAN, разрешающий светиться только этому индикатору.

**18. Схема организации памяти компьютера и основные параметры ОЗУ**

****

При перемещении слева направо происходит следующее:

1) снижается стоимость бита;

2) возрастает емкость;

3) возрастает время доступа и снижается частота обращений процессора к памяти.

***Основная память компьютера состоит из ЗУ двух видов***: оперативного (ОЗУ) и постоянного (ПЗУ).

***ОЗУ*** предназначено для хранения переменной информации и допускает изменение своего содержимого в ходе выполнения процессором вычислительных операций. При выключении питания содержимое оперативной памяти теряется.

***ПЗУ*** содержит информацию, которая не должна изменяться в процессе всего времени существования компьютера. Такую информацию составляют стандартные программы тестирования компьютера, драйверы устройств, значения физических констант и пр. Эта информация заносится в ПЗУ заранее, в процессе изготовления компьютера. ПЗУ является энергонезависимым устройством.

***Важнейшие параметры:***

1) Емкость памяти - число бит хранимой информации.

2) Время обращения к памяти - время от подачи сигнала на запись или считывание до того момента, когда закончатся все действия, связанные с выполняемой операцией и устройство готово реализовать следующую операцию.

**19. Статические ОЗУ с произвольным доступом**

В статическом ОЗУ информация в ячейках памяти хранится до тех пор, пока подается напряжение питания.

Ячейки представляют собой полупроводниковые триггеры. Достоинства — небольшое энергопотребление, высокое быстродействие. Отсутствие необходимости производить «регенерацию». Недостатки — малый объём, высокая стоимость. Благодаря принципиальным достоинствам широко используется в качестве кеш-памяти процессоров в компьютерах.

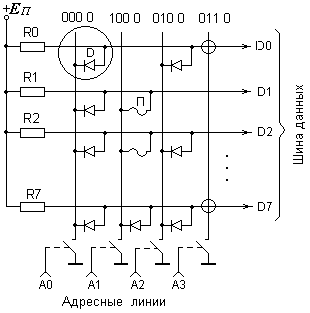
**20. ОЗУ динамического типа**

В запоминающих устройствах динамического типа информация хранится в виде заряда на конденсаторе. Поэтому питание на ОЗУ подается не постоянно, а только в очень короткие промежутки времени. Оно используется для восстановления заряда на конденсаторах матрицы ОЗУ. Благодаря импульсному питанию динамические ОЗУ потребляют в тысячи раз меньше мощности, чем аналогичные по емкости статические.

Достоинства — низкая стоимость, большой объём. Недостатки — необходимость периодического считывания и перезаписи каждой ячейки — т. н. «регенерации», и, как следствие, понижение быстродействия, большое энергопотребление. Процесс регенерации реализуется специальным контроллером, установленным на материнской плате или в центральном процессоре.

**21. Постоянные и перепрограммируемые запоминающие устройства**

Постоянные запоминающие устройства (ПЗУ) являются энергонезависимыми устройствами, служащими для хранения цифровых данных. ПЗУ могут быть построены на пассивных элементах (плавких перемычках П или диодах D) или активных (транзисторах). Схема ПЗУ представляет собой матрицу, где количество горизонтальных линий равно разрядности хранимого слова, а число вертикальных – количеству хранимых слов.

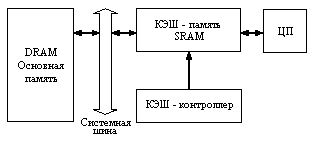


Запись информации в ПЗУ осуществляется пословно (побайтно). Для занесения информации в ячейку ПЗУ необходимо на линии данных, в которых должна быть "1", подать высокий потенциал (≈ 25 В) и выбрать соответствующую адресную линию, т.е. соединить ее с сигнальной землей. Протекающий ток расплавляет диод или плавкую перемычку, исключая тем самым шунтирующую цепь соответствующей линии данных.

Недостаток рассмотренной схемы ПЗУ состоит в том, что после занесения информации в это устройство ее нельзя изменить. Для устранения этого недостатка разработаны полупостоянные ***электрически*** ***перепрограммируемые постоянные запоминающие устройства*** (ЭППЗУ). Схема ЭППЗУ подобна ПЗУ на основе МОП транзисторов, однако транзисторы в таком устройстве имеют "плавающий" затвор, который электрически изолирован оксидным слоем полупроводникового материала. При подаче на "плавающий" затвор (ПЗ) положительного потенциала по отношению к стоку транзистора на ПЗ индуцируется электрический заряд, который за счет высококачественной изоляции может сохраняться до 10 лет и более.

**22. Кэш-память и ее организация**

***Кэш-память*** ― это быстродействующая память, располагаемая между МП и основной памятью. Она предназначена для повышения производительности компьютеров. В кэш-память помещают текущие программы и данные, хранящиеся обычно в основной памяти.



В начале работы программа, обращаясь к своему адресному пространству, передает устройству управления памятью или микропрограмме виртуальный адрес. Если страница или сегмент, содержащие этот адрес, находятся в основной памяти, виртуальный адрес преобразуется в адрес основной памяти. В противном случае возникает отказ страницы, и страница или сегмент загружается в основную память.

После этого аппаратно проверяется, не находится ли требуемый блок уже в кэш-памяти. Если это так, то содержимое адресуемой ячейки извлекается из кэша (либо результат вычисления заносится в ячейку). В противном случае формируется отказ кэш-блока, и информация копируется из основной памяти в КЭШ. После загрузки блока в кэш-память команда выполняется повторно. Схема загрузки из основной в кэш-память управляется внешней (по отношению к ЦП) схемой – кэш-контроллером и без участия ОС.

***Существуют три способа организации кэша:***

1. *Полностью ассоциативная кэш-память.* В этом случае в кэше содержатся наиболее часто используемые процессором блоки данных и их полные адреса. Когда процессор запрашивает данные, кэш-контроллер сравнивает адрес требуемых данных с каждым адресом в кэш-памяти. Для ускорения поиска используют схемы с параллельным поиском, которые содержат компараторы и контроллеры.

2*.* *Кэш-память с прямым отображением.* Каждый блок из основной памяти может занимать одно единственное место в кэше. Это уменьшает число шагов поиска и сравнений, необходимых для фиксации попаданий и промахов. Основной недостаток – конфликтование в кэш-памяти ячеек ОЗУ, которые имеют одно и тоже место хранения в кэше. В этом случае каждый раз нужно будет осуществлять попеременно подкачку этих ячеек. Однако такие случаи встречаются довольно редко, и эта кэш-память обладает приемлемой производительностью при невысокой стоимости.

3. *Частично ассоциативная кэш-память*. Этот способ кэширования является промежуточным между рассмотренными выше. Частично ассоциативная кэш-память характеризуется наличием нескольких наборов блоков с прямым отображением, которые функционируют параллельно. Когда из основной памяти поступает блок данных, он может быть записан на соответствующее этому блоку место любого набора. При выборе нового блока кэш-контроллер должен решить, на место которого блока в кэш-памяти его записать. При этом может быть реализован:

а) случайный выбор;

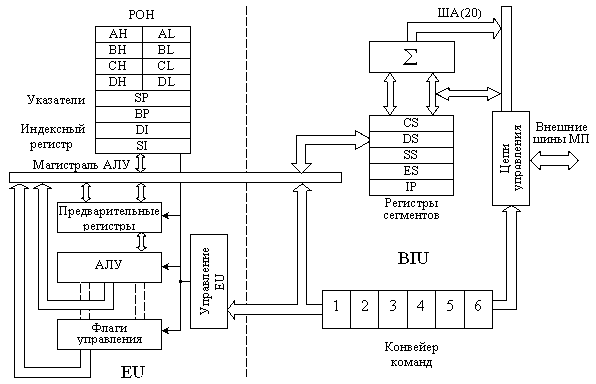
б) запись в последовательном порядке;

в) на место блока, который не адресовался дольше других.

**23-24. Архитектура 16-разрядных процессоров. Устройство и функционирование 16-разрядного процессора**

16-разрядный МП Intel 8086 имеет мультиплексную 20-разрядную ША и 16-разрядную ШД и рассчитан на работу как в одно, так и в многопроцессорных системах.

Эффективность работы МП существенно повышена за счет введения команд математических операций (включающих умножение и деление) над 8- и 16-разрядными числами, команд побитовой обработки чисел, команд работы с массивами данных, расширения видов прерываний работы МП, а также реализации конвейерного типа выполнения команд в самой БИС. МП может работать с памятью объемом до 1 Мбайт, обмениваться информацией с 64 Кбайт внешних устройств, имеет 256 типов различных прерываний.



Блок сопряжения с шинами BIU производит все пересылки данных и кодов для EU. Пересылки между памятью или внешними устройствами осуществляется по требованию EU. В то время как EU занят выполнением команды, блок BIU получает последующие в программе коды из памяти. Блок выполнения команд EU имеет 16-разрядные АЛУ с регистром состояния и флажками управления, а также РОНы. Все регистры и внутренние магистрали блоки 16-разрядные. Блок не имеет связи с внешними шинами МП.

На АЛУ поступают коды команд из конвейера команд, расположенного в BIU. Если в результате дешифрации кода команд в АЛУ необходимо получение одного или нескольких операндов по внешним магистралям МП, то EU запрашивает BIU на получение и размещение необходимых данных в BIU.

BIU считывает команды с памяти и сохраняет их в конвейере команд, где может быть размещено до 6 инструкций. Это позволяет BIU выдавать их в EU по мере надобности без дополнительной загрузки внешней шины. BIU организует получение нового кода команды как только два байта из конвейера будут переданы в EU.

Коды подаются в EU последовательно, так как они записаны в программе. Если EU выполняет команду передачи управления в другое место программы, то BIU очищает конвейер команд, получает код из нового адреса, передает его в EU и начинает заполнять конвейер заново. Если EU требует обращения к внешнему устройству, то BIU приостанавливает процесс получения команд в конвейер и организует необходимый цикл обмена данными.

Любая ячейка памяти МП имеет два типа адресов: ***физический*** и ***логический***. ***Физический*** адрес представляется 20-разрядным числом и однозначно определяет любую из 1 Мбайт ячеек памяти. В 16-разрядной системе адреса расположены в диапазоне от 0 до FFFFF. Весь обмен информацией МП с памятью осуществляется с использованием физических адресов. ***Логический*** адрес состоит из двух основных частей: значения базы сегмента и значения смещения в сегменте. Базовый адрес и смещение в сегменте отображаются 16-разрядными числами. Как только BIU обращается к памяти, базовый адрес формирует физический по принципу: Значение базы сегмента смещается на четыре разряда влево, и полученное 20-разрядное число складываются со значением смещения в сегменте. Таким образом, база сегмента задает для памяти сегменты длинной 64 Кбайт, а значение сегмента в смещении – расстояние от начала сегмента до искомого адреса памяти. Максимально возможное смещение в сегменте равно 64 Кбайт. В любой момент времени программа может осуществлять доступ к одному из четырех сегментов:

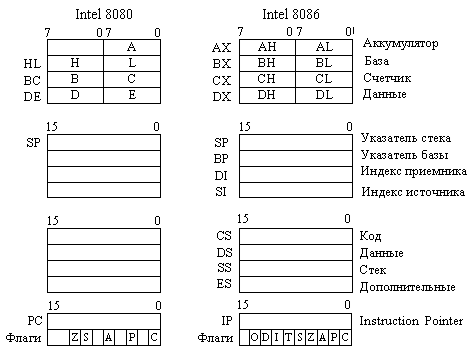
1) сегменту текущего кода (CS);

2) сегменту текущих данных (DS);

3) сегмент текущего значения стека (ES);

4) дополнительный сегмент текущих значений (ES).

**25. Регистры 16-разрядного процессора и система команд**



***МП 8086 содержит три группы регистров***:

1) *Группа РОН* включает в себя семь 8-разрядных регистров, к ним может быть организован программный доступ как к 8- или 16-разрядным регистрам. 16-разрядные регистры обозначаются AX, BX, CX и т.д. При обращении к ним, как к 8-разрядным регистрам, их обозначают AL, AH, BL, BH и т.д. Все эти регистры могут быть использованы при выполнении арифметических и логических команд. Однако, есть команды, использующие определенные регистры для специфических целей, тогда применяют мнемонические обозначения: аккумулятор, база, счет, данные.

2) *Группа указателей и индексных регистров* состоит из четырех 16-разрядных регистров. Регистры указатели: SP (Stack Pointer); BP (Base Pointer). Индексные регистры: SI (Source index); DI (Destination index). Обычно эти регистры содержат информацию о смещении по адресам в выбранном сегменте и позволяют компактно писать программы каждый раз непосредственно, не приводя используемого адреса. С их помощью производится вычисление адресов программ. Чаще всего в регистрах указателях записано адресное смещение по отношению к стековому сегменту, а в индексных регистрах – адресное смещение по отношению к сегменту данных.

3) *Группа регистров сегментов* состоит из четырех 16-разрядных регистров: CS; SS; DS и ES. Они задают начальные адреса самих сегментов памяти.

Также имеется флаговый регистр, который содержит шестнадцать триггеров, из которых используется только 9. Эти триггеры отображают состояние процессора при выполнении последней арифметической или логической команды. Пять триггеров аналогичны МП 580-й серии. Дополнительные триггеры сигнализируют:

а) OVERFLOW – переполнение;

б) DIRECTION – направление – указывает направление работы с массивами (автоматическое увеличение или уменьшение на единицу адреса массива);

в) INTERRUPT – прерывание – определяет для МП возможность реагирования на прерывание;

г) TRAP – устанавливает для МП пошаговый режим.

**26. Команды работы со строками**

***Система команд*** МП состоит из 91 мнемокоманд и по функциональному признаку разбивается на 6 групп:

1) Пересылка данных;

2) Арифметические операции;

3) Логические операции и сдвиги;

4) Передача управления;

5) Обработка цепочек;

6) Управление процессором.

Под ***цепочкой (строкой)*** понимают последовательность любых контекстно-связанных байт или слов, находящихся в смежных ячейках памяти.

1) Цепочной команде может предшествовать специальный однобайтовый префикс повторения REP. Число повторений задается регистром СХ. Например, последовательность команд:

MOV CX, 500

REP MOVS DST, SRC

заставит МП выполнять команду MOVS 500 раз, уменьшая значение регистра СХ после каждого повторения, до тех пор, пока <СХ> не станет равным 0.

Остальные префиксы повторения, решение о продолжении или прекращении повторений принимают в зависимости от значения флага ZF: REPE (repeat while equal) пока равно; REPZ (repeat while zero) повторять пока нуль; REPNE; REPNZ.

2) Команда LODS (load string) пересылает операнд строка-источник, адресованный регистром SI, из сегмента данных в регистр АL или АХ, а затем изменяет SI так, чтобы он указывал на следующий элемент строки. Его значение увеличивается, если флаг направления DF равен 0, и уменьшается, если DF=1. Формат команды имеет вид:

LODS SRC ;<AC> <- <SRC>.

3) Команда сохранения строки STOS (stove string) служит для запоминания содержимого аккумулятора в элементе строки, адресуемом регистром DI. После выполнения команды содержимое DI увеличивается на 1. Формат команды:

STOS DST ;DST:=<AC>

4) Команда пересылки строк MOVS используется для копирования байта или слова из одной части памяти в другую. Она имеет формат:

MOVS DST, SRC ;DST:=< SRC>

5) Команды сравнения строк CMPS (compare string). Формат команды имеет вид:

CMPS SRC, SRC ;< SRC > = < DST > ?

6) Команда сканирования SCAS имеет формат:

SCAS DST ;<AC> - <DST>

По этой команде вычитается элемент строки DST (байт или слово), адресуемое регистром DI, из содержимого аккумулятора AL или АХ. В соответствии с полученной разностью устанавливаются флажки, но значение операндов не изменяется.

**27. Команды арифметических и логических операций**

1) Команда ADD позволяет производить сложение 8- или 16-битовых двоичных чисел в режиме регистр-регистр, регистр-память и память­регистр, причем адресация памяти осуществляется в любом допустимом режиме.

ADD mem/reg1,mem/reg2

ADD mem/reg,data

2) Команда ADC выполняет сложение с переносом.

ADC mem/reg1,mem/reg2

ADC mem/reg,data

3) Команда INC позволяет увеличить на 1 содержимое любого общего регистра или ячейки памяти.

INC mem/reg

4) Команда SUB позволяет производить вычитание 8- или 16-битных двоичных чисел.

SUB mem/reg1,mem/reg2

SUB mem/reg,data

5) Команда SBB выполняет вычитание с переносом.

SBB mem/reg1,mem/reg2

SBB mem/reg,data

6) Команда DEC позволяет уменьшить на 1 содержимое любого общего регистра или ячейки памяти.

DEC mem/reg

7) Команда NEG изменяет знак числа, находящегося в регистре или ячейке памяти, на противоположный.

NEG mem/reg

8) Команда умножения беззнаковых целых чисел MUL выполняет умножение адресуемого операнда на содержимое аккумулятора.

MUL reg

MUL mem

9) Команда IMUL аналогична команде MUL, но сомножители и произведение интерпретируются как знаковые двоичные числа в дополнительном коде.

IMUL reg

IMUL mem

10) Команда деления беззнаковых чисел DIV производит деление содержимого аккумулятора и его расширения на содержимое адресуемого операнда.

DIV reg

DIV mem

11) Команда IDIV аналогична команде DIV, но делимое, делитель и частное интерпретируются как знаковые двоичные числа в дополнительном коде.

IDIV reg

IDIV mem

12) Команда сравнения CMP выполняет вычитание второго операнда из первого, но нигде не запоминает результат операции и влияет только на состояние флажков.

CMP mem/reg1,mem/reg2

CMP mem/reg,data

13) Логические операции представлены командами NOT (инверсия), AND (конъюнкция), OR (дизъюнкция), XOR (исключающее ИЛИ) и командой TEST, которая выполняет конъюнкцию операндов, но не изменяет их значений.

AND mem/reg1,mem/reg2

AND mem/reg,data

OR mem/reg1,mem/reg2

OR mem/reg,data

XOR mem/reg1,mem/reg2

XOR mem/reg,data

TEST mem/reg1,mem/reg2

TEST mem/reg,data

NOT mem/reg

14) Команды ROL и ROR реализуют простой циклический сдвиг влево и вправо соответственно, помещая значение из выдвигаемого бита в освобождающийся бит.

Поле операнда команд сдвига имеет вид mem/reg,count. Опреанд count определяет число сдвигов и может быть указан как константа 1 или как регистр CL. В первом случае выполняется сдвиг на один бит, а во втором число сдвигов определяется содержимым регистра CL, которое воспринимается как беззнаковое число.

ROL mem/reg,1

ROL mem/reg,CL

ROR mem/reg,1

ROR mem/reg,CL

15) Команды RCL и RCR называются командами циклического сдвига влево и вправо через перенос.

16) Команды SHL и SHR реализуют логический сдвиг влево и вправо соответственно.

17) Команды SAL и SAR предназначены для арифметического сдвига влево и вправо.

**28. Способы адресации памяти и устройств ввода/вывода**

***1. Регистровая адресация.***

MOV AX, SI ; <SI>→<AX>

ADD DI, BX ; <BX> + <DI>

***2. Непосредственная адресация***

SUB AL, 30H ; <AL> - 48 (30H = 48D)

MOV CL, 10 ; (10→<CL>)

***3. Прямая адресация***

MOV AX, GAMMA

ADD TEMP, BL ; <BL>+ <<DS>↑4+TEMP>

***4. Косвенная регистровая***

ADD AX, [DI] ; к <AX> прибавляется содержимое ячейки памяти, адрес который находится в DI

***5. Базовая адресация***

MOV AX, [BP]10 ; Обе команды передают шестое слово массива, адресуемое ВР,

MOV AX, [BP+10] ; в Аккумулятор,

ADD [BX]TEMP,CX ; Прибавить <СХ> к слову ТЕМР в массиве, адресуемом ВХ.

***6. Индексная адресация***

MOV ADRM [SI], AX ; Передать <АХ> в элемент массива с начальным адресом ADRM

***7. Базово-индексная адресация***

MOV AX, [BX+2][DI]

***8. Адресация портов ввода-вывода***

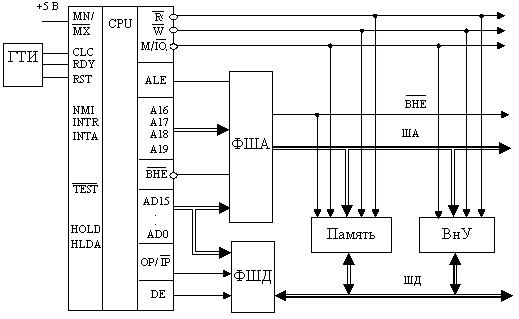
IN AL, 40H ; Ввод байта из порта номер 40Н

OUT DX, AX ; Вывод слова в порт с адресом, хранящемся в DХ

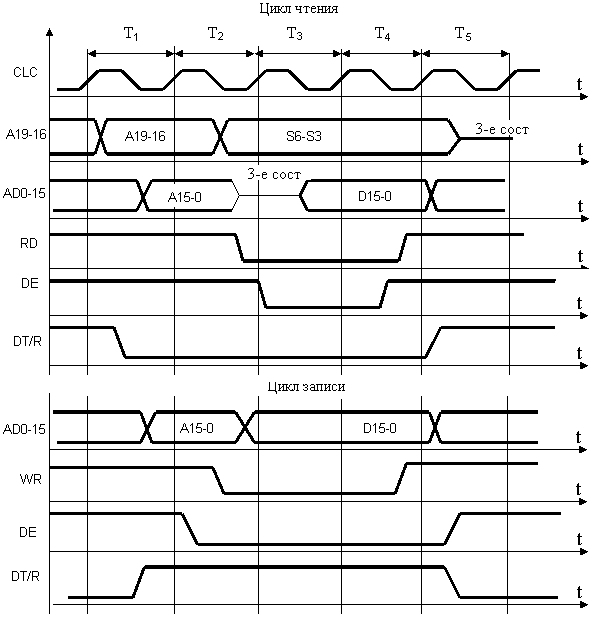
IN AX, DX ; Ввод слова из устройства, адрес которого хранится в регистре DХ.

**29. Структура и функционирование 16-разрядной микро-ЭВМ**

Микропроцессор обменивается информацией с внешними устройствами (ВнУ) и памятью по 16-разрядной шине данных с использованием 20-разрядной адресной шины, шины управления и шины состояния МП.



Минимальный режим работы процессора задается путем подачи высокого уровня сигнала на вывод микросхемы MN/MX. При минимальном режиме управляющие сигналы для памяти и внешних устройств генерирует сам процессор, а в максимальном режиме для управления шинами используется специальный системный контроллер, который формирует управляющие сигналы на основании значения линий состояния S0-S2.



Минимальный цикл обмена информацией микро-ЭВМ состоит из четырех машинных тактов. Цикл начинается с формированием на такте Т1 сигнала M/IO, определяющего тип устройства (ОЗУ или ВнУ), к которому производится обращение для пересылки данных. Длительность сигнала M/IO равна длительности цикла шины, и он используется для селекции адресуемого устройства. В такте Т1 и в начале такта Т2 микропроцессор выставляет адрес ОЗУ на линии А19-16 и АД15-0, либо адрес ВнУ, а также вырабатывает сигнал ВНЕ, который вместе с А0 определяет передачу слова или одного из байтов. Одновременно с этим МП выдает строб адреса ALE, по спаду которого адрес фиксируется во внешних регистрах-защелках.

В такте Т2 происходит переключение шин: на линии A19/S6 – A16/S3 поступают сигналы состояния S6-3, которые сохраняются до конца такта Т4. Значения этих сигналов зависят от вида выполняемого действия процессора. В минимальном режиме работы процессора эти сигналы практически не используются. В цикле чтения в такте Т2 линии АД15-0 переводятся в третье состояние, давая тем самым процессору перейти из режима записи к режиму приема команды или данных. В тактах Т2-Т4 вырабатывается сигнал чтения RD=0, который указывает адресуемому устройству на необходимость выдачи слова. Для управления формирователем шины данных (ФШД), который подключается к линиям АД15-0, в тактах Т2-Т4 формируется сигнал DE (Data Enable), разрешающий прием данных, действующий в течение всего цикла.

После выполнения чтения и установления сигнала RD=1 микропроцессор заканчивает такт следующим образом: линии АД15-0 переключаются в высокоомное состояние, сигналы M/IO, DE, DT/R, S7-3 переходят в неактивное состояние. Шинные формирователи данных отключены от канала.

**30-31. Защита памяти в процессорах второго и последующих поколений. Работа процессора в защищенном режиме**

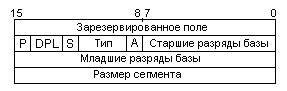
Для осуществления защиты памяти кроме базового адреса вводится ***указатель размера*** и ***атрибут сегмента***. Наиболее широко применяются следующие ***атрибуты сегментов***:

а) разделение на системную область и область пользователя;

б) разделение на область программ и данных;

в) в случае работы с областью программы производят разделение ее на участки, допускающие только считывание данных или также и запись.

Процессор в каждом цикле вырабатывает сигналы состояния, показывающие, является ли данный цикл считыванием или записью. Устройство управления памятью (УУП) при каждом обращении к памяти сравнивает атрибуты сегментов с этими сигналами. Если в размере сегмента или в его атрибуте обнаруживается ошибка, УУП выдает в ЦП сигнал сегментной ошибки, вызывающий специальное прерывание.



Для хранения базового адреса, размера и атрибута сегмента в УУП введены специальные регистры – ***дескрипторы сегментов***. Число таких регистров определяется количеством используемых сегментов. В дескрипторе содержится 24-разрядный базовый адрес, 16 разрядов размера сегмента, 3 разряда типа, определяющие атрибут сегмента. Кроме них имеются битовые поля: Р – присутствие в основной памяти; А– обращение к сегменту; DPL – уровень привилегий дескриптора; S – бит системного сегмента. S=0 — системный; S=1 — сегмент программы или данных.

***Существуют следующие уровни привилегий и защиты*** (наиболее привилегированным является уровень с меньшим номером):

3-Программы-приложения

2-Служебные программы

1-Служебные программы

0-Ядро ОС

В соответствии с уровнями привилегий и защищенности установлены следующие ***правила доступа для сегментов программ и данных***:

1) Данные из сегмента, имеющий уровень привилегий могут быть выбраны программой такого или более высокого уровня привилегий.

2) Сегмент программ, имеющий уровень защиты PL, может быть вызван программой, имеющий такой или более низкий уровень привилегий.

3) Уровень защиты и привилегий определяется двумя битами, значение которых указывает номер кольца защиты или уровня. Эти биты размещаются в байте доступа дескриптора или в селекторе.

4) Выполнение команды ветвления ограничивается сегментами в пределах одной программы. В командах вызова и перехода допускается переход в сегменты другой программы при условии, что уровень ее привилегий равен текущему уровню.

Следует заметить, что процессор разрешает менее привилегированной процедуре вызвать более привилегированную, но ограничивает доступ разрешенными точками входа. Допустимые точки входа идентифицируются специальными дескрипторами, называемыми шлюзами вызова. Привилегия в шлюзе задается достаточно низкой, что позволяет обращаться в точку входа операционной системы. Размер смещения в шлюзе не учитывается.

**32. Поддержка многозадачности и виртуальной памяти в ЭВМ**

В компьютерах, построенных на основе 16-разрядных процессоров первого поколения, работавших под управлением DOS, одновременно выполнялась только одна задача. Процессоры второго поколения получили возможность поддержки ***многозадачного режима***.

***При переключении на другую задачу осуществляются следующие действия***:

1) Состояние выполняемой задачи помещается в сегмент состояния задачи (ССЗ);

2) Селектор новой задачи – в селекторную часть TR;

3) Дескриптор ССЗ новой задачи в регистр дескрипторов TR;

4) Состояние новой задачи посылается в процессор. Кроме того, в процессор посылается селектор локальной таблицы дескрипторов (ЛТД), с помощью которого можно обратиться к новой ЛТД.

Для реализации ***мультипрограммного режима*** в ОС должны быть предусмотрены программы, выполняющие специальные функции. Одной из них является *программа управления задачей*, которая управляет переходом задачи из одного состояния в другое. Выбор, какая задача будет выполняться следующей, решается *планированием задач*. Замена задач в состоянии выполнения называется *переключением задач*. В этом случае в блок управления задачей необходимо передать содержимое ПС, ФР и РОН, а из блока управления новой задачей считать содержимое этих регистров. Однако для реализации этих действий программно требуется много времени. В целях сокращения времени, основные операции целесообразно выполнять аппаратными средствами.

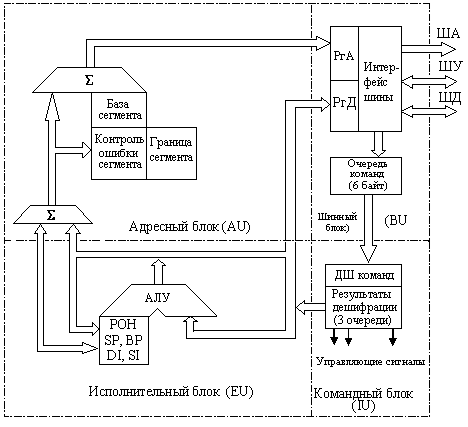
В процессорах, начиная со второго поколения и выше, реализована аппаратная поддержка виртуальной памяти. ***Виртуальная память*** – это способ организации основной памяти большой емкости с помощью внешней памяти. Она позволяет при составлении программы распоряжаться всем пространством адресов, зарезервированных в процессоре. Такие адреса называются виртуальными.

При выполнении программы УУП преобразует виртуальные адреса в физические. Обычно в основной памяти размещается небольшое количество сегментов, к которым в данной части программы осуществляется обращение, т.е. размещаются только необходимые в данное время сегменты. Индикатором нахождения данного сегмента в основной памяти является бит Р дескриптора сегмента. При ссылке на несуществующие сегменты выполнение программы приостанавливается и производится замена содержимого основной памяти. При этом сегменты, которые не предполагается использовать, посылают во внешнюю память, а на их место размещают требуемые сегменты. Затем возобновляется выполнение приостановленной программы. Функция замены осуществляется ОС.

Для отбора сегментов, которые используются наименее часто применяется бит А. При обращении к сегменту биту А присваивается значение 1. ОС в фиксированный момент времени проверяет значение А, и если оно равно 1, производит приращение содержимое строки справочной таблицы и сбрасывает А. Таким образом, с помощью справочной таблицы ОС отбирает сегменты, которые реже всего используются и поэтому в первую очередь подлежат замене.

**33. Архитектура 16-разрядного процессора второго поколения**

Процессор состоит из ***четырех блоков***: адресного AU, шинного BU, исполнительного EU и командного IU, причем все блоки могут работать параллельно.



***Шинный блок*** осуществляет считывание памяти и портов ввода/вывода. ***Адресный блок*** вычисляет все адреса и формирует физические адреса. За счет независимой параллельной работы блоков производительность возрастает в 2-3 раза по сравнению с процессором 8086. Для использования шин с максимальной эффективностью применяется 6-байтовая очередь команд. Результаты дешифрации помещаются в очередь результата. Поэтому ***командный блок*** после выполнения одной команды сразу же переходит к выполнению следующей. При наличии команд переходов очередь сбрасывается.

Процессор может работать в двух режимах: реальном и защищенном. ***Реальный режим*** используется для выполнения программ МП предыдущего поколения и инициализации регистров защищенного режима. ***Защищенный режим*** обеспечивает возможность поддержки виртуальной памяти, реализовывать механизм привилегий с целью защиты памяти и многозадачную работу.

***В защищенном режиме*** в сегментных регистрах располагается не информация о физическом адресе, а селекторы, указывающие на описатели сегментов. С помощью селекторов, расположенных в сегментных регистрах, центральный процессор получает из локальной или глобальной таблиц дескрипторов описатели, характеризующие размещение и длину используемых сегментов.

Если процедура получения дескрипторов будет осуществляться при выполнении каждой команды, то это приведет к существенному снижению быстродействия. Дескрипторы находятся в ОЗУ и при поступлении нового селектора считываются в регистры дескрипторов. Таким образом, информация о физическом адресе сегмента и его длине при загрузке сегментного регистра загружается в так называемый *теневой регистр*, который имеется для каждого сегментного регистра.

***Система команд процессора 80286*** включает все команды 8086-го процессора и дополнена рядом новых команд:

1) сохранение константы в стеке, сохранение и восстановление всех регистров PUSHA, POPA;

2) целочисленное умножение на константу;

3) сдвиги с указанием счетчика в константе;

4) вход и выход из процедур ENTER, LEAVE;

5) контроль диапазона BOUND;

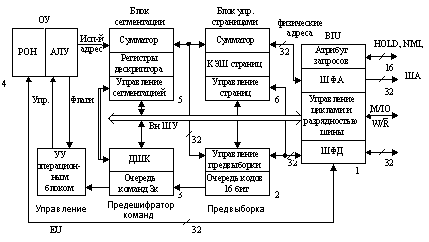
6) ввод/вывод байта (слова) в строку, одиночный INSB, INSW, OUTSB, OUTSW, или по счетчику в регистре СХ, повторение строковых команд по префиксу REP;

7) сброс флага переключения задач CTS;

8) команды управления защитой LGDT, SGDT, LIDT, LMSW, SMSW, доступные в обоих режимах и только в защищенном режиме: LLDT, SLDT, LTR, STR.

**34. Особенности архитектуры 32-разрядных процессоров. Страничная организация памяти.**

Процессор 80386 состоит из следующих ***шести функциональных блоков***, которые работают параллельно: интерфейсный блок; блок предварительной выборки команд; блок предварительной дешифрации команд; блок центрального процессора; блок сегментации; блок управления страницами.



***Центральный процессор*** включает в себя операционное ОУ и управляющее УУ устройства. ОУ состоит из АЛУ и восьми 32–разрядных РОН. Особенностью АЛУ является наличие 64 – разрядного сдвигателя, используемого при быстрых арифметических и циклических сдвигах, умножении и делении. В связи с этим 32–разрядное умножение выполняется менее чем за 1 мкс.

***Подсистема выборки команд*** реализует двухступенчатый алгоритм конвейеризации и состоит из блоков предвыборки кодов и преддешифрации команд. Первый из них осуществляет заполнение очереди команд длиной 16 байт. Во втором блоке производится преддешифрация, определяется тип и формат команд, выделяется поле относительного смещения, содержимое которого поступает в блок сегментации для вычисления линейного адреса. Команды, подготовленные к выполнению, хранятся в очереди команд, куда помещается в среднем 3 команды.

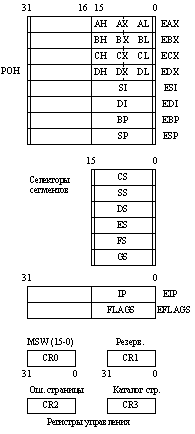
***Диспетчер памяти*** состоит из *блока сегментации* и *блока управления страницами*. Осуществляет двухступенчатое формирование физического адреса ячеек памяти. Наличие диспетчера памяти определяет два режима работы МП:

1. В реальном режиме МП 386 работает как очень быстрый 8086, но при необходимости с расширением разрядности операндов и адресов до 32;

2. В защищенном режиме могут осуществляться переключения и выполнения нескольких задач, предназначенных для режима виртуального МП 8086.

***Блок управления страницами*** действует на более низком уровне, по сравнению с сегментированием. Разбиение на страницы возможно только в защищенном режиме. Каждый сегмент делится на страницы фиксированного размера по 4 Кбайта каждая.

***Блок магистрального интерфейса*** реализует циклы обмена с памятью, сопроцессором, контроллерами. Обмен осуществляется с помощью 32-й ШД, 34-й ША и 16-разрядной шиной управления. Особенность шины данных является возможность динамического изменения ее разрядности. За один цикл могут быть переданы 8, 16 или 32 бита.



Все 16 – ***разрядные регистры*** МП 8086 и 80286 содержатся в 32 – разрядных регистрах МП 80386.

В состав процессора входят также шесть 16–разрядных ***сегментных регистров*** CS, SS, DS, ES, FS и GS, которые содержат значения селекторов сегментов, указывающих на текущие адресуемые сегменты памяти. Селекторы CS указывает текущий сегмент кода команд, SS – стека, а селекторы DS, ES, FS и GS – текущие сегменты данных.

Имя 32–разрядного ***флагового регистра*** EFLAGS. Кроме флагов С, Р, А, Z, S, T, I, D, и O добавлены новые флаги:

1) IOPL – уровень привилегии ввода – вывода;

2) NT – вложенная задача;

3) RF – флаг итога используется в пошаговом режиме при отладке. Если бит установлен, то любая ошибка отладки в следующей команде игнорируется;

4) VM – виртуальный режим 8086.

В процессоре имеется четыре 32 – разрядных ***регистра управления*** CR0, CR1, CR2 и CR3, используемые для фиксации общего состояния процессора:

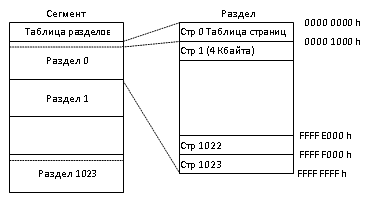
1) CR0 – регистр управления машины. Младшие 16 разрядов называют словом состояния (MSW). Другие биты используются для управления страницами, переключения задач, эмуляции сопроцессора, управление сопроцессором, включение защиты;

2) CR1 – зарезервирован;

3) CR2 – линейный адрес ошибки страницы. Содержит адрес, который вызвал последнюю ошибку страницы;

4) CR3 – базовый адрес каталога страниц.

***Страничная организация памяти*** обеспечивает более эффективное заполнение памяти по сравнению с сегментной, однако, требует дополнительного времени и специальных аппаратных средств для преобразования адресов. Страничная организация имеет место, если в регистре управления CR3 бит 31 имеет значение PG=1.



При страничной организации сегмент разбивается на отдельные ***разделы***, число которых может достигать 1024. Раздел может содержать 1024 ***страницы*** объемом по 4 Кбайта каждая. Границы страниц жестко фиксированы. Начальные адреса страниц имеют значения от 00000000Н до FFFFF000H.

***Линейный адрес*** при страничной организации рассматривается как совокупность трех полей:

Поле ***TABLE*** (А31–А22) указывает относительный адрес таблицы страниц выбираемого раздела в каталоге.

Поле ***PAGE*** (А21–А12) задает относительный адрес требуемой страницы раздела.

Поле ***BYTE*** (А11–А0) содержит относительный адрес выбираемого на странице байта.

Таким образом, страницы, содержащие отдельные фрагменты программ или данных, могут быть рассеяны по разным частям памяти, а их размещение определяется содержанием каталога разделов и таблиц страниц.

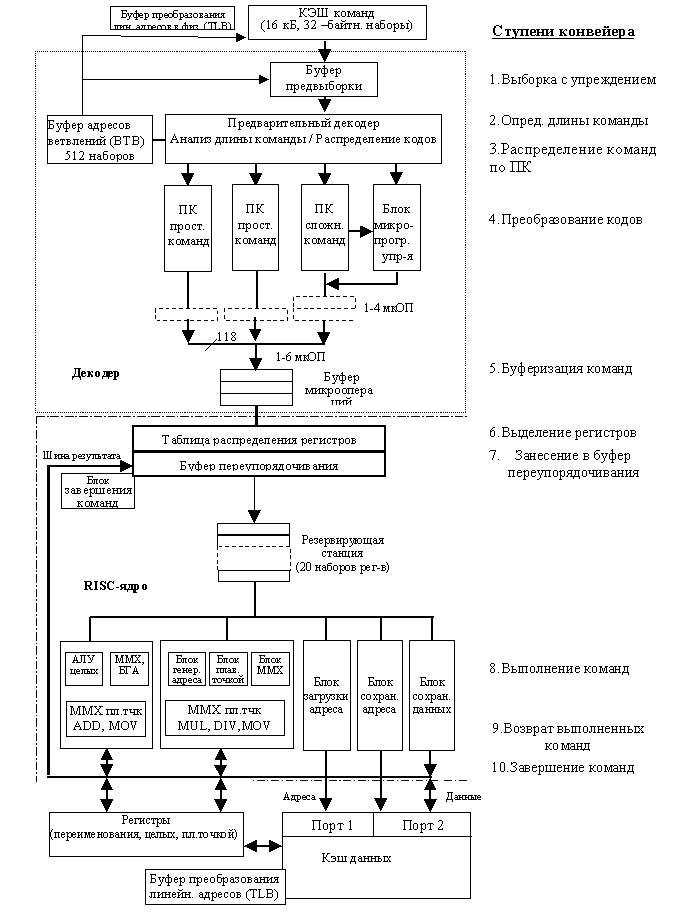
**35. Архитектура суперскалярных процессоров типа Pentium**

В этих процессорах наряду с внутренним КЭШем первого уровня (L1-кэш) размером 64 Кбайт, введен кэш второго уровня (L2-кэш) объемом 512 и более Кбайт.

Ядро процессора Pentium III состоит из двух частей: ***декодера*** и ***RISC–ядра***.

***Декодер*** получает из кэша команд ассемблерные команды, представляющие собой полный комплект команд компьютера (CISC-команды) и преобразует их в микрооперации сокращенного набора команд (RISC-команды). В декодере имеется три преобразователя кодов.

***RISC–ядро*** выполняет полученные микрооперации параллельно несколькими блоками обработки (АЛУ) в произвольном порядке и выдает результат обработки в порядке, заданном исходной ассемблерной программой. Pentium III обладает десятью блоками обработки, которые объединены в 5 групп. За один машинный такт на блоки обработки одновременно может быть подано максимум 5 микроопераций, причем на каждую из групп не более одной микрооперации. Первые две группы содержат по одному ММХ-АЛУ и блок генерации адреса. Оставшиеся три блока обработки служат для выполнения операций обращения к памяти: чтения и записи операндов или промежуточных результатов.

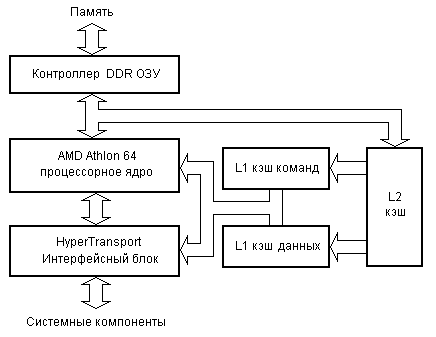


За один такт машинного цикла процессора на три преобразователя кодов ПК могут быть поданы максимум три х86-команды. Задачей ПК является преобразование х86-команд в RISC–микрооперации. Преобразователи кодов отличаются тем, что два из них обрабатывают только «простые» команды и генерируют одну RISC –операцию. Третий преобразователь предназначен для «сложных» команд и на каждую х86-инструкцию генерирует от 1 до 4-х микроопераций. Однако среди сложных команд имеются такие, которые для их выполнения требуют целую микропрограмму. Такие команды преобразуются в блоке микропрограммного управления.

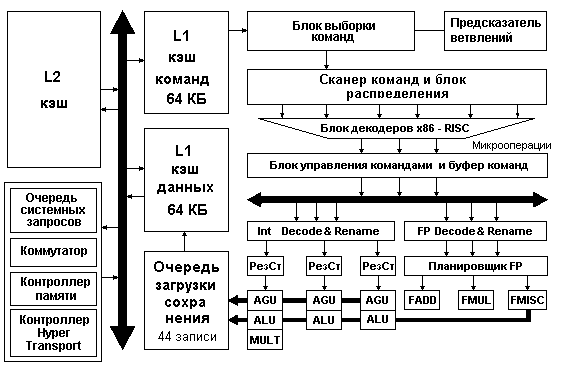
За один такт из трех ПК на RISC–ядро процессора могут быть подано максимум 6 микроопераций. Они предварительно заносятся в 6-словный буфер микроопераций и оттуда подаются в буфер динамически выделяемой памяти.

После того, как по соответствующим статусным битам установлено завершение команды, происходит обратное переименование «внутренний регистр - архитектурный регистр». Затем блок завершения команд восстанавливает результаты выполненных команд в порядке, установленном исходной программой.

**36. Архитектуры 64-разрядных процессоров.**



***Шина HyperTransport*** (HT)— это двунаправленная последовательно/параллельная компьютерная шина, с высокой пропускной способностью и малыми задержками. HyperTransport поддерживает автоматическое определение ширины шины, от 2-х битных линий до 32-х битных. Шина HyperTransport основана на передаче пакетов. HyperTransport может использоваться для передачи служебных сообщений системы, для передачи прерываний, для конфигурирования устройств, подключенных к шине и для передачи данных.

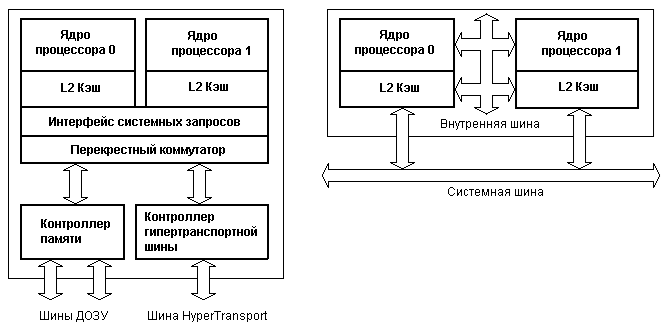


Блок декодеров превращает команды х86 переменной длины в RISC-микрооперации фиксированной длины. Причем за один такт каждый декодер может обработать инструкцию длиной до 16 байт и отправить планировщику на исполнение микрооперации, которые затем упаковываются в группы по три.

В ядре установлены три 8-входовые станции резервирования (РезСт). Все вместе они образуют 24-входовый планировщик заданий. Исполнительные устройства ядра располагают тремя целочисленными блоками ALU, 3-мя блоками генерации адресов и загрузки AGU, тремя блоками с плавающей точкой. Длина целочисленных конвейеров имеет 12 ступеней, FPU конвейеров — 17. Все данные 64-битовые. Арифметические и логические операции, кроме умножения, выполняются за один такт. На 32-битное умножение расходуется три такта и пять тактов на 64-битное умножение.

**37. Архитектура многоядерных процессоров.**

Существуют две архитектуры двуядерных процессоров: фирмы ***AMD (Athlon 64 X2)*** и ***Intel (Pentium D)***.

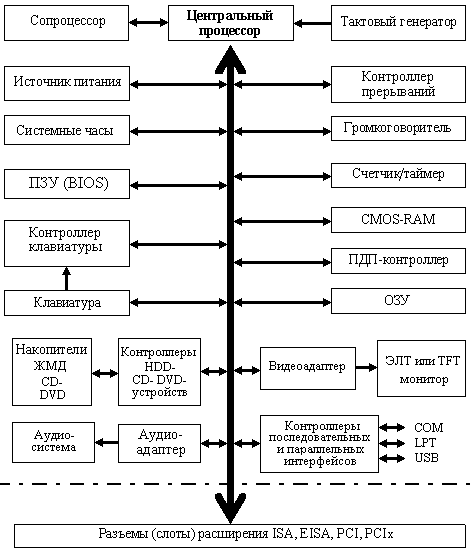


При всей несхожести архитектур ядер, оба процессора имеют одну общую черту – одинаково реализованную двуядерность: каждое ядро имеет собственный кэш второго уровня.

В процессоре ***Pentium D*** для связи ядер используется внутренняя шина. Такое решение не самое удобное в том случае, когда одному ядру надо получить данные, содержащиеся в КЭШе другого ядра. Ведь нередко ядра работают над одной и той же задачей, просто выполняя разные её части. Отсюда неминуемые задержки в доступе к информации, а также возможен простой одного из ядер по той причине, что его кэш полностью заполнен данными, которые всё ещё нужны второму ядру. Логичное решение этой проблемы – использование общей кэш-памяти. И именно таким образом работает двуядерный процессор, получивший название Core Duo.

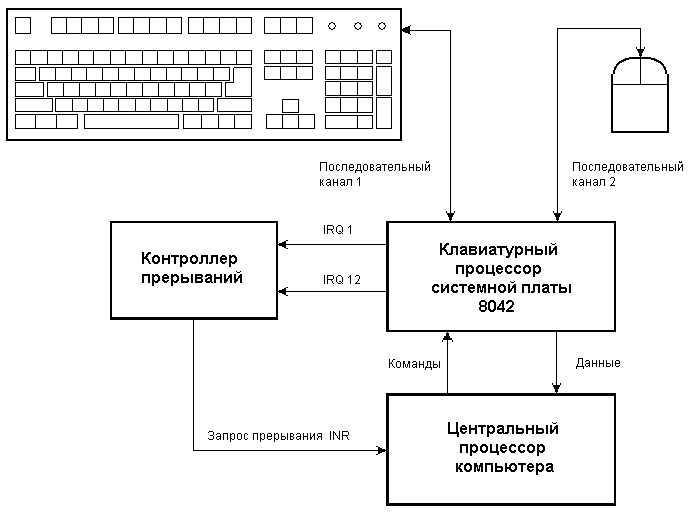
Каждое из ядер ***Athlon 64 X2*** также обладает собственным набором исполнительных устройств и выделенной кэш-памятью второго уровня; контроллер памяти и контроллер шины HyperTransport - общие. А вот взаимодействие каждого ядра с разделяемыми ресурсами происходит посредством специального перекрестного коммутатора и интерфейса системных запросов, в котором формируется очередь системных запросов. На этом же уровне организовано и взаимодействие ядер, благодаря чему снимается дополнительная нагрузка на системную шину и шину памяти.

**38. Обобщенная структурная схема компьютера. Состав и устройство системной платы.**



**39. Клавиатура компьютера и ее взаимодействие с процессором.**

Каждый раз, когда нажимается или отпускается одна из клавиш, схема клавиатуры генерирует однобайтовое число, называемое скен-кодом, которое идентифицирует клавишу. Клавиатура выдает различные скен-коды при нажатии и отпускании клавиши. При нажатии байт скен-кода содержит число в диапазоне от 1 до 83. При отпускании генерируется скен-код на 128 (80H) больше, чем скен-код при нажатии клавиши. Например, при нажатии клавиши Z скен-код 44, а при отпускании 172 = 44 + 128.



Нажатие или отжатие клавиши сообщается ROM-BIOS посредством прерывания INT9, которое вызывает прикладную программу обработки прерываний. При этом читается состояние порта 96 (60H), через который осуществляется связь клавиатуры с BIOS.

Прикладная программа получает скен-код и переводит его в 2-х байтный код. Младший байт этого кода содержит ASCII-код клавиши, а старший – скен-код клавиши. Специальные клавиши F1 – F10, Ins, Del и др., а также клавиши дополнительной цифровой клавиатуры имеют в младшем байте 0, а в старшем байте скен-код.

**40 Видеосистема компьютера. Принцип функционирования ЖК-монитора.**

Видеосистема компьютера состоит из ***видеоадаптера*** и ***монитора***. ***Видеоадаптер*** – это устройство, осуществляющее сопряжение устройства отображения информации – дисплея с компьютером. ***Дисплей*** осуществляет преобразование электрических сигналов, поступающих из видеоадаптера в видимое изображение. Основным блоком монитора является электронно-оптическое устройство – электронно-лучевая трубка (ЭЛТ), либо плоские экраны на жидкокристаллических элементах.

Все мониторы используют ***растровый принцип*** формирования изображения, в соответствии с которым изображение на экране образуется группой близко расположенных горизонтальных линий или строк, называемой растром. Электронный луч ЭЛТ последовательно проходит каждую строку слева направо, начиная с левого верхнего угла. Когда луч проходит по строке, цвет и яркость каждого пикселя изменяется и весь растр представляется как связанное изображение. После прохождения очередной строки слева направо луч смещается вниз к началу следующей строки, и так происходит до тех пор, пока луч полностью не сформирует растр.

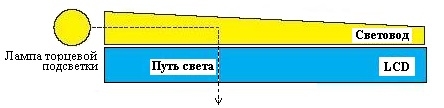
В настоящее время наиболее широко применяются мониторы с использованием ***жидкокристаллических панелей***, которые вытеснили мониторы на основе электронно-лучевых трубок.

В производстве LCD известны две технологии: с использованием ***пассивных матриц (STN)*** и ***активных матриц (TFT)***.

***STN-матрица***, состоит из ЖК-элементов с изменяемой прозрачностью, в которых используется твист-эффект. Суть этого эффекта состоит в изменении угла вращения плоскости поляризации проходящего света под воздействием электрического поля в жидких кристаллах.

В ***активной матрице*** ***на тонкопленочных транзисторах — TFT***, каждый пиксель управляется отдельным транзистором. Практически все современные ЖК-мониторы используют панели на тонкопленочных транзисторах, обеспечивающих более яркое и четкое изображение большего размера. По сравнению с пассивной матрицей, TFT-мониторы имеет более высокую контрастность, насыщенность, меньшее время переключения.

***Принцип действия и многослойная структура всех LCD TFT-дисплеев*** примерно одинаковы. Свет от лампы подсветки проходит через первый поляризатор и попадает в слой жидких кристаллов, над которыми размещены пластины миниатюрных конденсаторов, управляемых тонкопленочными транзисторами.



Транзистор с конденсатором создает электрическое поле, которое формирует ориентацию жидких кристаллов. Пройдя такую структуру, свет меняет свою поляризацию и будет или полностью поглощен вторым поляризационным фильтром (экран становится черным), или не будет поглощаться (экран - белый), или поглощение будет частичным (одна из градаций яркости цвета).

Цвет изображения определяют цветовые фильтры. Каждый пиксель матрицы состоит из трех субпикселей - красного, зеленого и голубого. А различные цвета получаются в результате изменения величины соответствующего электрического потенциала на пластинах конденсатора.

**41. Аудиосистема компьютера, состав и принцип действия. Синтез звука.**

Звуковая система компьютера состоит из ***звукового адаптера*** (звуковой карты) и ***электроакустических преобразователях звуковых колебаний*** (микрофона и звуковых колонок).

***Звуковые карты выполняют следующие функции:***

1) дискретизацию аналоговых сигналов с частотами 11,025 кГц, 22,05 и 44,1 кГц. Первая частота относится к 8 битовым картам, другие – к 16 битовым;

2) 8- или 16– битовое квантование, кодирование и декодирование с использованием линейной импульсно-кодовой модуляции (ИКМ);

3) одновременно производить запись и воспроизведение звуковой информации;

4) ввод сигналов через монофонический микрофон с автоматическим регулированием уровня входного сигнала;

5) ввод и вывод аудио сигналов через линейный вход/выход;

6) микширование сигналов от нескольких источников и выдача суммарного сигнала в выходной канал. В качестве источников используются:

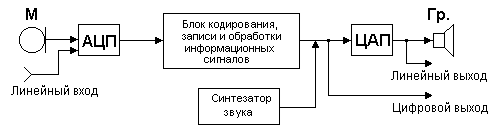
а) ЦАП;

б) музыкальный синтезатор;

в) внешний источник, подключенный к линейному входу.

7) управление уровнем суммарного сигнала и сигнала каждого из каналов в отдельности;

8) синтез звуковых колебаний с использованием частотной модуляции (FM) и волновых таблиц (WT).



Микрофон (М) осуществляет преобразование акустических колебаний в электрический, а громкоговоритель (Гр.) преобразование электрических колебаний в акустические. Входной сигнал с микрофона усиливается, а с линейного входа подается непосредственно на аналого-цифровой преобразователь.

***Синтез звуковых сигналов*** предназначен для генерации звуков музыкальных инструментов, соответствующие определенным нотам, а также создавать "немузыкальные" звуки: шум ветра, выстрела и т.п.

***Существует несколько методов синтеза***:

1. ***Аддитивный синтез***. Звук в процессе синтеза формируется путем сложения нескольких исходных звуковых волн.

2. Второй метод является разновидностью ***нелинейного синтеза***. Для получения одного музыкального звука используется сигнал одного генератора. Гармоническую окраску получают в результате нелинейных искажений исходного сигнала. Для этого синусоидальный сигнал, формируемый генератором, управляемым кодом (ГУК) с амплитудой A1 и частотой f1 пропускают через нелинейный элемент с некоторой характеристикой К(х). Зная амплитуду сигнала A1 и вид характеристики К(х), можно вычислить спектр сигнала на выходе.

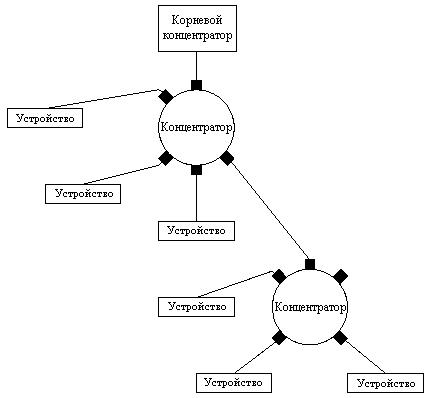
3. Следующим широко распространенным методом является синтез на основе ***частотной модуляции***. При частотной модуляции осуществляется изменение частоты f несущего колебания U(t) = Asin(2πf + φ) по закону модулирующего колебания x(t).

**42. Последовательный интерфейс USB. Общая характеристика, сигналы и форматы данных, виды транзакций.**

***Интерфейс USB*** (Universal Serial Bus) — это новый интерфейс, который является универсальным последовательным интерфейсом. Используется в компьютерах с 1999 года.

***Архитектурой USB*** предусматривается топология "звезда". Система должна состоять из одного ведущего концентратора с хост-контроллером, управляемым операционной системой, некоторого количества концентраторов и узлов (периферийных устройств). Хост-контроллер входит в состав системного блока компьютера. На шине USB допускается наличие только одного хоста. На системном блоке персонального компьютера может располагаться несколько хостов, каждый из которых управляет собственной шиной USB.

***Концентратор*** (Hub, хаб) — ключевой элемент системы в архитектуре USB. Он служит для создания дополнительных портов.



Устройства подключаются к концентраторам. Всего устройств может быть до 127, концентратор также считается устройством. На практике такое количество устройств вряд ли используется, а топология чаще остается линейной. В устройства, подключаемые к шине USB, часто встраивается концентратор с единственным выходным портом, что позволяет объединять устройства в цепочку. При построении цепочки следует иметь в виду, что наиболее скоростные устройства следует подключать ближе к корневому концентратору, а наименее скоростные – в конец цепочки. Это обеспечит приоритетное обслуживание высокоскоростных устройств.

Одним из преимуществ USB является возможность "горячего" подключения без перезагрузки системы. При обнаружении на шине нового устройства концентратор оповещает об этом корневой концентратор. Затем система опрашивает вновь подключенное устройство о его возможностях и потребностях и конфигурирует его. Вдобавок при этом загружаются необходимые драйверы.

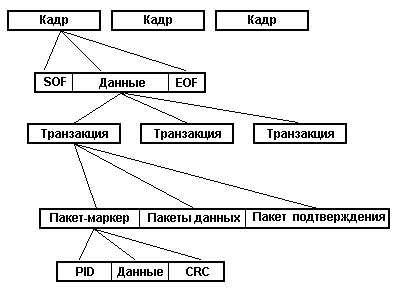
***Архитектура USB допускает четыре базовых типа передачи данных***:

1) Управляющие посылки до 64 байт, используются для конфигурирования устройств.

2) Сплошная передача небольших пакетов без жестких требований ко времени доставки (например, от сканера или к принтеру). Длина пакетов от 8 до 64 байт.

3) Прерывания – для передачи типа вводимых символов или координат.

4) Изохронная передача – непрерывная передача мультимедийных данных, без повтора в случае ошибок.



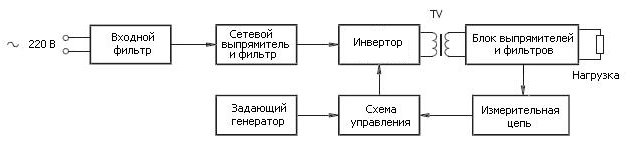
Любой обмен данными по шине USB инициируется хост-контроллером. Хост-контроллер формирует кадры из наиболее приоритетных данных. Каждый кадр начинается стартовой посылкой SOF (Start of Frame) и завершается посылкой EOF (End of Frame).

Любая передача данных состоит из одной или нескольких транзакций.

Все транзакции по USB интерфейсу состоят из трех пакетов. Каждая транзакция начинается по инициативе контроллера, который посылает ***пакет-маркер***, описывающий тип и направление передачи, адрес устройства USB и номер конечной точки. В каждой транзакции возможен обмен только между адресуемым устройством и ЭВМ. Адресуемое маркером устройство USB распознает свой адрес и подготавливается к обмену. Источник данных передает ***пакет данных*** или уведомление об отсутствии данных, предназначенных для передачи. После успешного приема пакета получатель данных посылает ***пакет подтверждения***. Для защиты передаваемой информации от ошибок используется помехозащищенное кодирование циклическим кодом. В случае обнаружения ошибок контроллер автоматически производит повторную передачу (до трех раз). Если повторы безуспешны, сообщение об ошибке передается клиентскому ПО для программной обработки.

**43. Особенности построения источников питания компьютеров.**

В настоящее время в компьютерах применяются источники питания только импульсного типа, преимуществами которых являются малые габариты при большой мощности и независимость от изменения частоты и номинального напряжения сети. Принцип работы импульсных блоков питания заключается в выпрямлении сетевого напряжения с последующим преобразованием его в переменное высокочастотное напряжение прямоугольной формы, которое понижается трансформатором до нужных значений, выпрямляется, фильтруется и поддерживается в заданных пределах.



***Входной фильтр*** предназначен для устранения кратковременных скачков питающего напряжения и импульсных помех.

***Сетевой выпрямитель*** преобразует переменное напряжение электросети в постоянное пульсирующее напряжение.

***Инвертор*** преобразует постоянное напряжение, полученное от выпрямителя в последовательность высокочастотных импульсов прямоугольной формы.

***Схема управления*** формирует импульсы управления мощным инвертором, обеспечивая стабилизацию выходных напряжений, защиту от короткого замыкания на выходе и пр.