

Лекция 2. Аппаратная и программная поддержка интерфейсов

Основные принципы программирования доступа к периферийным устройствам.
Методы управления обменом.

Регистровая программная модель периферийного устройства
Особенности адресации.

Аппаратные средства поддержки работы периферийных устройств:
контроллеры, адаптеры; мосты.

Основные принципы программирования доступа к периферийным устройствам

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

- **независимость от устройств.** Вид программы не должен зависеть от того, читает ли она данные с гибкого диска или с жесткого диска. Единые правила именования ПУ.
- **обработка ошибок как можно ближе к аппаратуре.** Если контроллер обнаруживает ошибку чтения, то он должен попытаться ее скорректировать. Если же это ему не удастся, то исправлением ошибок должен заняться драйвер устройства. Многие ошибки могут исчезать при повторных попытках выполнения операций ввода-вывода, например, ошибки, вызванные наличием пылинок на головках чтения или на диске. И только если нижний уровень не может справиться с ошибкой, он сообщает об ошибке верхнему уровню.

Основные принципы программирования доступа к периферийным устройствам

- **использование блокирующих (синхронных) и неблокирующих (асинхронных) передач.** Большинство операций физического ввода-вывода выполняется асинхронно - процессор начинает передачу и переходит на другую работу, пока не наступает прерывание. Пользовательские программы намного легче писать, если операции ввода-вывода блокирующие - после команды READ программа автоматически приостанавливается до тех пор, пока данные не попадут в буфер программы. ОС выполняет операции ввода/вывода асинхронно, но представляет их для пользовательских программ в синхронной форме.
- **одни устройства являются разделяемыми, а другие - выделенными.** Диски - это разделяемые устройства, так как одновременный доступ нескольких пользователей к диску не представляет собой проблему. Принтеры - это выделенные устройства, потому что нельзя смешивать строки, печатаемые различными пользователями. Наличие выделенных устройств создает для операционной системы некоторые проблемы.

Многоуровневая организация подсистемы ВВОДА-ВЫВОДА



Для решения поставленных проблем целесообразно разделить ПО В/В на четыре слоя:

Пользовательский слой ПО

Независимый от устройств слой операционной системы.

Драйверы устройств.

Обработка прерываний.

Обмен данными с ПУ

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.

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.

Применяются три способа (метода) организации ввода/вывода:

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

Программно управляемый

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).

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.

Программа управления вводом - выводом

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

В этой программе:

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.

Модуль ВВ исполняет затребованное действие, после чего устанавливает в единицу соответствующий бит в своем регистре состояния. Ничего другого, чтобы уведомить ЦП, модуль не предпринимает.

Для определения момента завершения операции или пересылки очередного элемента блока данных ЦП должен периодически опрашивать и анализировать содержимое регистра состояния шины (канала, магистрали) ВВ.

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 ВВ, то есть доступно для считывания.

Существуют четыре типа команд В/ВЫВ, которые может получить модуль ВВ: управление, проверка, чтение и запись.

Команды управления используются для активизации ПУ и указания требуемой операции. Например, в устройство памяти на магнитной ленте может быть выдана команда перемотки или продвижения на одну запись. Для каждого типа ПУ характерны специфичные для него команды управления.

Команда проверки применяется для проверки различных ситуаций, возникающих в модуле ВВ и ПУ в процессе ввода/вывода: включено ли ПУ, готово ли оно к работе, завершена ли последняя операция ввода/вывода и не возникли ли в ходе ее выполнения какие-либо ошибки. Действие команды сводится к установке или сбросу соответствующих разрядов регистра состояния модуля ВВ.

Команда чтения побуждает модуль получить элемент данных из ПУ и занести его в РД. ЦП может получить этот элемент данных, запросив модуль ВВ поместить его на шину данных.

Команда записи заставляет модуль принять элемент данных (байт или слово) с шины данных и переслать его в РД с последующей передачей в ПУ.

Метод прямого доступа к памяти

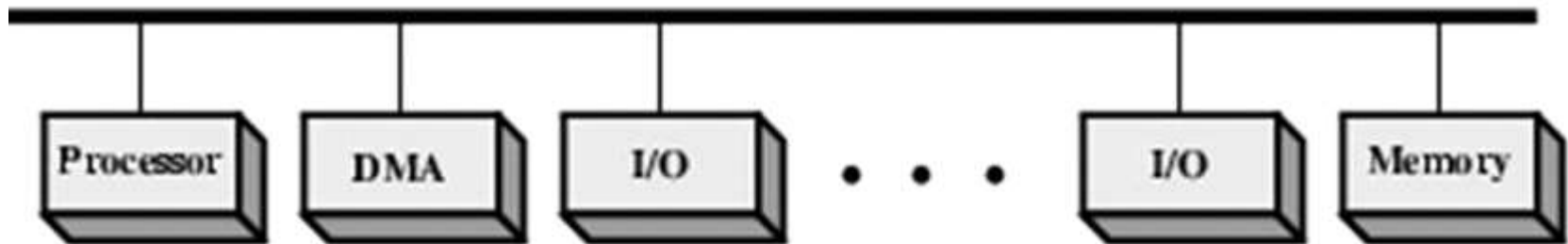
Метод *прямого доступа к памяти* (ПДП, DMA) позволяет выполнять обмен между ОП системы и ресурсами ПУ *асинхронно по отношению к вычислительному процессу*.

В этом режиме обмен данными между ПУ и основной памятью компьютера происходит без участия ЦП.

Обменом в режиме ПДП управляет не программа, выполняемая ЦП, а электронные схемы, внешние по отношению к ЦП. Обычно схемы, управляющие обменом в режиме ПДП, 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.

Схема 1 подключения ПДП

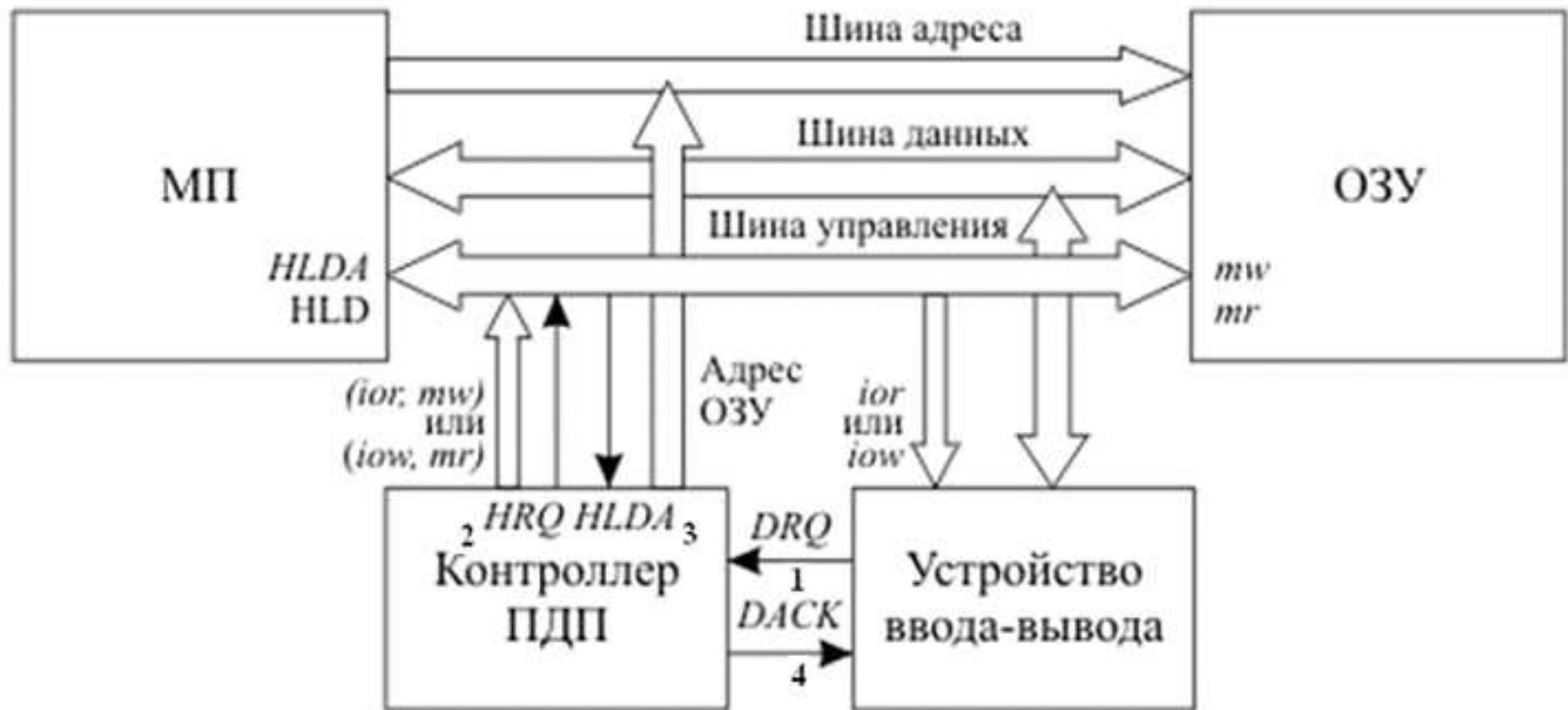


К общей шине подключается контроллер DMA и все вовлеченные в операцию устройства.

В каждой транзакции шина используется дважды
1) I/O к DMA; 2) DMA к ОЗУ.

При передаче по шине CPU дважды приостанавливает свою работу.

Структура ЭВМ с КПДП



Обмен данными в режиме ПДП

Алгоритм ПДП

Перед началом работы контроллер ПДП необходимо инициализировать: занести начальный адрес области ОП, с которой производится обмен, и длину передаваемого массива данных. В дальнейшем по сигналу запроса прямого доступа контроллер фактически выполняет все те действия, которые обеспечивал микропроцессор при *программно-управляемой передаче*.

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

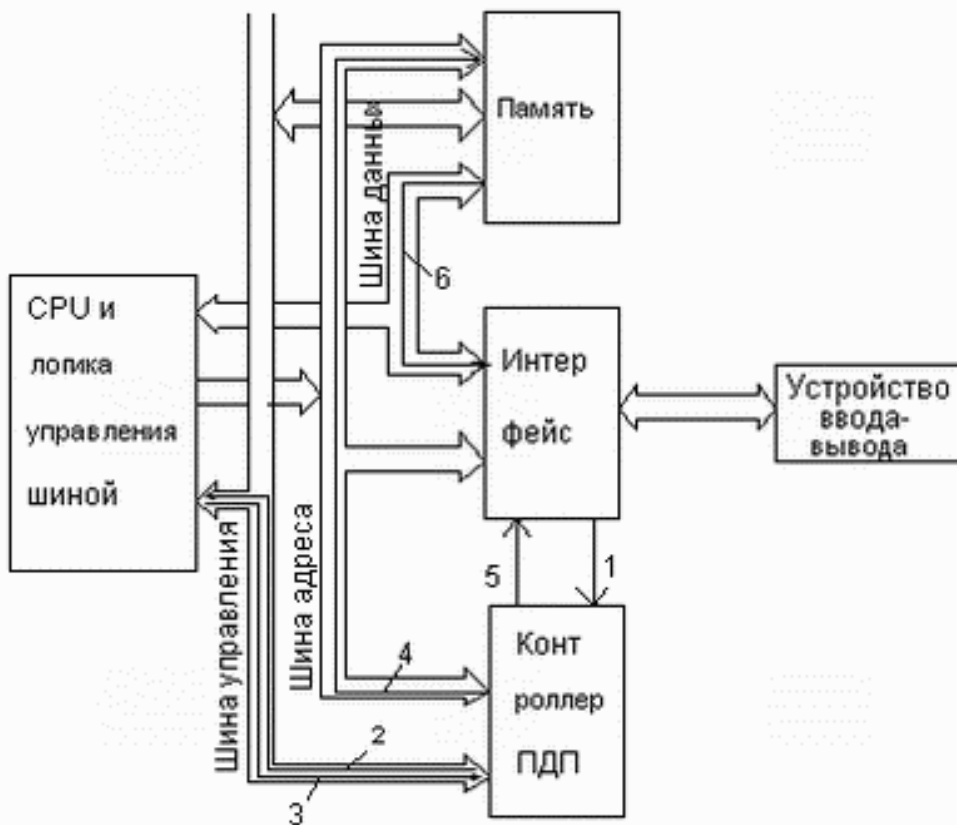
- 1) Принять** запрос на ПДП (сигнал DRQ) от УВВ.
- 2) Сформировать** запрос к МП на захват шин (сигнал HRQ).
- 3) Принять** сигнал от МП (HLDA), подтверждающий факт перевода микропроцессором своих шин в третье состояние.

Алгоритм ПДП (продолжение)

- 4) **Сформировать** сигнал, сообщаящий устройству ввода-вывода о начале выполнения циклов *прямого доступа к памяти* (DACK).
- 5) **Сформировать** на шине адреса компьютера адрес ячейки памяти, предназначенной для обмена.
- 6) **Выработать** сигналы, обеспечивающие управление обменом (IOR, MW для передачи данных из УВВ в оперативную память и IOW, MR 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 ПДП). Если условие окончания не выполнено, то изменить адрес в регистре текущего адреса на длину переданных данных и повторить шаги 5-8.

Работа контроллера ПДП

1. Интерфейс устройства В/В при наличии у него готовых данных для оперативной памяти или готовности принять данные от оперативной памяти, выдаёт сигнал запроса ПДП.
2. Контроллер ПДП формирует сигнал запроса шины и получает управление шиной от центрального процессора.
3. На шину адреса помещается адрес ячейки памяти, который контроллер ПДП выдаёт из своего регистра адреса.



4. Контроллер ПДП выдаёт подтверждение интерфейсу устройства В/В на В/В данных.
 5. Данные помещаются на шину данных (при выводе – из ОЗУ, при вводе - от интерфейса устройства В/В.
 6. Данные, в случае вывода фиксируются интерфейсом устройства ВВ, а в случае ввода, данные поступают в ячейку памяти, адрес которой помещён на шине адреса (счётчик байт вычитает из своего значения «1», а регистр адреса прибавляет «1»).
- Контроллер освобождает шину и управление шиной возвращается CPU

Регистры контроллера

Контроллер ПДП имеет в своём составе такие элементы:

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

Общая организация контроллера ПДП приведена на рисунке.



Регистры

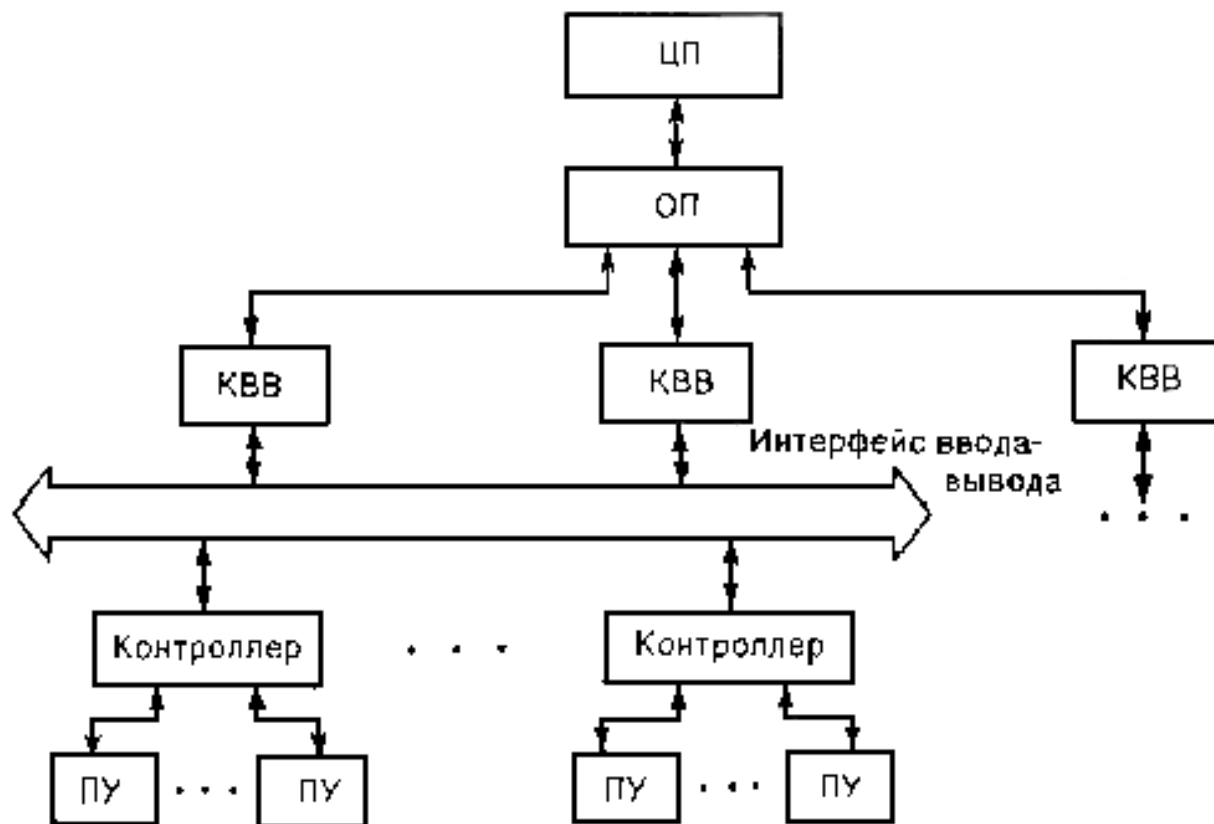
Регистр управления предназначен для хранения и выдачи на шину сигналов, которые управляют и информируют устройства, задействованные в обмене. Эти сигналы указывают устройствам о направлении передачи: ввод или вывод, между какими устройствами, так как контроллер способен управлять и передачами типа память—память, о непрерывности режима ПДП или с перерывами передачи, собственно разрешении режима ПДП, приоритетах каналов.

Регистр состояния служит для указания состояния контроллера ПДП, а именно: счётчиков каналов и наличия или отсутствия запросов ПДП.

В *регистр адреса* при инициализации контроллера передаётся начальный или конечный адрес массива памяти в который помещаются данные, а в регистр—счётчик байт число передаваемых байт. После каждого переданного байта счётчик уменьшает первоначальное значение числа передаваемых байт. Таким образом контролируется процесс передачи данных между памятью и устройствами ввода—вывода, при достижении в счётчике байт «нуля» режим работы ПДП завершается, контроллер ПДП возвращает управление шиной процессору. А затем, в зависимости от ситуации, устройства либо снова начинают процесс инициализации режима ПДП, либо управление осуществляет центральный процессор.

Канал ввода/вывода

Поскольку в современной компьютерной системе устройств В/В несколько, то для каждого устройства в составе контроллера ПДП организуется канал, в состав которого входят перечисленные выше регистры. Назначение регистров также видно из их названия.

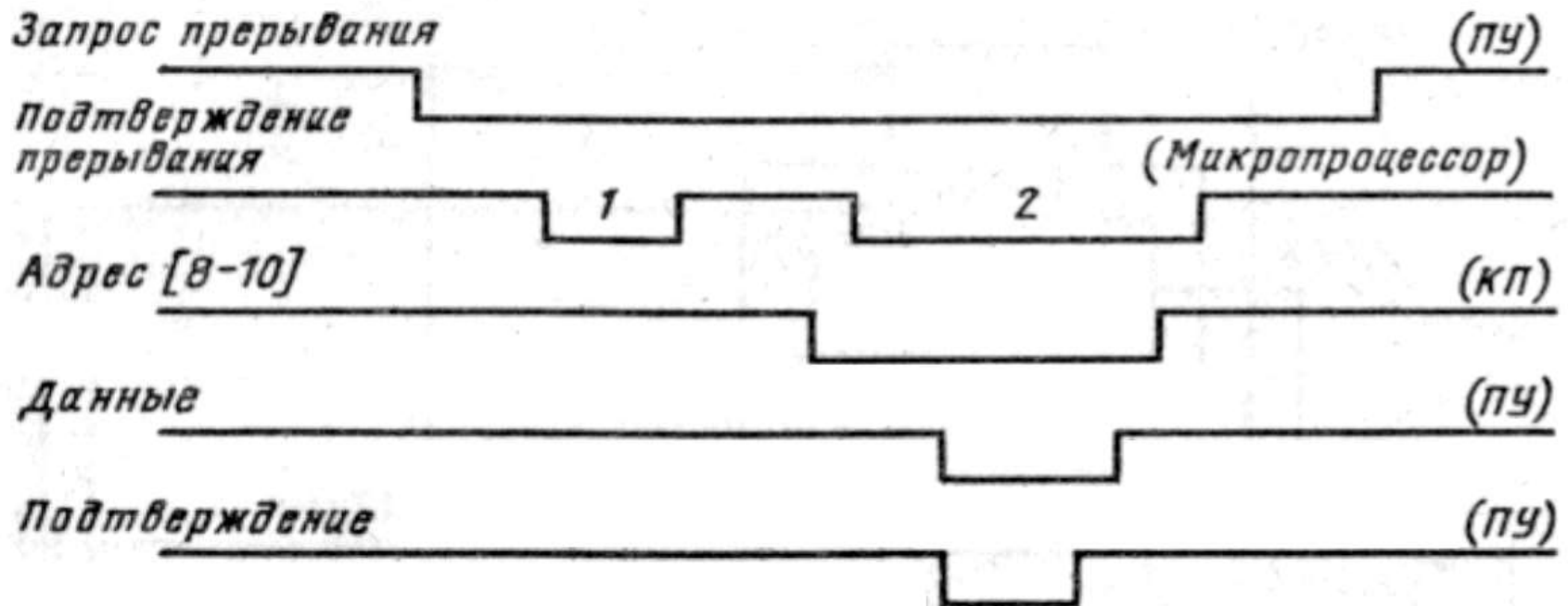


Основные функции канала

прием команд управления работой канала из ЦП;
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 ОП и ПУ;
контроль передаваемой информации на четность;
подсчет количества передаваемых байт информации;
прием и анализ информации о состоянии периферийных устройств;
формирование запросов в ЦП на прерывание;
управление последовательностью прерываний от ПУ и выполнение прерывания.

По прерыванию

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



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».

APIC (*Advanced Programmable Interrupt Controller*) — улучшенный ...
Он был добавлен в процессоре Pentium.

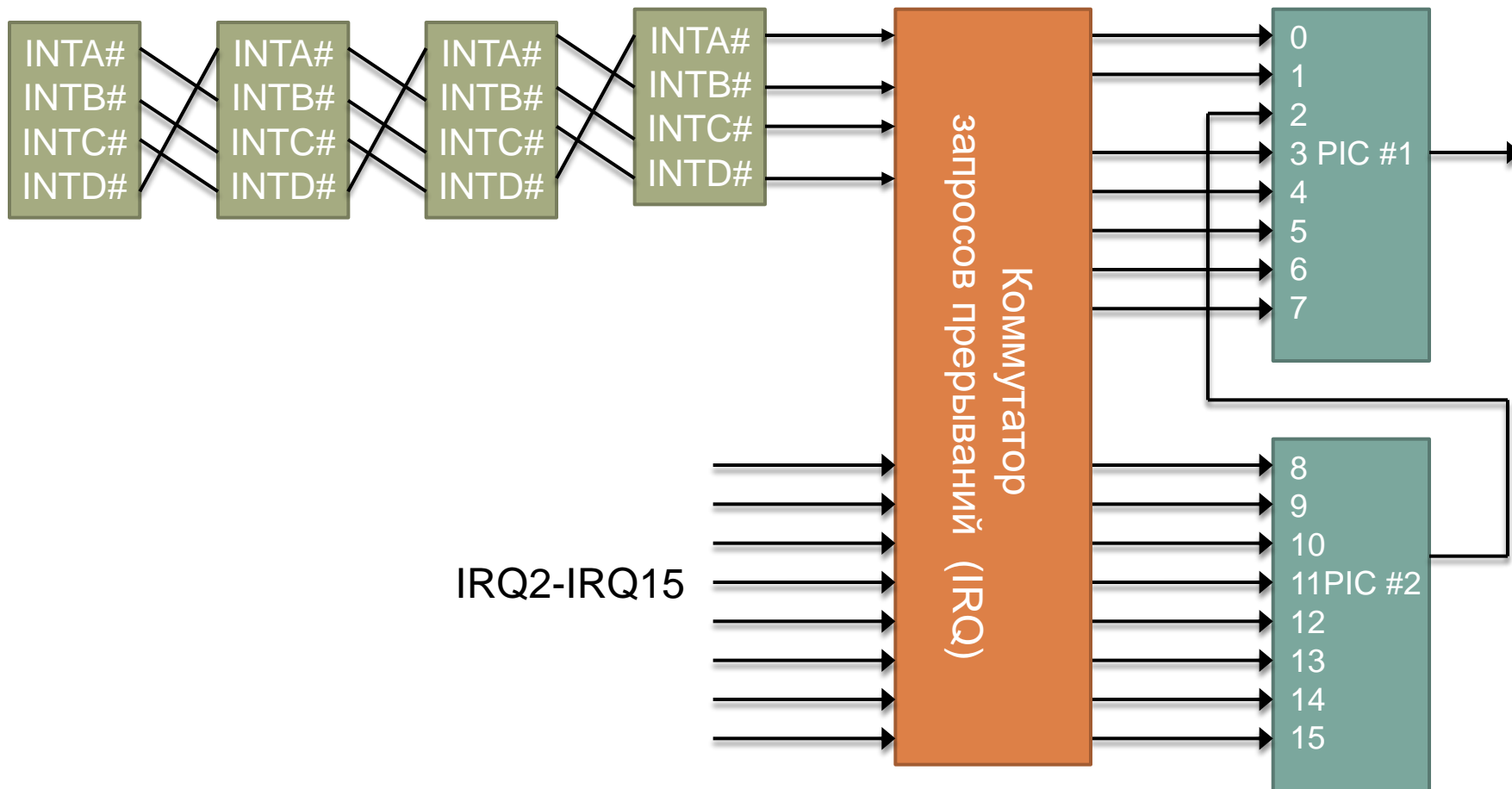
Преимущества расширенного контроллера прерываний:

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. Контроллер прерываний, совместимый с IBM PC, исполнялся как устройство шины ISA с очень медленным доступом к его регистрам (порт 0x20). I/O APIC — контроллер, расположенный на системной плате, обычно как часть микросхем обрэмления процессора (например, микросхема Intel 82489DX).

Схема подачи сигналов INT#



Сигнализация по линии INTx#

1. Устройство вводит сигнал прерывания, понижая уровень линии INTx#.
2. ЦП получает сигнал прерывания с вектором, соответствующем определенной линии IRQ.
3. Обработчик прерывания (драйвер) обращается к устройству и проверяет, установлен ли в его регистрах сигнал запроса прерывания.
4. Если это было именно его устройство, драйвер сбрасывает сигнал прерывания программным способом и начинает обработку.
5. После отработки прерывания линия запроса все еще может быть в низком уровне из-за прихода прерывания от другого устройства, разделяющего ту же линию – тогда процедура повторяется.

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.

Регистровая модель ПУ

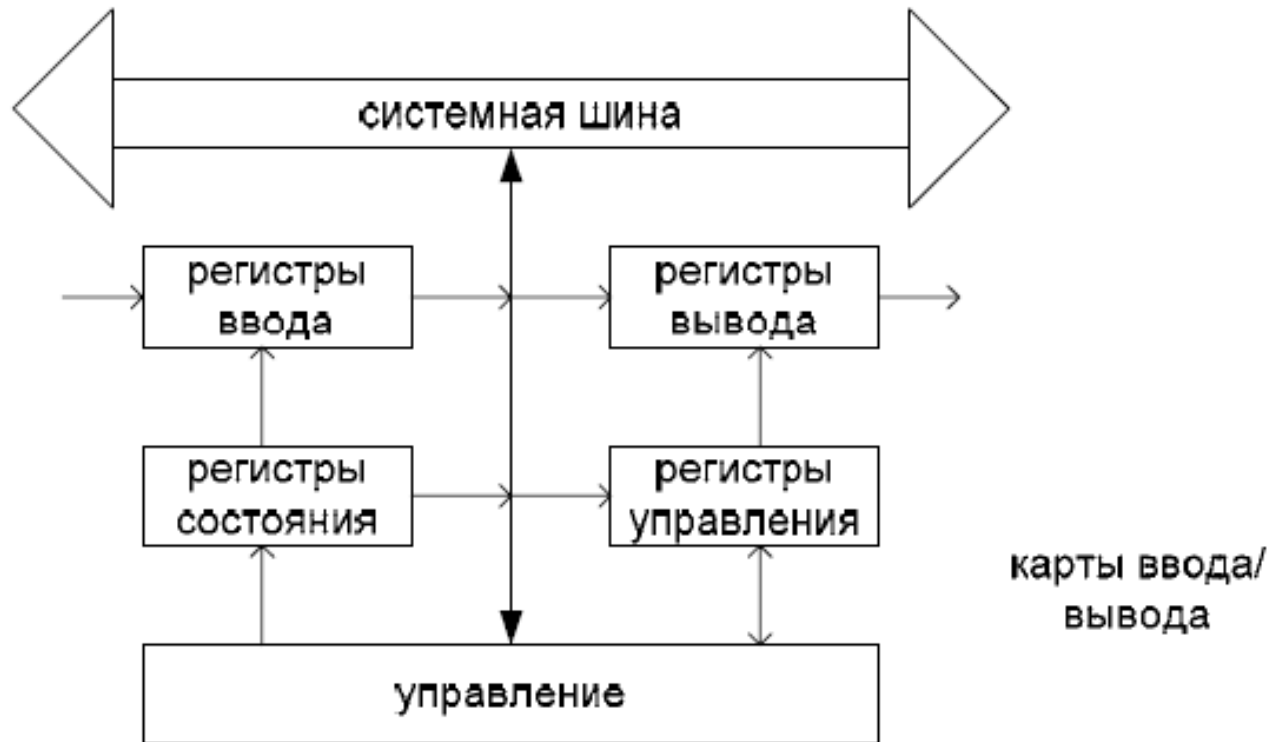
Изначально разработчики придерживались *регистровой программной модели* ПУ. Устройство представлялось программно доступным (в общем пространстве портов ввода-вывода) набором регистров, среди которых обязательно были три – состояния, управления и данных (модель CSD). Доступ предполагался методом PIO.

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.

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;

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.

Аппаратные средства поддержки работы периферийных устройств

Системная плата или **материнская плата**

Чипсет— набор микросхем, спроектированных для совместной работы с целью выполнения набора каких-либо функций

Контроллеры

Карта расширения (адаптер)— это печатная плата, которую помещают в слот расширения материнской платы компьютерной системы с целью добавления дополнительных функций. (видео-, звуковая- сетевая карат)

Мосты

Коммутаторы, расширители, повторители, концентраторы, хабы

Повторитель, Умножитель портов (Port Multiplier)

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 сообщаемыми 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.

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.

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.

Контроллеры

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-ex.

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

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

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

Адаптеры

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.

Видеоадаптеры

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.

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.