

Лекция XX

План лекции

1. Система прерываний
2. Контроллер прерываний
3. Программирование контроллера прерываний i8259A

1. Система прерываний

- Каждый сигнал от различных устройств компьютера, чтобы стать «осознанным», должен быть обработан какой-то программой. А выполнять программы может только микропроцессор. Но он один в компьютере, следовательно, в каждый конкретный момент времени микропроцессор может быть занят выполнением только одной программы. Как же ему узнавать о нажатиях клавиш и других сигналах, постоянно возникающих во время работы компьютера, если он выполняет некоторую программу?

- Возможным решением здесь может быть, например, периодическая остановка текущей программы и выполнение других программ, производящих опрос устройств компьютера и, в свою очередь, запускающих необходимые программы для обслуживания этих устройств. Это далеко не оптимальный путь, значительно снижающий производительность компьютера. Другой возможный подход к обслуживанию устройств — создание системной очереди на обслуживание. Этот подход предполагает некую очередь, в которую «выстраиваются» запросы на обслуживание от устройств. Микропроцессор периодически просматривает эту очередь и выполняет обслуживание запросов в ней. Этот вариант, хотя и лучше предыдущего, но тоже не оптимальный. В современных микропроцессорах, каковыми являются микропроцессоры фирмы Intel, принят подход, основанный на понятии прерывания. Прерывание — инициируемый определенным образом процесс, временно переключающий микропроцессор на выполнение другой программы с последующим возобновлением выполнения прерванной программы. Что дает использование механизма прерываний? Он позволяет обеспечить наиболее эффективное управление не только внешними устройствами, но, как мы увидим далее, и программами. Нажимая клавишу на клавиатуре вы фактически инициируете посредством прерывания немедленный вызов программы которая распознает нажатую клавишу заносит ее код в буфер клавиатуры откуда он в дальнейшем считывается некоторой другой программой или операционной системой. На время такой обработки микропроцессор прекращает выполнение некоторой программы и переключается на так называемую процедуру обработки прерывания. После того как данная процедура выполнит необходимые действия, прерванная программа продолжит выполнение с точки где было приостановлено ее выполнение. Некоторые операционные системы используют механизм прерываний не только для обслуживания внешних устройств, но и для предоставления своих «услуг». Так, хорошо известная и до сих пор достаточно широко используемая операционная система MS DOS взаимодействует с системными и прикладными программами преимущественно через систему прерываний.

- Чтобы устройства, входящие в состав компьютера могли взаимодействовать с центральным процессором, в IBM совместимых компьютерах предусмотрена система прерываний (Interrupts).
- Система прерываний позволяет компьютеру приостановить текущие действия и переключиться на другие в ответ на поступивший запрос например на нажатие клавиши на клавиатуре

- Без прерываний было бы очень трудно обеспечить эффективную работу компьютера. С одной стороны, желательно чтобы компьютер был занят возложенной на него работой, с другой стороны необходима его мгновенная реакция на любой требующий внимания запрос. Прерывания обеспечивают немедленную реакцию системы.
- Каждому компоненту компьютера, которому может потребоваться запросить внимание процессора выделяется особый номер прерывания. Например, клавиатура имеет свой номер прерывания, поэтому при нажатии клавиши процессор узнает об этом. Номера прерываний имеют также дисковые накопители и принтеры. Накопители на дисках с помощью прерывания сообщают об окончании запрошенной программной операции, а принтеры сигнализируют об отсутствии бумаги или о готовности печатать следующий символ.

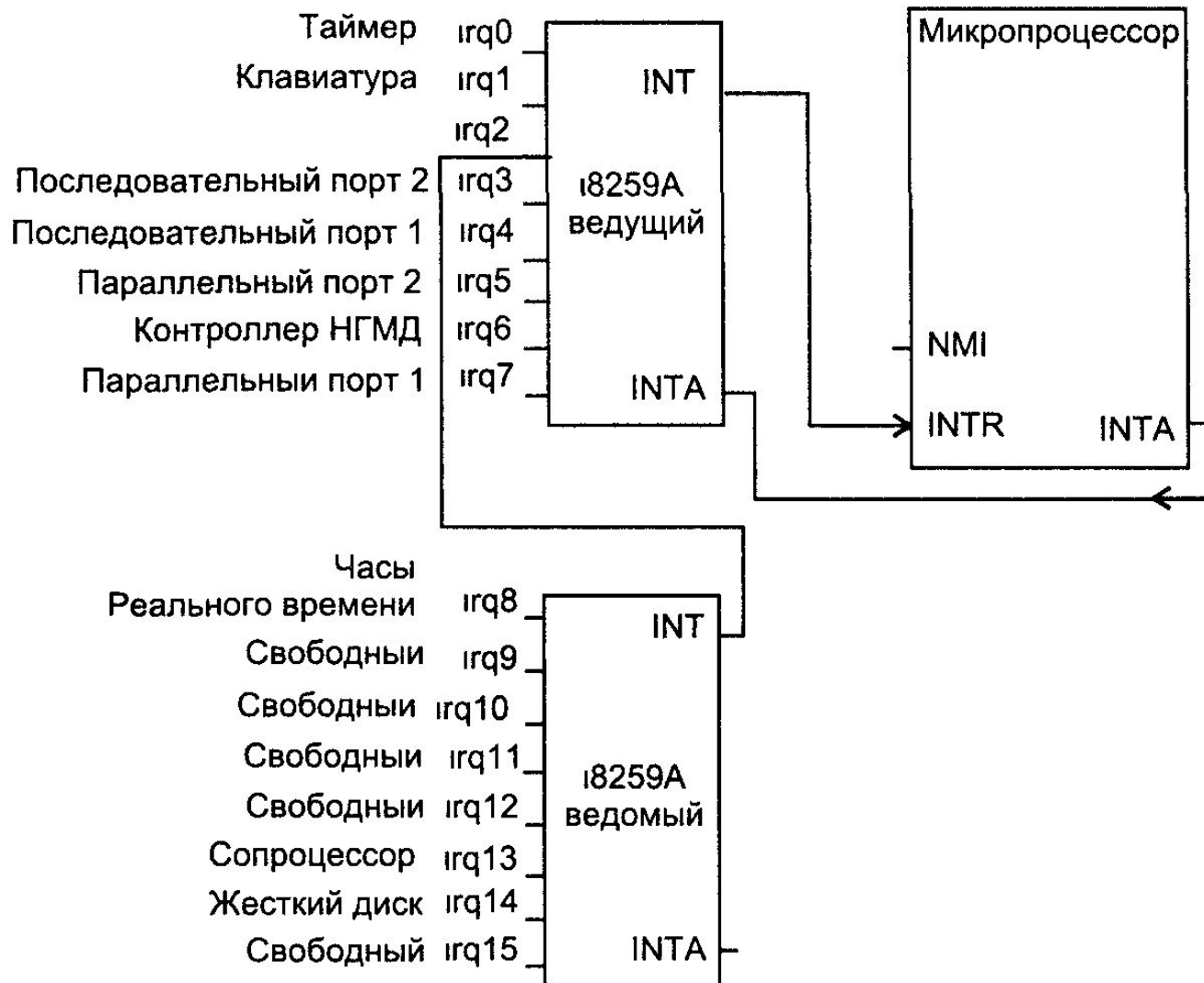
- В IBM PC-совместимых компьютерах имеется два вида прерываний аппаратные (Hardware interrupts) и программные (Software interrupts). К аппаратным относятся в первую очередь, прерывания встроенные в процессор — прерывания при делении на нуль прерывания при аварии питания и т.п. К аппаратным также относятся прерывания, с помощью которых процессору сообщается об аппаратных событиях, например, уже упомянутых отсутствии бумаги в принтере или окончании дисковой операции.
- С помощью прерываний прикладные программы могут выполнять операции запрограммированные в операционной системе ROM BIOS или в других сервисных программах. Отметим, что аппаратное прерывание может получить управление и при выполнении программного прерывания. Программные прерывания действуют практически так же, как и аппаратные, и отличаются только источником прерывания. Аппаратные прерывания работают следующим образом. Когда какому-либо устройству требуется участие процессора (например, когда на клавиатуре нажата клавиша), оно посылает особый сигнал — запрос на прерывание. Получив такой запрос, процессор прекращает выполнять текущую последовательность команд, а вместо нее начинает выполнять другую последовательность, соответствующую данному прерыванию (в нашем примере — передает код нажатой клавиши работающей программе).

- Для простой и эффективной локализации процедуры обслуживания прерываний служит таблица векторов прерываний, содержащаяся в первых ячейках памяти компьютера. Вектор — это полный адрес памяти (4 байта) той процедуры, которая вызывается при появлении запроса на прерывание. Таким образом, в таблице векторов прерываний в ячейках памяти с 0 по 3 находится адрес программы обработки прерывания с номером 0, в ячейках памяти с 4 по 7 — адрес программы обработки с номером 1 и т.д.
- В IBM PC-совместимых компьютерах управление аппаратными прерываниями осуществляется специальными микросхемами — контроллерами прерываний. Как уже отмечалось, на современных материнских платах контроллеры прерываний интегрированы в микросхему контроллера периферии Chipset.
- В PC XT с восьмиразрядной шиной ISA имеется только 8 внешних аппаратных прерываний. В PC AT с процессором 80286 количество линий прерываний было увеличено вдвое, благодаря использованию уже двух контроллеров.

- Каждый из контроллеров имеет по 8 входных линий для приема запросов на прерывание. Линии запроса на прерывание называются IRQ (Interrupt - Requests). Прерывания, используемые вторым контроллером, подаются на неиспользованный вход IRQ 2 первого (рис 5.18).
- Каждая линия имеет свой приоритет. Наивысший приоритет соответствует линии IRQ 0. Отметим, что контроллеры прерываний включены таким образом, что линии IRQ 8—IRQ 15 имеют приоритет ниже приоритета линии IRQ 2, но выше IRQ 3. Каждой линии IRQ соответствует вектор прерываний, указывающий местонахождение процедуры обслуживания запроса на прерывание в ROM BIOS. Линиям IRQ 1—IRQ 7 соответствуют вектора прерываний от 8H до 0FH, а линиям IRQ 8—IRQ 15 — вектора от 70H до 77H.
- Часть линий прерываний резервируется системой, а часть может быть предоставлена для использования дополнительным оборудованием, необходимым пользователю. Так, например, IRQ 13 отводится для математического сопроцессора и оно не может быть использовано другим устройством, даже если математический сопроцессор не установлен. Обычно для IBM PC-совместимых компьютеров свободны IRQ 3, IRQ 5, IRQ 7, IRQ 10, IRQ 11 и IRQ 12. Стандартное распределение аппаратных прерываний приведено в табл. 5.15

IRQ	Назначение	Тип адаптера
00	Прерывание от таймера	—
01	Прерывание от клавиатуры	—
02	Связь со вторым контроллером прерывания (каскадное прерывание)	—
03	COM2, COM4	8- или 16-разрядный
04	COM1, COM3	8- или 16-разрядный

- Исходя из вышеприведенных рассуждений, можно сказать, что прерывания могут быть внешними и внутренними.
- Внешние прерывания вызываются внешними по отношению к микропроцессору событиями. На рис 1 схематически изображена подсистема прерываний компьютера на базе микропроцессора Intel



- На рис 1 видно, что у микропроцессора есть два физических контакта — INTR и NMI. На них и формируются внешние по отношению к микропроцессору сигналы, возрастающие фронты которых извещают микропроцессор о том, что некоторое внешнее устройство просит уделить ему внимание. Вход INTR (INTRerrupt Request) предназначен для фиксации запросов от различных периферийных устройств, например таких, как системные часы, клавиатура, жесткий диск и т.д. Вход NMI (NonMaskable Interrupt) — немаскируемое прерывание. Этот вход используют для того, чтобы сообщить микропроцессору о некотором событии, требующем безотлагательной обработки, или катастрофической ошибке. Внешние прерывания относятся, естественно, к не планируемому прерываниям.
- Внутренние прерывания возникают внутри микропроцессора во время вычислительного процесса. К их возбуждению приводит одна из двух причин.
- ненормальное внутреннее состояние микропроцессора, возникшее при обработке некоторой команды программы. Такие события принято называть исключительными ситуациями, или просто исключениями. Этот вид прерываний отчасти также можно отнести к непланируемым;
- обработка машинной команды `int xx` Этот тип прерываний называется программным. Это — планируемые прерывания, так как с их помощью программист обращается в нужное для него время за обслуживанием своих запросов либо к операционной системе, либо к BIOS, либо к собственным программам обработки прерываний.

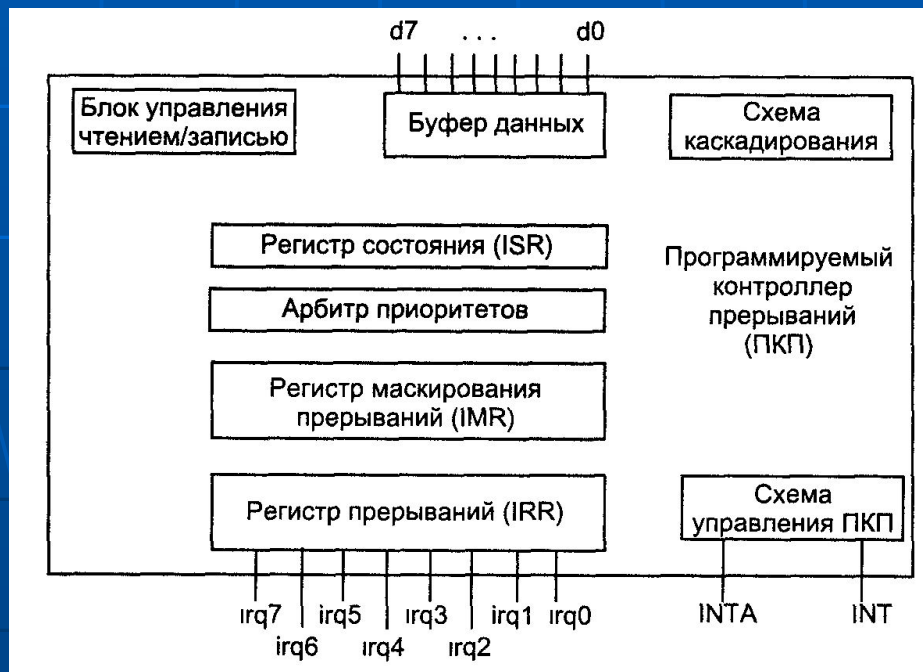
- Далее мы рассмотрим особенности обработки прерываний. Как уже отмечалось, микропроцессоры Intel имеют два режима работы — реальный и защищенный. В этих режимах обработка прерываний осуществляется принципиально разными методами. Поэтому на данном уроке мы дадим характеристику реального режима и рассмотрим обработку прерываний в этом режиме. На следующем уроке будет рассмотрен защищенный режим работы микропроцессора, и на последнем уроке мы рассмотрим обработку прерываний в этом режиме. Для глубокого понимания процессов, происходящих в компьютере при осуществлении прерывания, необходимо узнать о том, какие ресурсы компьютера при этом задействуются, каковы их характеристики и принципы функционирования.
- В общем случае система прерываний — это совокупность программных и аппаратных средств, реализующих механизм прерываний.
- К аппаратным, средствам системы прерываний относятся:

- 1. Выводы микропроцессора:
 - 1.1 INTR — вывод для входного сигнала внешнего прерывания. На этот вход поступает выходной сигнал от микросхемы контроллера прерываний 8259A;
 - 1.2 INTA — вывод микропроцессора для выходного сигнала подтверждения получения сигнала прерывания микропроцессором. Этот выходной сигнал поступает на одноименный вход INTA микросхемы контроллера прерываний 8259A;
 - 1.3 NMI — вывод микропроцессора для входного сигнала немаскируемого прерывания,
- 2. Микросхема программируемого контроллера прерываний 8259A. Она предназначена для фиксирования сигналов прерываний от восьми различных внешних устройств. В силу ее важной роли при работе всей вычислительной системы мы ее подробно рассмотрим ниже;
- 3. Внешние устройства: таймер, клавиатура, магнитные диски и т.д.
- К программным средствам системы прерываний реального режима относятся:
- 4. Таблица векторов прерываний. В этой таблице в определенном формате, который зависит от режима работы микропроцессора, содержатся указатели на процедуры обработки соответствующих прерываний,
- 5. Следующие флаги в регистре флагов `flags\eflags`:
 - 5.1 IF (Interrupt Flag) — флаг прерывания. Предназначен для так называемого маскирования (запрещения) аппаратных прерываний, то есть прерываний по входу INTR. На обработку прерываний остальных типов флаг IF влияния не оказывает. Если `IF=1`, микропроцессор обрабатывает внешние прерывания, если `IF=0`, микропроцессор игнорирует сигналы на входе INTR;
 - 5.2 TF (Trace Flag) — флаг трассировки. Единичное состояние флага TF переводит микропроцессор в режим покомандной работы. В режиме покомандной работы после выполнения каждой машинной команды в микропроцессоре генерируется внутреннее прерывание с номером 1, и далее следуют действия в соответствии с алгоритмом обработки данного прерывания;
- 6. Машинные команды микропроцессора: `int,into,iret,c1i,sti`.

2. Контроллер прерываний

- На наш взгляд, знакомство с системой прерываний микропроцессора Intel следует начать с обсуждения организации обработки аппаратных прерываний. Как видно из рис. 15.1, центральное место в схеме обработки аппаратных прерываний занимает программируемый контроллер прерываний (ПКП), выполненный в виде специальной микросхемы i8259A. Как мы уже говорили, эта микросхема может обрабатывать запросы от восьми источников внешних прерываний. Этого явно мало, поэтому в стандартной конфигурации вычислительной системы обычно используют две последовательно соединенные микросхемы i8259A. В результате такого соединения количество возможных источников внешних прерываний возрастает до 15.
- Для того чтобы разобраться с обработкой аппаратных прерываний, нам придется проникнуть внутрь микросхемы i8259A. Перечислим функции, выполняемые микросхемой контроллера прерываний:
- фиксирование запросов на обработку прерывания от восьми источников, формирование единого запроса на прерывание и подача его на вход INTR микропроцессора;
- формирование номера вектора прерывания и выдача его на шину данных;
- организация приоритетной обработки прерываний;
- запрещение (маскирование) прерываний с определенными номерами.
- На рис. 15.2 показано схематическое представление внутренней структуры и физических выводов микросхемы i8259A.

- Рассмотрим назначение представляющих для нас интерес выводов i8259A:
- d0...d7 — выходы i8259A, замыкающиеся на системную шину данных. По ним передается номер вектора прерывания и принимается управляющая информация;
- INT — вывод выходного сигнала запроса на прерывание, который подается на вход микропроцессора INTR;
- INTA — вывод для сигнала от микропроцессора, подтверждающего факт принятия им прерывания на обслуживание;
- irq0...irq7 — выходы для входных сигналов запросов на прерывания от внешних устройств.



- Важное свойство данного контроллера — возможность его программирования, что позволяет достаточно гибко изменять алгоритмы обработки аппаратных прерываний. Исходя из этого, микросхема i8259A имеет два состояния:
- 1. Состояние настройки параметров обслуживания прерываний, во время которого путем посылки в определенном порядке так называемых управляющих слов производится инициализация контроллера;
- 2. Состояние работы — это обычное состояние контроллера, в котором производится фиксация запросов на прерывание и формирование управляющей информации для микропроцессора в соответствии с параметрами настройки.
- Рассмотрим назначение основных структурных компонентов контроллера прерываний (см. рис. 15.2):
- 1. Регистр запросов на прерывания IRR (Interrupt Request Register) — восьмиразрядный регистр, фиксирующий поступление сигнала на один из входов i8259A — irq0...irq7. Фиксация выражается в установке соответствующего бита в единичное состояние;
- 2. Регистр маскирования прерываний IMR (Interrupt Mask Register) — восьмиразрядный регистр, с помощью которого можно запретить обработку запросов на прерывания, поступающих на соответствующие входы (уровни) irq0...irq7. Для запрещения (маскирования) определенных уровней прерываний необходимо установить соответствующие биты регистра IMR. Эта операция осуществляется путем программирования порта 21 h;
- 3. Регистр обслуживаемых прерываний ISR (Interrupt Service Register) — восьмиразрядный регистр, единичное состояние разрядов которого показывает, прерывания каких уровней обрабатываются в данный момент в микропроцессоре;
- 4. Арбитр приоритетов PR (Priority Resolver) — функцией данного блока является разрешение конфликта при одновременном поступлении запросов на входы irq0...irq7;
- 5. Блок управления — основной функцией данного блока является организация информационного обмена контроллера прерываний и микропроцессора через шину данных. На этот блок замыкаются как выводы d0...d7, так и некоторые другие (см рис 15 2)

- Рассмотрим возможные прохождения и обработку сигнала прерывания от некоторого внешнего устройства. При этом воспользуемся структурной схемой контроллера прерываний и обозначениями на ней (см. рис. 15.2).
- Допустим, на вход irq_0 поступает сигнал прерывания, что приводит к установке нулевого бита регистра IRR. Этот регистр связан с регистром маски IMR, состояние битов которого определяет, какие уровни прерываний запрещены (единичные биты) или разрешены к обработке (нулевые биты). Управление данным регистром осуществляется через порт $21h$. Таким образом, если бит 0 в IMR равен нулю, то прерывание уровня 0 разрешено. Далее сигнал поступает к арбитру приоритетов. Как мы уже отметили, функция этого блока - разрешение конфликтов при одновременном поступлении запросов на несколько уровней. Обычно самый высокий приоритет у уровня irq_0 , и далее приоритет уменьшается с возрастанием номера уровня. Если конфликта нет, то сигнал поступает на схему управления контроллером прерываний, которая формирует сигнал на выводе int . Этот вывод связан со входом микропроцессора INTR. Таким образом, сигнал на входе $i8259A$ достиг микропроцессора. Что происходит далее в микропроцессоре, мы рассмотрим ниже. Сейчас отметим только значимые для данного обсуждения моменты. Итак, при поступлении сигнала на вход INTR в микропроцессоре происходят следующие процессы:

- 1. Анализируется флаг IF. Если вы помните, единичное состояние этого флага говорит о том, что аппаратные прерывания разрешены, нулевое — запрещены.
- 2. Если прерывания запрещены, то запрос на прерывание «повисает» до момента установки IF в единицу.
- 3. Если прерывания разрешены, микропроцессор выполняет следующие действия
 - 3.1 Сбрасывает флаг IF в ноль;
 - 3.2 Формирует сигнал подтверждения прерывания на выводе микропроцессора INTA. Этот вывод микропроцессора замкнут на одноименный вывод микросхемы i8259A.
- Таким образом, сигнал о прерывании прошел через микропроцессор и вернулся обратно в контроллер прерываний i8259A через вывод INTA. Данный вывод внутри контроллера прерываний замкнут на его схему управления, которая выполняет сразу несколько действий при поступлении этого сигнала:
 - 1. Сбрасывает бит в регистре IRR, соответствующий уровню прерывания irq_0
 - 2. Устанавливает в 1 бит 0 регистра ISR, тем самым фиксируя факт обработки прерывания уровня 0 в микропроцессоре.
 - 3. Формирует с помощью блока управления номер вектора прерывания, значение которого формируется в буфере данных и далее поступает на выводы i8259A d0...d7. Выводы d0...d7 замкнуты на шину данных, по которой номер вектора поступает в микропроцессор. В микропроцессоре этот номер используется для вызова соответствующей процедуры обработки прерывания.

- На данном этапе обработки прерывания, после того как номер прерывания по шине данных поступил в микропроцессор, последнему стало известно все об источнике прерывания. Далее микропроцессор осуществляет процедуру обработки прерывания. Если в это время придет другой сигнал о прерывании того же уровня, то он будет запомнен установкой бита в IRR, и обслуживание этого прерывания будет отложено. Если приходит прерывание другого уровня, то его дальнейшая обработка зависит от приоритета, который оно имеет по отношению к уже обрабатываемым прерываниям. Если приоритет выше, то текущая процедура обработки прерывания останавливается, и вызывается процедура обработки более приоритетного прерывания.

- Очень важный момент связан с процессом завершения обработки прерывания. Проблема здесь состоит в следующем. После принятия микропроцессором запроса на обслуживание прерывания в контроллере устанавливается бит в регистре ISR, номер этого бита соответствует уровню прерывания. Установка бита с данным номером блокирует все прерывания уровня, начиная с текущего, и менее приоритетные в блоке-арбитре приоритетов. Если процедура прерывания закончит свою работу, то она сама должна этот бит сбросить, иначе все прерывания этого уровня и менее приоритетные будут игнорироваться. Для осуществления такого сброса необходимо послать код 20h в порт 20h. Есть и другая возможность — установить такой режим работы микросхемы i8259A, когда сброс этого бита будет производиться автоматически. Тонкий момент заключается в том, что происходит такой автоматический сброс будет одновременно с приходом сигнала INTA (то есть извещения о том, что запрос на обработку прерывания принят к обработке микропроцессором). Недостаток автоматического сброса в том, что существует вероятность прихода прерывания того же уровня, который уже обрабатывается в данный момент микропроцессором.

- В этом случае процедура обработки прерывания должна обладать свойством реентерабельности, то есть допускать повторное обращение к себе до завершения обработки предыдущего обращения. Для того чтобы процедура была реентерабельной, она должна иметь специфическую структуру, в частности, для каждого сеанса обращения к ней создается своя область для хранения переменных и значений регистров, а исполняемая часть процедуры находится в оперативной памяти только в одном экземпляре. Иногда может потребоваться подобный автоматический сброс, но надежнее и проще, конечно, контролировать этот процесс и самостоятельно сбрасывать бит в ISR. Это можно сделать либо в конце работы процедуры, либо в том месте процедуры, начиная с которого можно разрешить рекурсивный вызов данной процедуры, будучи уверенным в том, что она не разрушит никаких данных и работу программы в целом.

- Другой не менее интересный момент заключается в том, что микропроцессор при принятии к обработке запроса на прерывание сбросил флаг IF в ноль, тем самым запретив все последующие аппаратные прерывания. Этим обстоятельством программист может пользоваться по своему усмотрению. Вы, конечно, помните, что все запросы на прерывания с приоритетом, равным текущему или меньшим, будут запрещены в любом случае, — это обусловлено логикой работы контроллера i8259A. Поэтому программист должен решить, насколько его замыслам могут помешать запросы на более приоритетные прерывания. Если это некритично, то лучше сразу, в начале процедуры обработки прерывания установить флаг IF в единицу. В большинстве случаев эту операцию нужно делать как можно раньше. Для установки флага IF в единицу в системе команд микропроцессора есть специальная команда, не имеющая операндов.
- sti— разрешить аппаратные прерывания

- Наиболее наглядный пример, показывающий важность своевременной установки IF, связан с отсчетом времени. Если вы не знакомы с тем, как ведется учет времени в компьютере, то уделим этому немного внимания. Как после включения компьютер определяет текущее время суток или как он запоминает информацию о своей конфигурации после выключения? Все дело в том, что компьютер имеет небольшую энергонезависимую память, которая питается от аккумулятора и не зависит от подключения к электросети. Конструктивно эта память выполнена на специальном типе полупроводниковых элементов с так называемой CMOS-структурой (Complementar Metal Oxide Semiconductor — комплиментарная МОП-структура). Особенность таких элементов памяти — в их пониженной по сравнению с обычными микросхемами потребляемой мощности (при этом они являются и более медленными, что в данном случае не принципиально). Аккумулятор кроме CMOS-памяти питает еще и микросхему системных часов, в функции которой входит отсчет текущих даты и времени суток. Таким образом, текущие значения даты и времени постоянно хранятся в CMOS-памяти и поддерживаются в актуальном состоянии даже после выключения компьютера. Кроме того, в CMOS-памяти хранится некоторая другая информация, в частности, о конфигурации компьютера. Во время загрузки компьютера дата и время считываются в область данных BIOS. Дальнейший отсчет времени, после загрузки системы, ведется уже с помощью системного таймера — другой микросхемы на системной плате, в функции которой входит регулярно, примерно 18,2 раза в секунду, генерировать сигнал, который в качестве прерывания подается на уровень irq0 контроллера прерываний i8259A. Во время работы компьютера соответствующая программа BIOS обрабатывает прерывание данного уровня и ведет счет времени. Если терять такты по этому входу, то фактическое время на часах будет отставать, и поэтому в большинстве случаев в обработчиках прерываний есть смысл, как можно раньше выдавать команду sti.

3. Программирование контроллера прерываний i8259A

- Большая популярность применения этой микросхемы в качестве диспетчера аппаратных прерываний в компьютерах на базе микропроцессоров Intel объясняется наличием большого количества различных режимов ее работы, что позволяет сделать подсистему прерываний достаточно гибкой и эффективной. Действительно, если посмотреть на развитие аппаратной части компьютеров, начиная, например, с i8088/8086, то видно, что менялись самые разные компоненты, но подсистема прерываний, основанная на микросхеме i8259A, так и осталась неизменной.

- В процессе загрузки компьютера и в дальнейшем во время работы контроллер прерываний настраивается на работу в одном из четырех режимов:
- 1. FNM (Fully Nested Mode) — режим вложенных прерываний. В этом режиме каждому уровню (уровню) $irq0\dots irq7$ присваивается фиксированное значение приоритета, причем уровень $irq0$ имеет наивысший приоритет, а $irq7$ — наименьший. Приоритетность прерываний определяет их право на прерывание обработки менее приоритетного прерывания более приоритетным (при условии, конечно, что $IF=1$).
- 2. ARM (Automatic Rotation Mode) — режим циклической обработки прерываний. В этом режиме значения приоритетов уровней прерываний также линейно упорядочены, но уже не фиксированным образом, а изменяются после обработки очередного прерывания по следующему принципу: значению приоритета последнего обслуженного прерывания присваивается наименьшее значение. Следующий по порядку уровень прерывания получает наивысшее значение, и поэтому при одновременном приходе запросов на прерывания от нескольких источников преимущество будет иметь этот уровень. Это дает возможность обеспечить «равноправие» при обработке прерываний.
- 3. SRM (Specific Rotation Mode) — режим адресуемых приоритетов. Этот режим можно рассматривать как вариант режима ARM. В режиме SRM программист или система самостоятельно могут назначить уровень прерывания с наивысшим приоритетом.
- 4. PM (Polling Mode) — режим опроса. Этот режим запрещает контроллеру автоматически прерывать работу микропроцессора при появлении прерывания от некоторого внешнего устройства. Для того чтобы микропроцессор смог узнать о наличии того или иного запроса на прерывание, он должен сам обратиться к $i8259A$ для получения содержимого IRR, проанализировать его и далее действовать, но своему алгоритму. Данный режим моделирует так называемую опросную дисциплину обработки прерываний. Мы упоминали о ней в начале урока. Согласно этому подходу, инициатором обработки прерывания становится не само прерывание, как при векторной дисциплине, а микропроцессор, причем в определяемые им (точнее, операционной системой, выполняемой на нем) моменты времени.

- Программирование контроллера прерываний осуществляется через адресное пространство ввода-вывода посредством двух 8-битовых портов с адресами 20h и 21h. Управление контроллером осуществляется путем посылки в определенной последовательности в эти порты специальных приказов двух типов.
- ICW (Initialization Control Word) — управляющее слово инициализации. Всего имеются четыре таких слова с жесткой внутренней структурой — ICW1...ICW4. Эти слова предназначены для задания режима работы контроллера. Количество этих слов (4) определено количеством режимов (см. выше).
- OCW (Operation Control Word) — операционное управляющее слово. Таких слов всего три, и они несут информационную нагрузку для определенных выше режимов работы контроллера прерываний. Обычно их обозначают OCW1...OCW3.

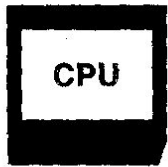
- Сначала рассмотрим программирование процедур работы с устройствами ввода-вывода. Остановимся на вопросах программирования ввода-вывода прерываний для отображения информации на дисплее, ввода с клавиатуры, вывода на принтер и работы с файлами.

- Программирование работы с дисплеем
- Задание режимов работы и обмен данными с дисплеем можно выполнять при прерываниях BIOS тип 10H, а вывод данных на дисплей и при прерываниях DOS тип 21H. Для уяснения отличий прерываний BIOS от прерываний DOS рассмотрим в качестве примера несколько функций видеопрерываний BIOS.
- Видеооперации с прерыванием 10H BIOS. Это прерывание обеспечивает выполнение 16 различных процедур работы с дисплеем (идентифицируются содержимым регистра AH). Приведем самые важные из них

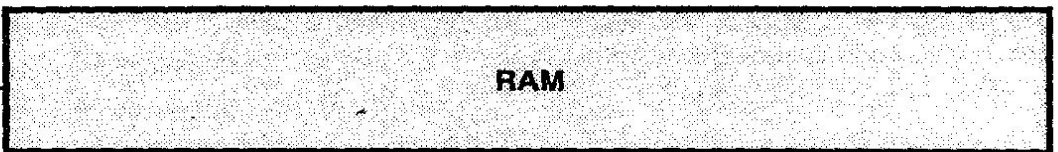
- 1 Перемещение курсора в заданную позицию- $AH=2$ Координаты курсора (строка, столбец) предварительно засылаются соответственно в регистры DH , DL В регистре BH указывается номер страницы буфера по умолчанию и в графическом режиме BH^O .
- 2 Очистка экрана дисплея' $AH=6$ и $AH=0$, или $AH=7$ и $AH=0$. 3. Чтение символа, находящегося в текущей позиции курсора, и его атрибута: $AH \gg 8$. В регистре BH указываются номер страницы буфера (для текстовых режимов). Считанный символ возвращается в AL , а его атрибут — в AH (атрибут символа — это его характеристики: цвет, цвет фона, яркость, инвертирование, подчеркивание, мигание и т. д.).
- 4. Задание режима изображения (текстовый, графический, цветность, формат, разрешающая способность и т. п.): $AH=0$. Режим изображения определяется со держимым регистра AL .
- 5. Запись нового атрибута символа и вывод символа в текущую позицию курсора: $AH=9$. Прочие регистры должны содержать:
 - $O AL$ — записываемый символ;
 - $Q BL$ — его атрибут;
 - $a BH$ — страницу буфера;
 - $O CX$ — счетчик записываемых символов (число повторений символа).
- 6. Вывод символа в текущую позицию курсора: $AH=AH$ или $AH=9$ и $BL=0$. Содержимое AL , BH и CX аналогично п. 5.
- 7. Чтение текущего видеостатуса: $AH=FH$. Возвращает в: AL — текущий режим; AH — число столбцов на экране; BH — активную страницу буфера.
- 8. Вывод строки символов: $AH=13H$ и $AL=0$. Прочие регистры должны содержать:
 - $a ES: BP$ — указатель строки (ее начальный адрес в памяти);
 - $O CX$ — длину строки;
 - $O DX$ — строку дисплея;
 - $a BH$ — страницу буфера.

- Видеооперации с прерыванием 21 H DOS
- 1. Вывод символа на экран дисплея: AH=2 или AH=6 и DL<>FFH. В регистре DL должен быть ASCII-код символа. Пример фрагмента программы (вывод символа «C»):
 - mov AH, 6
 - mov DL, 43H ; 43H - это ASCII код символа «C»
 - int 21H
- 2. Вывод строки символов: AH=9 (чаще всего используемая функция). В регистрах DS: DX должен быть начальный адрес строки символов, которая должна заканчиваться символом \$. Пример фрагмента программы (вывод строки 'вывод строки символов\$'):
 - Text db 'вывод строки символов\$' mov AH, 9
 - mov DX, offset text ; offset text - это адрес : выводимой строки
 - int 21H
- 3. Вывод на экран текстового файла: AH=40H. Стандартные файловые указатели определяют тип и устройство ввода-вывода:
 - O 0 Ввод с клавиатуры;
 - Q 1 Вывод на экран дисплея;

- Прямой доступ к памяти
- Передача данных в режиме прямого доступа к памяти {Direct Memory Access, DMA} требуется при обмене данными между оперативной памятью и высокоскоростными устройствами. В режиме прямого доступа периферийное устройство связано с оперативной памятью непосредственно через каналы прямого доступа (каналы DMA), а не через внутренние регистры микропроцессора. Наиболее эффективным такой режим обмена данными бывает в ситуациях, когда требуется высокая скорость для передачи большого объема информации (например, при загрузке данных в память с компакт-диска).
- Для инициализации процесса прямого доступа на системной шине используются соответствующие сигналы. Так, устройство, требующее прямой доступ к памяти, по одному из свободных каналов DMA обращается к контроллеру, сообщая ему путь (адрес), откуда или куда переслать данные, начальный адрес блока данных и объем данных (рис. 5.19). Инициализация обмена происходит с участием процессора, но собственно передача данных осуществляется уже под управлением контроллера DMA, а не процессора.
- Для организации прямого доступа в память в компьютерах IBM PC/XT использовалась одна 4-канальная микросхема DMA i8237, канал 0 которой предназначен для регенерации динамической памяти. Каналы 2 и 3 служат для управления высокоскоростной передачей данных между дисковыми гибкими дисками, винчестером и оперативной памятью соответственно. Доступным является только канал DMA 1.



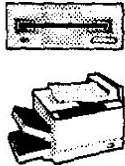
Шина памяти



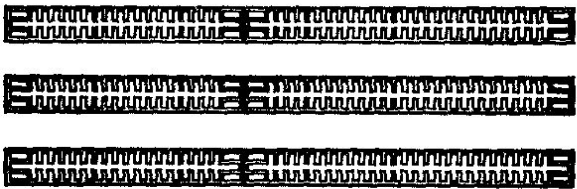
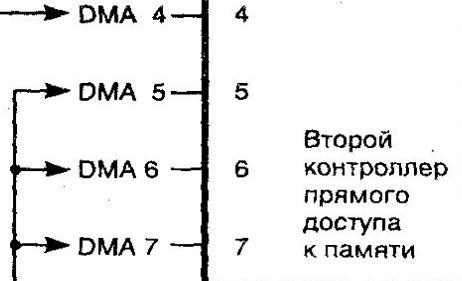
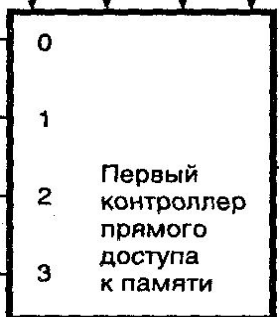
RAM

DMA 0
DMA 1
DMA 2
DMA 3

DMA 5
DMA 6
DMA 7



DMA 0
DMA 1
DMA 2
DMA 3



- IBM PC/AT-совместимые компьютеры имеют уже 7 каналов прямого доступа к памяти. Как и для контроллеров прерывания, это достигается путем каскадного включения двух микросхем i8237, интегрированных в микросхему контроллера периферии, одна из линий (канал DMA 4) используется для подключения каналов DMA 0—DMA 3.
- Из всех каналов DMA стандартно задействован только канал DMA 2 для обмена данными с дисководом гибких дисков. Каналы DMA 1 и DMA 2 обычно используются звуковыми картами.
- Один канал DMA может использоваться различными устройствами, но не одновременно, например, канал DMA 1 — для звуковой карты и стримера. Однако в этом случае вы не сможете одновременно слушать музыку и записывать данные на ленту.

- Порты ввода/вывода
- В РС любое устройство (контроллеры, установленные на картах расширения или на материнской плате) за исключением оперативной памяти рассматривается центральным процессором как периферийное. Обмен данными между CPU и периферийными устройствами осуществляется через так называемые порты ввода/вывода. Конструктивно порт ввода/вывода — это буферное устройство или регистр контроллера, процессора обработки сигналов и т.п., которые непосредственно подключены к шине ввода/вывода РС. Для управления обменом данными между аппаратными компонентами компьютера каждому порту ввода/вывода присваивается свой уникальный шестнадцатеричный номер (адрес порта), например 2F8h, 370h. Можно провести аналогию с почтой: чтобы письмо попало по назначению, на конверте должен быть правильно указан адрес получателя. Напомним — символ "h" (hex) означает, что адрес указывается в шестнадцатеричной системе исчисления

- В IBM-совместимых PC можно адресовать (использовать) 65536 (2¹⁶) портов ввода/вывода (хотя большинство из них, как правило, не используется). Причем адресное пространство портов ввода/вывода не совпадает с адресным пространством памяти, что дает возможность иметь полный объем памяти и полный набор портов ввода/вывода. Стандартный диапазон адресов портов ввода/вывода для IBM PC-совместимых компьютеров составляет 0—3FFh включительно, хотя реально можно использовать адреса вплоть до FFFFh (например, порты процессора WT-синтеза звуковой карты (см. главу 14) имеют адреса в диапазоне 620h—E23h). Сюда входят порты контроллеров клавиатуры, жестких и гибких дисков, видеоадаптеров EGA/VGA, звуковых карт, последовательных и параллельных интерфейсов, игровых портов и любого другого периферийного оборудования.

- Отметим, что периферийными устройствами могут использоваться несколько портов ввода/вывода. Иногда их количество может достигать нескольких десятков. Например, контроллер параллельного интерфейса, к которому обычно подключается принтер, имеет три регистра: регистр вывода данных, регистр состояния и регистр управления, адресуемые через свои порты ввода/вывода, а контроллер последовательного интерфейса — десять регистров, адресуемых через семь портов ввода/вывода.
- Естественно, адресация портов осуществляется центральным процессором при выполнении той или иной программы. Для того чтобы исключить необходимость указания конкретного адреса каждого порта ввода/вывода при программировании, а также для оперативного их изменения в зависимости от конкретной конфигурации компьютера, существует понятие базовый адрес порта ввода/вывода (Base Adress In/Out).

- Базовый адрес порта ввода/вывода присваивается каждому периферийному устройству и соответствует младшему адресу из группы портов (обычно адресу порта регистра данных). Адресация остальных портов периферийного устройства осуществляется путем задания смещения (целое число) относительно базового адреса. Например, для адресации порта регистра состояния контроллера параллельного интерфейса необходимо значение базового адреса порта LPT увеличить на единицу. Следует помнить, что под базовым адресом ввода/вывода любого периферийного устройства понимается весь диапазон адресов портов ввода/вывода.
- BIOS резервирует ряд диапазонов адресов портов ввода/вывода стандартных аппаратных компонентов PC которые не могут быть использованы другими периферийными устройствами (табл. 5.16).

Диапазон адреса	Назначение
000–01F	Первый контроллер DMA
020–03F	Первый контроллер прерываний
040–05F	Таймер
060–06F	Контроллер клавиатуры
070–07F	Текущее время
080–09F	Регистр страниц DMA
0A0–0BF	Второй контроллер прерывания
0C0–0DF	Второй контроллер DMA
0F0–0FF	Сопроцессор
100–1EF	Свободен
1F0–1F8	Контроллер винчестера
1F9–1FF	Свободен
200–20F	Игровой порт (джойстик)
200–217	Свободен
220–24F	Свободен
250–277	Свободен
278–27F	LPT2
280–2EF	Свободен
2F8–2FF	COM2
300–31F	Свободен
330–35F	Свободен
360–36F	Сетевая карта
370–377	Свободен
378–37F	LPT1
380–38F	SDLC 2 (синхронное управление передачей данных 2)

390—39F	Свободен
3A0—3AF	SDLC 1 (синхронное управление передачей данных 1)
3B0—3BF	Монохромная графическая карта, параллельный интерфейс (IBM PS/2)
3C0—3CF	Графическая карта EGA
3D0—3DF	Графическая карта CGA
3E0—3EF	Свободен
3F0—3F7	Контроллер дисководов
3F8—3FF	COM1

- Например, под интерфейс LPT1 резервируется диапазон адресов 378h— 37Eh, в пределах которого можно выбирать базовый адрес порта ввода/вывода. Тогда будет исключена возможность конфликтов при адресации портов на аппаратно-программном уровне. Поскольку для управления и контроля LPT1 используются три регистра, в пределах этого диапазона адресов имеется 5 возможных значений базового адреса — 378h—37Ch. Диапазон адресов 200h—207h зарезервирован под игровой порт для джойстика, хотя фактически из восьми адресов обычно используется только один — 200h или 207h. Подобная ситуация наблюдается и для других устройств.
- К сожалению, на практике стандартные адреса портов и прерывания используются не всеми изготовителями аппаратного обеспечения так, как было описано. Тем не менее, вы можете самостоятельно сконфигурировать плату расширения, чтобы это не привело к конфликту между двумя прерываниями или даже адресами портов.
- Изменить базовый адрес порта ввода/вывода можно с помощью джамперов на карте контроллера (материнской плате), программно (например, используя команду Пуск (Start), Настройка (Setting), Панель управления (Control Panel), Система (System) в Windows 95/98) или через CMOS Setup.