

Раздел 1. Лекция 2

Аппаратная и программная
поддержка интерфейсов

Основные вопросы лекции

1. Основные принципы программирования доступа к периферийным устройствам.
2. Методы управления обменом.
3. Регистровая программная модель периферийного устройства. Особенности адресации.
4. Аппаратные средства поддержки работы периферийных устройств: контроллеры, адаптеры, мосты.
5. BIOS.
6. UEFI.

1. Основные принципы программирования доступа к периферийным устройства

Разбиение ввода-вывода на несколько уровней, причем нижние – учитывают особенности аппаратуры, верхние – обеспечивают удобный интерфейс для пользователей.

- *независимость от устройств;*
- *обработка ошибок как можно ближе к аппаратуре;*
- *использование блокирующих (синхронных) и неблокирующих (асинхронных) передач;*
- *одни устройства являются разделяемыми, а другие – выделенными.*

1. Основные принципы программирования доступа к периферийным устройства

Для решения поставленных проблем целесообразно разделить программное обеспечение ввода-вывода на четыре слоя:



1. Пользовательский слой программного обеспечения.
2. Независимый от устройств слой операционной системы.
3. Драйверы устройств.
4. Обработка прерываний.

1. Обмен данными с ПУ

Программирование доступа к ПУ в общем случае является нетривиальной задачей, даже если не касаться особенностей работы с ПУ, связанных с архитектурой ОС (которая в общем случае реализует виртуализацию ПУ через систему драйверов). Единого интерфейса программирования (API) для работы с ПУ не существует, зачастую даже стандартный интерфейс для определенного типа устройств разрабатывается не сразу.

Ранее разработчики ПО полагались на API, предоставляемый системным BIOS (или BIOS самого устройства), а в сложных случаях прибегали к «ручному» программированию устройства. Однако в многозадачных средах такой подход не работает – требуется обеспечить множественный доступ к одному и тому же устройству. Реализуется это либо программно, через *драйверы*, либо через *интеллектуальный хост-контроллер*, функции которого распределены между «железом» и драйверами.

1. Методы управления вводом/выводом

В ВС находят применение три способа организации ввода/вывода:

- программно-управляемый ввод/вывод;
- ввод/вывод по прерываниям;
- прямой доступ к памяти.

1. Программно-управляемый ввод/вывод

Наиболее простым методом обмена является *программно-управляемый доступ* (*программный доступ, ввод/вывод с опросом*), или PIO.

Управляет обменом (определяет моменты передачи данных, подает адреса и т.д.) *процессор*, чаще всего центральный (но может быть и выделенный процессор ввода-вывода). При этом фактически происходит пересылка данных между регистрами процессора и регистрами/памятью ПУ (или контроллера интерфейса).

1. Ввод-вывод по прерываниям

ЦП выполняет программу, которая обеспечивает прямое управление процессом ввода/вывода, включая проверку состояния устройства, выдачу команд ввода или вывода. Выдав на шину ВВ команду, центральный процессор должен ожидать завершения ее выполнения.

В такой программе будет следующая последовательность действий:

1. ЦП с помощью команды ввода/вывода сообщает модулю ввода/вывода, а через него и ПУ о предстоящей операции. Адрес модуля и ПУ, к которому производится обращение, указывается в адресной части команды ввода или вывода.
2. Модуль исполняет затребованное действие, после чего устанавливает в единицу соответствующий бит в своем регистре состояния. Ничего другого, чтобы уведомить ЦП, модуль не предпринимает.
3. Для определения момента завершения операции или пересылки очередного элемента блока данных процессор должен периодически опрашивать и анализировать содержимое регистра состояния шины (канала, магистрали) ВВ.

1. Команды ввода-вывода

Данные читаются пословно. Для каждого читаемого слова ЦП должен оставаться в *цикле проверки*, пока не определит, что слово находится в регистре данных (РД) модуля Ввода-Вывода, то есть доступно для считывания.

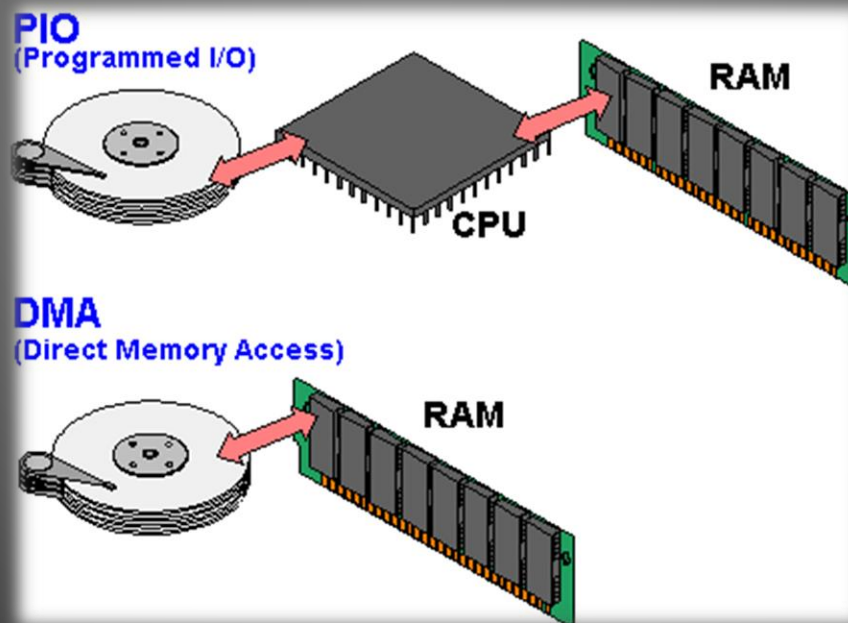
Существуют четыре типа команд ввода-вывода, которые может получить модуль Ввода-Вывода:

1. Команда управления.
2. Команда проверки.
3. Команда чтения.
4. Команда записи.

2. Методы управления обменом

Выделяют несколько методов:

1. Программно-управляемый доступ, или PIO (Programmed input/output).
2. Прямой доступ к памяти, или DMA (Direct memory access)/



2. PIO

Принцип работы: управляет обменом процессор, чаще всего центральный.

Преимущество PIO — в простоте аппаратной реализации ПУ. Требуется обеспечить лишь выставление на шину / чтение с шины содержимого регистров или ячеек памяти по сигналу доступа.

Недостаток — в низком быстродействии и необходимости задействовать процессор, который в общем случае будет простаивать ввиду более высокого быстродействия по сравнению с ПУ.

Текущее состояние: устаревший режим, так как для его работы ему необходимо задействовать центральный процессор, что приводит к значительной потере производительности.

2. DMA

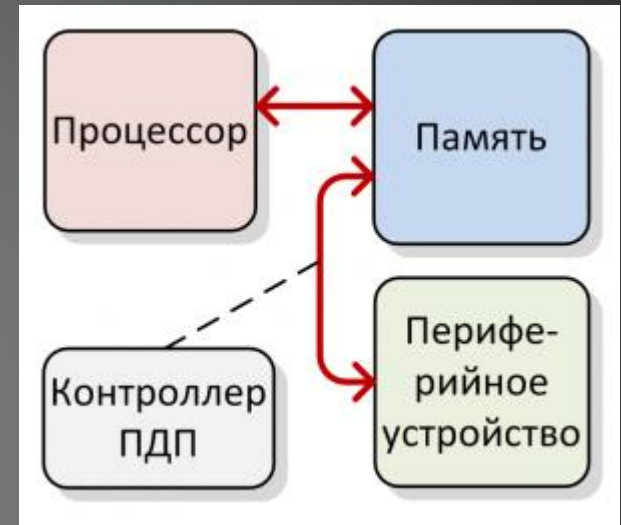
Принцип работы: позволяет выполнять обмен между ОП системы и ресурсами ПУ асинхронно по отношению к вычислительному процессору.

Преимущество: В этом режиме обмен данными между ПУ и основной памятью компьютера происходит без участия процессора.

Недостаток: Не работает в защищенном режиме.

Обменом в режиме DMA управляет не программа, выполняемая процессором, а электронные схемы, внешние по отношению к процессору. Обычно схемы, управляющие обменом в режиме DMA, размещаются в специальном контроллере, который называется *контроллером прямого доступа к памяти*.

Контроллер DMA может быть как общесистемным, так и входить в состав ПУ. Его требуется запрограммировать на пересылку данных между двумя адресатами, после чего он самостоятельно вырабатывает сигналы передачи данных.



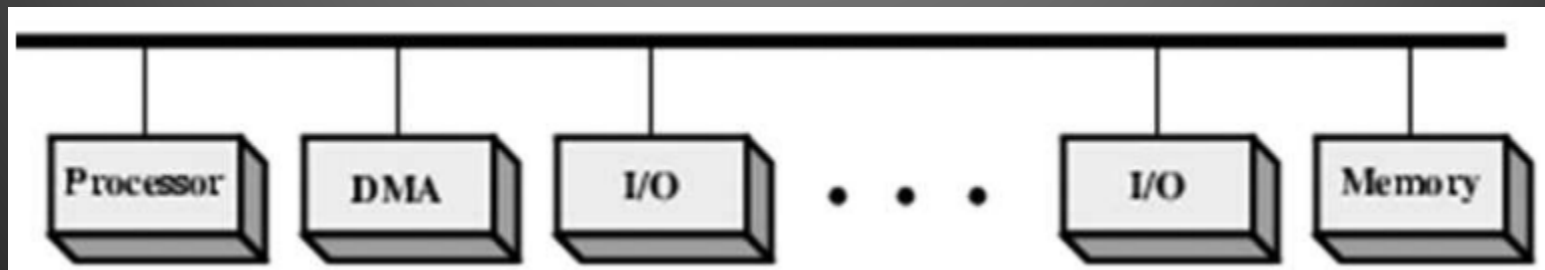
2. Схема 1 подключения DMA

К общей шине подключается контроллер DMA и все вовлеченные в операцию устройства.

В каждой транзакции шина используется дважды:

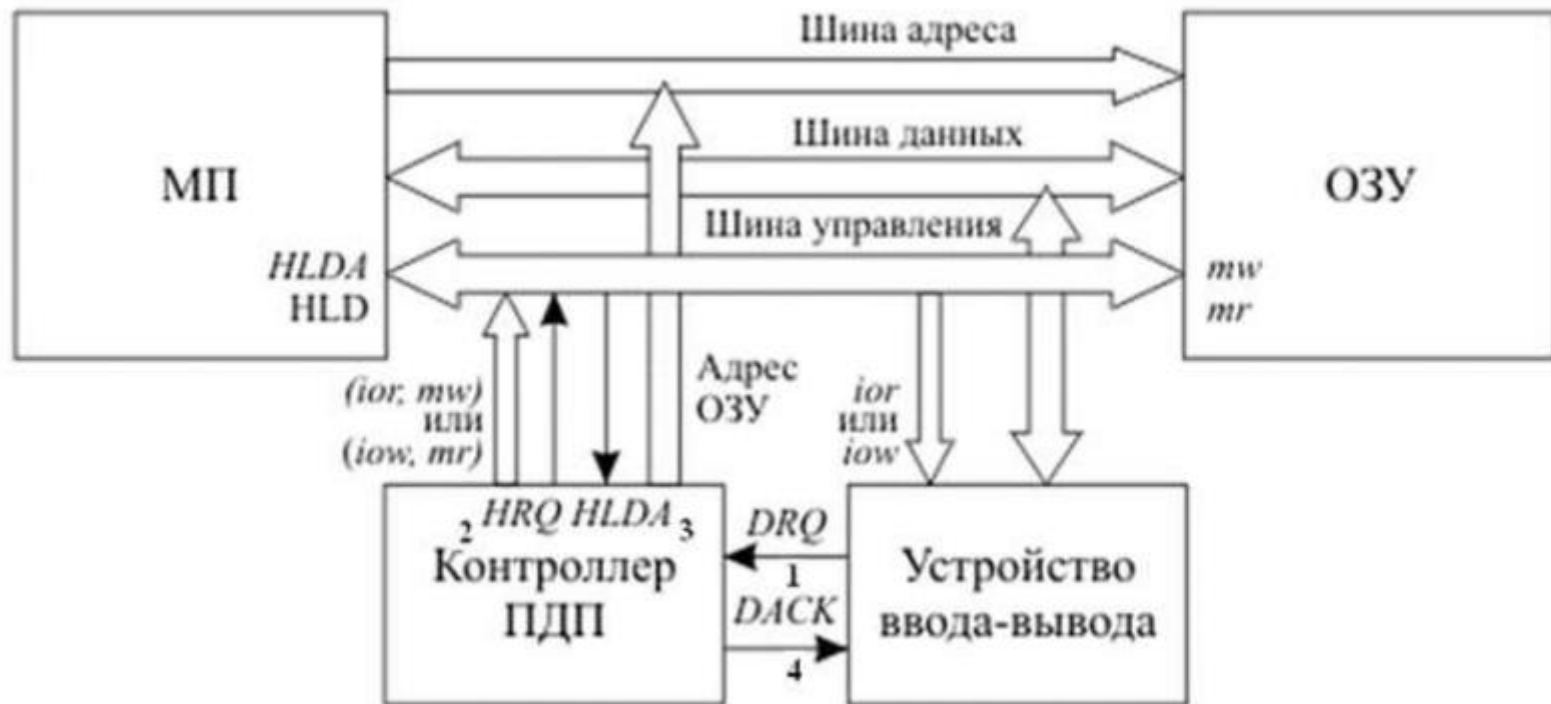
- 1) I/O к DMA;
- 2) DMA к ОЗУ.

При передаче на шине CPU дважды приостанавливает свою работу.



2. Структура ЭВМ с CDMA (или КПДП)

На схеме представлен обмен данными в режиме DMA.



2. Обмен данными в режиме DMA

Последовательность действий КПДП при запросе на *прямой доступ к памяти* со стороны устройства ввода-вывода:

1. **Принять запрос на ПДП** (сигнал DRQ) от УВВ.
2. **Сформировать запрос к микропроцессору на захват шин** (сигнал HRQ).
3. **Принять сигнал от микропроцессора (HLDA), подтверждающий факт перевода микропроцессором своих шин в третье состояние.**
4. **Сформировать сигнал, сообщаящий устройству ввода-вывода о начале выполнения циклов *прямого доступа к памяти* (DACK).**

2. Обмен данными в режиме DMA

Последовательность действий КПДП при запросе на *прямой доступ к памяти* со стороны устройства ввода-вывода:

5. Сформировать на шине адреса компьютера адрес ячейки памяти, предназначенной для обмена.
6. Выработать сигналы, обеспечивающие управление обменом (IOR, MW для передачи данных из оперативной памяти в УВВ).
7. Уменьшить значение в счетчике данных на длину переданных данных.
8. Проверить условие окончания сеанса прямого доступа (обнуление счетчика данных или снятие сигнала запроса на DMA).

Если условие окончания не выполнено, то изменить адрес в регистре текущего адреса на длину переданных данных и повторить шаги 5-8.

2. Регистры контроллера

Контроллер DMA имеет в своем составе такие элементы:

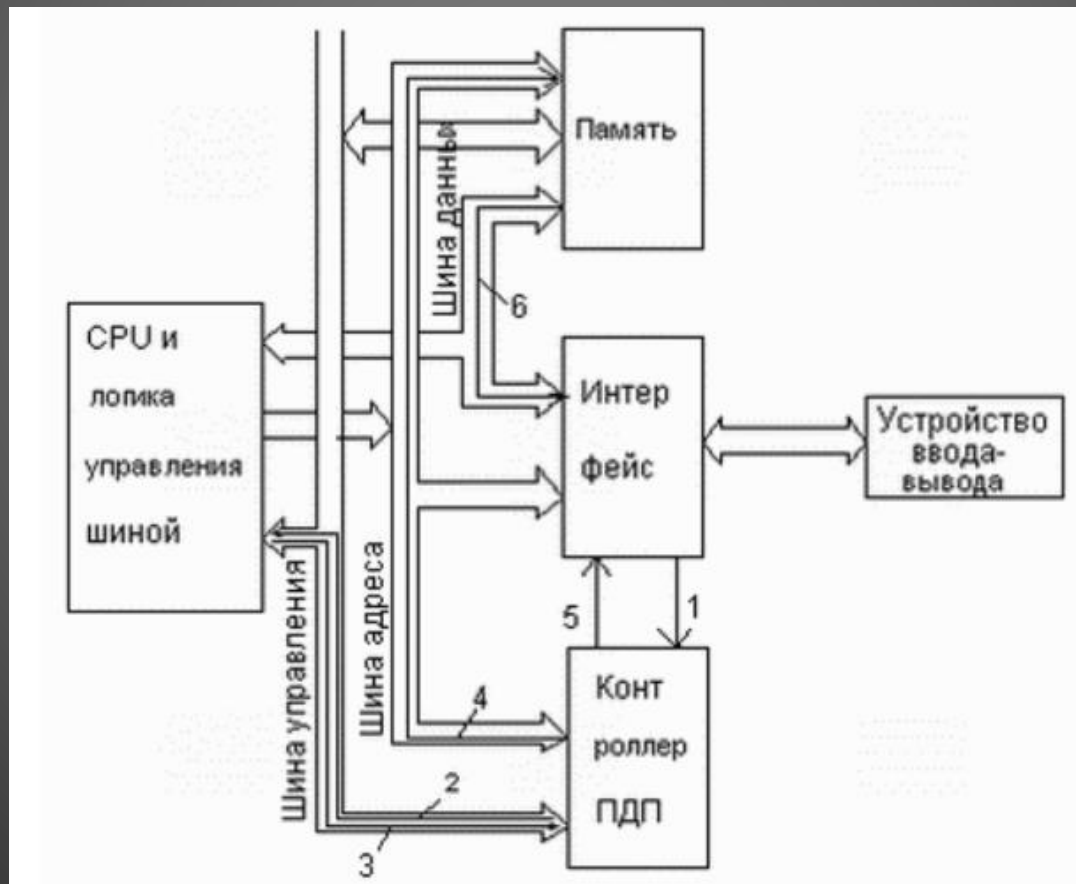
- регистры управления и состояния;
- регистр адреса;
- регистр-счётчик байт.

Общая организация контроллера DMA приведена на рисунке.



2. Регистры контроллера

Работа контроллера прямого доступа памяти происходит в представленной последовательности.



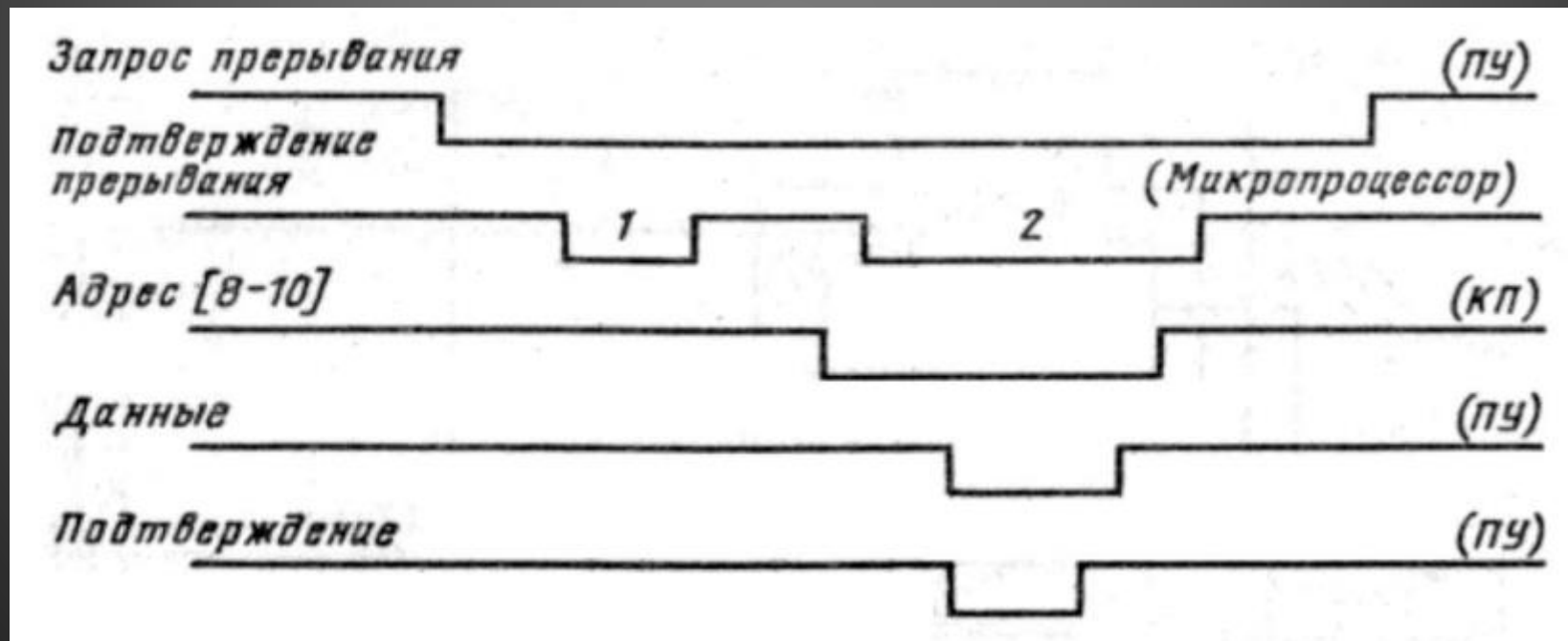
2. Канал ввода/вывода

Поскольку в современной компьютерной системе устройств ввода–вывода несколько, то для каждого устройства в составе контроллера DMA организуется канал, в состав которого входят перечисленные выше регистры. Назначение регистров также видно из их названия:

1. *Регистр управления.*
2. *Регистр состояния.*
3. *Регистр адреса.*

2. Про прерывания

ЦП выдает команду Ввода-Вывода, а затем продолжает делать другую полезную работу. Когда ПУ готово к обмену данными, оно через модуль Ввода-Вывода извещает об этом процессор с помощью запроса на прерывание. ЦП осуществляет передачу очередного элемента данных, после чего возобновляет выполнение прерванной программы.



2. Контроллеры прерываний PIC, APIC

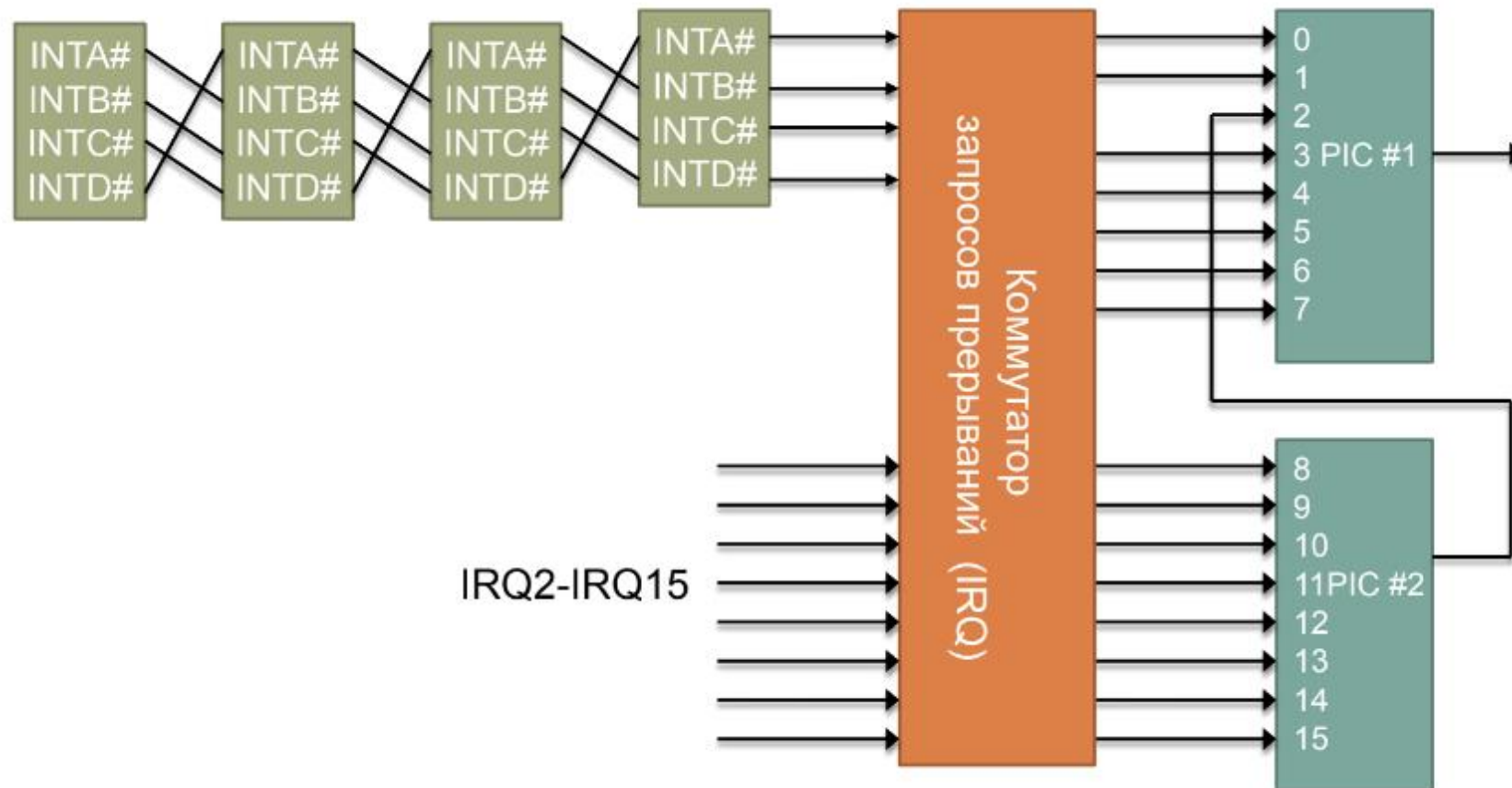
Программируемый контроллер прерываний (*Programmable Interrupt Controller, PIC*) - отвечает за приём запросов прерываний от различных устройств, их хранение в ожидании обработки, выделение наиболее приоритетного из одновременно присутствующих запросов и выдачу его вектора в процессор, когда последний пожелает обработать прерывание. Слово «программируемый» в названии контроллера означает, что режимы его работы устанавливаются программно, а не являются жёстко «зашитыми».

Улучшенный программируемый контроллер прерываний (*Advanced Programmable Interrupt Controller, APIC*). Он был добавлен в процессоре Pentium.

Преимущества расширенного контроллера прерываний:

- возможность реализации межпроцессорных прерываний — сигналов от одного процессора другому;
- поддержка до 256 входов IRQ, в отличие от 15 на классической IBM PC;
- крайне быстрый доступ к регистрам текущего приоритета прерывания и подтверждения прерывания.

2. Схема подачи сигналов INT#



3. Программные интерфейсы ПУ

Разработчики новых устройств зачастую создают собственные программные модели и интерфейсы программирования, что приводит к проблемам совместимости с прикладным и системным ПО.

Тем не менее, для целого ряда современных устройств разработаны стандартные интерфейсы программирования. В особенности это касается универсальных внешних интерфейсов.

3. Состав ПУ

В состав ПУ входят следующие узлы:

1. регистр ввода для приема входных данных от МП;
2. регистр вывода для выдачи данных из МП в ПУ;
3. регистр состояния ПУ;
4. регистр управления (регистр режима);
5. схема управления и селекции ПУ;



3. Регистровая модель ПУ

Изначально разработчики придерживались *регистровой программной модели* ПУ. Устройство представлялось программно доступным (в общем пространстве портов ввода-вывода) набором регистров, среди которых обязательно были три – состояния, управления и данных (т.н. модель CSD). Доступ предполагался методом PIO.

Адресное пространство ввода/вывода может быть совмещено с адресным пространством памяти или быть выделенным.

По мере усложнения архитектуры и повышения требований к устройствам и интерфейсам появилась необходимость реализации более сложной многоуровневой модели программирования с применением объектно-ориентированного подхода.

Современные интерфейсы программирования устройств включают не только аппаратные, но и программные компоненты, входящие в состав ядра операционной системы. Программисту приходится иметь дело не с регистрами, а с системными объектами, а всю низкоуровневую работу с аппаратными ресурсами выполняет драйвер со стандартным интерфейсом программирования.

3. Система ввода-вывода, понятие и задачи. Совмещенное адресное пространство СВВ, достоинства и недостатки. Выделенное (изолированное) адресное пространство СВВ, достоинства и недостатки

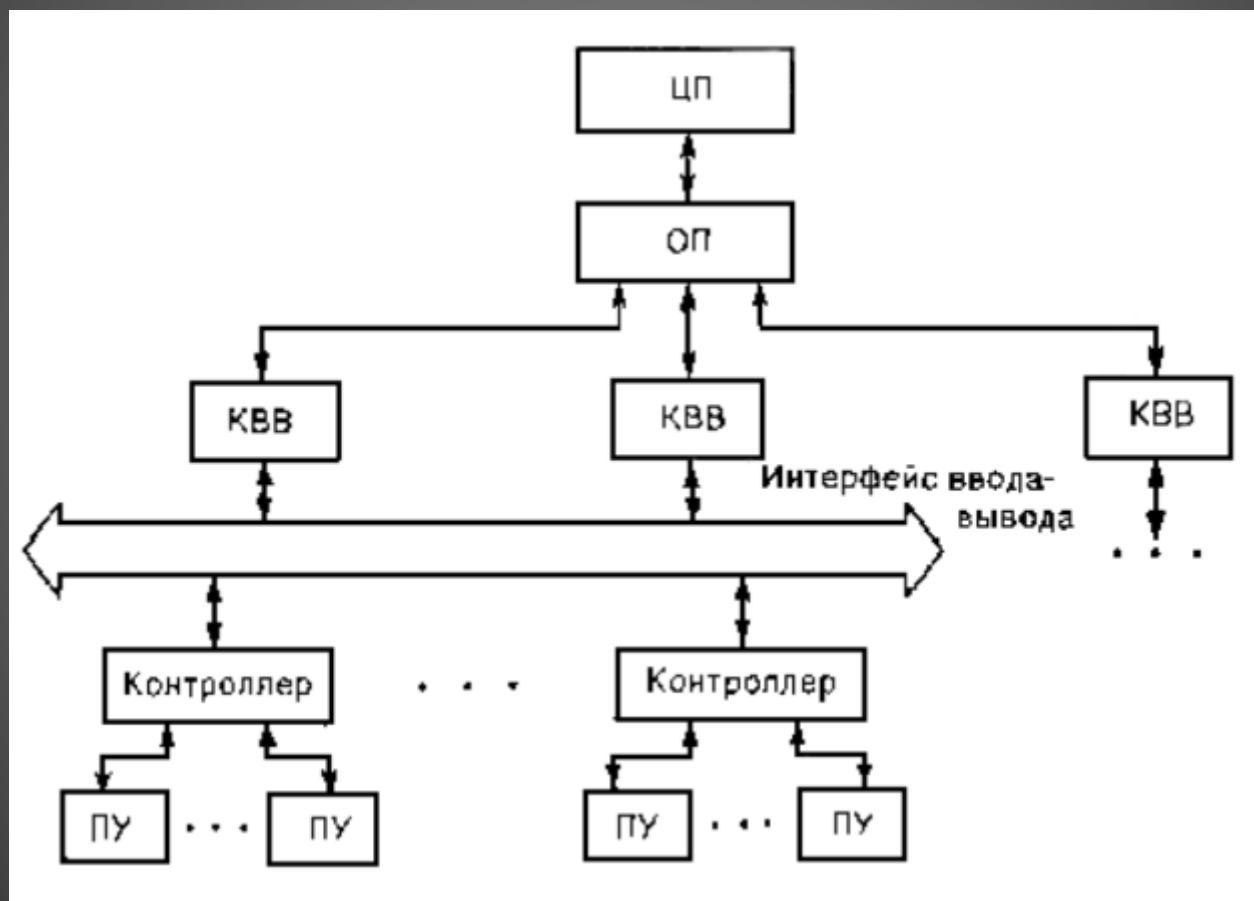
Модульность. ВС проектируются на основе модульного (или агрегатного) принципа. Он заключается в том, что отдельные устройства выполняются в виде конструктивно законченных модулей (агрегатов), которые могут сравнительно просто в нужных количествах и номенклатуре объединяться, образуя ЭВМ.

Унифицированные (не зависящие от типа ПУ) **форматы данных**, которыми ПУ обмениваются с ядром ЭВМ, в том числе и унифицированный формат сообщения, которое периферийное устройство посылает в ядро о своем состоянии. Преобразование в индивидуальные форматы данных осуществляют контроллеры и адаптеры.

Унифицированный интерфейс, т.е. унифицированный по составу и назначению набор линий и шин, унифицированные схемы подключения, сигналы и алгоритмы (протоколы) управления обменом информацией между ПУ и ядром ЭВМ.

Унифицированные (не зависящие от типа ПУ) **формат и выбор команд** процессора для операций ввода-вывода. Операция ввода-вывода с любым ПУ представляет для процессора просто операцию передачи данных независимо от особенностей принципа действия данного ПУ, типа его носителя и т.п..

3. Каналы ввода-вывода



3. Основные функции канала

- прием команд управления работой канала из центрального процессора;
- адресация внешнего устройства, указанного в принятой команде;
- выполнение действий, заданных в командах;
- установка управляющих сигналов на шинах интерфейса ввода- вывода;
- прием управляющих сигналов, поступающих от ПУ по шинам интерфейса;
- непосредственная передача информации между оперативной памятью и ПУ;
- контроль передаваемой информации на четность;
- подсчет количества передаваемых байт информации;
- прием и анализ информации о состоянии периферийных устройств;
- формирование запросов в центральный процессор на прерывание;
- управление последовательностью прерываний от ПУ и выполнение прерывания.

3. Модули ввода/вывода. Функции модуля

Модуль ввода/вывода в составе вычислительной машины отвечает за управление одним или несколькими ПУ и за обмен данными между этими устройствами с одной стороны, и основной памятью или регистрами ЦП — с другой.

Основные функции модуля:

- локализация данных;
- управление и синхронизация;
- обмен информацией;
- буферизация данных;
- обнаружение ошибок.

4. Контроллеры и адаптеры

Контроллеры и адаптеры представляют собой устройства, выполняющие роль своеобразного переходника соединяющего компьютер с каким либо периферийным устройством.

С помощью ***адаптеров и контроллеров*** можно ***связать несколько внешних электронных устройств*** и успешно управлять их работой. Для работы определенной группы устройств существует своя группа переходников или плат расширения. Посредством этих устройств происходит обмен информационными потоками сообщаящими компьютеру о текущем состоянии электроники и происходящих в ней сбоях.

Традиционно выделяют две самостоятельных группы адаптеров и контроллеров:

- ***Встраиваемые устройства***, располагаются внутри системного блока. Это и есть так называемые платы расширения. Подключаются такие устройства благодаря, имеющимся на материнской плате, слотам.
- ***Выносные устройства***, подключаемые посредством имеющихся в компьютере интерфейсов. Например, широко используются USB интерфейсы.

4. Назначение и принцип работы контроллера

Контроллер, как правило, представляет собой электронное устройство, иногда выполненное как часть самого процессора или же сложных микросхем его обрaмления, входы которого присоединены электрически к соответствующим выходам различных устройств.

Номер входа контроллера прерываний обозначается «IRQ». Следует отличать этот номер от приоритета прерывания, а также от номера входа в таблицу векторов прерываний (INT). Так, например, в IBM PC в реальном режиме работы (в этом режиме работает MS-DOS) процессора прерывание от стандартной клавиатуры использует IRQ_1 и INT_9.

4. Контроллер портов ввода-вывода

Одним из контроллеров, которые присутствуют во всех компьютерах, является *контроллер портов ввода-вывода*.

Типы портов:

- параллельные (LPT1-LPT4), к ним обычно присоединяют принтеры и сканеры;
- последовательные асинхронные порты (COM1-COM4), к ним подсоединяются мышь, модем и т. д.;
- игровой порт - для подключения джойстика;
- порт USB (USB 2, 3), к нему подключаются новые модели принтеров, сканеров, модемов, мониторов и т.д. Одним из его достоинств является возможность подключения целой цепочки устройств. Например, через один порт USB подключен принтер, через принтер подключен сканер и т.д.

Некоторые устройства могут подключать и к параллельным, и к последовательным портам, и к порту USB.

4. Виды контроллеров

Контроллер - предназначен для соединения различных устройств (видеокамера, DVD плеер, переносные жесткие диски, звуковые карты).

Контроллер USB - позволяет оснастить системный блок дополнительными или отсутствующими портами USB для соединения с устройствами поддерживающими этот интерфейс (принтеры, сканеры, мыши, клавиатуры, модемы, игровые устройства).

Контроллер SATA - позволяет подключить накопитель с интерфейсом SATA к материнской плате на которой отсутствует данный разъем или как расширение для добавления дополнительных жестких дисков. Контроллер устанавливается как в слот PCI так и PCI-Express.

Контроллер IDE на SATA - позволяет подключить накопитель с IDE в SATA разъем.

4. Интеллектуальный хост-контроллер

Современные контроллеры интерфейсов снабжены **интеллектуальным хост-контроллером** – устройством, обеспечивающим более гибкое управление процессом обмена данными. В частности, такой хост-контроллер самостоятельно обрабатывает списки задач, формируемые в памяти системы, не требуя от процессора контроля за состоянием ПУ

4. Назначение и принцип работы адаптера

Адаптер - устройство сопряжения ЭВМ (ПЭВМ) и ПУ.

Средство сопряжения различных устройств ЭВМ в том числе использующих различные способы представления данных.

В коммутационных, соединительных и кабельных устройствах (также – «переходник») - элемент с разнотипными разъемами, служащий для соединения разнотипных штекеров и/или гнезд, либо для подключения их к телекоммуникационным розеткам различных размеров и типов, изменения разводки или числа проводов в разъемах, соединения разнотипных кабелей.

Основное назначение адаптера контроллера:

1. Подключение внешних запоминающих устройств.
2. Контроль за работой определенного устройства.
3. Передача информации от одного ПК к другому.
4. Преобразование и согласование сигналов между устройствами ПК.

4. Виды адаптеров

Разветвители USB - для расширения количества USB портов, например из 1 в 4. Разветвители имеют самые различные формы.

Адаптер DVI->VGA - позволяет подключить устройство которое не имеет DVI входа, а только VGA. Наиболее часто требуется подключить монитор не оснащенный DVI разъемом к видеокарте у которой в наличии только DVI.

Адаптер DVI->HDMI - для подключения видеокарты (компьютера) к ЖК или плазменному телевизору или другому устройству поддерживающими данный тип разъема.

Адаптер - для подсоединения USB клавиатуры или мыши в разъем PS/2.

Адаптер PS/2->USB - для обратного подсоединения PS/2 устройства к USB разъему. Очень часто данный вариант соединения отказывается работать.

4. Видеоадаптер и его виды

Видеоадаптер - устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора. Первые мониторы, построенные на электронно-лучевых трубках, работали по телевизионному принципу сканирования экрана электронным лучом, и для отображения требовался видеосигнал, генерируемый видеокартой.

Назначение: преобразование цифрового сигнала, циркулирующего внутри РС, в аналоговые электрические сигналы, подаваемые на монитор. Другими словами, видеоадаптер выполняет роль интерфейса между компьютером и устройством отображения информации (монитором).

Типы: MDA, CGA, HGC, EGA, VGA, SuperVGA, IBM 8514.

Режимы работы: графический режим, текстовый режим.

4. Назначение и принцип работы моста

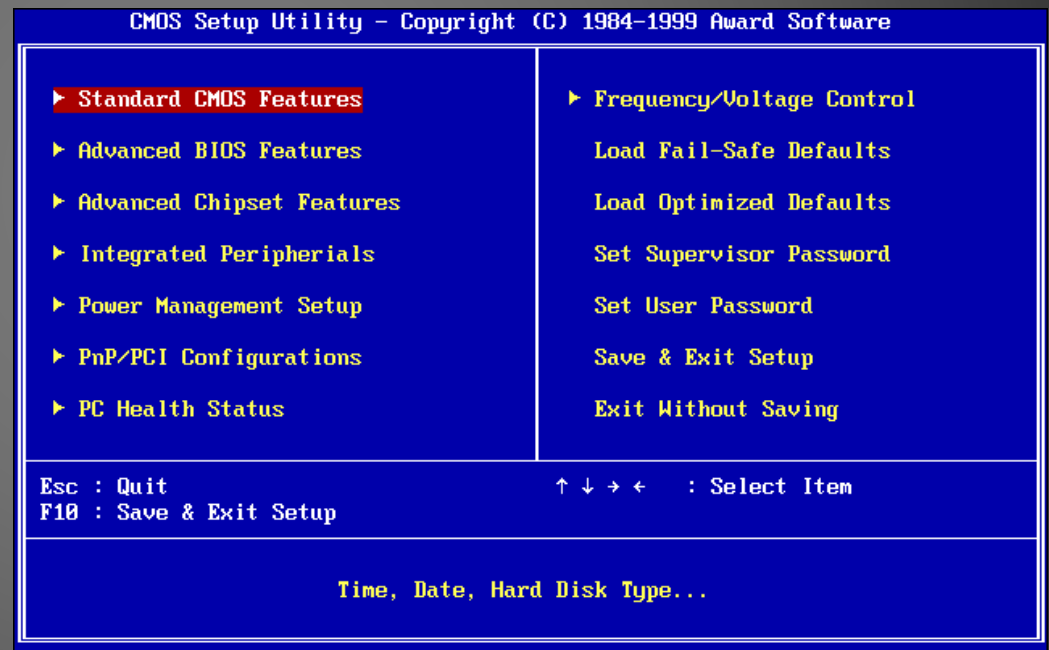
Мост - устройство передачи данных, соединяющее две и более компьютерные шины, например мост PCI-PCI.

Основная функция моста - передать информацию от основной шины к дополнительной.

5. BIOS

BIOS (Basic Input-Output System) — это базовая система ввода-вывода. Это программа низкого уровня, хранящаяся на чипе материнской платы компьютера.

BIOS загружается при включении компьютера и отвечает за пробуждение его аппаратных компонентов, убеждается в том, что они правильно работают, а потом запускает программу-загрузчик, запускающую операционную систему.



5. Принцип работы BIOS

Перед загрузкой операционки BIOS проходит через POST, или Power-On Self Test, самотестирование после включения. Она проверяет корректность настройки аппаратного обеспечения и его работоспособность. Если что-то не так, на экране появится серия сообщений об ошибках или услышите из системного блока загадочный писк. Что именно означают звуковые сигналы описано в инструкции к компьютеру.

При загрузке компьютера по окончании POST BIOS ищет Master Boot Record, (MBR) — главную загрузочную запись. Она хранится на загрузочном устройстве и используется для запуска загрузчика ОС.

BIOS хранит различные настройки в комплементарной структуре металл-оксид-полупроводнике, который называется Complementary Metal-Oxide-Semiconductor (CMOS). Использование её устарело, поскольку такой метод уже заменили флэш-памятью (также её называют EEPROM).

5. Почему BIOS устарел?

Не смотря на все изменения и улучшения BIOS имеет серьезные ограничения:

- Может загружаться только с жестких дисков объемов не более 2,1 Тб (это ограничение BIOS MBR);
- Может работать в 16-битном процессоре и ему доступен всего 1 Мб памяти. У него проблемы с одновременной инициализацией нескольких устройств, что ведёт к замедлению процесса загрузки, во время которого инициализируются все аппаратные интерфейсы и устройства.

Компании начали искать решения и в итоге Intel начала работу над Extensible Firmware Interface (EFI) ещё в 1998 году. Apple выбрала EFI, перейдя на архитектуру Intel на своих MacBook в 2006-м, но другие производители не пошли за ней.

6. Исторический переход от BIOS к UEFI

В 2007 Intel, AMD, Microsoft и производители PC договорились о новой спецификации **Unified Extensible Firmware Interface (UEFI)**, унифицированный интерфейс расширяемой прошивки.

UFE индустриальный стандарт, обслуживаемый форумом UEFI и он зависит не только от Intel. Поддержка UEFI в ОС Windows появилась с выходом Windows Vista Service Pack 1 и Windows 7. Большая часть компьютеров, которые вы можете купить сегодня, используют UEFI вместо BIOS.

UEFI — это небольшая операционная система, работающая над прошивкой PC, поэтому она способна на гораздо большее, чем BIOS. Её можно хранить в флэш-памяти на материнской плате или загружать с жёсткого диска или с сети.

6. Принцип работы UEFI

Прошивка UEFI может грузиться с дисков объёмом более 2,2 Тб – теоретический предел для них составляет 9,4 зеттабайт. Это примерно в три раза больше всех данных, содержащихся в сегодняшнем Интернете. UEFI поддерживает такие объёмы из-за использования разбивки на разделы GPT вместо MBR. Также у неё стандартизирован процесс загрузки, и она запускает исполняемые программы EFI вместо кода, расположенного в MBR.



6. UEFI

Достоинства UEFI относительно BIOS:

- Поддерживает жесткие диски большего объема;
- Быстрее грузится за счет большего адресного пространства;
- Более безопасен;
- Обладает графическим интерфейсом;
- Поддерживает работу с использованием мыши.

Более подробно можно почитать про UEFI на [habr.com](https://habr.com/ru/articles/111111/) «Что такое UEFI, и чем отличается от BIOS?».