

## Основные понятия.

Существует множество различных формулировок понятия ЭВМ от достаточно простых и понятных до чрезмерно вычурных, которые, однако, схожи по своей сути.

По Э.Таненбауму:

**Цифровой компьютер** – машина, которая может решать задачи, выполняя данные ей команды. Последовательность команд, описывающих решение определённой задачи, называется программой.

По Б. Я. Цилькеру:

**ЭВМ** – устройство, которое принимает данные, обрабатывает их в соответствии с хранимой программой, генерирует результаты и обычно состоит из блоков ввода/вывода, памяти, арифметики, логики и управления.

Некорректное определение:

**ЭВМ** – функциональный блок, способный выполнять реальные вычисления, включающие множественные арифметические и логические операции без участия человека в этих процессах.

По Новикову, Майорову (наилучший вариант):

**ЭВМ** - искусственная (инженерная), предназначенная для вычислений на основе алгоритмов.

Принципы построения ЭВМ, с одной стороны, определены назначением ЭВМ и, с другой стороны, элементной базой (набором элементов, которые используются для создания ЭВМ).

Основным назначением ЭВМ является выполнение вычислений на основе алгоритмов, и поэтому свойства алгоритмов предопределяют принципы построения ЭВМ или, точнее, ее архитектуру (организацию).

## Понятие архитектуры и организации ЭВМ.

В компьютерной литературе существует большое количество разнообразных трактовок понятия архитектура ЭВМ, которые отличаются как по смыслу, так и по разнообразию элементов, включаемых в понятие архитектура. (См. электронный конспект). По мнению специалистов, впервые термин архитектура компьютера (Computer Architecture) был употреблен фирмой IBM (International Business Machines) при разработке семейства машин IBM 360 в середине 60 – ых годов.

## Обобщенное понятие архитектуры.

Под архитектурой ЭВМ обычно понимается ее представление и описание возможностей с точки зрения пользователя, разрабатывающего программу на машинно-ориентированном языке.

Архитектура, как правило, отображает те аспекты структуры и принципы функционирования ЭВМ, которые являются видимыми для пользователя и, следовательно, для разрабатываемых им программ.

Термины архитектура ЭВМ и организация ЭВМ во многом кажутся подобными. В связи с этим, некоторые специалисты используют их как синонимы. Сторонником этого является Э.Таненбаум:

*“Архитектура связана с аспектами, которые видны программисту. Например, сведения о том, сколько памяти можно использовать при написании программы – часть*

*архитектуры, а аспекты разработки (например, какая технология используется при создании памяти) не является частью архитектуры. Изучение того, как разрабатываются те части компьютерной системы, которые видны программистам, называется изучением компьютерной архитектуры. Термины компьютерная архитектура и компьютерная организация означают, в сущности, одно и то же...”.*

Однако существует и другой подход, при котором эти понятия если не противопоставляются, то, по крайней мере, отличаются. Это различие состоит в том, что если понятие архитектура ЭВМ определяет возможности ЭВМ, то понятие организация ЭВМ определяет, как эти возможности реализованы в рамках конкретных моделей ЭВМ. Одним из сторонников подобного подхода является я У. Столлингс:

*“При описании компьютерных систем принято различать их структурную организацию и архитектуру. Хотя точное определение этим понятиям дать довольно трудно, среди специалистов существует общепринятое мнение о смысле этих понятий и различий между ними*

*Термин архитектура компьютерной системы (компьютера) относится к тем характеристикам системы, которые доступны извне, то есть со стороны программы или, с другой точки зрения, оказывает непосредственное влияние на логику выполнения программ.*

*Под термином структурная организация компьютерной системы подразумевается совокупность операционных блоков (устройств) и их взаимосвязей, обеспечивающих реализацию спецификаций, заданных архитектурой компьютера.*

*В число характеристик архитектуры входят набор машинных команд, формат разрядной сетки для представления данных разных типов, механизм обращения к средствам ввода/вывода и метод адресации памяти.*

*Характеристики структурной организации включают скрытые от программиста детали аппаратной реализации системы: управляющие сигналы, аппаратный интерфейс между компьютером и периферийным оборудованием, технологию функционирования памяти”.*

Существенное отличие между архитектурой и структурной организацией ЭВМ проявляется для моделей компьютеров, принадлежащих к одному семейству. Все они, как правило, обладают единой архитектурой, с некоторым расширением от младшим моделям к старшим, при обязательном условии совместимости, но разной структурной организацией, в результате чего старшие модели обладают большей производительностью и, соответственно, стоимостью по сравнению с младшими.

В принципе, понятие организация (не только применительно к ЭВМ) обычно используется в двух аспектах: структурная и функциональная.

**Структурная организация** ЭВМ определяет, как устроена ЭВМ, т. е. задает ее структуру на уровне устройств, входящих в состав ЭВМ, и организацию связей между этими устройствами на уровне аппаратных интерфейсов.

**Функциональная организация** определяет, в свою очередь, принципы функционирования ЭВМ, т. е. как в ней протекают вычислительные процессы при решении различных задач.

В некотором смысле существует аналогия между понятиями архитектура ЭВМ и функциональная организация. В связи с тем, что возможности ЭВМ постоянно развиваются и совершенствуются, то, и понятие архитектура ЭВМ включает в себя все

большее число аспектов, отражающих принципы построения и функционирования ЭВМ.

## Виды архитектуры ЭВМ и их составные элементы.

Одним из подходов к уровням представления архитектуры ЭВМ является её разделение на 2 уровня (2 класса):

- программная архитектура, которая включает в себя аспекты, видимые программистам и, соответственно, программам
- аппаратная архитектура, включающая аспекты, невидимые для программиста.

В этом смысле понятие аппаратной архитектуры и структурной организации ЭВМ можно рассматривать как синонимы.

В связи с принятым во всем мире деления программистов на прикладных и системных, программную архитектуру также целесообразно разделить на 2 вида: прикладную и системную. Основные элементы архитектуры и структурной организации представлены на рис 1.

### Краткое представление основных элементов прикладной архитектуры компьютеров.

#### Типы, форматы и способы представления данных, аппаратно поддерживаемых в ЭВМ.

Для многих компьютеров основной задачей является обработка данных различного типа, которые внутри компьютера должны представляться определённых форматах.

Под форматом данных обычно понимают внутреннее представление данных, в первую очередь, разрядность и назначение битов. Например, для знаковых целых чисел крайний левый бит формата отводится для представления знаков. Для обозначения форматов стандартной длины принято использовать следующие наименования:

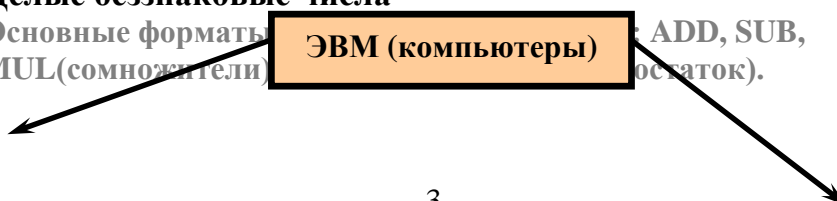
- Байт (B – Byte) – 8 бит
- Слово (W – Word) – 16 бит
- Двойное слово (DW – Double Word) – 32 бита
- Учетверенное слово (QW – Quadro Word) – 64 бита

Ключевым понятием в отношении данных, представляемых в ЭВМ, является наличие или отсутствие аппаратной поддержки для конкретного типа и формата данных. Под аппаратной поддержкой подразумевается наличие в системе команд ЭВМ некоторого множества машинных команд, предназначенных для обработки данных определенного типа, представленных в соответствующих форматах.

Применительно к базовой модели Intel 8086 аппаратной поддержкой обладают следующие типы и форматы данных:

- Целые знаковые числа  
Основные форматы: B, W. Примеры команд: ADD, SUB, IMUL(сомножители), IDIV(делитель, частное, остаток).  
Неосновной формат: DW. Примеры команд: IMUL(произведение), IDIV(делимое).

- Целые беззнаковые числа  
Основные форматы: B, W. Примеры команд: ADD, SUB, MUL(сомножители), DIV(делитель, частное, остаток).



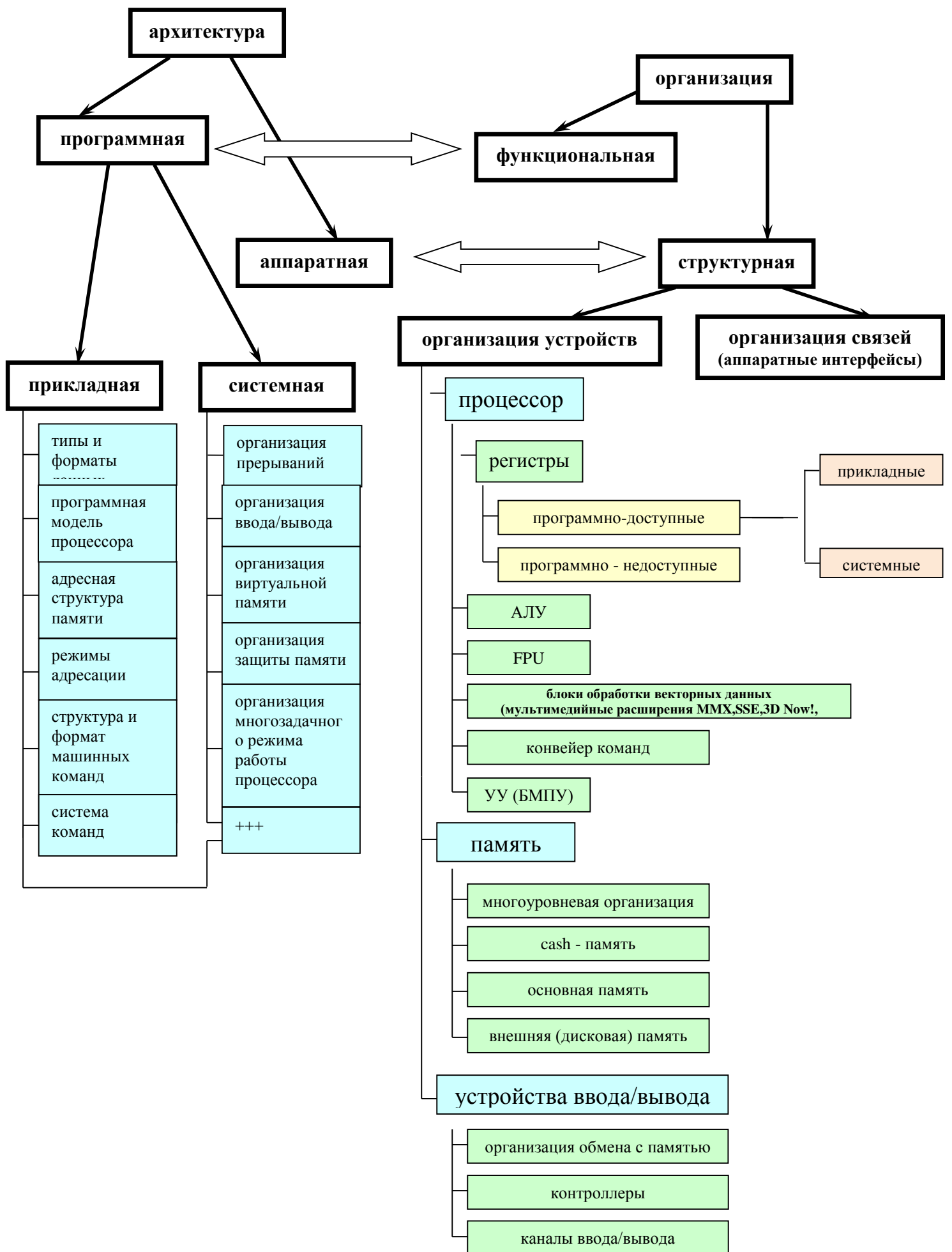


Рис.1.Обобщенное представление архитектуры, организации ЭВМ и их элементов.

Неосновной формат: DW. Примеры команд:  
MUL(результат),DIV(делимое).

- **Числа с плавающей запятой**  
Короткий формат: 32 бита  
Длинный формат: 64 бита  
Расширенный формат: 80 бит  
Примеры команд: FADD, FSUB
- **Десятичные числа**  
Упакованный формат – BCD:В. Примеры команд: DAA, DAS  
Неупакованный формат – ASCII. Примеры команд: AAA, AAS, AAM, AAD
- **Логические значения**  
Основной формат:В, W. Примеры команд: AND, OR, XOR, TEST, NOT
- **Символьные данные**  
Основной формат:В. Примеры команд: MOVS, LODS, STOS, CMPS, SCAS  
XLAT.

**Для числовых данных необходимо представлять диапазон и точность.**

### Программная модель (регистровая структура) процессора.

Регистровая структура процессора включает в себя набор программно доступных регистров. В соответствии с этим, этот аспект достаточно часто называют программной моделью процессора. Фактически, рассмотрение этого аспекта связано с перечислением программно доступных регистров и описанием их назначения для использования.

Программная доступность регистров означает, что со стороны программы, с использованием некоторых машинных команд, может осуществляться обращение к этому регистру, либо по чтению, либо по записи.

Важной характеристикой регистра является его разрядность. Как правило, именно разрядность внутренних регистров и определяет разрядность самого процессора (Intel 8086 – 16 –разрядный). Разрядность регистра определяет количество бит информации, которое можно представить (хранить) в данном регистре.

Любой процессор в современной ЭВМ содержит собственную внутреннюю память для хранения, в основном, операндов и адресов, а также результатов выполняемых операций. Эту внутреннюю память называют регистровой памятью, или сверхоперативной памятью, чтобы подчеркнуть значительно большее её быстродействие по сравнению с оперативной (основной) памятью. Быстродействие памяти определяется так называемым временем доступа (обращения). Время доступа к регистровой памяти – единицы наносекунд, а к оперативной памяти - десятки.

В состав регистровой памяти любого процессора входят как программно доступные, так и программно недоступные регистры. Типичным примером программно недоступного регистра может служить регистр команд, в который производится выборка машинной команды из памяти перед её выполнением. Программно доступные регистры, в свою очередь разделяются на прикладные (доступные как прикладным, так и системным программам) и системные (доступные только системным программам). Системные регистры появляются в процессорах семейства Intel 80X86, начиная с модели i286, в которой впервые был введён защищённый режим.

В старших моделях процессора Intel используются следующие группы системных регистров.

- Управляющие регистры CR – Control Registers
- Регистры управления памятью
- Регистры отладки DR – Debug Registers
- Регистры проверки TR – Test Registers (аппаратная поддержка механизмов тестирования внутренних блоков)

Программная модель базового процессора Intel 8086 включает в себя 14 шестнадцатиразрядных регистров, 8 из них входят в состав Регистров Общего Назначения (РОН) – General Purpose Registers (GPR). Группа этих регистров предназначена как для хранения операндов (результатов), так и адресов.

В принципе, существует два диаметрально противоположных подхода к использованию регистров процессора:

- 1) Полная специализация регистров, когда каждый регистр используется только по одному специальному назначению.
- 2) Полная универсализация, когда каждый регистр можно использовать по любому назначению.

В процессорах фирмы Intel используется промежуточный подход. Это означает, что, в принципе, за каждым регистром закреплена его определенная функциональная специализация. Например, функционально специализированный регистр CX – Counter Register. Его специализация проявляется при выполнении команд циклов (LOOP), команд сдвигов (SAR) или команд обработки строк (MOVS, CMPS). Эта специализация отражается в наименовании регистра. Однако наличие специализации у регистров не мешает их использованию для других целей (не по прямому назначению). Например, в регистр CX может быть помещён операнд для какой – либо арифметической команды.

Использование регистров по их прямому назначению позволяет существенно сократить длину машинного (объектного) кода программы за счёт использования неявной адресации (операнд или адрес не задаётся, а подразумевается по умолчанию).

Следующая группа регистров программной модели – 4 Сегментных Регистра – Segment Registers (SR). С использованием этих регистров реализуется простейшая модель сегментирования памяти. В сегментных регистрах содержатся базовые (начальные) адреса 4 сегментов памяти по наименованию регистров:

- Code Segment (сегмент кода)
- Stack Segment(сегмент стека)
- Data Segment(сегмент данных)
- Extra Segment(дополнительный сегмент)

Модель памяти в процессоре Intel 8086 предполагает формирование физического адреса как суммы двух компонент: базовый адрес сегмента и Offset (внутрисегментное смещение). При суммировании компонент первая составляющая сдвигается влево на 4 разряда, в итоге получается двадцатиразрядная сумма, представляющая собой физический адрес, который и выставляется на внешнюю шину адреса. В процессоре Intel 8086 используется мультиплексированная шина адрес/данные, т. е. по одним и тем же проводам, но в разные моменты времени передаются адрес и данные. В соответствии с принципами формирования физического адреса, в сегментных регистрах находятся старшие 16 – ти разрядные компоненты 20 – ти разрядных базовых адресов сегментов.

В соответствии с этим подходом, границы сегментов физической памяти выравниваются на 16 – ти байтную границу (4 младших нуля в адресе), которую принято называть границей параграфа.

Выбор внутрисегментного смещения (Offset) определяется видом обращения к памяти. Например, при выборке команды в качестве компонент адреса используется пара CS – IP.

### Регистр IP (Instruction Pointer).

Любой процессор, входящий в состав компьютера с неймановской архитектурой, в качестве обязательного элемента, содержит так называемый счётчик команд, который иначе называется программным счётчиком PC (Program Counter) или указатель команд.

Содержимое IP используется процессором при выборке очередной команды из памяти. В момент выполнения машинной команды, содержимое IP определяет адрес следующей команды.

Понятие “следующая”, в отношении команды, характеризует последовательность команд не столько в смысле их выполнения, сколько в смысле их положения в памяти. Так, например, при выполнении команд перехода, вызовов, возвратов, содержимое IP изменяется на значение адреса перехода, вызова или возврата.

### Регистр FR (Flag Register).

Содержимое этого регистра используется, во-первых, для фиксации так называемых признаков результата (арифметические флаги), а так же для управления режимом работы процессора (флаги управления). Содержимое арифметических флагов используется при выполнении команд условных переходов. Значения арифметических флагов изменяются при выполнении большинства арифметических и логических команд, к флагам управления относятся:

- IF (Interrupt Flag) – Флаг Прерывания
- TF (Trace Flag) – Флаг Трассировки
- DF (Direction Flag) – Флаг Направления

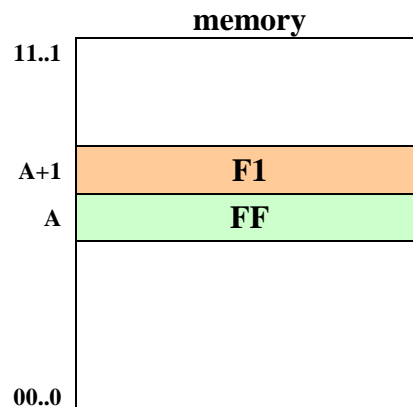
### Адресная структура основной памяти и принципы размещения информации в ней. Принципы формирования физического адреса.

Этот аспект определяет память с позиции программиста, разрабатывающего программу на ассемблере (*память, как она видна программисту*).

В подавляющем большинстве адресация памяти осуществляется на уровне байт. В соответствии с этим для программы память представляется в виде массивов последовательно адресуемых байтов. При размещении единицы информации, кратных байту, например Word или Double Word, в памяти компьютера, адрес единицы информации определяется адресом одного из байтов, либо старшего, либо младшего. В зависимости от этого, в англоязычной литературе система адресации, начиная от старшего байта, обозначают термином “Big Endian”, а с младшего байта “Little Endian”. Сторонниками первого принципа является фирма IBM и Motorola, сторонниками второго - Intel и DEC. Например: -15 в формате Word

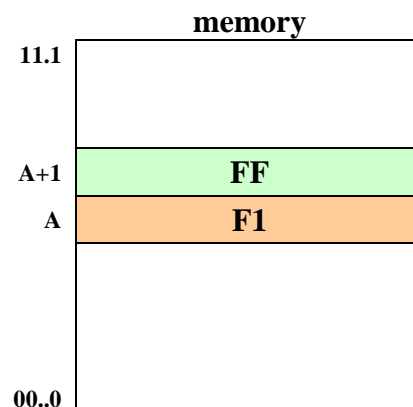
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
F				F				F				1			

**Big Endian –**  
байт с большей значимостью по меньшему адресу



1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
F				F				F				1			

**Little Endian –**  
байт с большей значимостью по большему адресу



При размещении единицы информации фиксированной длины в памяти компьютера достаточно широко используется понятие целочисленной границы. В некоторых моделях компьютеров несоблюдение целочисленной границы при размещении данных, а иногда даже и команд, приводит к прерыванию программы.

**Принцип целочисленной границы гласит:**

Адрес любой фиксированной единицы информации, содержащий  $2^k$  байт, должен быть кратен  $2^k$ .

Это означает, что  $k$  младших разрядов адреса должны быть равны нулю. В процессорах семейства Intel 80X86 проверка соблюдения целочисленной границы только в отношении данных аппаратно поддерживается, начиная с модели i486.

При одном обращении к памяти, количество байт, перемещаемых из процессора в память или обратно, соответствует ширине (разрядности) шины данных. При этом 2 или 4 байта, участвующих в обмене являются структурой, выровненной на целочисленную границу (естественное требование аппаратного интерфейса памяти). В соответствии с этим, для уменьшения числа обращений к памяти необходимо соблюдать целочисленную границу. Реализация этого правила учтена в большинстве компиляторах.

**Режимы адресации.**

При выполнении любой машинной команды, производя заданную операцию, будь это сложение, вычитание, конъюнкция, дизъюнкция и т.д., данные, над которыми производятся операции и называемые операндами, задаются своими адресами. В соответствии с этим, единственным идентификатором данных (операндов) и



результатов внутри ЭВМ является их адрес. С помощью адреса определяется местоположение операндов в памяти компьютера (основной или регистровой).

Под режимом адресации принято понимать способ формирования так называемого исполнительного адреса операнда (или результата) на основе информации, находящейся в адресной части команды.

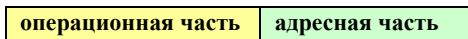
Исполнительный адрес достаточно часто называют программным адресом, т.к. он формируется программой. В терминологии фирмы Intel, исполнительный (программный) адрес называется эффективным адресом Effective Address (EA).

В подавляющем большинстве случаев исполнительный адрес не является физическим. Под физическим адресом принято понимать адрес, по которому производится фактическое обращение к памяти. Именно физический адрес выставляется процессором на внешнюю шину адреса.

Получение физического адреса на основе исполнительного сопровождается преобразованием последнего с использованием в общем случае механизмов сегментации и страничного преобразования. Применительно к процессору фирмы Intel, работающему в так называемом защищенном режиме, в котором осуществляется поддержка виртуальной организации памяти, как на уровне сегментов, так и на уровне страниц, преобразование эффективного адреса в физический осуществляется по следующей схеме:



Структура машинной команды в общем случае состоит из двух основных частей: операционной и адресной:



Операционная часть задаёт тип выполняемой операции и обычно называется кодом операции, Operation Code (OpC).

Адресная часть задаёт адреса операндов, участвующих в операции, а так же адрес результата.

В зависимости от числа операндов, задаваемых в адресной части команды, машинные команды разделяются на трёхадресные, двухадресные, одноадресные и безадресные (нольадресные). В процессорах семейства Intel 80X86 используются безадресные, одно – и двухадресные команды. Использование безадресных команд (однобайтные команды, содержащие только код операции) может быть связано с двумя аспектами:

- использование неявной адресации операндов (их адреса не задаются в команде, но подразумеваются по умолчанию)
- для выполнения команды операндов не требуется (NOP, HLT)

В терминологии фирмы Intel операнды двухадресной команды называются источником Source (SRC) и приемником Destination (DST). Результат операции помещается по адресу операнда – приемника.

Основными режимами адресации, используемыми в ЭВМ, принято считать:

## 1. Прямая

- памяти
  - регистровая
2. Косвенная
    - памяти
    - регистровая
  3. Относительная
    - базовая
    - индексная
    - базово - индексная без смещения
    - базово – индексная со смещением
    - относительно текущего счетчика команд
  4. Непосредственная
  5. Неявная.

В максимальном случае для процессора 8086 относительный адрес может состоять из трёх компонент  $EA = Base + Index + Displacement (Disp)$

В старших моделях процессоров Intel реализовано дальнейшее развитие относительной адресации в виде базово – индексной адресации с масштабированием. При использовании этого режима индексная компонента EA умножается на соответствующий масштаб.  $EA = Base + Index * 2^{Scale} + Disp$ , Scale = 0,1,2,3..

#### Структура и форматы машинных команд.

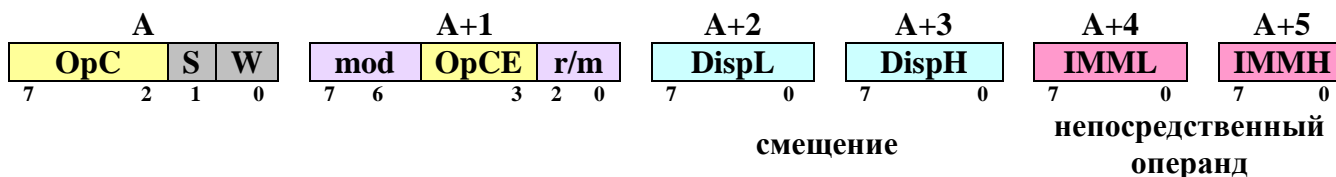
Структура машинной команды задаёт основные части (поля) машинной команды и определяет их назначение. В свою очередь, формат машинной команды определяет разрядность, как всей команды, так и отдельных её полей. Под полем принято понимать совокупность последовательных битов формата (команды или данных), которые имеют определённое общее назначение. Примерами отдельных полей машинных команд могут являться

- поле кода операции (OpC);
- адрес базового или индексного регистров (Base, Index);
- смещение(Disp);
- непосредственный операнд (IMM);

Длина машинной команды без учета возможных префиксных байтов составляет от 1 до 6 байт. Префиксные байты имеют специальный код, отличный от кода операции и оказывают то или иное влияние на выполнение только одной машинной команды, следующей за префиксом. Виды префиксов : Rep (повторения), Seg (замена сегмента), Lock (блокировка шины).

В базовой модели Intel 8086 используется более 10 разнообразных форматов команд. Пример формата команды максимальной длины (двухоперандная команда с постбайтом адресации и непосредственным операндом)

адреса в ОП

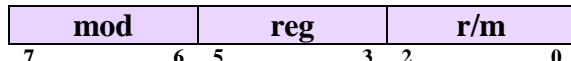


Специальные биты OpC - W(Word) и S(Sign Extended). Расширение Word определяет длину операндов (W=0 -> байт, W=1 -> слово). А S=1 - необходимость знакового расширения непосредственного операнда.

Знаковое расширение операнда имеет место только для комбинации S,W = 1,1, при этом в машинной команде задаётся только один младший байт (IMML) непосредственного операнда, т. к. операнд команды, задаваемый постбайтом адресации, является двухбайтным, то для приведения непосредственного операнда к тому же формату (слово) производится предварительное его расширение на старший байт. При этом операнды рассматриваются как целые знаковые числа, естественно, представленные в дополнительном коде. В связи с чем, старший байт IMMH заполняется нулями для положительного операнда или единицами для отрицательного. Фактически, все биты старшего байта становятся равными знаковому биту младшего байта, что и называется знаковым расширением операнда. При W=0, бит S становится неактуальным.

Постбайт адресации используется в команде для задания режимов адресации, в принципе, для обоих операндов. В приведенном примере формата, постбайт адресует только один операнд, поскольку второй из операндов задан непосредственно в команде.

Структура постбайта адресации для двухадресной команды имеет вид:



Т. к. поле reg в рассматриваемом формате не используется, то на его месте задается расширение кода операции. Назначение полей постбайта см. Эл. Консп.

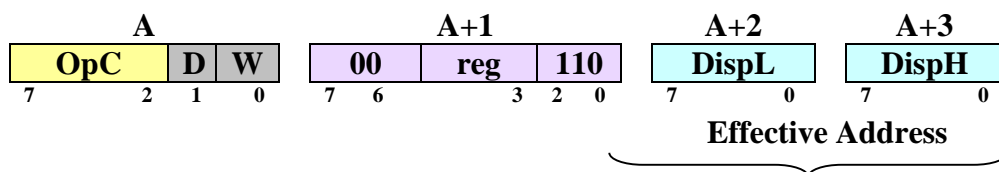
Для операнда, находящегося в памяти, поле r/m (register/memory) задаёт компоненты эффективного адреса Base и/или Index неявным образом (см. таблицу кодирования). В базовой модели для базовой компоненты EA могут использоваться только регистры BX и BP. А для индексной компоненты только регистры SI и DI.

С помощью постбайта адресации могут быть реализованы следующие режимы адресации:

- прямая регистровая;
- прямая адресация памяти EA = Disp;
- базовая EA = Base + Disp;
- индексная EA = Index + Disp;
- базово – индексная без смещения EA = Base + Index;
- базово – индексная со смещением EA = Base + Index + Disp;
- косвенная регистровая EA = [РОН], где РОН = BX, SI, DI.

Прямая адресация памяти реализована, как исключительный случай при следующих значениях полей постбайта адресации: mod = 00, r/m = 110. В этом случае формат команды принимает вид

адреса в ОП



В старших моделях процессоров, точнее, начиная с модели i386, используется 32 – битная адресация и 32 – битные операнды. Для сохранения совместимости с 16 – битными моделями в старших моделях имеется возможность использования, как 16 – битной адресации, так и 16 – битных операндов.

Глобально, размер адресации и операндов задаётся специальным битом D – **Default Size** (размер по умолчанию), этот бит находится в **Description Segment** (описатель сегмента). Локальную установку разрядности адреса и/или операнда по сравнению с принятой по умолчанию (по значению бита D) можно осуществить с использованием соответствующих префиксов AS – Address Size, OS – Operand Size.

Для 32 – битных моделей максимальный размер машинной команды составляет 12 байт (без учета возможных префиксов)

2байта – код операции

1байт – постбайт адресации

1байт – дополнительный байт адресации (SIB – Scale Index Base)

4 байта – операнд

Байт SIB используется для задания базово – индексной адресации с масштабированием



В отличие от 16 – битного адреса, где базовый и/или индексные регистры задаются неявно, в 32 – битной адресации поля Base и Index задают прямые адреса РОНов, в которых размещаются базовая и индексная компонента EA.

Еще одним существенным отличием является возможность использования практически любых РОНов в качестве базы и/или индексов.

### Базовая система команд.

Система команд компьютера включает в себя перечень машинных команд, реализуемых непосредственно аппаратными средствами. Именно система команд определяет возможности компьютера в плане решения разнообразных задач обработки данных.

Основной характеристикой системы команд является её мощность. Обычно под этим термином понимается количество разнообразных мнемкокодов, используемых для символического обозначения на ассемблере.

Существует и другой подход к определению мощности системы команд, когда в разнообразие машинных команд включают не только разнообразие мнемкокодов, но и разнообразие используемых режимов адресации и форматов команд.

Например, с использованием первого подхода, мощность системы команд базовой модели Intel 8086 составляет 113 мнемкокодов. С учетом же возможности использования разнообразных режимов адресации и форматов машинных команд при втором подходе, мощность системы команд составляет не менее 4000 машинных команд.

При переходе от младших моделей к старшим система команд процессора неуклонно расширяется. Основными причинами расширения для семейств Intel 90X86 Pentium являются:

1. Поддержка защищённого режима (i286).
2. Внедрение блока FPU (Float Pointer Unit) в один кристалл CPU. Система команд блока FPU, кроме арифметических команд обработки данных с плавающей точкой, включают в себя большой набор трансцендентных команд для вычислений значений большинства элементарных функций ( $\sqrt{\quad}$ , ln, exp, sin, cos, tg, arctg, arcos, arcsin).
3. Включение в Pentium блока MMX (Multimedia Extension), система команд, которая содержит порядка 60 команд для поддержки принципа векторной обработки на уровне целочисленных данных.

Отличие векторной обработки и, соответственно, векторных команд от скалярной обработки и, соответственно, скалярных команд, состоит в том, что операндами векторных команд являются вектора, последовательно расположенные в памяти.

Суммарная длина векторных данных составляет 64 бита, т. е. количество элементов вектора весьма ограничено. Блок MMX принято считать первым использованием принципов векторной обработки в микропроцессорах. Сами идеи векторной обработки появились еще в 60 – е годы и были реализованы уже в 70 – е годы в первых суперкомпьютерах ILLIAC – IV, CRAY – 1. В отличие от микропроцессорной реализации длина вектора в этих компьютерах составляла 64 элемента. MMX – векторная обработка, но для целочисленных данных.

#### 4. Внедрение в кристалл процессора блока SSE (SSE2)

SSE – Streaming SIMD Extension - потоковое SIMD расширение

SIMD – Single Instruction Multiple Data

Внедрение порядка 80 команд, поддерживающих векторную обработку в отношении данных с плавающей запятой.

С учетом разнообразных расширений, мощность степени команд последних моделей Pentium составляет более 400 мнемкокодов.

По функциональному назначению команды базовой модели принято разделять на следующие группы:

- арифметические команды(ADD,SUB,IMUL,MUL,DEC,INC,NEG,COMP..);
- логические команды (побитовой обработки)(AND,OR,XOR,NOT,TEST);
- команды сдвигов(SAR,SAL,SHL,SHR,ROL,ROR,RCL,RCR);
- команды управления программами(JMP,J/cond,LOOP,CALL,RET).

## CISC- и RISC – архитектура.

Непосредственно с мощностью использования системы команд связаны два основных направления в архитектуре компьютера:

CISC – Complex Instruction Set Computer – компьютер с расширенной системой команд

RISC – Reduced Instruction Set Computer - компьютер с сокращенной системой команд.

Вся история развития компьютеров с CISC - архитектурой сопровождалась постоянным развитием и расширением их системы команд. Статистические исследования, широко проводимые в 70 –е годы в отношении частоты использования различных команд, позволили сформулировать достаточно актуальный в своё время принцип “80X20”.Суть которого в том, что на 20% машинных команд из общей системы команд процессор затрачивает порядка 80% своего бюджета, т. е. очень многие машинные команды оказываются практически невостребованными при разработке программных продуктов.

Сложность используемой системы команд в первую очередь сказывается на сложности устройства управления. При реализации CISC – процессора на одном кристалле в виде СБИС, на долю устройства управления приходится от 30% до 60% площади кристалла.

В соответствии с реализацией принципов микропрограммного управления, используемого во всех CISC - процессорах, в состав устройства управления включена достаточно большая микропрограммная память для хранения микрокодов по реализации различных машинных команд. Например, в ЭВМ VAX – 11 ёмкость микропрограммной памяти составляет 50 Кбайт.

#### Основные особенности RISC – архитектуры.

Термин RISC был впервые введён в 1980 г. Дэвидом Паттерсоном, профессором Калифорнийского Университета.

- 1) Использование сравнительно небольшого множества машинных команд, наиболее часто используемых при решении задач обработки данных. Первые модели RISC – процессоров (середина и конец 80-ых г.г.) имели мощность системы команд менее 100, в современных моделях RISC – процессоров - примерно 150. Система команд современного RISC – процессоров включает в себя целочисленную арифметику, арифметику с плавающей точкой, мультимедийные расширения (векторные команды).
- 2) Стремление к выполнению большинства машинных команд за 1 машинный такт (машинный цикл).

Под одним машинным тактом понимается интервал времени между двумя последовательными синхросигналами, подаваемыми на вход процессора от генератора (как правило, внешняя микросхема). Величина, обратная длительности машинного такта, называется тактовой частотой, это одна из самых важных характеристик процессора, как и компьютера, определяющая его быстродействие (производительность).

За 1 машинный такт в процессоре выполняются элементарные действия, называемые микрооперациями, например, пересылка между двумя регистрами или выполнение элементарных операций в АЛУ, таких как сложение или логическое умножение. Управляющее слово, инициирующее выполнение одной или нескольких совместимых во времени микроопераций, называется микрокомандой.

Совокупность микрокоманд для реализации какой – нибудь сложной машинной команды называется микропрограммой. Микропрограммы составляются и отлаживаются при проектировании компьютера и хранятся в постоянной памяти, называемой памятью микропрограммы. Она входит в состав блока (устройства управления). В RISC – процессорах грань между машинной командой и микрокомандой практически стирается.

- 3) (как следствие из 2)) В RISC – процессорах для реализации устройства управления используется принцип жесткой логики как альтернатива принципа микропрограммной логики. Это означает, что устройство управления реализуется как схемный автомат с использованием автоматной модели Мили Мура. В виду упрощения устройства управления площадь, занимаемая им на кристалле для RISC – процессоров составляет 5 – 10 %, а для CISC – процессоров – 30 -50%

Освобождённая площадь кристалла используется для:

- Увеличения внутренней регистровой памяти. В современных моделях RISC – процессорах число внутренних регистров может быть несколько сотен (до 500).
- Увеличение объёма внутренней Кэш – памяти. Типичный размер внутрикристалльный Кэш первого уровня 64 – 128 Кбайт.

Замечание: в некоторых моделях используется внутрикристалльная Кэш – память и второго уровня.

- Внедрение в кристалл дополнительных блоков для реализации векторной обработки (мультимедийные расширители).
- 4) Использование большого числа внутренних регистров создаёт дополнительную сложность при выходе на обработку прерывания, связанные с необходимостью сохранения контекста прерываемой программы. Для уменьшения времени издержек на переключение программ в RISC – процессорах зачастую используется механизм регистровых окон. Идея состоит в том, что каждой программе выделяется некоторое подмножество регистров, образующих окно.
  - 5) Ограниченное число форматов машинных команд и используемых режимов адресации. Используется 3 – 5 форматов машинных команд фиксированной длины и 2 - 3 основных режима адресации (косвенная адресация в RISC – процессорах не используется). Всё это делается для максимального упрощения процесса декодирования команды и формирования адресов операндов, что в свою очередь приводит на минимизацию затрат на блок управления.
  - 6) Все команды обработки данных реализуются только над регистровыми операндами (команды типа reg/reg). Для обмена с памятью используются специальные команды типа LOAD (memory -> reg) и STORE (reg -> memory).
  - 7) Широкое использование принципов суперскалярной и суперконвейерной обработки.

Основные модели RISC – процессоров:

Модель процессора	Полное название	Фирма изготовитель	Полное название
SPARC	Scaleable Processor ARChitecture	SUN Microsystems	SUN Microsystems
Micro SPARC	32 разрядные		
Super SPARC			
Hyper SPARC			
Ultra SPARC	64 разрядный		
ALPHA	ALPHA	DEC/HP	Digital Equipment Corporation/Hewlett Packard
Power PC	Performance Optimized With Enhanced RISC	IBM/Motorola	International Business Machines/Motorola
PA RISC	Precision Architecture (64разрядный)	HP	Hewlett Packard

**Режимы работы процессора. Привилегированные команды.**

В целях разграничения доступа к системным ресурсам со стороны прикладных и системных программ, в современных моделях процессоров в том или ином виде существует два основных режима функционирования:

- прикладной
- системный.

В системном режиме допускается исполнение любых машинных команд. В прикладном режиме не допускается исполнение так называемых привилегированных команд. В простейшем случае режим работы процессора задаётся с помощью специального бита, находящегося в каком – либо системном регистре. Например, в процессоре IBM 370 бит режима находится в слове состояния программы PSW (Program Status Word). Два альтернативных состояния процессора называются Task/Supervisor.

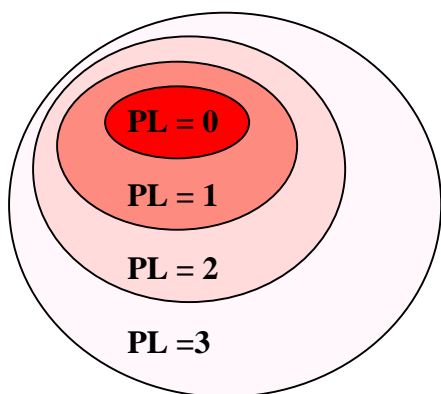
Аналогичный бит режима System/User имеет место в моделях системы VAX(DEC), который находится в PS (Processor Status).

В процессорах семейства 80X86 Pentium используется более сложная интерпретация (способ) для задания режима работы процессора в плане разделения системного и пользовательского режимов. С помощью специального бита PE (Protect Enabled) в управляющем регистре CR0 задаётся или реальный режим (PE = 0) или защищённый режим (PE = 1).

При использовании защищённого режима вступают в действие различные средства защиты. Одним из средств защиты, поддерживаемым на аппаратном уровне, является защита по уровням привилегий (кольцам защиты).

Аппаратно поддерживаются 4 уровня привилегий для сегментов и 2 уровня привилегий для страниц. Идея защиты по уровням привилегий состоит в присвоении различным сегментным объектам, в частности сегментам кода, данных, стека, определённого уровня привилегий. Этот уровень отражается соответствующим двухбитным полем DPL(Descriptor Privilege Level), размещаемом в дескрипторе (описателе сегмента).

Уровень привилегий определяет степень важности и доступности сегмента. Наивысшим уровнем привилегий является PL = 0, и он присваивается программам ядра, наинизший уровень PL = 3, он присваивается прикладным программам.



Общее правило защиты предполагает возможность обращений или доступа из внутренних колец. Во внешние. Попытки обращений из внешних колец во внутренние в общем случае пресекаются средствами защиты с выходом на прерывание специального типа Тип 13 (“Нарушение общей защиты”).



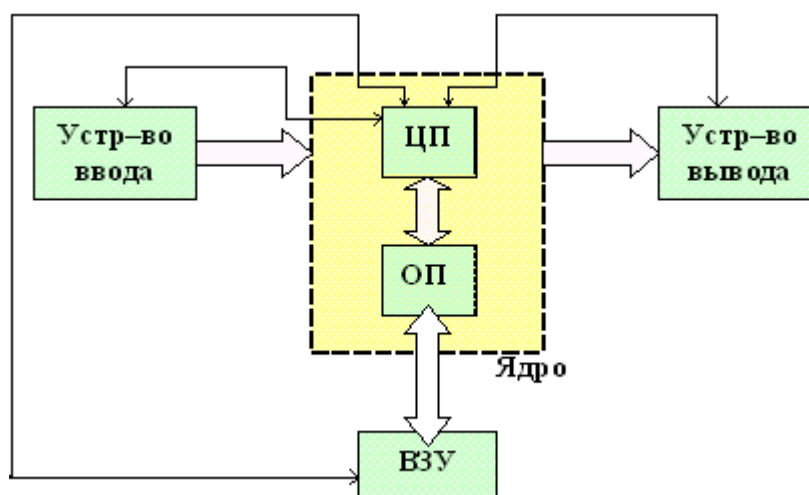
Пользовательский режим ассоциируется с уровнем привилегии  $PL = 3$ , системный режим с уровнем привилегий  $PL = 0$ .

Современные операционные системы, в частности Windows и Unix поддерживают только 2 уровня привилегий (User/Supervisor). Соответствующие биты для двух уровней используются на страничном уровне и размещаются в страничных дескрипторах. В процессорах семейства Intel к привилегированным командам относятся:

- 1) Команды загрузки и сохранения системных регистров.
- 2) Команды манипуляции флагом IF: CLI (0 -> IF), STI (1 ->IF).
- 3) Команды ввода/вывода: IN/OUT, INS/OUTS.
- 4) Команда останова процессора - HLT.

Попытка выполнения привилегированных команд в пользовательском режиме ( $PL = 3$ ) приводит к выводу на прерывание 13. Почти все привилегированные команды, за исключением команд ввода/вывода и манипуляций флагом IF требуют для своего выполнения наивысшего уровня привилегий  $PL = 0$ .

### Упрощенная структура компьютера (ЭВМ).



Независимо от принадлежности компьютера некоторому классу или типу, его в первом приближении можно разделить на 2 части:

1. центральную;
2. периферийную.

Центральную часть принято называть ядром. В ядро входят два основных устройства компьютера: ЦП и ОП.

Периферийную часть можно условно представить устройствами трех типов:

- внешние запоминающие устройства, которые образуют внешнюю память (к ним относятся накопители на магнитных дисках и на магнитных лентах);
- устройства ввода;
- устройства вывода.

Обмен информацией между ядром и периферией, а так же между устройствами ядра осуществляется на уровне аппаратных интерфейсов. Организация обмена между ядром и периферийной частью компьютера возлагается на систему ввода/вывода

(I/O S – Input/Output System). Система ввода/вывода представляет собой аппаратно - программный комплекс.

Аппаратная часть I/O S включает в себя:

1. собственно периферийные устройства (разделённые на УВ/В и ВЗУ);
2. контроллеры ПУ(устройства управления);
3. контроллеры для организации обмена, в частности, контроллер DMA – Direct Memory Acces (ПДП-прямой доступ к памяти) PIC (Program Interrupt Controller – Программируемый Контроллер Прерываний);
4. интерфейсы (шины);
5. система прерываний.

Программная часть I/O S включает в себя:

1. супервизор (Supervisor) в/в;
2. драйверы ВУ.

Для современных программных средств I/O S типичным свойством является многоуровневая (иерархическая) организация, в частности, многоуровневые драйверы.

Программное обеспечение I/O S разделяется на устройство-зависимую часть и устройство независимую часть. Устройство-независимое ПО выполняет следующие функции:

- буферизация;
- защита (сообщения об ошибках);
- блокирование (блочный характер передачи);
- обеспечение единообразного программного интерфейса для драйверов устройств.

## **Организация ввода/вывода.**

### **Понятие, основные характеристики и уровни представления интерфейса.**

В общем плане под интерфейсом принято понимать способ сопряжения и взаимодействия между несколькими объектами. В отношении компьютеров принято рассматривать множество понятий интерфейса: аппаратный, программный, пользовательский. В отношении аппаратных интерфейсов используются следующие понятия: интерфейс памяти, интерфейс ввода/вывода, интерфейс периферийных устройств (малый интерфейс). Существует большее количество подходов к определению аппаратного интерфейса. Основными компонентами в различных понятиях аппаратного интерфейса, являются:

1) Совокупность линий, шин, обеспечивающих обмен информацией между устройствами.

2) Алгоритм (протокол) обмена, определяющий последовательность организации передачи информации по линиям интерфейса.

3) Разделение интерфейса на ряд уровней представлений.

На обобщённой структуре двойными линиями обозначаются структуры, по которым осуществляются передача информации данных между компонентами компьютера, а одинарными – обозначаются связи по управлению. Тем самым подчёркивается, что центральный процессор выполняет в компьютере двойную функцию: как устройство обработки (выполняет заданные программы) и как устройство управления всеми компонентами компьютера.

От ЦП к остальным устройствам по линиям связи передаются управляющие сигналы (приказы, команды в/в); в обратную сторону передаются сигналы о

состоянии устройств, в частности об их готовности к обмену, а также запросы прерываний (например, для идентификации момента завершения операции в/в).

Основными типами линий (шин), входящих в состав аппаратного интерфейса, являются:

- ⇒ шина адреса;
- ⇒ шина данных;
- ⇒ шина управления (для передачи сигналов, управляющих обменом);
- ⇒ линии синхронизации (по шинам передаются сигналы, синхронизирующие передачу информации по интерфейсу);
- ⇒ линии запросов прерываний (по ним передаются сигналы прерывания от устройств, подключаемых к интерфейсу);
- ⇒ линии питания;
- ⇒ линии заземления.

#### Основные характеристики интерфейса:

1. Пропускная способность определяется максимальным количеством бит или байт данных, передаваемых по интерфейсу за одну секунду.
2. Информационная ширина (количество бит или байт данных, передаваемых параллельно по шине данных, т.е. разрядность линии).
3. Максимально возможное удаление устройств, подключаемых к интерфейсу.

Алгоритм (протокол) обмена обычно представляется с помощью временных диаграмм. Определяется порядок следования и допустимые параметры (амплитуда, длительность) для управляющих и информационных сигналов при работе интерфейсов в различных режимах.

Уровни представления интерфейсов:

- Логический уровень определяет состав, наименование, назначение шин интерфейса, а также порядок передачи информации по этим линиям (протокол обмена).
- Физический уровень определяется параметрами электрических, оптических сигналов, передаваемых по линиям интерфейса.
- Конструктивный уровень определяет физическую реализацию шин интерфейса: скрученная (витая) пара, коаксиальный кабель; а также определяет виды разъемов и распределение линий интерфейсов по контактам разъема.

### Шины (интерфейсы) персональных компьютеров на базе процессоров Pentium.

Структура современных ПК отличается большим разнообразием используемых шин или интерфейсов.

Название шины	Полное название	Перевод	Последовательная /параллельная	Комментарий
FSB	Front – Side Bus	Шина переднего плана	параллельная	обеспечивает связь между ЦП и ОП
BSB	Back – Side Bus	Шина заднего плана	параллельная	обеспечивает связь между ЦП и внешнего КэшL2, отличается большой пропускной способностью
PCI Intel - 1990	Peripheral Component Interconnect	Соединение периферийных компонент	параллельная	шина расширения
ISA	Industry Standard Architecture	Стандартная промышленная архитектура	параллельная	шина расширения; ISA –16разрядн. EISA-32разрядн. Для подключения принтера, модема, звуковой карты
SCSI	Small Computer System Interface	Интерфейс малых вычислительных систем	параллельная	для подключения периферийных интерфейсов (в частности, магнитных дисков)
USB	Universal Serial Bus	Последовательная универсальная шина	последовательная	для подключения медленных устройств (например клавиатуры)

**Основные аспекты организации ввода/вывода.**

**1. Структура компьютера в плане организации связей между ядром и периферийными устройствами:**

- а) структура с единым интерфейсом (с магистральным интерфейсом, с общей шиной);**
- б) многошинная структура, рис. 1.11 (Таненбаум);**
- в) структура с каналами (процессорами) ввода/вывода (Цилькер);**

**2. Адресация к ВУ или ПУ. Основным аспектом, связанным с адресацией ВУ, является объединение или разделение адресных пространств памяти и ввода/вывода.**

**3. Способ организации ввода/вывода:**

- а) программный (программно управляемый, программируемый) ввод/вывод;**
- б) ввод/вывод по прерыванию (управляемый прерываниями);**
- в) ввод/вывод с использованием прямого доступа к памяти;**
- г) канальный ввод/вывод.**

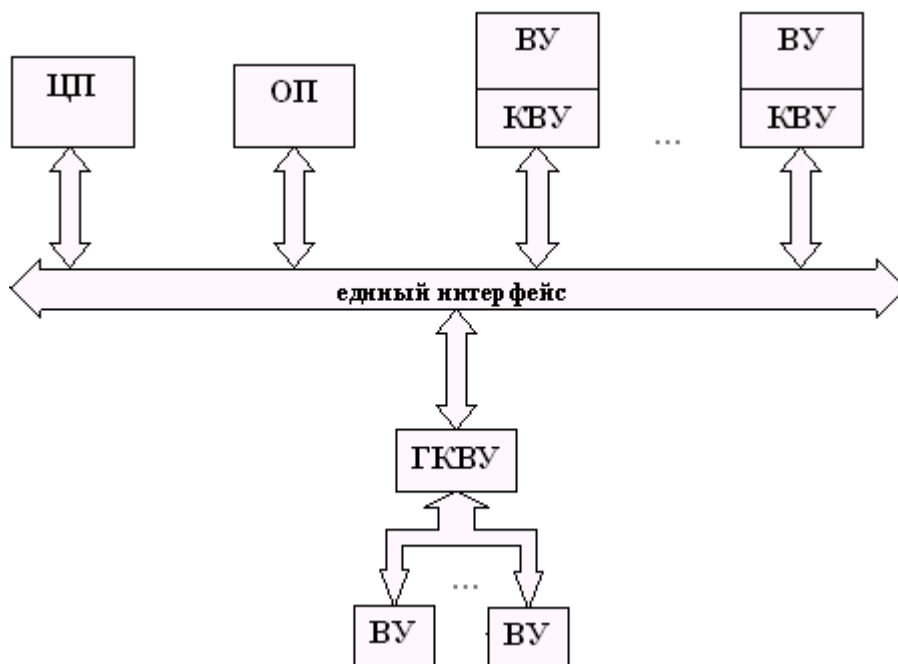
**Упрощенная структура компьютера с единым интерфейсом.**

Ещё в конце прошлого века подобная структура являлась канонической и стандартной для большинства моделей мини ЭВМ, микро ЭВМ и ПК. Примером единого интерфейса может служить стандартный интерфейс Unibus, который использовался в компьютерах фирмы DEC (PDP – 11, VAX – 11, CM ЭВМ).

Основными особенностями компьютеров с единым интерфейсом являются:

- 1) все устройства, как центральные, так и периферийные, для связи между собой используют одни и те же шины адреса, данных и управления;**
- 2) в любой момент времени по единому интерфейсу может быть организована передача данных только между двумя устройствами;**
- 3) в соответствии с п.2 при наличии большого числа устройств, единый интерфейс становится “узким местом” (bottle neck), в связи с чем подобная структура совершенствовалась путём использования локальных дополнительных шин;**
- 4) как правило, использование единого интерфейса предполагает единообразие операции с памятью (чтение и запись) и ввода/вывода, в связи с этим предполагается использование объединённого адресного пространства для памяти и ввода/вывода (ввод/вывод, отображённый на память).**

**В современных ПК подобная структура не используется.**



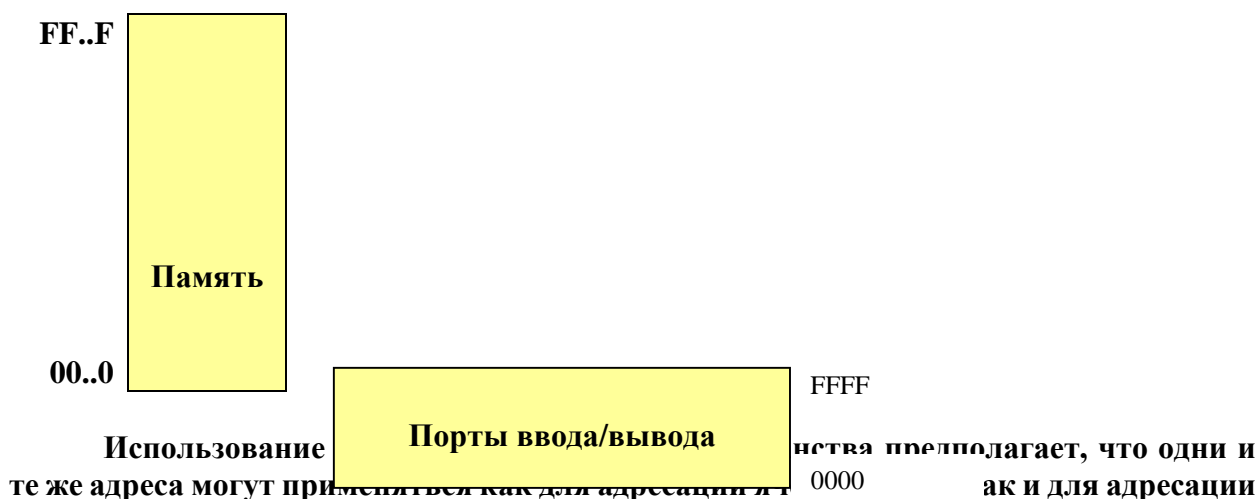
### Адресация ВУ.

Адресация собственно ВУ в современных компьютерах используется достаточно редко. Примером использования фактического адреса ВУ могут служить команды ввода/вывода IBM 370. В современных ПК адресация ВУ осуществляется на уровне программно доступных регистров контролеров ВУ, которые называются портами ввода/вывода.

### Способы адресации портов ввода/вывода и их сравнительный анализ.

Различие способов адресации связано с использованием отдельного или единого адресного пространства для памяти и портов ввода/вывода.

#### Раздельное адресное пространство.



портов ввода/вывода. При этом в системе команд процессора должны использоваться специальные команды для ввода/вывода, отличные от команд обмена с памятью. Достоинством подобного подхода принято считать возможность использования более короткого адреса порта ввода/вывода по сравнению с адресом ячейки памяти.

Так, например, в системе команд процессора Intel для адресации портов ввода/вывода может использоваться либо 1 байт (при прямой адресации порта), либо 2 байта (при косвенной адресации порта).

При косвенном задании адреса используется регистр DX (неявно адресуемый), в котором и находится адрес порта ввода/вывода.

### Единое адресное пространство.

Использование единого адресного пространства существенно влияет как на систему команд процессора, так и на управление вводом/выводом на аппаратном уровне.



Некоторая область адресного пространства в старших адресах используется не для адресации памяти, а для адресации портов ввода/вывода. Подобная идея была впервые реализована в мини ЭВМ PDP – 11(DEC). Подобный способ хорошо вписывается в рамки так называемого магистрального интерфейса.

В системе команд отсутствуют специальные команды ввода/вывода, и обмен между памятью и портами ввода/вывода реализуется по аналогии с обменом между процессором и памятью с использованием обычной команды типа MOV.

Организация ввода/вывода с отображением на память обладает следующими достоинствами:

- 1) Не требуются специальные команды ввода/вывода (упрощение системы команд).
- 2) Возможность использования любых видов обработки (реализуется и/или логическими командами применительно к содержимому портов ввода/вывода).

Недостатками использование совмещенного адресного пространства являются:

- 1) Усложнение управления кэшированием, связанное с необходимостью запрета кэширования адресного пространства, выделенного для портов ввода/вывода.
- 2) Сложность реализации этого способа для многошинной архитектуры.

Для одношинной архитектуры любой адрес, выставляемый на шину адреса, сравнивается всеми модулями памяти и ввода/вывода на предмет собственной принадлежности. При отдельных шинах памяти и ввода/вывода требуются дополнительные затраты для проверки адреса на его принадлежность к памяти или вводу/выводу.

Возможные способы решения проблемы:

- первоначальный запрос направляется к памяти по быстрой шине, а затем, если память не может ответить на него, то запрос перенаправляется в шину ввода/вывода;
- фильтрация адресов специальной микросхемой, отделяющей адреса памяти от адресов портов ввода/вывода; в частном случае подобную

функцию может выполнять мост PCI, в состав которого входят специальные регистры диапазона; при этом все адреса, попадающие в мост и принадлежащие выделенному диапазону, передаются в дальнейшем не к памяти, а непосредственно в шину PCI, которая служит шиной ввода/вывода.

## Способы организации ввода/вывода.

### Программно управляемый ввод/вывод (ПУВВ). (гл.8 Цилькер)

Ввод/вывод осуществляется при непосредственном участии ЦП. Реализация ввода/вывода производится специальной программой (драйвером ВУ), в котором выполняются следующие действия:

- 1) Пересылка порции данных между ОП и портом ввода/вывода (как правило, в качестве промежуточного звена используется какой – либо регистр ЦП).
- 2) Проверка готовности ВУ к обмену (сводится к опросу регистра состояния контроллера ВУ).
- 3) Ожидание готовности ВУ.
- 4) Изменение (модификация) параметров пересылки, в частности текущего адреса области ОП для ввода/вывода и счетчика длины пересылаемого блока.
- 5) Проверка завершения передачи путем сравнения счетчика с некоторым конечным значением (например, с нулем для декрементного счетчика).

Достоинства и недостатки ПУВВ PIO(Programming Input/Output):

Основным достоинством PIO принято считать относительную простоту его реализации, а основным недостатком – неэффективное использование ресурсов ЦП.

### Ввод/вывод по прерыванию.

Использование этого способа организации ввода/вывода позволяет устранить основной недостаток предыдущего способа.

Основная идея: после осуществления элементарных действий по пересылке очередной порции данных, ПЦ вместо переключения в состояние активного ожидания с опросом готовности ВУ (как в первом способе) переходит к выполнению другой программы (планировщик заданий Операционной Системы запускает другой процесс); в свою очередь, устройство ввода/вывода о своей готовности продолжать операцию ввода/вывода с очередной порцией данных сообщает ПЦ с помощью сигнала прерывания; сигналы прерывания от ВУ в ПЦ Intel приходят на специальный вход INTR(Interrupt Request); при получении сигнала от ВУ ПЦ выполняет следующие действия:

- 1) завершает выполнение текущей команды программы;
- 2) сохраняет контекст прерываемой программы;
- 3) идентифицирует источник прерывания и преобразует код источника в начальный адрес программы обработчика прерываний;
- 4) загружает адрес обработчика в счетчик команд;
- 5) переходит к выполнению программы обработчика прерываний.

Замечание: обработчики прерываний от устройств ввода/вывода связаны с драйверами соответствующих устройств.

Несмотря на очевидное преимущество этого способа по сравнению с предыдущим, его недостатком являются большие издержки времени, связанные с контекстным переключением из прерванной программы на обработчик прерываний



и обратно. Для устройств с посимвольным обменом эти переключения имеют место после пересылки, если не каждого байта, то сравнительно небольшого их числа (4-8ба), определяемых шириной шины данных.

### Прямой доступ к памяти – DMA (Direct Memory Access).

Основное отличие DMA от предыдущих способов в том, что участие ЦП в организации ввода/вывода сводится к минимуму. ЦП организует лишь так называемую инициализацию DMA, а также реакцию на завершение операции ввода/вывода.

Режим DMA используется для организации так называемых блочных пересылок. Типичным ВУ с блочным обменом являются накопители на магнитных дисках и магнитных лентах. Управление обменом в режиме DMA осуществляется специальным устройством (микросхемой), называемым контроллером DMA – DMAC. При этом контроллер DMA реализует обмен не на программном, а чисто на аппаратном уровне (DMAC – микропрограммный автомат).

DMAC содержит некоторое число программно доступных регистров, представленных для ЦП адресуемыми портами ввода/вывода. (гл.8 Цилькер, Орлов) Инициализация DMA сводится к заданию режима работы и необходимых адресов путем пересылки требуемой информации из ЦП в регистры контроллера DMA. На этапе инициализации задаются следующие основные данные:

- начальный адрес блока памяти (области памяти), используемого при обмене;
- объем передаваемого блока памяти в байтах (типичный размер блока при обмене с жестким диском составляет 512 байт, длина или объем сектора);
- код операции обмена (ввод или вывод);
- адрес устройства прямого доступа (адрес ВУ задается в связи с тем, что стандартный контроллер DMA включает в себя 8 каналов прямого доступа).

DMA может быть реализован в одном из следующих основных режимов (по Цилькеру):

Название режима	Описание режима
Блочная пересылка	Захват шины на весь период пересылки блока; на весь период DMA ЦП не имеет доступа к шине памяти, в этот период процессор может продолжать работу по программе с обращением к КЭШ – памяти.
Пропуск цикла	После пересылки слова DMAC освобождает шину на один цикл, предоставляя ее ЦП.
Прозрачный режим	DMAC имеет доступ к шине только в тех циклах, в которых ЦП в ней не нуждается.

- 1) Поддержка DMA на аппаратном уровне в процессоре фирмы Intel осуществляется на уровне входного сигнала HOLD(захват шины) и

выходного сигнала HLDA(Hold Acknowledgment) – сигнал подтверждения захвата. Сигнал HOLD инициализирует DMAC при начале цикла обмена; ЦП, получив этот сигнал, отключается от шины и выставляет активный уровень сигнала подтверждения HLDA, получив этот сигнал, DMAC начинает цикл блочного обмена.

2) Стандартные контроллеры DMA позволяют реализацию следующих видов обмена:

Port -> Mem  
Mem -> Port  
Mem -> Mem (обмен с видеопамятью)  
Port -> Port

2) В современных моделях ПК для обмена с жесткими дисками наряду с DMA также используется и PIO.

### **Канальный ввод/вывод (КВВ).**

Канальный ввод/вывод основан на использовании в архитектуре ЭВМ специализированных процессоров, ориентированных на организацию ввода/вывода. Эти процессоры обычно называются каналами ввода/вывода. Канальный ввод/вывод является программно управляемый, реализуется с помощью специальной программы, которая носит название канальной.

Канальные программы для организации обмена с различными типами ВУ хранятся в основной памяти. В связи с тем, что КВВ является процессором, правда специализированным, управление порядком выполнения команд канальной программы осуществляется с помощью своеобразного счетчика команд.

Команды канальной программы называются УСК (Управляющими Словами Канала). УСК содержат следующую основную информацию:

1. код команды (например: прочитать или записать, проверить состояние ВУ, т.п.);

2. начальный адрес области ОП, с которой осуществляется обмен;

3. длина передаваемого блока в байтах;

4. различные идентификаторы и признаки, влияющие на организацию обмена:

- признак цепочки данных;

при его установке следующая команда канальной программы выполняет то же действие, что и данная, но с другой областью памяти; с помощью цепочки данных обеспечивается непрерывный обмен с несмежными областями ОП без привлечения ЦП;

- признак цепочки команд;

установка этого признака в текущей команде указывает каналу на необходимость продолжения канальной программы после завершения текущей команды путем выборки следующей команды; сброшенный признак цепочки команд означает, что текущая команда является последней в канальной программе (аналог - HLT);

- признак программно управляемого прерывания;

установка этого признака в какой – либо команде сопровождается выдачей сигнала прерывания из КВВ в ЦП в момент выборки этой команды на исполнение; получив сигнал прерывания, ЦП может, например, приступить к обработке блока данных, передача которого завершилась при выполнении предыдущей части канальной программы.

## **Основные функции КВВ:**

- 1) **Функции по установлению логической связи между ВУ и ОП.**
  - а) прием и декодирование команд ввода/вывода от ЦП;
  - б) инициирование выполнения канальной программы при получении команды SIO (Start Input/Output) от ЦП;
  - в) проверка состояния ВУ, участвующего в обмене, и передача в ЦП информации о его готовности или неготовности к обмену;
- 2) **Функции, связанные с непосредственной передачей данных между ВУ и ОП.**
  - а) последовательная выборка команд канальной программы из ОП, их декодирование и выполнение;
  - б) обеспечение приема, передачи, контроля и промежуточного хранения данных при обмене между ОП и ВУ;
  - в) формирование текущих адресов ОП, по которым записываются или считываются передаваемые данные;
  - г) согласование форматов данных, передающихся по интерфейсу ввода/вывода, с форматом интерфейса ОП (как правило, ширина интерфейса ввода/вывода составляет 1, 2 или 4 байта, что меньше ширины интерфейса ОП: 4, 8, 16 байт);
  - д) подсчет числа передаваемых байт данных с целью определения момента завершения передачи блока данных;
  - е) выработка последовательности синхронизирующих и управляющих сигналов в соответствии со стандартом интерфейса ввода/вывода;
  - ж) анализ особых ситуаций в ВУ во время обмена (ошибка передаваемых данных, сбой устройства и т.п.) и информирование ЦП об этих ситуациях (с помощью запроса прерывания);
- 3) **Функции, связанные с завершением обмена и разрушением логической связи между ВУ и ОП.**
  - а) определение момента завершения в программе по организации обмена между ВУ и ОП;
  - б) передача в ЦП сигнала прерывания о завершении обмена.

**Участие ЦП в организации КВВ сводится к выполнению следующих функций:**

- Инициирование операции ввода/вывода (реализуется командой SIO основной программы).
- Проверка состояния канала ввода/вывода (реализуется командой TCH – Test Channel).
- Проверка состояния ВУ (реализуется командой TIO – Test Input/Output).
- Остановка операций ввода/вывода (реализуется командой HIO – Halt Input/Output) возможно, для инициирования более приоритетной операции.

Команды ввода/вывода являются привилегированными, т.е. могут выполняться только программами операционной системы в режиме Supervisor (SVR).

Использование каналов ввода/вывода в архитектуре компьютеров, относительно к классу Main Frame и суперкомпьютеров, является мощной аппаратной поддержкой реализации мультипрограммного режима обработки. Архитектура компьютера с каналами ввода/вывода, даже при наличии одного ЦП,

позволяет параллельно реализовать выполнение одной программы в ЦП и процессов ввода/вывода для нескольких других программ с использованием ресурсов КВВ.

## **Классификация КВВ.**

По режиму функционирования, они разделяются на мультиплексные и селекторные. Мультиплексный канал обеспечивает параллельную работу многих ВУ. Селекторный канал работает в монопольном режиме, обеспечивая работу единственного ВУ.

Взаимодействие мультиплексного канала с одним из ВУ, подключенным к ВУ, принято называть сеансом связи с ВУ. В зависимости от порции данных, передаваемых через канал между ВУ и ОП за один сеанс связи, мультиплексные каналы разделяются на два вида: байт-мультиплексные и блок-мультиплексные.

Часть ресурсов мультиплексного канала, используемая отдельным ВУ, называется подканалом. В каждом подканале используется некоторая область памяти канала, в которой хранится информация о текущем состоянии обмена с данным ВУ. Переключение мультиплексного канала с обслуживания одного ВУ на другое сопровождается сохранением информации в памяти подканала для текущего ВУ и выборкой информации из памяти подканала для следующего ВУ.

## **Сравнение Канального ВВ с PIO и с DMA.**

Так как КВВ осуществляет организацию обмена по собственной программе, то КВВ следует считать программно управляемой, однако, в отличие от PIO, программу выполняет не ЦП, а КВВ.

Многие авторы сопоставляют КВВ и DMA. Аналогия между КВВ и DMA состоит в том, что оба эти способа обмена реализуются практически без участия ЦП. Так же, как и для DMA, КВВ требует участия ЦП на этапе инициализации. В частности, при инициализации от ЦП в КВВ передается начальный адрес канальной программы. Существенным же отличием КВВ от DMA является программная реализация первого и чисто аппаратная второго.

## **Организация прерываний.**

### **Основные отличия организации прерываний в защищенном режиме по сравнению с реальным режимом.**

Организация прерываний в реальном режиме практически ничем не отличается от организации прерываний в процессоре Intel 8086. В защищенном режиме, а также в его модификации в виде виртуального режима (V – режима), механизм прерываний при особых случаях, сохранив общую реакцию на их возникновение, значительно усовершенствован.

Основные усовершенствования сводятся к следующим.

- 1) Трансформация таблицы векторов прерываний в дескрипторную таблицу прерываний (IDT – Interrupt Description Table).
- 2) Более сложный процесс перехода к обработчику особого случая или прерывания с привлечением системных объектов в виде шлюзов.

- 3) Возможность передачи обработчику дополнительной информации о причине возникновения особого случая с помощью так называемого кода ошибки (Error Code).
- 4) Использование дополнительных видов особых случаев, связанных исключительно с защищенным режимом, например, неприсутствие страницы или сегмента в ОП, нарушение общей защиты и т.п.

Для защищенного режима используется следующая классификация программных прерываний, которые обобщаются единым наименованием exception, (особый случай). В зависимости от способа возникновения особых случаев и возможности перезапуска (рестарта ЦП) после их обработки с вызвавшей их командой, принято различать три вида особых случаев:

а) *Fault (нарушение)* – особые случаи, которые выявляются и обслуживаются либо перед выполнением, либо во время выполнения “виновной” команды.

При обнаружении нарушения, сохраняемые в стеке значения CS, EIP, указывают на команду, вызвавшую это нарушение, а не на следующую команду программы. Это дает возможность осуществлять рестарт программы после устранения причины нарушения именно с виновной команды.

Типичными примерами нарушений могут служить отсутствие сегмента или страницы в ОП.

б) *Trap (ловушка)* – особый случай, который возникает непосредственно после выполнения команды, вызвавшей этот особый случай. Значение регистров CS и EIP, сохраняемые при срабатывании ловушек, указывают на команду следующую по отношению к команде, вызвавшую это срабатывание.

Типичным примером ловушки может служить ловушка пошагового выполнения программы, ее генератором является установленный флаг TF.

в) *Abort (авария, выход из процесса)* – особый случай, который не позволяет точно локализовать вызвавшую его команду осуществить рестарт программы.

Примером такого особого случая может служить так называемая двойная ошибка, которая имеет место в том случае, когда при обработке нарушений, связанных с неприсутствием сегмента или страницы, возникает новое нарушение. Назначением аварии является устранение возможности бесконечных циклов прерываний.

Сначала шанс выйти из цикла дается программе обработки исключений по двойной ошибке, а если этого не достигается, т. е. при ее выполнении обнаруживается еще одно исключение, происходит аппаратное отключение процессора, из состояния отключения процессор может быть выведен сигналом сброса или немаскируемым прерыванием.

## Программируемый контроллер прерываний (PIC i8259A).

Одна микросхема PIC может обслуживать 8 запросов прерываний. В современных компьютерах на базе процессоров Intel используются две микросхемы PIC, объединенных с помощью так называемого каскадного включения, что позволяет в принципе обслуживать до 15 источников прерывания.

Одна из микросхем является ведущей, а вторая – ведомая. Ведущий PIC связан с CPU, а ведомый PIC с ведущим. Максимальные возможности каскадного включения PIC позволяют обслуживать до 64 внешних источников запросов прерываний. Связь между ведущим PIC и CPU осуществляется по двум линиям: 1-ая линия INT PIC – INTR(CPU), 2 – ая линия INTA (CPU) – INTA (PIC).

### Основные функции PIC.

- 1) Фиксация запросов, поступающих от подключенных к нему ВУ в специальном регистре запросов - IRR.
- 2) Осуществление внутреннего маскирования запросов с помощью специального регистра – маски IMR (0 – разрешение, 1 - запрет).
- 3) Выделение наиболее приоритетного запроса из всех поступивших и незамаскированных запросов.
- 4) Выдача в CPU сигнала о наличии хотя бы одного незамаскированного запроса (по линии 1).
- 5) Выдача в CPU номера (кода) запроса в цикле подтверждения прерывания, который, в свою очередь, модифицируется CPU в адрес вектора прерываний (начальный адрес программы обработчика соответствующего прерываний).

### Внутренняя структура PIC.

В состав PIC входят 7 байтных регистров, основными из которых являются:

- IRR – Interrupt Request Register
- IMR – Interrupt Mask Register
- ISR – Interrupt Service Register (фиксируются запросы, принимаемые на обслуживание или обработку в CPU)

Кроме регистров, в состав PIC входят комбинационные схемы, в частности:

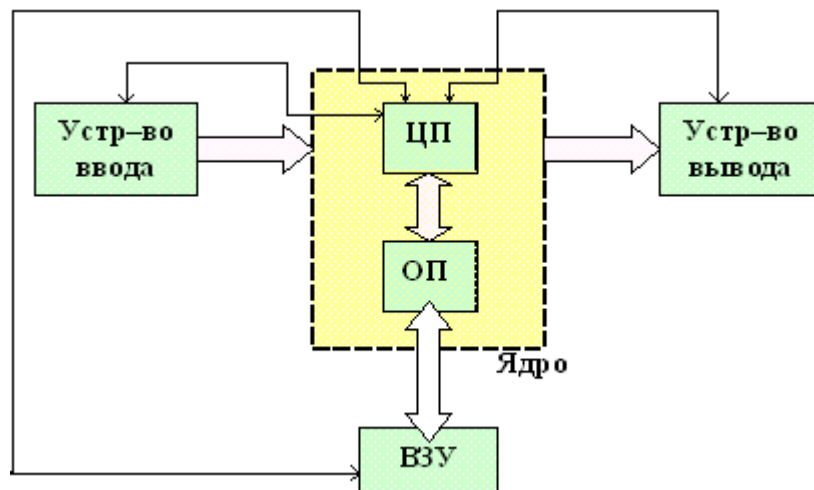
1. Схема выделения наиболее приоритетного незамаскированного запроса.
2. Шифратор выделенного запроса.

Шифратор представляет собой комбинационную схему, осуществляющую преобразование двоичного унитарного кода (код с единственной единицей) в двоичный позиционный код, в данном случае шифратор имеет 8 входов и 3 выхода.

3. Схема для реализации каскадирования.

Внешние запросы о ВУ поступают на входы ir0,..ir7.

### Упрощенная структура компьютера (ЭВМ).



Независимо от принадлежности компьютера некоторому классу или типу, его в первом приближении можно разделить на 2 части:

1. центральную;
2. периферийную.

Центральную часть принято называть *ядром*. В ядро входят два основных устройства компьютера: ЦП и ОП.

Периферийную часть можно условно представить устройствами трех типов:

- внешние запоминающие устройства, которые образуют внешнюю память (к ним относятся накопители на магнитных дисках и на магнитных лентах);
- устройства ввода;
- устройства вывода.

Обмен информацией между ядром и периферией, а так же между устройствами ядра осуществляется на уровне аппаратных интерфейсов. Организация обмена между ядром и периферийной частью компьютера возлагается на систему ввода/вывода (I/O System – Input/Output System). Система ввода/вывода представляет собой аппаратно - программный комплекс.

## **Периферийные (внешние) устройства компьютера.**

Внешние запоминающие устройства образуют внешнюю память компьютера, основным назначением этих устройств является долговременное хранение большого объема программ, данных и другой информации, необходимой для обеспечения функционирования в течение длительного времени. Во внешней памяти хранится практически все программное обеспечение компьютера.

Отличительными особенностями внешней памяти по сравнению с основной памятью являются:

- энергонезависимость;
- файловая организация данных и последовательный доступ к данным в файлах ;
- низкая скорость обмена как следствие использования механики и последовательного доступа;
- неограниченный объем по сравнению с ОП;
- минимальная стоимость хранения данных по сравнению с другими типами памяти.

Основные устройства (ВЗУ) в составе ВП:

1. Накопители на жестких дисках (винчестеры).
2. Накопители на гибких магнитных дисках.
3. Накопители на оптических дисках (CD ROM).
4. Накопители на кассетной магнитной ленте.
5. Устройства флэш-памяти.

Под вводом данных обычно понимается их передача из ПУ в основную память. Под выводом данных – передача данных из ОП в ПУ.

## **Периферийные устройства ввода/вывода.**

Устройства могут выполнять только функции ввода, специальные устройства вывода и могут совмещать эти функции. Стандартная серийная периферия ЭВМ и разнообразные специальные устройства в прикладных системах.

Устройства ввода – клавиатура, устройство считывания с перфоленты и перфокарт, мышь, микрофон, джойстик, сканер. Источниками данных могут быть различные бытовые приборы и устройства – фотокамеры, кинокамеры, видеокамеры

Устройства вывода – принтер, дисплей, перфоратор перфоленты или перфокарт, графопостроитель.

Устройства, совмещающие эти функции – телетайп,

Большое многообразие специальных устройств ввода/вывода – разнообразные датчики в контрольно-измерительных приборах, системах промышленной автоматики – нагреватели, вентиляторы.

Прямое управление элементами и блоками устройств – двигатели шаговые, постоянного тока, переменного тока, реле, пускатели

Непосредственный контроль – состояния контактов, переключателей, датчики положения, преобразователи угол-код,

### Основные способы организации ввода/вывода.

1. Программно-управляемый ввод-вывод (В/В) (PIO).
2. Каналы передачи данных
3. Мультишина в режиме DMA (прямой доступ к памяти).
4. Процессоры В/В

### Ввод/вывод с прямым программным управлением (PIO).

Для первых Неймановских ЭВМ использовалась структура с непосредственными связями и прямое программное управление В/В. Как наиболее простой и быстрый сохраняется во встроенных применениях – микроконтроллерах.

Ввод/вывод PIO осуществляется при непосредственном участии ЦП. Реализация ввода/вывода производится специальной программой (драйвером ВУ), в котором выполняются следующие действия:

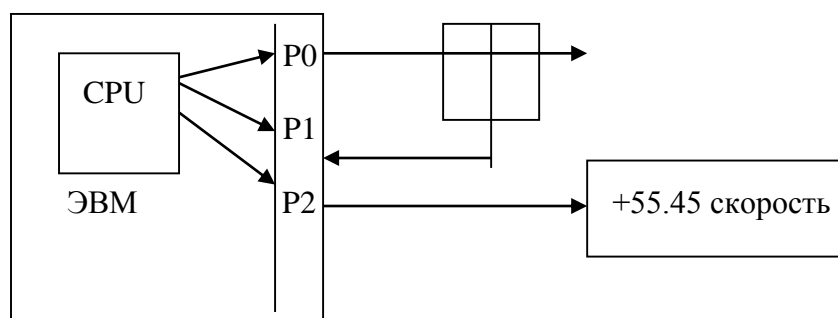
- 6) Пересылка данных между ОП и ВУ включает две стадии – чтение данных из ОП в регистры CPU, запись или чтение данных либо из адресуемых внешних регистров ВУ или непосредственно с датчиков в составе ВУ.
- 7) Преимущественно асинхронный принцип обмена, связанный с использованием механики, электромеханических элементов и существенными не предсказуемыми задержками в переходных процессах управления.

При запуске принтера ожидаем его готовности - разгон двигателя, контроль наличия бумаги в устройстве подачи и др. По готовности передаются команды управления обменом и контролируются результат их выполнения.

- 8) Использование квитирования и/или прерываний в асинхронном обмене

Значительные задержки ожидания случайных по продолжительности и времени возникновения событий завершения переходных процессов в ПУ устраняются использованием **прерываний** по их завершении Процессоры при этом могут параллельно выполнять обработку данных.

Схема включения простых средств управления клавиатурой по прерываниям и ЖКИ-индикатором с микроэвм (контроллером) MCS51 и организация локального пульта/





ЭВМ сканирует строки матрицы клавиш и при нажатии на клавишу принимает запрос прерывания, идентифицируя нажатую клавишу и принимая решение по ее значению. Очевидно, что событие нажатия клавиши случайное и не предсказуемое по длительности. Механическое нажатие клавиши сопровождается дребезгом, который устраняется программным ожиданием.

Стандартный модуль ЖКИ-дисплея включает контроллер управления разверткой изображения на дисплее, использует коды разверток во внутреннем ПЗУ и память для хранения ASCII –кодов идентифицируемых символов. Программное управление ЖКИ включает передачу и выполнение команд адресации по строкам и столбцам, специальные режимы инициализации, сброса и их завершения контролируется либо асинхронно - по сигналам завершения в слове состояния, либо временными задержками.

В современных ЭВМ функции управления клавиатурой отделены от CPU и выполняются специальным устройством со встроенным локальным управлением микроконтроллером, где все вопросы решаются программно и независимо от работы центральной машины.

Это устройство формирует прерывания, когда после нажатия символ уже распознан и его код передается в программу редактирования или командного управления.

Однако и с такой клавиатурой ожидание сохраняется и связано уже с принципиально медленной работой оператора. Высокопроизводительные ЭВМ, выполняя одну эту работу, простаивают и их стараются загрузить параллельно другой работой, а средства ручного ввода и индикации выносят в автономные терминалы под управлением ОС.

Если при этом работают несколько терминалов и некоторые задачи обращаются к внешним файлам, то параллельно-последовательное их прямое обслуживание не возможно и может привести к потерям данных и событий.

Основное назначение В/В – обмен данными с ОП и эти операции могут быть выполнены независимыми друг от друга .

Таким образом, фирма IBM впервые обратила внимание на эти проблемы и нашла, по тому времени, эффективное ее решение.

Управление вводом-выводом с использованием **каналов и мультипрограммная загрузка** ЭВМ обеспечивают до 90% загрузки Процессора и Памяти как наиболее дорогостоящего оборудования ЭВМ.

## **Каналы В/В IBM360/370**

В первую очередь, рутинные операции прямого управления электро-механическими устройствами (включение, выключение, управление двигателями, контроль состояния старт-стопных устройств и т.д.) выделены в **контроллеры Увв. Унифицированы интерфейсы и система кодирования управляющих команд контроллеров**. Таким образом, получена единая схема (интерфейс) включения различных УВВ. После этого выделена **система команд автономного управления и аппаратура** для реализации программного управления Увв в виде **оборудования каналов**.

Фирма IBM в системах IBM360/370 использовала для этой цели систему каналов и мультиплексор коммутации потоков данных.

В IBM360 эта проблема решалась на аппаратном уровне – совместным использованием общего оборудования CPU:

1) выделяется блок мультиплексора данных, в котором коммутируются потоки данных **CPU ↔ Mem, Mem ↔ I/O**.

2) Если в схеме Неймана предполагается прямое управление I/O, что вносит существенную задержку и даже остановку (недогрузку ) Процессора, то в архитектуре IBM для управления Увв организуются процессоры В/В в виде (**Мультиплексных**) и (**Селекторных**) Каналов.

**Каналы** являются программно-независимыми блоками управления с прямым доступом к контроллерам устройств ввода-вывода. Каналы работают по своей **канальной программе**, состоящей из специальных канальных команд, хранящихся в Основной памяти. Эту программу инициирует CPU, выполняя команду ввода-вывода, которая устанавливает связь с Увв через Канал.

Канал проверяет состояние готовности к вводу-выводу, запускает электромеханические элементы Увв (Принтер, Перфоратор, Лента) и, в случае их готовности, обращается в Основную память к собственной программе. Обмен осуществляется **информационными блоками байтов** в режиме **приостановки** Процессора на время быстрого обмена байтами между Памятью и Каналом. Канал параллельно и независимо от Процессора считывает или записывает байт в Увв.

Выполнение канальной программы завершается **прерыванием** от Канала, при этом завершается ввод-вывод или включается новая программа Канала.

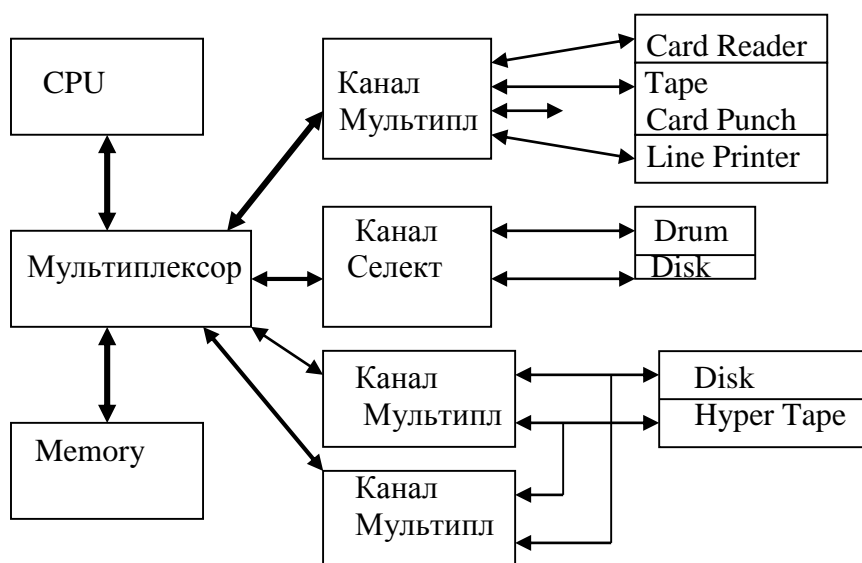


Рис.4. Структура IBM7094

**Приостановка** Процессора или занятие цикла (такта) – выполнение одной микрокоманды записи или чтения байта из ОП. Практически для выполнения программы эта задержка не существенна в случае медленных Увв(Принтер, Лента, Считыватель с перфокарт, Дисплей с клавиатурой,..)

**Мультиплексный канал** может обслуживать несколько Увв последовательно. Часть ресурсов мультиплексного канала, используемая отдельным ВУ, называется подканалом. В каждом подканале используется некоторая область памяти канала, в которой хранится информация о текущем состоянии обмена с данным ВУ. Переключение мультиплексного канала с обслуживания одного ВУ на другое сопровождается сохранением информации в памяти подканала для текущего ВУ и выборкой информации из памяти подканала для следующего ВУ.

В случае быстрых Увв (Диск, Барабан) работает только одно Увв через **Селекторный Канал**. Быстрые схемы Селекторного Канала размещаются в CPU и с учетом разделенного использования БМУ монополизируют и останавливают работу CPU.

Работа канала

- а) Выборка команд канальной программы из ОП, их декодирование и выполнение;
- б) обеспечение приема, передачи, контроля и промежуточного хранения данных при обмене между ОП и ВУ;
- в) формирование текущих адресов ОП, по которым записываются или считываются передаваемые данные;

- г) согласование форматов данных, передающихся по интерфейсу ввода/вывода, с форматом интерфейса ОП (как правило, ширина интерфейса ввода/вывода составляет 1, 2 или 4 байта, что меньше ширины интерфейса ОП: 4, 8, 16 байт);
- д) подсчет числа передаваемых байт данных с целью определения момента завершения передачи блока данных;
- е) выработка последовательности синхронизирующих и управляющих сигналов в соответствии со стандартом интерфейса ввода/вывода;
- ж) анализ особых ситуаций в ВУ во время обмена (ошибка передаваемых данных, сбой устройства и т.п.) и информирование ЦП об этих ситуациях (с помощью запроса прерывания);

Участие ЦП сводится к выполнению следующих функций:

- Инициирование операции ввода/вывода (реализуется командой SIO основной программы).
- Проверка состояния канала ввода/вывода (реализуется командой TCH – Test Channel).
- Проверка состояния ВУ (реализуется командой TIO – Test Input/Output).
- Остановка операций ввода/вывода (реализуется командой HIO – Halt Input/Output) возможно, для инициирования более приоритетной операции.

Команды ввода/вывода являются привилегированными, т.е. могут выполняться только программами операционной системы в режиме Supervisor (SVR).

В дальнейшем такое распределение оборудования и **совмещенное** использование СУ Процессором и несколькими Каналами становится в СБИС не возможным - управление вводом/выводом локализовано в контроллерах, память и контроллеры функционально и конструктивно разделены в современных процессорах и управление каналами заменяет режим **Прямого Доступа** .

В современных архитектурах класса **Main Frame** выделенные процессоры В/В **объединяются с CPU в мультипроцессорную систему и работают с общей или локальной памятью, где хранятся их программы и общие данные.** Процессоры управления вводом/выводом для высокопроизводительных систем позволяют разделить и распараллелить быструю обработку данных и медленные операции ввода.

### Прямой доступ к памяти – DMA (Direct Memory Access).

В начале 70-х годов фирма DEC разработала **Мини-ЭВМ PDP8-11**, принципиально отличающуюся от компьютеров IBM . Архитектура ЭВМ имеет модульную организацию на основе стандартного системного интерфейса **Общая шина:**

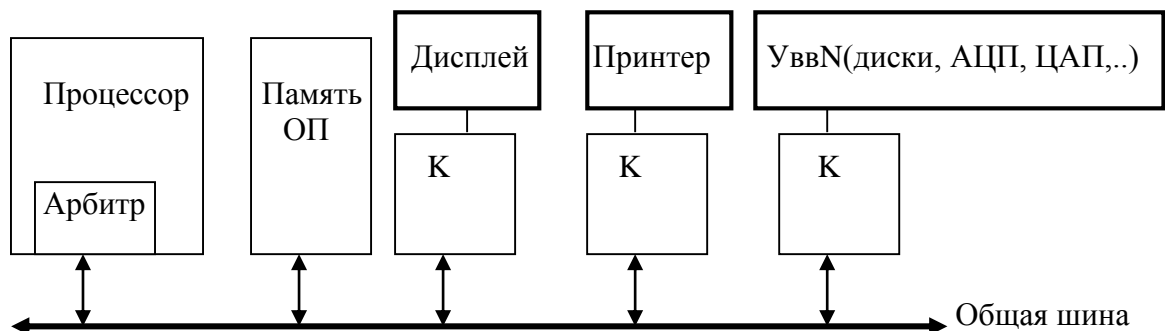


Рис.5. Структура PDP 11.

Процессор включает 16-битное АЛУ, что определяет **основной формат (разрядность)** данных. Стандартная система команд включает арифметические и логические операции с 16-битовыми данными с фиксированной точкой.

**Общая шина – стандартный системный интерфейс**, обеспечивающий единообразное электрическое и конструктивное подключение блоков ЭВМ в виде модулей. Корзина модулей с Общей шиной может быть сконфигурирована пользователем. Шина содержит 16- битную шину данных, 20-бит шину адреса и сигнальные линии, используемые **арбитром** шины для подключения асинхронных запросов к ОП.

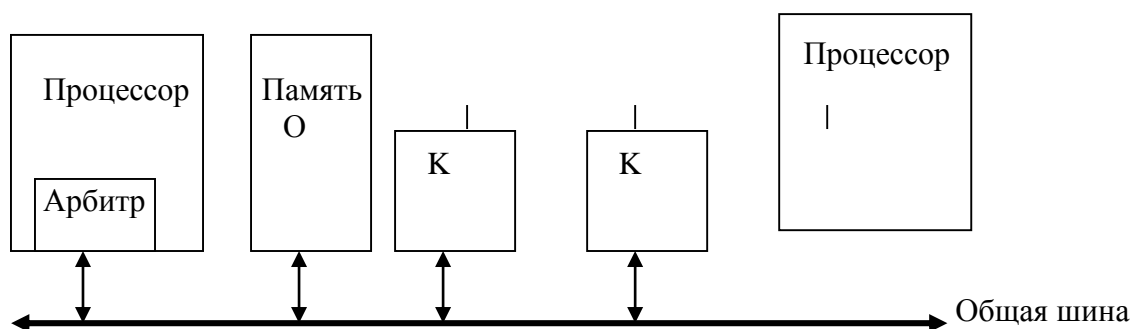
Выделенную линию Захвата Шины с открытым коллектором контролирует Арбитр. Контроллер Увв, захвативший шину, закорачивает ее на себя и освобождает по окончании обмена- устанавливается высокий уровень. Арбитр принимает Запрос шины по линиям Запроса и в соответствии с приоритетами разрешает доступ к шине конкретному Внешнему устройству.

Общая шина допускает обмен данными CPU  $\leftrightarrow$  ОП и Увв  $\leftrightarrow$  ОП в режиме **прямого доступа к памяти(ПДП)** с приостановкой процессора. При этом CPU не участвует в обменах Увв с ОП., обеспечивается высокая скорость обмена данными при минимальных затратах оборудования. Завершение многобайтовых операций ввода-вывода также контролируется по прерыванию и каждый модуль с контроллером ввода-вывода содержит схемы управления ПДП.

ЦП инициализирует DMA и завершает операцию ввода/вывода.

В ОШ могут работать несколько активных устройств (процессоров) и таким образом, реальным фактом стало создание мультипроцессорных или мультимашинных систем.

Потенциальная возможность квазипараллельной работы нескольких процессоров с ОП через общую шину - ограниченная по времени занятость шины при многотактном исполнении команд. Только 2-3 такта 12-тактной команды занята шина обменом данными с процессором, остальные такты шина свободна и может быть захвачена другим активным устройством и процессором в мультипроцессорной системе. Если для каждого процессора выделить локальную память для хранения собственной программы и локальные устройства ввода/вывода, общую память сохранить только для данных, то количество обращений к шине и ее загрузка в мультимашинной системе становится еще меньше и параллельная работа становится более производительной.



Режим DMA используется как для передачи одиночных байтов, так и для блочного обмена. Типичным ВУ с блочным обменом являются накопители на магнитных дисках и магнитных лентах. Управление обменом в режиме DMA осуществляется специальным устройством (микросхемой), называемым контроллером DMA – DMAC.

DMAC содержит некоторое число программно доступных регистров, представленных для ЦП адресуемыми портами ввода/вывода. (гл.8 Цилькер, Орлов)  
Инициализация DMA сводится к заданию режима работы и необходимых адресов путем пересылки требуемой информации из ЦП в регистры контроллера DMA. На этапе инициализации задаются следующие основные данные:

- начальный адрес блока памяти (области памяти), используемого при обмене;
- объем передаваемого блока памяти в байтах (типичный размер блока при обмене с жестким диском составляет 512 байт, длина или объем сектора);
- код операции обмена (ввод или вывод);
- адрес устройства прямого доступа (адрес ВУ задается в связи с тем, что стандартный контроллер DMA включает в себя 8 каналов прямого доступа).

DMA может быть реализован в одном из следующих основных режимов (по Цилькеру):

Название режима	Описание режима
Блочная пересылка	Захват шины на весь период пересылки блока; на весь период DMA ЦП не имеет доступа к шине памяти, в этот период процессор может продолжать работу по программе с обращением к КЭШ – памяти.
Пропуск цикла	После пересылки слова DMAC освобождает шину на один цикл, предоставляя ее ЦП.
Прозрачный режим	DMAC имеет доступ к шине только в тех циклах, в которых ЦП в ней не нуждается.

Архитектура современной ЭВМ с микропроцессором напоминает архитектуру PDP11 – Общую шину заменили другие стандарты – **MultiBus** (Intel), **VME**(Motorolla) и **иерархическая мультишина** ПК. Модули, подключаемые к системной шине, - **микропроцессор, ОП , БИС** сопряжения шины с периферийными устройствами – **параллельные и последовательные порты, контроллер ПЦП и прерываний.**

В процессорах фирмы Intel предусмотрен входной сигнал HOLD(захват шины) и сигнал подтверждения захвата HLDA(Hold Acknowledgment). Сигнал HOLD инициализирует DMAC при начале цикла обмена; ЦП, получив этот сигнал, отключается от шины и выставляет активный уровень сигнала подтверждения HLDA, получив этот сигнал, DMAC начинает цикл блочного обмена.

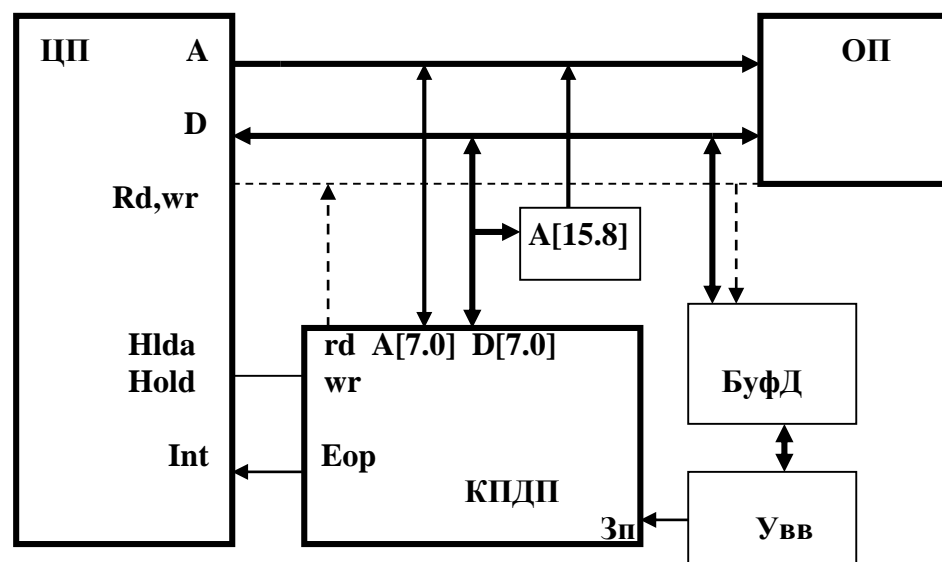
Арбитраж осуществляет специальный внешний модуль контроллера DMA, например, схема **K1810BT37** (отечественный аналог) [ Подклетнов Г.С.] управляет четырьмя каналами.

Каждый канал содержит регистр базового адреса ОП, регистр текущего адреса ОП, счетчик байтов, регистр режима, регистр команд, слово состояния.

**Основные режимы** – чтение, запись, проверка. При выполнении операции выбирается режим инкремента или декремента адреса. Передача – одиночная, блочная и каскадный – расширение числа каналов.

**Команда управления** – разрешение или запрет общего ПДП, приоритет фиксированный или вращение, обмен Память-Память или обычный Память-Увв  
**Регистр маски каналов** – разрешение или запрет i-ого канала.

**Схема включения.**



КПДП инициализируется и программируется ЦП записью управляющих байтов в регистры по адресам на линиях **A[3.0]**. По запросу ПДП из Увв (сигнал **Зп**) формируется сигнал **Hold** и подтверждение от ЦП – **Hlda**. КПДП выставляет адрес памяти за два такта – в первом старшие разряды **A[15.8]** сохраняются в буферном регистре, во втором – младшие разряды адреса **A[7.0]**, сигналы **rd** или **wr**, читаются или записываются данные, разрешенные буферу **БуфД** для Увв. Завершение обменом одиночных байтов – снимается запрос **Hold**. При обмене блоками по сбросу счетчика формируется прерывание сигналом **Eop**.

**Контроллеры DMA позволяют реализовать следующие передачи байтов:**

- Port -> Mem**
- Mem -> Port**
- Mem -> Mem (обмен с видеопамью)**
- Port -> Port**

Контроллеры ПДП берут на себя только непосредственно операции обмена данными и контроль обмена блоками данных. Вместе с тем каналы В/В выполняли и другие необходимые для ввода/вывода вспомогательные операции, например, кодирование и декодирование, упаковку байтов в форматы, согласованные с форматами слова ОП, быструю независимую инициализацию. Эти задачи решали программы каналов, а теперь их должны выполнять CPU.

Функции каналов реализуют специализированные процессоры ввода/вывода.

**Процессор ввода/вывода K1810BM89** (аналог микросхемы фирмы Intel).

## Способы адресации портов ввода/вывода и их сравнительный анализ.

Различие способов адресации связано с использованием отдельного или единого адресного пространства для памяти и портов ввода/вывода.

### Раздельное адресное пространство.



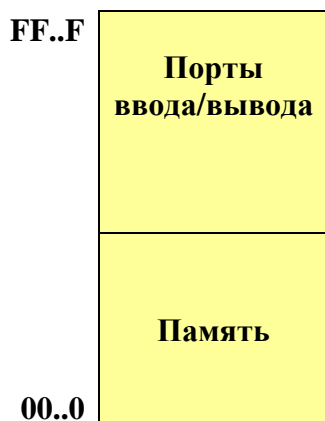
Использование раздельного адресного пространства предполагает, что одни и те же адреса могут применяться как для адресации памяти, так и для адресации портов ввода/вывода. При этом в системе команд процессора должны использоваться специальные команды для ввода/вывода, отличные от команд обмена с памятью. Достоинством подобного подхода принято считать возможность использования более короткого адреса порта ввода/вывода по сравнению с адресом ячейки памяти.

Так, например, в системе команд процессора Intel для адресации портов ввода/вывода может использоваться либо 1 байт (при прямой адресации порта), либо 2 байта (при косвенной адресации порта).

При косвенном задании адреса используется регистр DX (неявно адресуемый), в котором и находится адрес порта ввода/вывода.

### Единое адресное пространство.

Использование единого адресного пространства существенно влияет как на систему команд процессора, так и на управление вводом/выводом на аппаратном уровне.



Некоторая область адресного пространства в старших адресах используется не для адресации памяти, а для адресации портов ввода/вывода. Подобная идея была впервые реализована в мини ЭВМ PDP – 11(DEC). Подобный способ хорошо вписывается в рамки так называемого магистрального интерфейса.

В системе команд отсутствуют специальные команды ввода/вывода, и обмен между памятью и портами ввода/вывода реализуется по аналогии с обменом между процессором и памятью с использованием обычной команды типа MOV.

**Организация ввода/вывода с отображением на память обладает следующими достоинствами:**

- 3) Не требуются специальные команды ввода/вывода (упрощение системы команд).**
- 4) Возможность использования любых видов обработки (реализуется и/или логическими командами применительно к содержимому портов ввода/вывода).**

**Недостатками использование совмещенного адресного пространства являются:**

- 3) Усложнение управления кэшированием, связанное с необходимостью запрета кэширования адресного пространства, выделенного для портов ввода/вывода.**
- 4) Сложность реализации этого способа для многошинной архитектуры.**

Для одношинной архитектуры любой адрес, выставляемый на шину адреса, сравнивается всеми модулями памяти и ввода/вывода на предмет собственной принадлежности. При раздельных шинах памяти и ввода/вывода требуются дополнительные затраты для проверки адреса на его принадлежность к памяти или вводу/выводу.

**Возможные способы решения проблемы:**

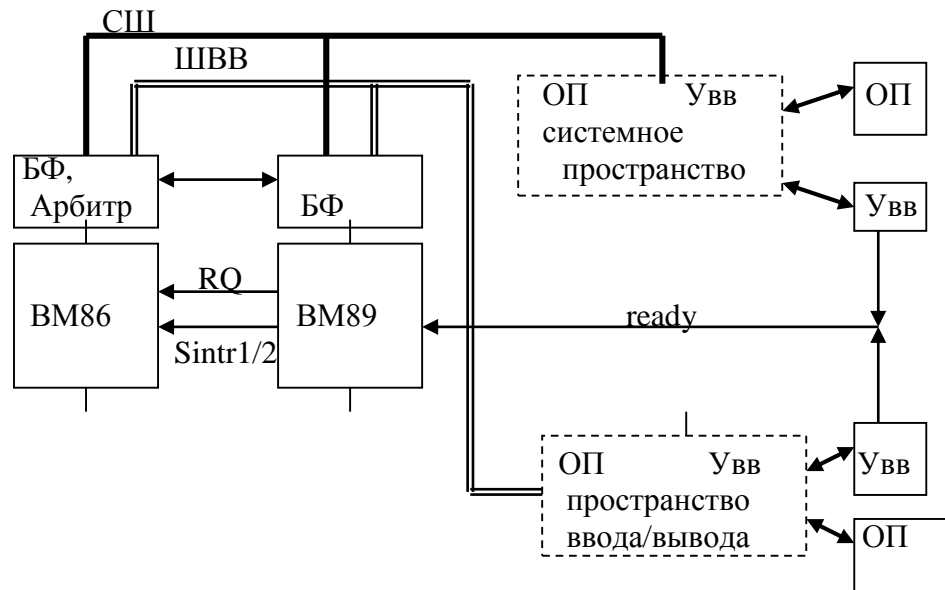
- первоначальный запрос направляется к памяти по быстрой шине, а затем, если память не может ответить на него, то запрос перенаправляется в шину ввода/вывода;
- фильтрация адресов специальной микросхемой, отделяющей адреса памяти от адресов портов ввода/вывода; в частном случае подобную функцию может выполнять мост PCI, в состав которого входят специальные регистры диапазона; при этом все адреса, попадающие в мост и принадлежащие выделенному диапазону, передаются в дальнейшем не к памяти, а непосредственно в шину PCI, которая служит шиной ввода/вывода.

## **ПРОЦЕССОР ВВОДА -ВЫВОДА**

Процессор использовался в мультипроцессорной конфигурации с i86. (Подклетнов Г.С.) Процессор может параллельно работать с двумя каналами ПДП со скоростью 0.28-1.5Мбайт. Процессор имеет 16-битовую шину данных и 20-битовую адресную шину, позволяют сопрягать 8- и 16-битовые шины. Используются две конфигурации включения – локальная и удаленная.

Локальная конфигурация рис.





СШ – системная шина, адресное пространство с Увв, отображаемыми на ОП, 1Мбайт  
 ШВВ – шина ввода/вывода и адресное пространство 64 Кбайт

RQ – сигнал запроса шины.

Sintr1/2 – сигналы запросов прерываний

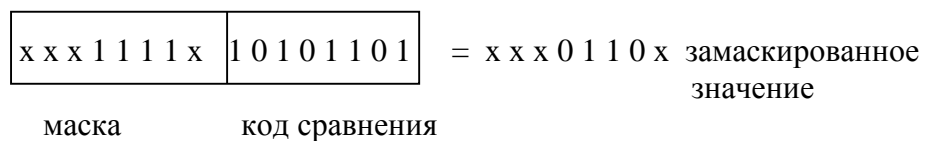
Ready – запрос ПДП

Две Multibus-шины СШ и ШВВ с двумя процессорами, каждый из которых может быть Host, Slave – всегда память ОП или Увв.

Буферами (усилителями) шин управляют два контроллера шин ВГ88 и Арбитр Multibus ВВ89.

Каждый канал в локальной конфигурации может выполнять ПДП-обмен по 8/16 битовой шине данных, выполнять программу, отвечать на запросы готовности или приостанавливаться.

20-битовые регистры каналов содержат 2 адреса (источников или приемников), адрес таблицы перекодировки, счетчик байтов, битовую маску для сравнения выделенных разрядов байта с заданным значением



Регистр управления каналом определяет,

- откуда и куда передаются данные (М-М, Y-M, M-Y, Y-Y)
- перекодировка (да, нет) – производится заменой передаваемого байта через выделенную таблицу 256 байт
- способ синхронизации (асинхронная, синхронная от источника или приемника)
- в каком регистре адрес источника (один из двух регистров)
- одиночный обмен или по счетчику
- как заканчивается обмен – по внешнему сигналу со смещением перехода 0, 4, 8
  - по счетчику со смещением перехода 0, 4, 8
  - по маскированному сравнению со смещением перехода (да, нет)

Команды выбираются из ОП линейно – по байтам . Смещение задается в байтах  
Блок управляющих параметров подготавливается в системном пространстве для каждого канала. ЦП выставляет адрес A[19..1]= сигналы CA,CA запроса готовности канала и вместе с ними сигналы SEL(A[0]) – выбора канала и запуск инициализации

Первая пара сигналов

10 – если используются несколько процессоров, то выбирается как ведущий

11 - ..... как ведомый

Далее процессор В/В вводит последовательно первые байты инициализации процессора

- настройка на заданную шину ШВВ или ШС

- выбирает ширину шины

Вторая пара сигналов

10 – выбран первый канал

11- выбран второй канал

Далее процессор В/В вводит последовательно байты инициализации выбранного канала и начинает выполнять программу ввода/вывода.

Система команд процессора В/В включает 53 команды следующих типов

Mov dst,src

Add dst,src

And dst,src

Or dst,src

Not dst

Setb bit

Clr bit

Call addr

Jmp addr,

Jz(nz,ce,nce,b, nb,) addr

Hlt - останов программы

Xfer - запуск ПДП (завершается автоматически по условиям - прерыванием или

командами перехода, останова)

Адресация косвенная регистровая – через внутренние регистры, длина команды 2-6 байт

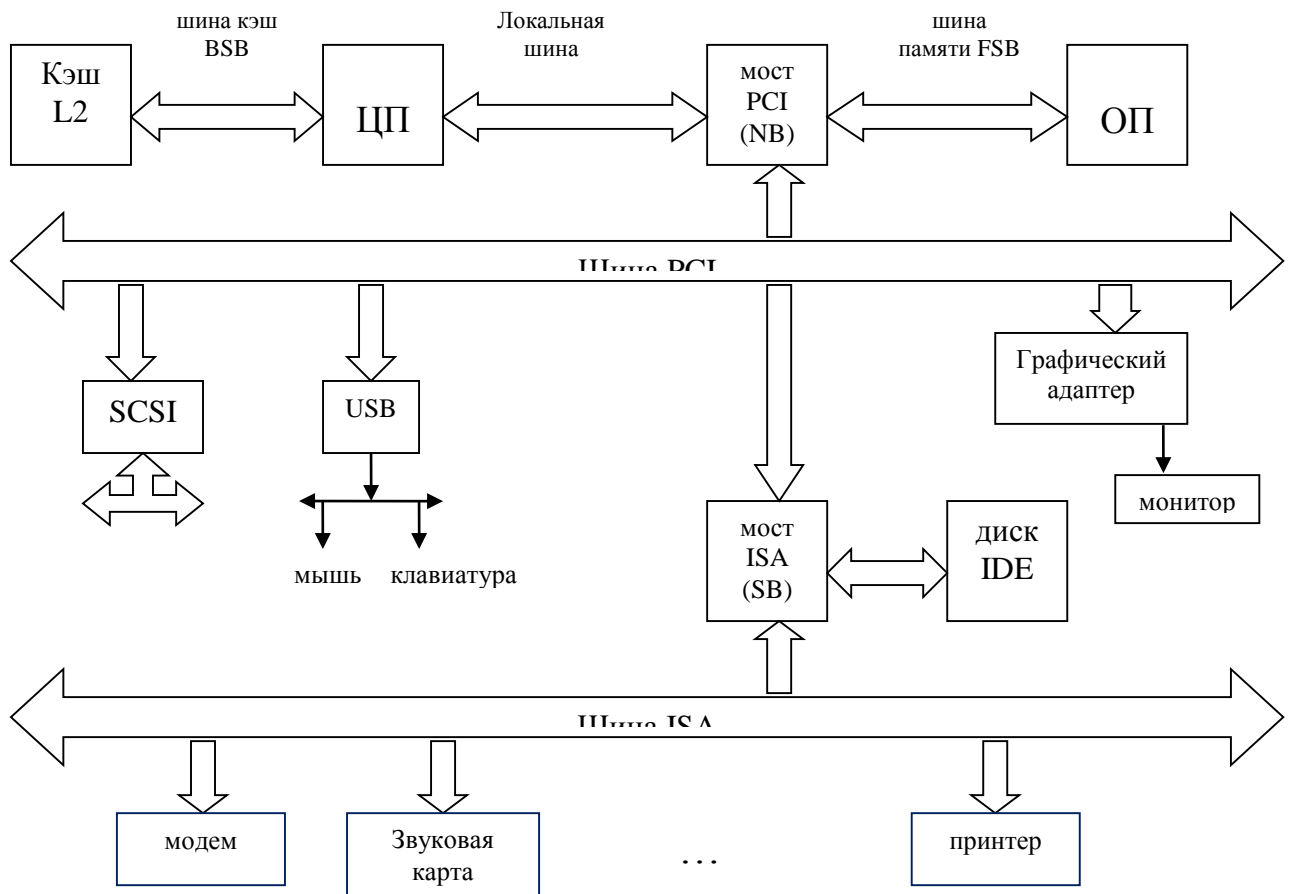
## Иерархия интерфейсов РС на базе процессора Pentium 4

Вследствие наличия существенного недостатка у одноименной структуры компьютера в 80-е годы прошлого столетия в различных моделях компьютера была реализована модификация этой структуры, сначала структура с двумя видами шин, далее структура с тремя видами шин и наконец - многошинная структура компьютера, используемая в современных компьютерах. В структуре с двумя видами шин производится дополнение общей шины (системная шина) дополнительными шинами В/В. Примерами шин (интерфейсов) В/В могут служить:

- SCSI – Small Computer System Interface;
- IDE/ATA – Integrated Drive Electronics(устройство с встроенным контроллером)/AT Attachment (AT – Advanced Technology).

Дальнейшее развитие структуры привело к использованию дополнительного вида шины, называемой *шиной расширения*. Шина расширения является промежуточным звеном между шиной процессор-память (основной шиной) и шиной В/В. Примером шины (интерфейса расширения) может служить PCI (Peripheral Component Interconnect).

### **3.1. Пример многошинной структуры ПК на базе первых моделей процессора Pentium(середина-конец 90-х гг.).**



DIB – Dual Independent Bus – двойная независимая шина:

- FSB – Front Side Bus – шина переднего плана;
- BSB – Back Side Bus – шина заднего плана;

NB – North Bridge

SB – South Bridge

USB – Universal Serial Bus – универсальная последовательная шина;

ISA – Industrial Standart Architecture

По мнению М.Гука мост представляет собой аппаратное средство для соединения шин (разнородных или однородных), мосты входят в состав чипсетов системных плат. Мосты являются программируемыми устройствами. При программировании задается диапазон адресов пространств памяти и В/В, отведенных устройствам, соединяемым мостом шин. Если адрес целого устройства текущей транзакции на одной шине (стороне моста) относится к шине противоположной стороны, мост перенаправляет транзакцию на соответствующую шину и выполняет действие по согласованию протоколов шин. Таким образом, совокупность мостов выполняет маршрутизацию транзакций по связанным шинам.

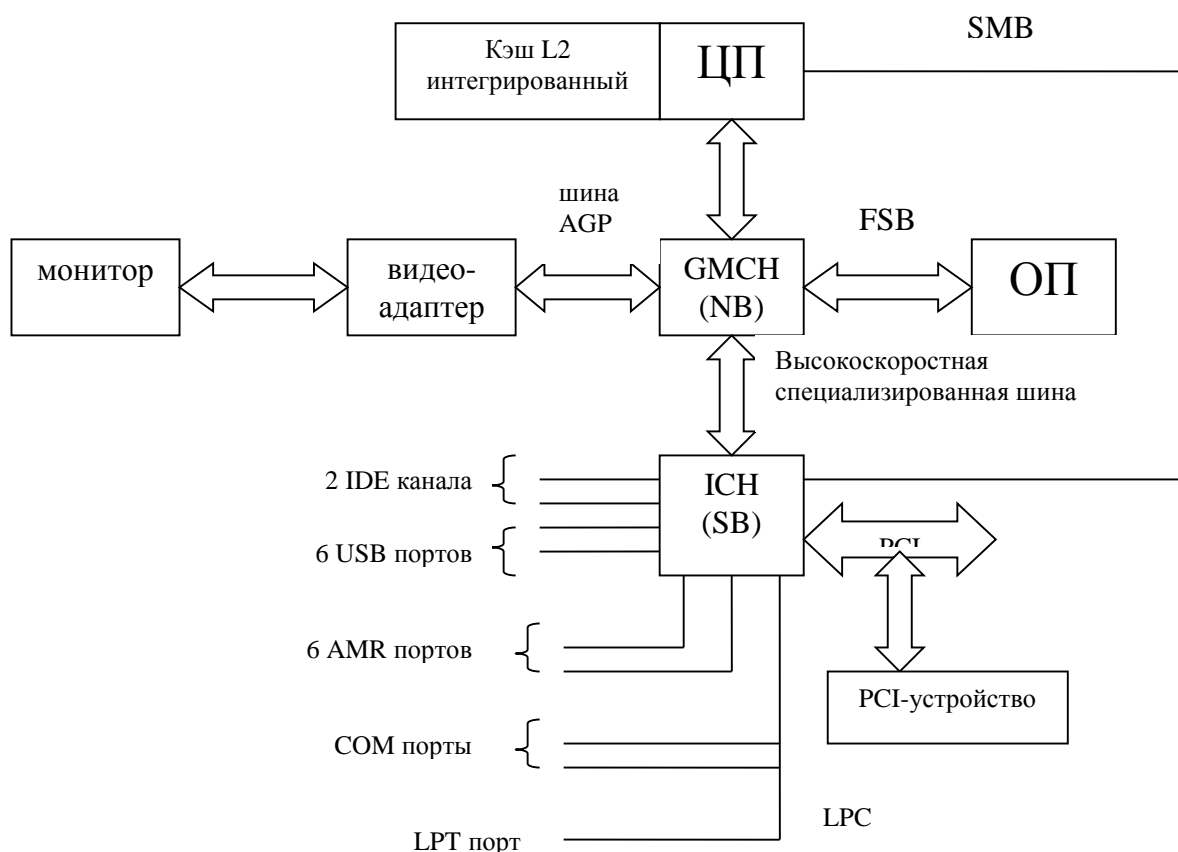
По Гуку: чипсет – набор специализированных интегральных схем, при соединении которых формируется функциональный блок компьютера.

Чипсеты применяются в системных платах, графических контроллерах и других устройствах, функции которых нельзя реализовать в одной микросхеме.

В данной структуре используется 7 видов интерфейсов (шин), из которых 5 (ISA, PCI, IDE, SCSI, USB) являются стандартными и 2 (FSB, BSB) – моделезависимыми (нестандартными).

Основная тенденция в архитектуре высокопроизводительных систем – многоуровневая иерархия интерфейсов, ориентированных на обслуживание обменов, существенно различающихся динамикой (скоростью и интенсивностью)

На рис приведена иерархия интерфейсов PC



Структура современных ПК отличается большим разнообразием используемых шин или интерфейсов.

Название шины	Полное название	Перевод	Последовательная /параллельная	Комментарий
FSB	Front – Side Bus	Шина переднего плана	параллельная	обеспечивает связь между ЦП и ОП
BSB	Back – Side Bus	Шина заднего плана	параллельная	обеспечивает связь между ЦП и внешнего КэшL2, отличается большой пропускной способностью

<b>PCI Intel - 1990</b>	<b>Peripheral Component Interconnect</b>	<b>Соединение периферийных компонент</b>	<b>параллельная</b>	<b>шина расширения</b>
<b>ISA</b>	<b>Industry Standard Architecture</b>	<b>Стандартная промышленная архитектура</b>	<b>параллельная</b>	<b>шина расширения; ISA –16разрядн. EISA-32разрядн. Для подключения принтера, модема, звуковой карты</b>
<b>SCSI</b>	<b>Small Computer System Interface</b>	<b>Интерфейс малых вычислительных систем</b>	<b>параллельная</b>	<b>для подключения периферийных интерфейсов (в частности, магнитных дисков)</b>
<b>USB</b>	<b>Universal Serial Bus</b>	<b>Последовательная универсальная шина</b>	<b>последовательная</b>	<b>для подключения медленных устройств (например клавиатуры)</b>

### **3.1.1. Основные особенности структуры**

1. Расширены функции мостов, в результате чего в рамках приведенной структуры они называются не мостами, а концентраторами (хаб). Для NB используется наименование GMCH – Graphics and Memory Controllers Hub – концентратор контроллеров графики и памяти. Для SB – ICH – Integrated Controllers Hub или Input/Output Controllers Hub – концентратор контроллеров В/В.

2. Связь между хабами GMCH и ICH осуществляется не по стандартизированной шине PCI, как в предыдущей структуре, а по отдельной высокоскоростной специализированной шине (шина является закрытой).

3. Отсутствует шина ISA, вместо нее введена шина LPC – Low Pin Count – малое число контактов, особенностью которой является отсутствие слотов (разъемов).

4. Добавлены AHR-порты для подключения модемов и звуковых карт через интерфейс AC-Link. В соответствии со спецификацией AC'97 – Audio Codec фирмы Intel на архитектуру и параметры звуковых карт существует разделение модемов и звуковых карт на аналоговые и цифровые чипы. Это позволяет создавать дешевые модемы и звуковые карты, на которых присутствует только аналоговый чип. Цифровая обработка реализуется ЦП. Для таких звуковых карт разработан AMR-порт, что означает Audio Modem Riser. Интерфейс порта AC-Link встраивается в хаб, при этом реализуется поддержка шести портов (каналов) AMR.

5. Использование специальной шины AGP для подключения монитора. AGP – Accelerated Graphics Port – ускоренный графический порт. Разработан фирмой Intel в 1997 году, является специализированным портом В/В для реализации высокопроизводительной графики.

AGP предназначен только для подключения видеоадаптера, позволяя ему использовать ОП и избавляя его от необходимости делить с другими устройствами шину PCI. На практике у современных видеоадаптеров имеется большой объем локальной видеопамати,

в результате чего поток данных циркулирует внутри видеоадаптера, слабо нагружая внешнюю шину, однако при построении 3D-изображения видеоадаптеру становится “тесно” в ограниченном объеме видеопамати и его поток данных выходит на внешнюю шину, по составу сигналов напоминающую PCI. Чипсет системной платы через GMCH связывает AGP с ОП через системную шину FSB, не пересекаясь с “узким местом” в виде шины PCI. Ускоренность AGP обеспечивается специальными особенностями принципов ее функционирования:

- 1) конвейеризация обращений к памяти;
- 2) вдвоенная передача данных;
- 3) демультимплексирование шин адреса и данных.

Максимальная пропускная способность шины в режиме счетверенной передачи составляет 1064 Мб/с. Реализация стандарта шины AGP потребовала существенного усложнения чипсета по сравнению с базовым вариантом подключения адаптера к шине PCI, но при этом существенно снизилась нагрузка на шину PCI

## **3.2. Основные характеристики и особенности в стандартных интерфейсах ПК.**

### **3.2.1. ISA/EISA.**

Является разработкой фирмы IBM, как развитие более ранних интерфейсов Microbus, Multibus применительно к ПК IBM PC.

Первоначальная версия ISA включала восьмиразрядную шину данных (ШД) и 20-ти разрядную шину адреса (ША), т.е. предназначалась для ПК на базе процессора Intel 8088. Дальнейшее развитие было произведено в 1984 году в связи с появлением модели (ориентированной на модель) Intel 80286, в этом случае ШД 16-ти разрядная, а ША по сравнению с 86 была увеличена до 24 разрядов. Пропускная способность шины в зависимости от модификации составляет от 4 до 16 Мб/сек.

*Основные недостатки* шины ISA:

1. Неспособность обеспечивать режим автоконфигурирования. В результате этого пользователю приходится вручную устанавливать номера прерываний и адреса устройств, что требует соответствующей квалификации. От этого недостатка свободна, например шина PCI.
2. Низкая пропускная способность шины. Обменяется тем, что передача данных по ней реализована без подтверждения (квитирования), в связи, с чем передача всегда выполняется со скоростью самого медленного устройства.

В связи с этими недостатками согласно спецификации PC'99, принятой фирмами Intel, Microsoft и другими производителями ПК и ОП, шина ISA не должна использоваться в ПК.

Расширением шины ISA является **EISA**, которая имеет ШД и ША по 32 бита. EISA является разработкой большой группы фирм, в которые входит в частности HP, Compaq, NEC и т.д., как альтернатива шине MCA (Micro Chanel Architecture), разработанной фирмой в 1987 году.

Основное достоинство по сравнению с MCA – возможность подключения ранее разработанных для ISA контроллеров (адаптеров) ВУ. Тем не менее, MCA находит в настоящее время применение в мощных файл-серверах, где требуется высоконадежный и производительный В/В.

### **3.2.2. PCI.**

Представляет собой типичный пример шины расширения. Она была разработана фирмой Intel, которая, запатентовав шину, организовала промышленный консорциум PCI SIG, в который вошли все ведущие фирмы в 1990 г.

Занимает особое место в архитектуре ПК, являясь высокоскоростной шиной расширения, соединяет системную шину с шинами (интерфейсами) ВВ.

Именно шина PCI считают своеобразной “центральной шиной (экватором)” структуры ПК при определении наименования моста.

В принципе шина PCI разрабатывалась применительно к ПК на базе процессоров Pentium, однако, являясь независимой от процессора, она находит также применение в компьютерах компании SUN Micro system, в серверах на процессорах Alpha и Rower PC.

Используются *две версии* шины PCI с 32-х и 64-х разрядной шиной данных, 132/264 Мб/с с частотой 33 МГц. Более поздние версии с частотой PCI 2.1 обеспечивают пропускную способность 528 Мб/с на частоте 66 МГц.

Шина PCI сейчас является *самой высокоскоростной* шиной расширения в ПК (не считая AGP – Accelerated Graphic Port), которая используется только для высокоскоростных графических мониторов.

### **3.2.3. IDE (ATA). SCSI.**

При использовании IDE основной контроллер диска встроен в чипсет (входит в состав южного моста), ответная часть контроллера размещена в самом устройстве (накопителе на жестком диске). К интерфейсам IDE можно подключать до четырех устройств.

По сравнению с интерфейсом SCSI, который требует отдельного контроллера. Скоростная возможность IDE мало уступает SCSI, однако IDE дисководов примерно в два раза дешевле. В связи с этим в большинстве случаев в ПК в качестве дискового интерфейса используется IDE. В свою очередь SCSI используется в серверах в качестве интерфейса высокоскоростных дисков, а также сканеров и стримеров (ленточных накопителей).

Различные версии IDE/ATA имеют существенно различающиеся пропускные способности. Одна из первых версий интерфейса 1986 года имела пропускную способность 4 Мб/с. Последняя версия ATA/ATAPI-4 (PI – Package Interface) имеет пропускную способность 66 – 100 Мб/с.

Разработка SCSI – 1986 г., интерфейс стандартизован ANSI – American National Standarts Institute. Используются две основные версии: Narrow (ШД – 8 б.) и Wide (ШД – 16 б.). Более современная версия 32 бита, определена стандартом, однако, обладает высокой стоимостью, из-за чего практически не используется.

Интерфейс использует последовательное (шлейфное) подключение устройств в количестве до 8 или 16, в зависимости от версии. Максимальное удаление до 25 метров.

Одним из устройств, подключенных к шине SCSI, является SCSI-контроллер (хост-адаптер), обеспечивающий связь с шиной SCSI с шиной расширения или системной шиной. С учетом этого фактическое число подключаемых к SCSI внешних устройств равно 7 или 15 в зависимости от модификации.

*Сравнение интерфейсов IDE/ATA и SCSI:*

*Достоинства SCSI:*

- 1) большее число подключаемых ВУ по сравнению с IDE (в IDE до четырех);
- 2) возможность большого удаления подключения устройств до 25 метров (для IDE не более 0,5 метров);
- 3) возможность параллельной работы всех устройств, подключенных к интерфейсу (для IDE может быть активным только одно устройство).

*Недостатки SCSI:*

Высокая стоимость, в частности дорогие кабели, примерно в 2 раза дороже, чем у IDE.

### **3.2.4. USB.**

Является промышленным стандартом расширения архитектуры ПК, ориентированным на интеграцию с телефонией и устройствами бытовой электроники. Шина разработана рядом компьютерных и коммерческих компаний, в число которых входят Intel, Microsoft, HP, Philips в 1996 г.

*Основными целями разработки USB являлись:*

1. Возможность “горячего” подключения (без выключения ПК) различных внешних (находящихся вне корпуса) устройств с низким и средним трафиком (скоростью обмена). К ним относятся: мышь, модем, принтер, клавиатура, джойстик, сканер, цифровая камера, звуковые колонки, цифровой телефон, флэш-память и другие. Шина USB является полным воплощением концепции PnP.
2. Замена коммуникационных портов, для которых характерны более низкие скорости и отсутствие “горячего” подключения.
3. Сокращение общей длины кабелей при подключении ВУ. Каждое USB-устройство может содержать разъем для последовательного подключения в цепочку других устройств. Более того, одно USB-устройство можно сделать концентратором (хабом), к которому можно подключить несколько других устройств.

Шина USB позволяет организовать многоуровневое каскадирование подключенных устройств и обеспечивает логическую топологию “дерево”, вершиной которой является корневой хаб (хост-контроллер).

В иерархии USB шин и устройств имеется единственный управляющий блок в виде хост-контроллера, который подключен к одной из шин расширения компьютера (обычно к PCI). Контроллер USB входит в состав южного моста и является двух-портовым. К каждому порту может быть подключены ВУ или промежуточный хаб при этом допускается до пяти уровней подключения ВУ к хост-контроллеру через промежуточные хабы. Общее число подключаемых устройств к USB может достигать 127.

#### **3.2.4.1. Характеристики USB.**

Версия **USB 1.0** имеет два режима передачи: низкоскоростной, пропускная способность составляет 1,5 Мбит/с и полноскоростной, с пропускной способностью 12 Мбит/с.

Версия **USB 2.0** в высокоскоростном режиме передачи обеспечивает пропускную способность 480 Мбит/с, что существенно расширяет класс устройств, подключаемых к шине.

Программная поддержка в виде драйвера шины вошла в состав ОС Windows 98, что являлось переломным моментом в истории шин.

#### **3.2.4.2. Физическая реализация шины.**

Последовательная шина USB по своей организации существенно отличается от параллельных шин (интерфейсов), в ней нет отдельных линий для данных, адреса управления. Все протокольные функции связаны с обменом по шине, выполняются с помощью одной пары сигнальных проводов путем пересылки определенным образом организованных цепочек байт, называемых пакетами.

Кабель USB состоит всего из четырех проводов. Два из них предназначены для передачи питающего напряжения. Физическая реализация кабеля USB - экранированная витая пара с длиной сегмента до 5 метров для полной скорости передачи или



неэкранированная невитая пара с длиной сегмента до 3-х метров для низкой скорости передачи.

Линии питания в USB обеспечивают подачу питающего напряжения на устройства, подключенные к ней, и не требуют для этих устройств дополнительного источника питания. Передача данных по соответствующим линиям USB осуществляется в полудуплексном режиме.

### **3.3. Функции мостов.**

Северный мост иначе называется системным контроллером (TSC). Северный мост как системный контроллер выполняет функции по взаимодействию и обмену между устройствами, подключенными к нему: ЦП, ОП, внешняя кэш память L2 и шина PCI.

**SB** называется контроллером шин и выполняет следующие *системные функции*:

- 1) организация моста между шинами PCI и ISA с согласованием частот синхронизации;
- 2) реализация высокопроизводительного (обычно двух канального) дискового интерфейса IDE/ATA;
- 3) реализация стандартных для ПК средств для В/В: два контроллера прерывания PIC, два контроллера доступа к памяти DMAC, трехканальный счетчик таймера, логика немаскируемого прерывания NMI;
- 4) коммутация запросов прерывания от устройств на шинах PCI и ISA, а также устройств на материнской плате на входы запросов контроллеров прерывания (PIC);
- 5) коммутация каналов DMA;
- 6) реализация моста с внутренней шиной X-bus используется традиционно в ПК для подключения контроллера клавиатуры, БИС энергонезависимой памяти с BIOS (Flash BIOS), часов реального времени;
- 7) реализация контроллера интерфейса USB;
- 8) поддержка системного мониторинга (управление SM Bus – System Monitoring Bus).

### **3.4. Средства мониторинга. System Monitoring Bus.**

Современные чипсеты включают в состав, как правило, встроенный модуль мониторинга. Реализация мониторинга производится путем непрерывного контроля значений, снимаемых с датчиков температуры (до 8 штук) и напряжения (до 8 штук) и сравнение их с пороговыми значениями. При выходе за пределы включается сигнализация.

Реализация расширенного мониторинга подразумевает использование обратных связей, в частности снижения частоты ЦП до нормализации температуры. Наиболее развитой является обратная связь от датчиков температуры к управлению вентиляторами (до 3 штук).

## 1.2. Общие представления об аппаратных интерфейсах. Понятие интерфейса.

В дальнейшем под интерфейсом будем понимать *аппаратный интерфейс*.

В литературе существует достаточно большое число определений интерфейса. В общем плане под интерфейсом принято понимать *способ сопряжения и взаимодействия между несколькими объектами и субъектами*.

В отношении ЭВМ принято рассматривать множество понятий интерфейсов, например, *аппаратный, программный, пользовательский* и т.д.

В обобщенном плане под *аппаратным интерфейсом* принято понимать совокупность линий и шин электрических схем и алгоритмов, предназначенную для осуществления обмена информацией между устройствами.

*Аппаратные интерфейсы*, используемые в ЭВМ, как правило, обладают свойством *унифицируемости*, подчиняются определенным стандартам.

Унификация затрагивает следующие моменты:

- унифицированность набора линий и шин по составу и назначению;
- унифицированность сигналов и протоколов обмена по линиям (шинам) интерфейса;
- унифицированность конструктивных характеристик средств сопряжения.

Многие авторы отождествляют понятие интерфейса и шины, кроме того, неоднозначность термина тоже имеет место и в отношении шин различных типов. Так, например системная шина достаточно часто называется главной шиной, шиной процессора, шиной памяти и т.п.

В определение интерфейса не вписываются так называемые беспроводные интерфейсы.

Основные виды линий (шин) входящие в состав интерфейсов:

- шина адреса;
- шина данных;
- шина управления (для передачи сигналов управляющих обменом);
- линии синхронизации (для передачи сигналов, синхронизирующих обмен данных по интерфейсу);
- линии запросов прерывания (для передачи сигналов прерываний от ВУ подключенных к интерфейсу в ЦП или РС);
- линии разрешения прерывания (для передачи сигналов разрешения от ЦП или РС к ВУ);
- линии питания;
- линии заземления.

Общее число линий в современных интерфейсах составляет ~150-200.

### 2.3.1. Уровни представления интерфейсов.

1) **логический**: определяет состав, наименование и предназначение линий (шин), а также порядок передачи информации (сигналов) по этим линиям (протокол обмена, который обычно представляется в виде временных диаграмм).

2) **физический**: определяет параметры сигналов (электрических, оптических), переданных по линиям интерфейсов;

3) **конструктивный**: определяет физическую реализацию шин интерфейсов (печатные проводники, витая пара, коаксиальный кабель), а также виды разъемов и распределения линий интерфейсов по контактам разъемов.

### 2.3.2. Классификация интерфейсов.

1. *По способу соединения компонент:*

- магистральный;
- радиальный;
- цепочный;
- комбинированный (смешанный).

С помощью *цепочного* интерфейса обычно реализуются линии управления, которые проходят последовательно через ряд подключенных к ним ВУ (в частности, сигналы разрешения). *Сигнал разрешения*, проходя последовательно через подключенные к интерфейсу ВУ, может быть заблокирован первым из ВУ на этой линии, которая предварительно послала запрос прерывания.

Реализация цепочного интерфейса предоставляет преимущество в обслуживании (приоритет) тем устройствам, которые находятся ближе по электрической связи к источнику сигнала разрешения. Этим источником, как правило, является **арбитр** – специализированный блок, входящий в состав ЦП. Как правило, линии цепочного интерфейса объединяются с линиями магистрального интерфейса, образуя комбинированный.

2. *По способу передачи информации:*

- параллельные;
- последовательные;
- параллельно-последовательные.

3. *По принципу обмена:*

- синхронный;
- асинхронный.

4. *По режиму передачи информации:*

- односторонний (симплексный);
- двухсторонний (дуплексный);
- двухсторонний - поочередный (полудуплексный).

5. *По функциональному назначению:*

- системные;
- интерфейсы периферийных устройств (малые интерфейсы);
- интерфейсы ввода/вывода;
- интерфейсы программно-управляемых модульных систем и приборов (приборные).

### 2.3.3. Основные характеристики интерфейсов.

1. **Пропускная способность.** Определяется максимальным количеством бит (чаще байт), передаваемых по интерфейсу за единицу времени (за 1 сек.).

2. **Информационная ширина.** Определяется числом бит (реже байт), переданных по линиям интерфейса параллельно (разрядность шины данных).

3. **Максимальное возможное удаление устройств подключенных к устройству.**

Подобную структуру называют также иерархической, так как организация ввода/вывода в ней реализована по иерархической схеме: ЦП      КВВ      ВУ.

В свое время подобная структура являлась типичной для больших универсальных ЭВМ, основные из которых являлись разработками фирмы IBM 360/370/390. Немного позднее в 90-е годы ЭВМ подобного типа получили название Main Frame.

### **3.5. Сравнение канального ввода/вывода с Programmable I/O.**

Так как канал В/В осуществляет организацию обмена с ВУ по собственной программе, то КВВ следует считать программно управляемым, однако, в отличие от PIO программу, связанную с обменом, выполняет не ЦП, а специализированный процессор – КВВ.

### **3.6. Сравнение с DMA.**

Многие специалисты сопоставляют (как синонимы) канальный В/В и DMA. Аналогия между ними состоит в том, что оба этих способа организации В/В реализуются практически без участия ЦП. Так же как и для DMA КВВ требует некоторого участия ЦП лишь на этапе инициализации В/В, при этом из ЦП в КВВ передается начальный адрес программы в памяти, а также при особых ситуациях в работе канала или ВУ. Существенным отличием КВВ от DMA является программная реализация первого и чисто аппаратная второго.

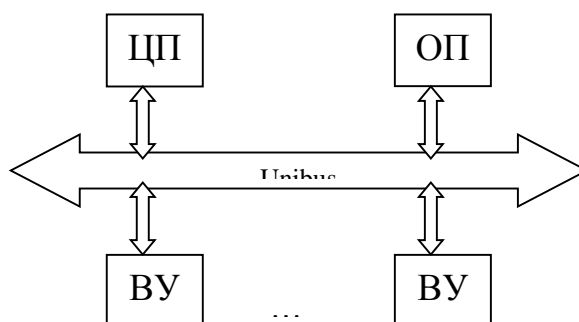
## 4. Магистральная структура компьютера

Такая структура является типичной для мини- и микро- ЭВМ, в том числе и персональных компьютеров 70-80 гг. XX века.

В качестве примеров реализации единого интерфейса можно привести следующие:

- omnibus (PDP - 8) – DEC;
- unibus (PDP – 11) – DEC;
- multibus (Intel 8086);
- IBM PC (PC/XT) – 8086;
- ISA (PC/AT) – INT 80286.

### 4.1. Обобщенная структура компьютера с общей шиной.



*Основные особенности:*

1. Для связи между любыми компонентами используются одни и те же линии интерфейса (ША, ШД, ШУ и т.п.), а также сигналы, управляющие передачей. Этот факт в значительной степени упрощает организации связи между устройством и обеспечивает простоту наращивания структуры путем подключения дополнительных устройств.

2. Использование единого интерфейса предполагает, что в любой момент времени по нему может быть организован обмен только между двумя устройствами, одно из которых является ведущим, а другое исполнительным, ведомым. Ведущим устройством не может быть ОП. Подобная структура является источником конфликтов между различными активными устройствами, требующими практически одновременного взаимодействия с шиной для передачи данных. Компьютеры с подобной структурой не предполагают значительного наращивания числа устройств. Подобная структура для увеличения производительности требует введения дополнительных шин для разгрузки основной.

3. В компьютерах с общей шиной могут быть реализованы различные способы организации В/В, такие как PIO, В/В по прерыванию, а также В/В в режиме DMA.

В структуре с общей шиной могут быть реализованы оба подхода (в рамках конкретной модели – один из них) к адресации ВУ (также портов В/В):

- 1) использование отдельного адресного пространства для памяти и портов В/В;
- 2) использование единого адресного пространства.

Первый способ адресации является типичным для ПК, второй типичным для PIC.

Использование единого адресного пространства для памяти и портов В/В предполагает единообразие операций обмена с памятью и ВУ. Это означает, что в системе команд компьютеров, предполагающих использование единого адресного пространства, отсутствуют специальные команды В/В (типа IN и OUT). Это означает, что В/В, т.е. пересылка данных из регистра процессора в регистр контроллера ВУ и в обратном направлении реализуются той же универсальной командой (типа MOVE), как и обмен между процессором и памятью.

## 4.2. Усовершенствования структуры с единым интерфейсом.

### 4.3. Основные недостатки многошинной структуры ПК на базе процессора Pentium.

1. Использование для связей мостов NB и SB сравнительно низкоскоростной шины PCI.
2. Использование морально устаревшей шины ISA в качестве дополнительной шины расширения для подключения некоторых низкоскоростных устройств.

### Понятие, основные характеристики и уровни представления интерфейса.

В общем плане под интерфейсом принято понимать способ сопряжения и взаимодействия между несколькими объектами. В отношении компьютеров принято рассматривать множество понятий интерфейса: аппаратный, программный, пользовательский. В отношении аппаратных интерфейсов используются следующие понятия: интерфейс памяти, интерфейс ввода/вывода, интерфейс периферийных устройств (малый интерфейс). Существует большее количество подходов к определению аппаратного интерфейса. Основными компонентами в различных понятиях аппаратного интерфейса, являются:

1) Совокупность линий, шин, обеспечивающих обмен информацией между устройствами.

2) Алгоритм (протокол) обмена, определяющий последовательность организации передачи информации по линиям интерфейса.

3) Разделение интерфейса на ряд уровней представлений.

На обобщённой структуре двойными линиями обозначаются структуры, по которым осуществляются передача информации данных между компонентами компьютера, а одинарными – обозначаются связи по управлению. Тем самым подчёркивается, что центральный процессор выполняет в компьютере двойную функцию: как устройство обработки (выполняет заданные программы) и как устройство управления всеми компонентами компьютера.

От ЦП к остальным устройствам по линиям связи передаются управляющие сигналы (приказы, команды в/в); в обратную сторону передаются сигналы о состоянии устройств, в частности об их готовности к обмену, а также запросы прерываний (например, для идентификации момента завершения операции в/в).

Основными типами линий (шин), входящих в состав аппаратного интерфейса, являются:

- ⇒ шина адреса;
- ⇒ шина данных;
- ⇒ шина управления (для передачи сигналов, управляющих обменом);
- ⇒ линии синхронизации (по шинам передаются сигналы, синхронизирующие передачу информации по интерфейсу);

- ⇒ линии запросов прерываний (по ним передаются сигналы прерывания от устройств, подключаемых к интерфейсу);
- ⇒ линии питания;
- ⇒ линии заземления.

#### **Основные характеристики интерфейса:**

4. Пропускная способность определяется максимальным количеством бит или байт данных, передаваемых по интерфейсу за одну секунду.
5. Информационная ширина (количество бит или байт данных, передаваемых параллельно по шине данных, т.е. разрядность линии).
6. Максимально возможное удаление устройств, подключаемых к интерфейсу.

Алгоритм (протокол) обмена обычно представляется с помощью временных диаграмм. Определяется порядок следования и допустимые параметры (амплитуда, длительность) для управляющих и информационных сигналов при работе интерфейсов в различных режимах.

#### **Уровни представления интерфейсов:**

- Логический уровень определяет состав, наименование, назначение шин интерфейса, а также порядок передачи информации по этим линиям (протокол обмена).
- Физический уровень определяется параметрами электрических, оптических сигналов, передаваемых по линиям интерфейса.
- Конструктивный уровень определяет физическую реализацию шин интерфейса: скрученная (витая) пара, коаксиальный кабель; а также определяет виды разъемов и распределение линий интерфейсов по контактам разъема.

### **Шины (интерфейсы) персональных компьютеров на базе процессоров Pentium.**

#### **Основные аспекты организации ввода/вывода.**

**1. Структура компьютера в плане организации связей между ядром и периферийными устройствами:**

- а) структура с единым интерфейсом (с магистральным интерфейсом, с общей шиной);
- б) многошинная структура, рис. 1.11 (Таненбаум);
- в) структура с каналами (процессорами) ввода/вывода (Цилькер);

**2. Адресация к ВУ или ПУ.** Основным аспектом, связанным с адресацией ВУ, является объединение или разделение адресных пространств памяти и ввода/вывода.

**3. Способ организации ввода/вывода:**

- а) программный (программно управляемый, программируемый) ввод/вывод;
- б) ввод/вывод по прерыванию (управляемый прерываниями);
- в) ввод/вывод с использованием прямого доступа к памяти;
- г) канальный ввод/вывод.

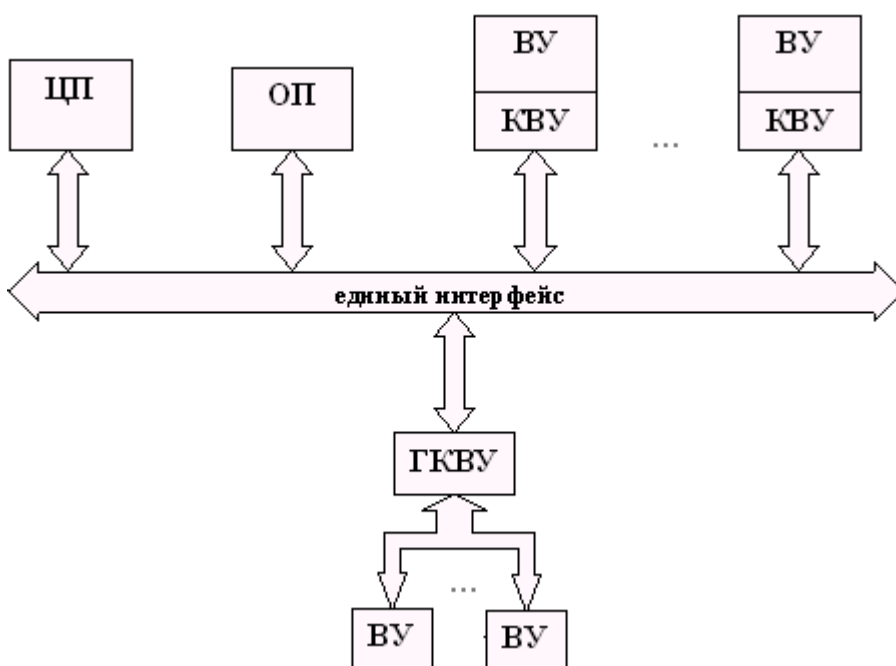
#### **Упрощенная структура компьютера с единым интерфейсом.**

Ещё в конце прошлого века подобная структура являлась канонической и стандартной для большинства моделей мини ЭВМ, микро ЭВМ и ПК. Примером единого интерфейса может служить стандартный интерфейс Unibus, который использовался в компьютерах фирмы DEC (PDP – 11, VAX – 11, CM ЭВМ).

Основными особенностями компьютеров с единым интерфейсом являются:

- 1) все устройства, как центральные, так и периферийные, для связи между собой используют одни и те же шины адреса, данных и управления;
- 2) в любой момент времени по единому интерфейсу может быть организована передача данных только между двумя устройствами;
- 3) в соответствии с п.2 при наличии большого числа устройств, единый интерфейс становится “узким местом” (bottle neck), в связи с чем подобная структура совершенствовалась путём использования локальных дополнительных шин;
- 4) как правило, использование единого интерфейса предполагает единообразие операции с памятью (чтение и запись) и ввода/вывода, в связи с этим предполагается использование объединённого адресного пространства для памяти и ввода/вывода (ввод/вывод, отображённый на память).

В современных ПК подобная структура не используется.



## Адресация ВУ.

Адресация собственно ВУ в современных компьютерах используется достаточно редко. Примером использования фактического адреса ВУ могут служить команды ввода/вывода IBM 370. В современных ПК адресация ВУ осуществляется на уровне программно доступных регистров контролеров ВУ, которые называются портами ввода/вывода.

## Способы организации ввода/вывода.



КУА

### Сравнение ПВВ с РЮ и с DMA.

Так как КВВ осуществляет организацию обмена по собственной программе, то КВВ следует считать программно управляемой, однако, в отличие от РЮ, программу выполняет не ЦП, а КВВ.

Многие авторы сопоставляют КВВ и DMA. Аналогия между КВВ и DMA состоит в том, что оба эти способа обмена реализуются практически без участия ЦП. Так же, как и для DMA, КВВ требует участия ЦП на этапе инициализации. В частности, при инициализации от ЦП в КВВ передается начальный адрес канальной программы. Существенным же отличием КВВ от DMA является программная реализация первого и чисто аппаратная второго.