

Обработка прерываний

Оглавление

Введение	2
Понятие прерывания.....	2
Цель и механизм обработки прерываний	2
Элементы механизма прерывания	2
Функции механизма прерывания	3
Реакция на появление сигнала запроса прерывания	3
Классификация прерываний	5
Классы прерываний.....	5
Типы внешних прерываний.....	5
Режимы работы процессора	6
Аппаратные (внешние) прерывания.....	7
Программные (внутренние) прерывания	8
Исключения (внутренние прерывания).....	8
Аппаратные и программные средства системы прерываний	8
Таблица векторов прерываний	9
Таблица векторов	9
Приоритезация прерываний	10
Обработка прерываний	11
Обработка прерывания в реальном (однозадачном) режиме	11
Работа системы прерываний в защищенном (многозадачном) режиме	11
Особенности обработки аппаратных прерываний.....	12
Контроллер прерывания.....	15
Выводы	16

Введение

Понятие прерывания

Для обработки событий, происходящих асинхронно (см. ниже) по отношению к выполнению программы, лучше всего подходит механизм прерываний. Прерывание можно рассматривать как некоторое особое событие в системе, требующее моментальной реакции.

Практически **все системы ввода/вывода в компьютере работают с использованием прерываний**. В частности, когда вы нажимаете клавиши или щелкаете мышью, аппаратура вырабатывает прерывания. В ответ на них система, соответственно, считывает код нажатой клавиши или запоминает координаты курсора мыши. **Прерывания вырабатываются контроллером** диска, портами последовательной передачи данных, звуковым адаптером и другими устройствами.

Кажется очевидным, что возможны самые разнообразные прерывания по самым различным причинам. Поэтому с прерыванием связывают число - так называемый номер прерывания.

Этот номер однозначно соответствует тому или иному событию. Система умеет распознавать прерывания и при их возникновении запускает процедуру, соответствующую номеру прерывания.

Некоторые прерывания (первые пять по порядку номеров) зарезервированы для использования центральным процессором на случай каких-либо особых событий вроде попытки деления на нуль, переполнения и т. п.

Прерывание (англ. *interrupt*) — сигнал, сообщаящий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается и управление передается **обработчику** прерывания, который реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код.

Цель и механизм обработки прерываний

Основная цель введения прерываний – реализация асинхронного режима работы и распараллеливания работы отдельных устройств вычислительного комплекса. Механизм прерываний реализуется аппаратно-программными средствами.

Структуры систем прерываний могут быть самыми различными, но все они имеют общую особенность – прерывание непременно ведет за собой изменение порядка выполнения команд процессором.

Элементы механизма прерывания

Механизм обработки прерываний включает в себя следующие элементы:

1. **Установление** факта прерывания (прием и идентификация сигнала на прерывание).

2. **Запоминание** состояния прерванного процесса (состояние процесса определяется значением счетчика команд, содержимым регистра процессора, спецификацией режима: пользовательский или привилегированный)
3. **Управление** аппаратно передается программе обработки прерывания. В этом случае, в счетчик команд заносится начальный адрес подпрограммы обработки прерывания, а в соответствующие регистры из слова состояния.
4. **Сохранение** информации в прерванной программе, которую не удалось спасти с помощью действий аппаратуры.
5. **Обработка** прерывания. Работа может быть выполнена той же подпрограммой, которой было передано управление на 3-ем шаге, но в ОС чаще всего эта обработка реализуется путем вызова соотв. подпрограммы.
6. **Восстановление** информации относящейся к прерванному процессу.
7. **Возврат** в прерванную программу.

Первые 3 шага реализуются аппаратными средствами, а остальные – программно.

Функции механизма прерывания

Главные функции механизма прерывания:

1. Распознавание или классификация прерывания.
2. Передача управления обработчику прерывания.
3. Корректное возвращение к прерванной программе

Переход от прерванной программе к обработчику и обратно должен производиться **как можно быстрее**.

Одним из **быстрых методов** является использование таблицы, содержащей перечень всех допустимых для компьютера прерываний и адреса соответствующих обработчиков. Для корректного возвращения к прерванной программе, перед передачей управления обработчику, содержимое регистров процессора запоминается либо в памяти с прямым доступом либо в системном стеке.

Реакция на появление сигнала запроса прерывания

Рассмотрим функционирование компьютера при появлении сигнала запроса прерывания, опираясь в основном на обработку аппаратных прерываний ([рис. 1](#)).



Рис. 1. Выполнение прерывания в компьютере: t_p - время реакции процессора на запрос прерывания; t_c - время сохранения состояния прерываемой программы и вызова обработчика прерывания; t_v - время восстановления прерванной программы

После появления сигнала запроса прерывания ЭВМ переходит к выполнению программы - **обработчика прерывания**. Обработчик выполняет те действия, которые необходимы в связи с возникшей особой ситуацией. Например, такой ситуацией может быть нажатие клавиши на клавиатуре компьютера. Тогда обработчик должен передать код нажатой клавиши из контроллера клавиатуры в процессор и, возможно, проанализировать этот код. По окончании работы обработчика управление передается прерванной программе.

Время реакции - это время между появлением сигнала запроса прерывания и началом выполнения прерывающей программы (обработчика прерывания) в том случае, если данное прерывание разрешено к обслуживанию.

Два подхода ко времени реакции

Время реакции зависит от момента, когда процессор определяет факт наличия запроса прерывания. Опрос запросов прерываний может проводиться либо:

- по окончании выполнения очередного этапа команды (например, считывание команды, считывание первого операнда и т.д.),
- либо после завершения каждой команды программы.

Первый подход обеспечивает более быструю реакцию, но при этом необходимо при переходе к обработчику прерывания сохранять большой объем информации о прерываемой программе, включающей состояние буферных регистров процессора, номера завершившегося этапа и т.д. При возврате из обработчика также необходимо выполнить большой объем работы по восстановлению состояния процессора.

Во втором случае время реакции может быть достаточно большим. Однако при переходе к обработчику прерывания требуется запоминание минимального контекста прерываемой программы (обычно это счетчик команд и регистр флагов). В настоящее время в компьютерах чаще используется распознавание запроса прерывания после завершения очередной команды

Классификация прерываний

Классы прерываний

В зависимости от **источника возникновения** сигнала прерывания делятся на:

- **Аппаратные (внешние, асинхронные)** — события, которые исходят от внешних источников (например, периферийных устройств) и могут произойти в любой произвольный момент (принцип асинхронности): сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши;
- **Программные (внутренние)** — инициируются исполнением специальной инструкции в коде программы. Программные прерывания как правило используются для обращения к функциям встроенного программного обеспечения (firmware), драйверов и операционной системы. вызываются искусственно с помощью соответствующей команды из программы (int), предназначены для выполнения некоторых действий операционной системы, являются синхронными.
- **Исключения (внутренние)** — события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода: деление на ноль или переполнение, обращение к недопустимым адресам или недопустимый код операции;

Вторая классификация – по **местоположению** источника возникновения запроса:

- Внутренние
- Внешние

Типы внешних прерываний

Внешние прерывания возникают по сигналу какого-нибудь внешнего устройства.

Иногда желательно сделать систему нечувствительной ко всем или отдельным **аппаратным прерываниям** (т.е. от внешних устройств). Для этого используют так называемое маскирование внешних прерываний.

Внешние прерывания подразделяются на:

- **немаскируемые** (англ. *Non maskable interrupt, NMI*) — обрабатываются всегда, независимо от запретов на другие прерывания. К примеру, такое прерывание может вызвать сбой в микросхеме памяти
- **маскируемые** - которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний (в x86-процессорах — сбросом флага IF в регистре флагов);

Немаскируемые прерывания (говорят, что оно одно, т.к. подается на вывод микропроцессора NMI) инициируют источники, требующие безотлагательного вмешательства со стороны микропроцессора.

Маскируемые прерывания генерируются контроллером прерываний по заявке определенных периферийных устройств. Контроллер прерываний (выполнен в виде специальной микросхемы i8259A) поддерживает восемь уровней (линий) приоритета; к каждому уровню “привязано” одно периферийное устройство. Именно маскируемые прерывания часто называют аппаратными прерываниями



- В реальном и защищенном режиме работы микропроцессора обработка прерываний осуществляется принципиально разными методами.

Существуют два специальных внешних сигнала среди входных сигналов процессора, при помощи которых можно прервать выполнение текущей программы и тем самым переключить работу центрального процессора. Это сигналы NMI (no mask interrupt, немаскируемое прерывание) и INTR (interrupt request, запрос на прерывание).

Режимы работы процессора

Все 32-разрядные и более поздние процессоры Intel, начиная с 386, могут выполнять программы в нескольких режимах. Режимы процессора предназначены для выполнения программ в различных средах; в разных режимах работы возможности чипа не одинаковы, потому, что команды выполняются по-разному. В зависимости от режима процессора изменяется схема управления памятью системы и задачами.

Процессоры могут работать в трех режимах: реальном, защищенном и виртуальном реальном режиме (реальном внутри защищенного).

Реальный режим

В первоначальном IBM PC использовался процессор (8088), который мог выполнять **16-разрядные команды**, используя **16-разрядные внутренние регистры** и адресовать только **1 Мб памяти**, используя **20-и разрядную шину адреса**. Все программное обеспечение PC первоначально было предназначено для этого процессора, оно было разработано на основе 16-разрядной системы команд и модели памяти, объемом 1 Мб. Например DOS, все программное обеспечение DOS написано в расчете на 16-разрядные команды.

Более поздние процессоры (например, 286), могли также выполнять те же самые 16-разрядные команды, что и первоначальный 8088, **но намного быстрее**.

Другими словами процессор 286 был полностью совместим с первоначальным 8088. 16-разрядный режим, в котором выполнялись команды процессоров 8088 и 80286 был назван реальным режимом. Все программы, выполняющиеся в реальном режиме, должны использовать **только 16-разрядные команды и 20-разрядный адрес**.

Для программного обеспечения такого типа используется **однозадачный режим**, т.е. одновременно **должна выполняться только одна программа**. Нет никакой встроенной защиты для предотвращения перезаписи ячеек памяти, занятых одной программой или даже самой операционной системой, другими программами: это означает, что при выполнении нескольких программ вполне могут быть испорчены данные или код одной из программ, что может привести к остановке системы.

Защищенный режим

(режим защищенной виртуальной адресации)

Шло развитие аппаратной части.

Первым **32-разрядным процессором**, предназначенным для РС, был 386-ой.

Этот чип мог выполнять абсолютно **новую 32-разрядную систему команд**. Для того чтобы полностью использовать преимущество этой новой системы команд, были необходимы 32-разрядная операционная система и 32-разрядные приложения (**многозадачный режим**). Этот новый режим называли защищенным, так как выполняющиеся в нем программы защищены от перезаписи используемых ими областей памяти другими программами.

Такая защита делает систему более надежной, так как уже ни одна программа с ошибками не сможет повредить другие программы или операционную систему.

Аппаратные (внешние) прерывания

Аппаратные прерывания (IRQ, от Interrupt Requests) вызываются физическими устройствами и потому приходят асинхронно (в любой момент) по отношению к выполнению любых программ. Эти прерывания информируют систему о событиях, связанных с работой устройств. Например, о том, что наконец-то завершилась печать символа на принтере и неплохо было бы выдать следующий символ, или о том, что сектор диска уже прочитан и его содержимое доступно программе.

При аппаратном прерывании **иницируется сигнал** (запрос на прерывание), который сообщает процессору, что в системе произошло некоторое событие (например нажата клавиша), требующее его внимания. Когда вызывается прерывание, то процессор оставляет свою работу, выполняет прерывание, а затем возвращается на прежнее место.

Сигналы аппаратных прерываний, возникающих в устройствах компьютера, **поступают в процессор** через **2 контроллера прерываний**, входящих в состав многофункциональной микросхемы **периферийного контроллера**.

К входным выводам контроллеров (один из которых называют ведущим, а другой ведомым) подключаются выходы устройств, на которых возникают сигналы прерываний. Выход ведомого контроллера подключается к входу ведущего, а выход ведущего к входу INT микропроцессора. Кроме сигнала INT контроллеры передают в микропроцессор номер вектора (рассмотрим ниже), который образуется в контроллере путем сложения базового номера, записанного в одном из его регистров с номером входной линии, на который поступил запрос прерывания. Номера базовых векторов заносятся в контроллер в процессе начальной загрузки (и для ведущего контроллера в реальном режиме он равен 8, а для ведомого 70h). Номера аппаратных прерываний однозначно связаны с номерами линий, а через них с конкретными устройствами.





Рис. Аппаратная организация прерываний

Процессор получив сигнал по линии INT на выполнение прерывания сохраняет в стеке (элемент временного хранения данных) выполняемой программы содержимое регистров FLAGS, CS:IP. Загружает CS и IP из соответствующего вектора прерываний, и тем самым на обработчик прерываний, связанный с этим вектором и передает ей управление. Обработчик прерываний всегда завершается командой iret (interrupt return, возврат из прерывания), которая возвращает старые значения CS, IP и регистра флагов, тем самым приводя к возврату в точку программы, где она была прервана.

Программные (внутренние) прерывания

Программы могут сами вызывать прерывания с заданным номером. Для этого они используют команду INT. Это так называемые программные прерывания. Программные прерывания не являются асинхронными, так как вызываются из программы.

Программные прерывания удобно использовать для организации доступа к отдельным, общим для всех программ функциям. Например, функции операционной системы доступны прикладным программам именно через прерывания. При вызове этих модулей нет необходимости знать их текущий адрес в памяти.

Прикладные программы и драйверы могут сами устанавливать свои обработчики прерываний для их последующего использования другими программами. Для этого встраиваемые обработчики прерываний должны быть резидентными в памяти.

Исключения (внутренние прерывания)

Эти прерывания создаются цепями самого процессора при возникновении одной из специально оговоренных ситуаций. Если процессор сталкивается с одной из таких сбойных ситуаций, то он выполняет процедуру прерывания, используя закрепленной за ней вектор прерывания, который известен процессору.

Например :

Прерывание 0 вырабатывается в случае переполнения при операции деления на 0

Аппаратные и программные средства системы прерываний

Система прерываний - это совокупность программных и аппаратных средств, реализующих механизм прерываний.

К **аппаратным средствам** системы прерываний относятся:

- выводы микропроцессора - на них формируются сигналы, извещающие микропроцессор либо о том, что некоторое внешнее устройство «просит уделить ему внимание» (INTR, заявка на прерывание), либо о том, что требуется безотлагательная обработка некоторого события или катастрофическая ошибка (NMI)
- INTR - вывод для входного сигнала запроса на прерывание,
- NMI - вывод для входного сигнала немаскируемого прерывания
- INTA - вывод для выходного сигнала подтверждения получения сигнала прерывания микропроцессором;
- программируемый контроллер прерываний 8259A (предназначен для фиксирования сигналов прерываний от восьми различных внешних устройств; он выполнен в виде микросхемы; именно он формирует номер вектора прерывания и выдает его шину данных);
- внешние устройства (таймер, клавиатура, магнитные диски и т.п.)

К **программным средствам** системы прерываний реального режима относятся:

- **таблица векторов прерываний.**
Занимает первый килобайт ОП (адреса 00000h-003FFh).
Она содержит адреса (векторы - «векторы», т.к. два значения для указания адреса) обработчиков прерываний и состоит из 256 (0..255) элементов по 4 байта каждый:
- 2 байта - новое значение для регистра IP
- 2 байта - новое значение для регистра CS.
Расположение таблицы векторов прерываний в процессорах i80286 и старше определяется значением регистра IDTR.
Таблица векторов прерываний инициализируется при запуске системы, но в принципе может быть изменена и перемещена.
Каждый вектор имеет свой номер и называется номером прерывания.
- **два флага в регистре флагов flags/eflags:**
IF (Interrupt Flag) - флаг прерывания. Предназначен для маскирования (запрещения) аппаратных прерываний. Если IF=1, микропроцессор обрабатывает внешние прерывания, если = 0, то игнорирует;
TF(Trace Flag) - флаг трассировки. Если он=1, то микропроцессор переходит в режим покомандной работы. В этом режиме в микропроцессоре генерируется внутреннее прерывание с номером 1;
- **машинные команды микропроцессора: int, into** (прерывание по переполнению), **iret, cli, sti**

Таблица векторов прерываний

Таблица векторов

Для того чтобы связать адрес обработчика прерывания с номером прерывания, используется таблица векторов прерываний, занимающая первый килобайт оперативной памяти. Эта таблица находится в диапазоне адресов от 0000:0000 до 0000:03FFh и состоит из 256 элементов - дальних адресов обработчиков прерываний.

Элементы таблицы векторов прерываний называются векторами прерываний.

В первом слове элемента таблицы записана компонента смещения, а во втором - сегментная компонента адреса обработчика прерывания.

Вектор прерывания с номером 0 находится по адресу 0000:0000, с номером 1 - по адресу 0000:0004 и т. д.

Инициализация таблицы выполняется частично системой базового ввода/вывода BIOS после тестирования аппаратуры и перед началом загрузки операционной системой, частично при загрузке MS-DOS.

Ниже приведены некоторые (из 256) номера прерываний.

Номер	Описание
0	Ошибка деления. Вызывается автоматически, если в результате деления происходит переполнение (например, при делении на 0). Обычно при обработке этого прерывания MS-DOS выводит сообщение об ошибке и останавливает выполнение программы.
1	Прерывание пошагового режима. Вырабатывается после выполнения каждой машинной команды, если в слове флагов установлен бит пошаговой трассировки TF. Используется для отладки программ.
5	Печать копии экрана. Генерируется, если пользователь нажал клавишу <PrtSc>. В программах MS-DOS обычно используется для печати образа экрана.
9	IRQ1 - прерывание от клавиатуры. Генерируется, когда пользователь нажимает и отжимает клавиши. Используется для чтения данных из клавиатуры
16	Обслуживание клавиатуры
17	Обслуживание принтера
19	Перезагрузка операционной системы

Приоритезация прерываний

До окончания обработки прерывания обычно устанавливается запрет на обработку этого типа прерывания, чтобы процессор не входил в цикл обработки одного прерывания.

Приоритезация означает, что все источники прерываний делятся на классы и каждому классу назначается свой уровень приоритета запроса на прерывание. Т.е. процессор может вести себя по разному при поступлении более приоритетного запроса.

Приоритеты могут обслуживаться как относительные и абсолютные.

Относительное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то это прерывание будет обработано только после завершения текущей процедуры обработки прерывания.

Абсолютное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то текущая процедура обработки прерывания вытесняется, и процессор начинает выполнять обработку вновь поступившего

более приоритетного прерывания. После завершения этой процедуры процессор возвращается к выполнению вытесненной процедуры обработки прерывания.

Обработка прерываний

Обработка прерывания в реальном (однозадачном) режиме

производится в **три этапа**:

1) прекращение выполнения текущей программы;

Должно произойти так, чтобы потом вернуться и продолжить работу. Для этого необходимо сохранить содержимое регистров, так как они являются ресурсами, разделяемыми между программами.

Эти регистры сохраняются микропроцессором автоматически. Наиболее удобным местом хранения регистров является стек.

2) переход к выполнению и выполнение программы обработки прерывания;

Здесь определяется источник прерывания и вызывается соответствующий обработчик прерывания.

В реальном режиме микропроцессора допускается 256 источников - по кол-ву элементов таблицы векторов прерываний.

3) возврат управления прерванной программе.

Необходимо привести стек в состояние, в котором он был сразу после передачи управления данной процедуре.

Работа системы прерываний в защищенном (многозадачном) режиме

Обработка прерываний в защищенном режиме отличается от обработки в реальном режиме так же сильно, как и защищенный режим отличается от реального.

Потому что:

1. В защищенном режиме немного изменено распределение номеров векторов прерываний.
2. Принципиально иным является механизм обработки прерываний.

Классификация прерываний в защищенном режиме

Прерывания и исключения можно разделить на несколько групп:

- сбой;
- ловушка;
- аварийное завершение.

Это деление производится в соответствии со следующими признаками:

- какая информация сохраняется о месте возникновения прерывания (исключения)?
- возможно ли возобновление прерванной программы?

Исходя из этих признаков, можно дать следующие характеристики вышеперечисленным группам:

Сбой (ошибка) — прерывание или исключение, при возникновении которого в стек (в память) записываются значения регистров, указывающие на команду, вызвавшую данное прерывание.

Это позволяет, получив доступ к сегменту кода, исправить ошибочную команду в обработчике прерывания и, вернув управление программе, фактически осуществить ее рестарт (вспомните, что в реальном режиме при возникновении прерывания в стеке всегда запоминается адрес команды, следующей за той, которая вызвала это прерывание).

Стековой называют память, доступ к которой организован по принципу: "последним записан - первым считан" (Last Input First Output - **LIFO**).

Ловушка — прерывание или исключение, при возникновении которого в стек записываются значения регистров, указывающие на команду, следующую за командой, вызвавшей данное прерывание.

Так же, как и в случае ошибок возможен рестарт программы..

Аварийное завершение — прерывание, при котором информация о месте его возникновения недоступна или неполна и поэтому рестарт практически невозможен, если только данная ситуация не была запланирована заранее.

⇒ соответствующие программы-обработчики ошибок, ловушек и аварий будут отличаться алгоритмами работы.

Микропроцессор жестко определяет, какие прерывания являются ошибками, ловушками и авариями.

Особенности обработки аппаратных прерываний

Аппаратные прерывания вырабатываются устройствами компьютера, как правило, при завершении ими операций обмена данными или при изменении состояния. В зависимости от типа устройства обработчик прерывания может выполнять те или иные функции. Например, по прерыванию таймера соответствующий обработчик увеличивает содержимое счетчика, расположенного в оперативной памяти. По содержимому этого счетчика программы могут определить текущее время.

В отличие от программных прерываний, вызываемых запланировано программой или драйвером, аппаратные прерывания всегда происходят асинхронно по отношению к

выполняющимся программам. Кроме того, может возникнуть одновременно сразу несколько прерываний!

Для того чтобы система "не растерялась", решая какое прерывание обслуживать в первую очередь, существует **специальная схема приоритетов**. Каждому прерыванию назначается свой приоритет. Если происходит одновременно несколько прерываний, система отдает предпочтение самому высокоприоритетному, откладывая на время обработку остальных прерываний.

Система приоритетов реализована на двух микросхемах Intel 8259 (или аналогичных). Каждая микросхема является контроллером прерывания и обслуживает до восьми приоритетов. Микросхемы можно объединять (каскадировать) для увеличения количества уровней приоритетов в системе.

Уровни приоритетов обозначаются сокращенно IRQ0 - IRQ15 .

В компьютере типа IBM PC/XT была установлена только одна микросхема контроллера прерывания. Приоритеты линейно зависели от номера уровня прерывания. Прерывание IRQ0 соответствовало самому высокому приоритету, за ним шли прерывания IRQ1 , IRQ2 , IRQ3 и так далее.

Прерывание IRQ2 в компьютерах IBM PC/XT было зарезервировано для дальнейшего расширения системы. В компьютерах IBM PC/AT прерывание IRQ2 стало использоваться для каскадирования двух контроллеров прерывания 8259. Добавленные приоритетные уровни прерываний IRQ8 - IRQ15 в этих компьютерах располагаются по приоритету между прерываниями IRQ1 и IRQ3 .

Справочно:

Приведем список аппаратных прерываний, расположенных в порядке убывания приоритета:

Запрос	Источник	Приоритет	Номер вектора
NMI	Ошибка памяти или другая неисправимая ошибка в системе		02h
IRQ0	Системный таймер	1	08h
IRQ1	Клавиатура	2	09h
IRQ8	Часы реального времени	3	70h
IRQ9	Устройство на шине ISA	4	71h
IRQ10	Устройство на шине ISA	5	72h
IRQ11	Устройство на шине ISA	6	73h
IRQ12	Устройство на шине ISA	7	74h

IRQ13	Ошибка сопроцессора	9	75h
IRQ14	IDE контроллер	9	76h
IRQ15	Устройство на шине ISA	10	77h
IRQ3	Последовательный порт (COM2 или COM4)	11	0Bh
IRQ4	Последовательный порт (COM1 или COM3)	12	0Ch
IRQ5	Параллельный порт (LPT2) или IDE контроллер (вторичный)	13	0Dh
IRQ6	Контроллер дисковода	14	0Eh
IRQ7	Параллельный порт (LPT1)	15	0Fh

Номер	Описание
8	IRQ0 - прерывание интервального таймера, возникает 18,2 раза в секунду
9	IRQ1 - прерывание от клавиатуры
A	IRQ2 - используется для каскадирования аппаратных прерываний
70	IRQ8 - прерывание от часов реального времени
71	IRQ9 - прерывание от контроллера EGA
72	IRQ10 - зарезервировано
73	IRQ11 - зарезервировано
74	IRQ12 - зарезервировано
75	IRQ13 - прерывание от арифметического сопроцессора
76	IRQ14 - прерывание от контроллера жесткого диска
77	IRQ15 - зарезервировано
B	IRQ3 - прерывание асинхронного порта COM2
C	IRQ4 - прерывание асинхронного порта COM1
D	IRQ5 - прерывание от контроллера жесткого диска (только в компьютерах IBM PC/XT)

E	IRQ6 - прерывание генерируется контроллером НГМД
F	IRQ7 - прерывание принтера

Из этого списка видно, что самый высокий приоритет у прерываний от интервального таймера, затем идет прерывание от клавиатуры. Наименьший приоритет имеет прерывание принтера.

Контроллер прерывания

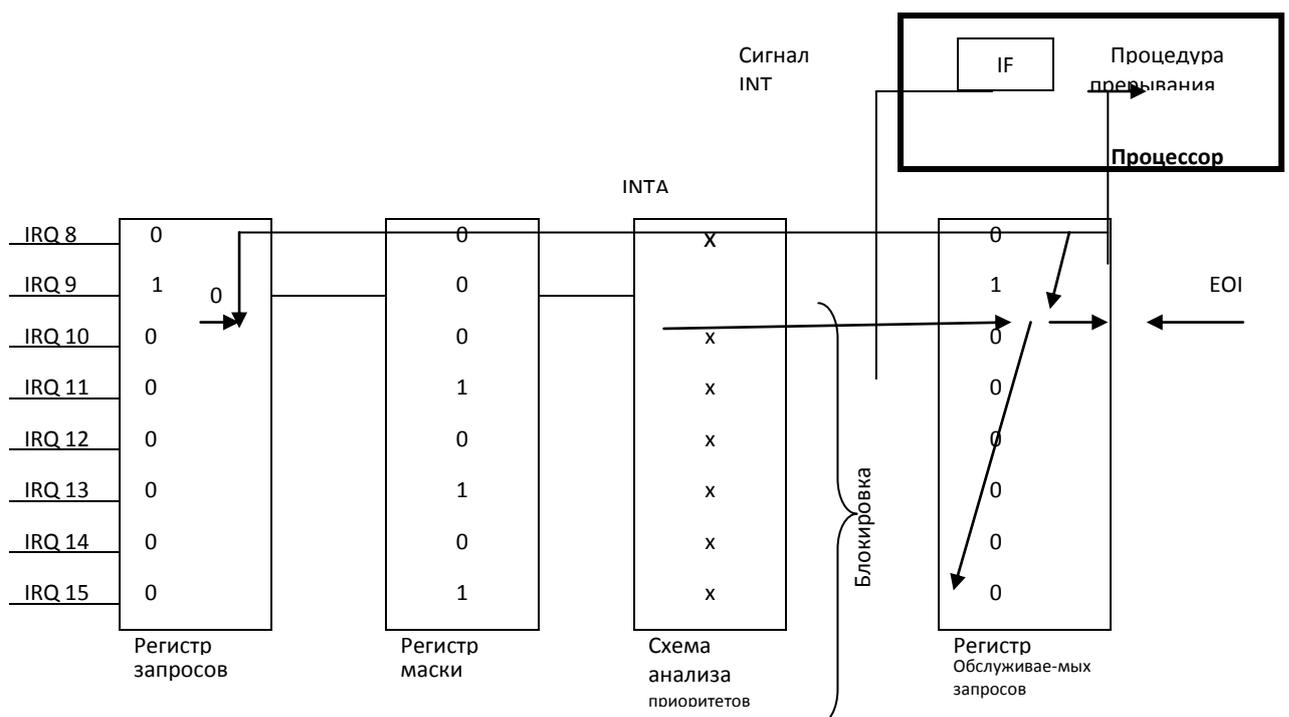
Необходимо быть знакомым с особенностями функционирования и программирования контроллера прерываний.

Контроллер прерываний может выполнять следующий набор операций:

- маскирование запросов на прерывание, то есть временное запрещение реакции на них;
- установка приоритетов запросов по различным входам, то есть разрешение конфликтов при одновременном приходе нескольких запросов на прерывание;
- работа в качестве основного контроллера (Master) или подчиненного (Slave).

Логически в контроллере можно выделить 4 основных узла :

- Регистр входных запросов
- Регистр маски
- Схема приоритетов
- Регистр обслуживаемых запросов



Последовательность обработки внешнего аппаратного прерывания имеет следующий вид:

1. Сигнал запроса прерывания поступает на вход регистра запросов и устанавливает в 1 соответствующий бит.
2. Значение регистра маски, равное 0, разрешает прохождение сигнала на схему анализатора приоритетов. (значение равное 1 запрещает) .
3. Схема приоритетов формирует сигнал, поступающий на вход регистра обслуживаемых запросов.
4. Сигнал запроса прерывания поступает на вход регистра обслуживаемых запросов и дает разрешение на установку в 1 его бита (однако не устанавливает). Одновременно сигнал поступает на вход INT . Процессор регистрирует поступление сигнала, если установлен флаг разрешения прерываний IF в регистре флагов (сброс флага IF командой cli запрещает аппаратные прерывания)
5. Микропроцессор, получив сигнал INT, формирует выходной сигнал INTA, который устанавливает бит регистра обслуживаемых запросов, разрешенный сигналом запроса прерывания (см. п.4) и сбрасывает аналогичный бит регистра запросов. Одновременно процессор сбрасывает флаг IF в регистре флагов, запрещая все аппаратные прерывания .

Поскольку процессор, получив сигнал INT, сбрасывает IF , при входе в обработчик все прерывания оказываются запрещенными и программа не может быть прервана внешними сигналами. Командой sti устанавливают флаг IF и разрешают прохождение запросов прерывания в процессор . Однако все уровни прерываний , начиная с текущего, остаются заблокированными в контроллере. В результате работа обработчика может быть прервана только при поступлении запроса прерывания более высокого приоритета . Такая ситуация называется **вложенным прерыванием**.

Выполнение в обработчике команд , реализующих приказ **конца прерываний EOI**, снимает блокировку в контроллере , и начиная с этого момента запрос прерывания любого уровня прервет выполнение обработчика. Особенно неприятной может оказаться ситуация , когда обработчик прерывается сигналом запроса прерывания того же уровня. Это приводит к тому, что программа обработчика, не дойдя до конца , опять начинает выполняться с самого начала. Для избежания данной ситуации обработчик прерываний должен быть предусматривать повторную входимость.

Выводы

- Прерывание можно рассматривать как некоторое особое событие в системе, требующее моментальной реакции.
- Механизм прерываний отлично подходит для управления событиями в вычислительной системе.
- Система распознает прерывания и при их возникновении запускает процедуру, соответствующую номеру прерывания.
- Существуют зарезервированные для использования центральным процессором прерывания.

- Прерывания бывают аппаратными, программными и исключениями.
- Чтобы связать адрес обработчика прерывания с номером прерывания используется таблица векторов прерываний.