

## 6.2 Организация системы прерывания

Нажимая на клавиши клавиатуры, задумывались ли вы над тем, что в это время происходит в компьютере, как он узнает о том, что была нажата та или иная клавиша? Или, к примеру, каким образом ведется отсчет времени в компьютере? Ведь каждый сигнал от различных устройств компьютера, чтобы стать «осознанным», должен быть обработан какой-то программой.

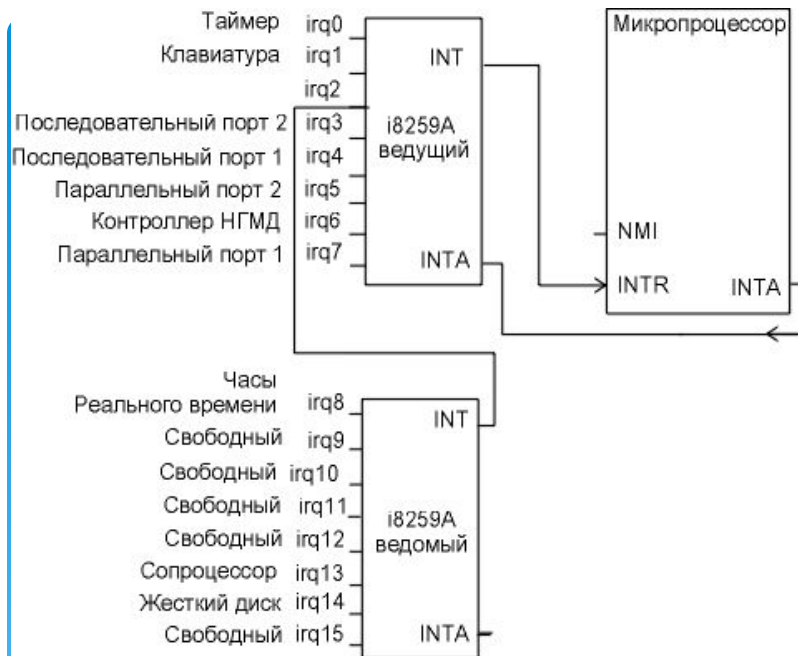
А выполнять программы может только микропроцессор. Но он один в компьютере, следовательно, в каждый конкретный момент времени микропроцессор может быть занят выполнением только одной программы. Как же ему узнавать о нажатиях клавиш и других сигналах, постоянно возникающих во время работы компьютера, 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.

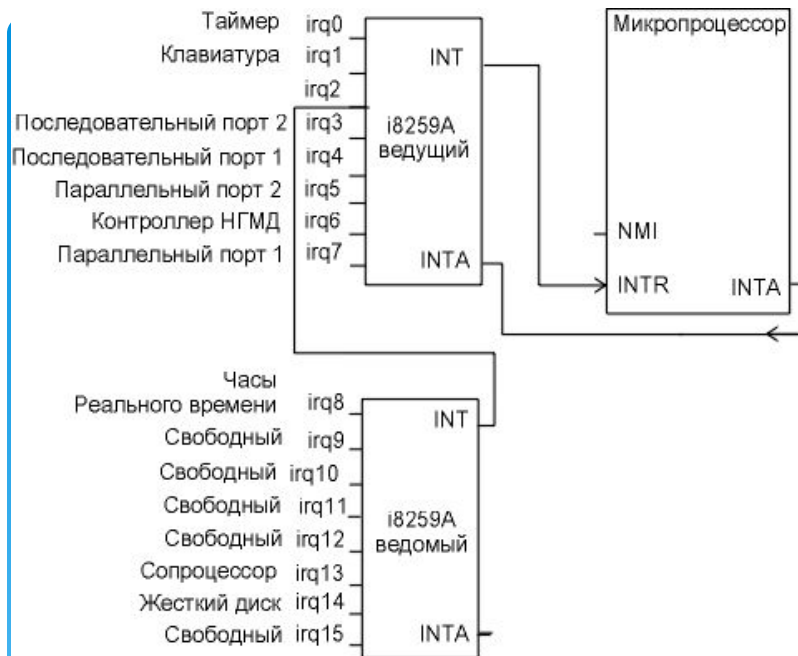
Исходя из вышеприведенных рассуждений, можно сказать, что прерывания могут быть внешними и внутренними.

Внешние прерывания вызываются внешними по отношению к микропроцессору событиями.



На рис. 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 (INTerrupt 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 непланируемым 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:

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:

\* выводы микропроцессора:

о INTR - вывод для входного сигнала внешнего прерывания. На этот вход поступает выходной сигнал от микросхемы контроллера прерываний 8259А;

о INTA - вывод микропроцессора для выходного сигнала подтверждения получения сигнала прерывания микропроцессором. Этот выходной сигнал поступает на одноименный вход INTA микросхемы контроллера прерываний 8259А;

о NMI - вывод микропроцессора для входного сигнала немаскируемого прерывания;

\* 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;

\* внешние устройства: таймер, клавиатура, магнитные диски и т. д.

К программным средствам системы прерываний реального режима относятся:

\* таблица векторов прерываний. В этой таблице в определенном формате, который зависит от режима работы микропроцессора, содержатся указатели на процедуры обработки соответствующих прерываний;

\* следующие флаги в регистре флагов `flags\eflags`:

о 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;

о 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;

\* машинные команды микропроцессора: `int`, `into`, `iret`, `cli`, `sti`.



## Таблица векторов прерываний

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.

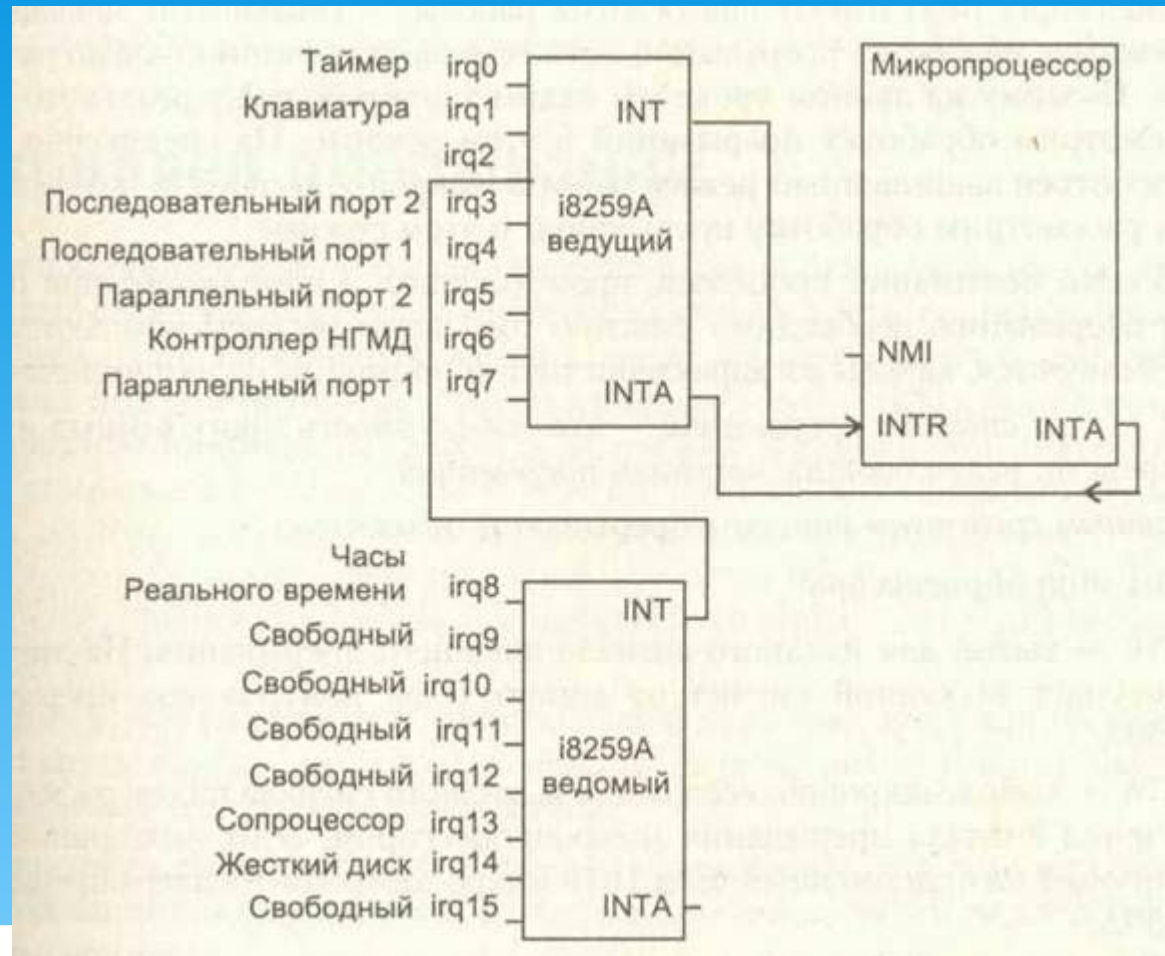
## Расскажем о назначении наиболее важных векторов прерываний.

Номер	Описание
0	Ошибка деления. Вызывается автоматически после выполнения команд DIV или IDIV, 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. При этом для процессора i8086 адрес возврата указывает на команду, следующую после команды деления, а для процессора i80286 и более старших моделей - на первый байт команды, вызвавшей прерывание
1	Прерывание пошагового режима. Вырабатывается после выполнения каждой машинной команды, если в слове флагов установлен бит пошаговой трассировки TF. Используется для отладки программ. Это прерывание не вырабатывается после пересылки данных в сегментные регистры командами MOV и POP
2	Аппаратное немаскируемое прерывание. Это прерывание может использоваться по-разному в разных машинах. Обычно оно вырабатывается при ошибке четности в оперативной памяти и при запросе прерывания от сопроцессора
3	Прерывание для трассировки. Генерируется при выполнении однобайтовой машинной команды с кодом CCh и обычно используется отладчиками для установки точки прерывания
4	Переполнение. Генерируется машинной командой INTO, если установлен флаг переполнения OF. Если флаг не установлен, команда INTO выполняется как NOP. Это прерывание используется для обработки ошибок при выполнении арифметических операций

5	Печать копии экрана. Генерируется, если пользователь нажал клавишу <PrtSc>. В программах MS-DOS обычно используется для печати образа экрана. Для процессора i80286 и более старших моделей генерируется при выполнении машинной команды BOUND, если проверяемое значение вышло за пределы заданного диапазона
6	Неопределенный код операции или длина команды больше 10 байт
7	Особый случай отсутствия арифметического сопроцессора
8	IRQ0 - прерывание интервального таймера, возникает 18,2 раза в секунду
9	IRQ1 - прерывание от клавиатуры. Генерируется, когда пользователь нажимает и отжимает клавиши. Используется для чтения данных из клавиатуры
A	IRQ2 - используется для каскадирования аппаратных прерываний
B	IRQ3 - прерывание асинхронного порта COM2
C	IRQ4 - прерывание асинхронного порта COM1
D	IRQ5 - прерывание от контроллера жесткого диска (только для компьютеров IBM PC/XT)
E	IRQ6 - прерывание генерируется контроллером НГМД после завершения операции ввода/вывода
F	IRQ7 - прерывание от параллельного адаптера. Генерируется, когда подключенный к адаптеру принтер готов к выполнению очередной операции. Обычно не используется
10	Обслуживание видеоадаптера
11	Определение конфигурации устройств в системе
12	Определение размера оперативной памяти
13	Обслуживание дисковой системы
14	Работа с асинхронным последовательным адаптером
80-85	Зарезервировано для BASIC
86-F0	Используются интерпретатором BASIC
F1-FF	Не используются

15	Расширенный сервис
16	Обслуживание клавиатуры
17	Обслуживание принтера
18	Запуск BASIC в ПЗУ, если он есть
19	Перезагрузка операционной системы
1A	Обслуживание часов
1B	Обработчик прерывания, возникающего, если пользователь нажал комбинацию клавиш <Ctrl+Break>
1C	Программное прерывание, вызывается 18,2 раза в секунду обработчиком аппаратного прерывания таймера
1D	Адрес видеотаблицы для контроллера видеоадаптера 6845
1E	Указатель на таблицу параметров дискеты
1F	Указатель на графическую таблицу для символов с кодами ASCII 128-255
20-5F	Используется MS-DOS или зарезервировано для MS-DOS
60-67	Прерывания, зарезервированные для программ пользователя
68-6F	Не используются
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
78 - 7F	Не используются

# Структура контроллера прерываний



## **Следующая специальная область памяти — это таблица векторов прерываний.**

Вообще, понятие прерывания довольно многозначно. Под прерыванием в общем случае понимается не только обслуживание запроса внешнего устройства, но и любое нарушение последовательной работы процессора.

Например, может быть предусмотрено прерывание по факту некорректного выполнения арифметической операции типа деления на ноль. Или же прерывание может быть программным, когда в программе используется команда перехода на какую-то подпрограмму, из которой затем последует возврат в основную программу.

В последнем случае общее с истинным прерыванием только то, как осуществляется переход на подпрограмму и возврат из нее.

В случае аппаратных прерываний номер прерывания или задается устройством, запросившим прерывание (при векторных прерываниях), или же задается номером линии запроса прерываний (при радиальных прерываниях).

Процессор, получив аппаратное прерывание, заканчивает выполнение текущей команды и обращается к памяти в область таблицы векторов прерываний, в ту ее строку, которая определяется номером запрошенного прерывания.

Затем процессор читает содержимое этой строки (код вектора прерывания) и переходит в адрес памяти, задаваемый этим вектором. Начиная с этого адреса в памяти должна располагаться программа обработки прерывания с данным номером.

В конце программы обработки прерываний обязательно должна располагаться команда выхода из прерывания, выполнив которую, процессор возвращается к выполнению прерванной основной программы.

Любое прерывание обрабатывается через таблицу векторов (указателей) прерываний. В этой таблице в простейшем случае находятся адреса начала прерывания, сохраняются в стеке программ обработки прерываний, которые и называются векторами. Длина таблицы может быть довольно большой (до нескольких сот элементов).



Обычно таблица векторов прерываний располагается в начале пространства памяти (в ячейках памяти с малыми адресами). Адрес каждого вектора (или адрес начального элемента каждого вектора) представляет собой номер прерывания.

Пусть, например, процессор выполнял основную программу и команду, находящуюся в адресе памяти 5000 (условно). В этот момент он получил запрос прерывания с номером (адресом вектора) 4. Процессор заканчивает выполнение команды из адреса 5000. Затем он сохраняет в стеке текущее значение счетчика команд (5001) и текущее значение PSW. После этого процессор читает из адреса 4



Таблицы векторов прерывания. Пусть этот код равен 6000. Процессор переходит в адрес памяти 6000 и приступает к выполнению программы обработки прерывания, начинающейся с этого адреса. Пусть эта программа заканчивается в адресе 6100. Дойдя до этого адреса, процессор возвращается к выполнению прерванной программы. Для этого он извлекает из стека значение адреса (5001), на котором его прервали, и бывшее в тот момент PSW. Затем процессор читает команду из адреса 5001 и дальше последовательно выполняет команды основной программы.



Прерывание в случае аварийной ситуации обрабатывается точно так же, только адрес вектора прерывания (номер строки в таблице векторов) жестко привязан к данному типу аварийной ситуации.

Программное прерывание тоже обслуживается через таблицу векторов прерываний, но номер прерывания указывается в составе команды, вызывающей прерывание.

Такая сложная, на первый взгляд, организация прерываний позволяет программисту легко менять программы обработки прерываний, располагать их в любой области памяти, делать их любого размера и любой сложности.

Во время выполнения программы обработки прерывания может поступить новый запрос на прерывание. В этом случае он обрабатывается точно так же, как описано, но основной программой считается прерванная программа обработки предыдущего прерывания. Это называется многократным вложением прерываний. Механизм стека позволяет без проблем обслуживать это многократное вложение, так как первым из стека извлекается тот код, который был сохранен последним, то есть возврат из обработки данного прерывания происходит в программу обработки предыдущего прерывания.

Отметим, что в более сложных случаях в таблице векторов прерываний могут находиться не адреса начала программ обработки прерываний, а так называемые дескрипторы (описатели) прерываний. Но конечным результатом

Во время выполнения программы обработки прерывания может поступить новый запрос на прерывание. В этом случае он обрабатывается точно так же, как описано, но основной программой считается прерванная программа обработки предыдущего прерывания. Это называется многократным вложением прерываний. Механизм стека позволяет без проблем обслуживать это многократное вложение, так как первым из стека извлекается тот код, который был сохранен последним, то есть возврат из обработки данного прерывания происходит в программу обработки предыдущего прерывания.

Отметим, что в более сложных случаях в таблице векторов прерываний могут находиться не адреса начала программ обработки прерываний, а так называемые дескрипторы (описатели) прерываний. Но конечным результатом обработки этого дескриптора все равно будет адрес начала программы обработки прерываний