Лабораторная работа № 5.

ИСПОЛЬЗОВАНИЕ АППАРАТНЫХ ПРЕРЫВАНИЙ

Цель работы – знакомство с различного вида аппаратными прерываниями и создание собственных подпрограмм обработки прерываний.

5.1. Общие положения

Микропроцессоры 8086/88 поддерживают механизм прерываний. В самом общем виде это наличие в аппаратуре специальных средств, с помощью которых выполнение текущей программы приостанавливается и процессор переходит к так называемой программе обслуживания прерывания (Interrupt Servise Routine - ISR). Механизм прерываний позволяет организовать выполнение тех или иных функций ядра и быструю реакцию процессора на возникновение каких-то внешних событий: ошибок в арифметических операциях, изменению состояния периферийных устройств и пр.

Микропроцессоры 8086/88 поддерживают 256 прерываний. Каждое из них имеет свой номер и ISR. Адрес точки входа в ISR называется вектором прерывания и хранится в специальной таблице, называемой таблицей векторов прерывания (ТВП). Код ISR может располагаться в любом месте памяти. Поэтому вектор прерывания занимает 4 байта: 2 байта отводится на значение сегментного регистра, устанавливаемое в CS (старшее слово), 2 байта - на значение смещения, устанавливаемое в IP (младшее слово). Вся ТВП занимает 256 * 4 = 1024 байт и располагается в оперативной памяти, начиная с адреса 0000:0000.

При возникновении прерывания процессор помешает в стек 6 байт: текущее значение CS, текущее значение IP (пара этих регистров определяет точку, с которой выполнение прерываемой программы возобновится), а также 2 байта флагов процессора. В CS и IP устанавливаются значения из ТВП, которые задают адрес начала ISR. Прерыванию 0 соответствует вектор прерывания по адресу 0000:0000, прерыванию 1 - по адресу 0000:0004h, прерыванию 2 - по адресу 0000:0008h и т.д.

Сама ISR - это программа, построенная с соблюдением специальных правил:

1)в самом начале она сохраняет в стеке все регистры процессора, которые будут использоваться в этой программе;

2) перед завершением работы программы значения регистров восстанавливаются;

3) последней инструкцией ISR, как правило, является инструкция возврата из прерывания IRET. Выполняя IRET, процессор извлекает из стека шесть слов информации, которые последовательно помещает в регистры IP, CS и регистр флагов, возвращаясь к исполнению прерванной программы.

Часто обработчикам программных прерываний требуется передать какие-то значения, задающие конкретное действие, характеристики ситуации и т.п., и получить какие-то результаты по завершению исполнения ISR. Для такого обмена данными используются внутренние регистры процессора.

Некоторые векторы прерывания в ТВП на самом деле задают не точки входа в ISR, а используются для хранения важной системной информации: адресов данных и таблиц. Кроме того, за некоторые векторы "зацеплены" ISR, не выполняющие никаких действий. Они служат заглушками для подключения дополнительных обработчиков. Так, например, в нормальном состоянии обработчик прерывания 1Ch не выполняет никаких действий и содержит единственную инструкцию возврата из прерывания IRET. Прерывание 1Ch вызывается из пределов ISR таймера (обработчик прерывания 8). Прерывание от таймера, в свою очередь, генерируется 18.2 раза в секунду аппаратурой системного таймера. Есть и другие обработчики - заглушки, вызываемые при функционировании ISR BIOS и MS-DOS.

5.2. Аппаратные прерывания

В процессе функционирования персонального компьютера могут встретиться четыре типа прерываний:

1) аппаратные;
2) программные;
3) исключительные ситуации процессора (processor exceptions);
4) немаскируемые.

Аппаратные прерывания возникают как результат некоторых внешних событий и в их генерации принимает участие специальная микросхема персонального компьютера - программируемый контроллер прерываний, или PIC (Programmable Interrupt Controller). Наиболее часто для этих целей используется одна или несколько микросхем 8259A либо их функциональные эквиваленты. В архитектуре компьютеры IBM PC AT используют PIC, построенный на двух микросхемах 8259A (рис. 5.1).

Микросхема 8259A рассчитана на 8 входов запросов прерываний, обозначаемых IRQ (Interrupt Request). Сигналы на них возбуждают внешние устройства: адаптеры асинхронной последовательной и параллельной связи, плата системного таймера и др. Контроллер прерываний имеет в своем составе ряд программируемых внутренних регистров, определяющих особенности обработки запросов прерываний.

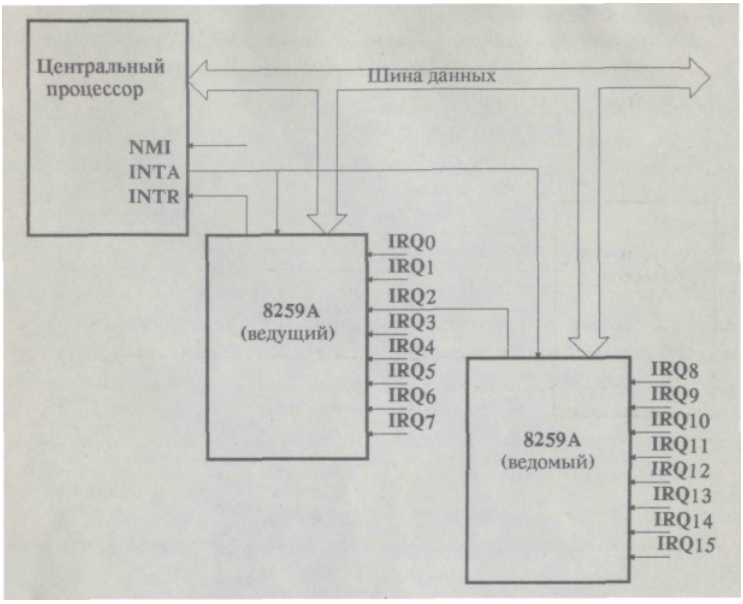

Рис 5.1. Двухкаскадная схема построения контроллера прерываний

Выход ведущей (единственной в однокаскадной схеме) микросхемы 8259A контроллера прерываний подается на специальный вход процессора (INTR). Этот вход процессора является маскируемым: если флаг маскирования прерываний IF равен единице, процессор способен "ощущать" изменение состояния линии INTR (прерывания разрешены); если же IF сброшен в 0, изменения на линии INTR не влияют на работу центрального процессора. Поэтому часто аппаратные прерывания, в формировании которых принимает участие PIC, называют маскируемыми. Если прерывания разрешены и устанавливается высокий потенциал на линии INTR, процессор завершает исполнение текущей инструкции и отвечает двумя циклами сигнала INTA.

Первый цикл сигнала INTA - это, по существу, пустой цикл, который готовит PIC к следующему циклу. Во время второго цикла PIC помещает на шину данных байт, задающий номер аппаратного прерывания. Получив байт номера прерывания, процессор умножает его на 4, формируя смещения до вектора прерываний в ТВП.

Процессор сохраняет в стеке текущее значение регистров флагов CS и IP, затем устанавливает в 0 флаг IF, а в CS и IP - значения из вектора прерывания. В результате управление передается в ISR.

Для того чтобы различать сигналы прерываний от различных внешних устройств, система прерываний IBM PC построена следующим образом. Каждое внешнее устройство подключено к собственной линии запроса прерываний IRQ. При получении сигнала на линии IRQ контроллер прерываний передает в процессор уникальный для данной IRQ байт номера прерывания. Соответствие линий

IRQ и номеров прерывания задается программированием контроллера прерываний. Такое программирование выполняется в ходе начальной загрузки системы специальной процедурой BlOSa и в дальнейшем обычно не изменяется. В принципе, перепрограммирование PIC может выполняться в любой момент и некоторые программы (Windows, OS/2) используют это при своей загрузке. В ходе программирования PIC задаются старшие 5 бит номера прерывания, а младшие 3 бита генерирует микросхема 8259A, определяя двоичный код номера линии IRQ. Ведущая (единственная) микросхема программируется BIOSом так, чтобы передавать в процессор прерывания от 08h до 0Fh. Ведомая 8259A в IBM PC AT настраивается на передачу номеров прерываний от 70h до 77h.

Кроме отображения IRQ на номера прерывания, PIC выполняет упорядочивание по приоритету одновременно возникающих запросов. Обычно наивысший приоритет имеет запрос на линии IRQ0, затем в порядке убывания IRQI, IRQ2, ..., IRQ7. Вход процессора INTR является так называемым "уровнем чувствительным". Это значит, что если процессор ощущает высокий уровень, он всегда начинает цикл обработки прерывания. Если начатая ISR устанавливает IF в единицу (а это, как правило, так и бывает), сохранение сигнала на линии INTR вызовет повторное вхождение в ту же самую ISR, а затем вхождение в третий, четвертый и далее раз до тех пор, пока не переполнится стек. Для того чтобы этого не происходило, контроллер прерываний блокирует генерацию сигнала INTR для текущей активной линии IRQ до тех пор, пока исполняемая ISR не даст явного указания сделать это. Обычно так ISR обозначают свое завершение, посылая в PIC команду завершения прерывания, или EOI (End Of Interrupt). Если ISR не сделает этого, контроллер продолжает блокировать выработку сигнала INTR для всех последующих запросов прерывания как по данной линии, так и по другим, менее приоритетным линиям.

Любая из линий запросов IRQi может быть маскирована. Специальный внутренний регистр PIC хранит битовую маску входов IRQi: бит 0 регистра маски управляет IRQ0 (IRQ8 в ведомой микросхеме 8259A), бит 1 - IRQI (IRQ9), ..., бит 7 - IRQ7 (IRQ15). Если бит равен нулю контроллер генерирует сигнал на линии INTR, если бит равен единице, контроллер не "чувствует" запрос на маскированной битом линии IRQi.

Использование двухкаскадной схемы для построения контроллера прерываний расширяет до 15 чисто обслуживаемых внешних устройств. Для двухкаскадной схемы выход INTR ведомой микросхемы 8259A подается на линию LRQ2 ведущей микросхемы. В результате линии запросов упорядочиваются по приоритету следующим образом: максимальный приоритет имеет IRQ0, затем в порядке

убывания IRQI, IRQ8, ..., IRQ15, IRQ3, ..., IRQ7. Как правило, PIC в ходе начальной загрузки настраивается так, что для линий IRQ0 - IRQ7 генерируются прерывания с номерами 08h - 0Fh соответственно, а для линий IRQ8 - IRQ15 - прерывания с номерами 70h - 77h. Подключение внешних устройств персональных компьютеров к линиям IRQ и, следовательно, закрепление аппаратных прерываний для большинства персональных компьютеров типа IBM PC фактически стандартизовано. В табл. 5.1 приводится закрепление внешних устройств и аппаратных прерываний для IBM PC AT.

Табл.5.1. Использование прерываний в IBM PC AT

Линия запроса	Номер прерывания	Обычное использование
IRQO	8h	Системный таймер
IRQ1	9h	Клавиатура
IRQ2	0Ah	Переадресация от ведомой 8259A
IRQ3	0Bh	COM2 (или COM4)
IRQ4	0Ch	COM1 (или COM3)
IRQ5	0Dh	LPT2
IRQ6	0Eh	Контроллер накопителей на гибком диске
IRQ7	0Fh	LPT1
IRQ8	70h	Таймер реального времени
IRQ9	71h	Прерывание обратного хода луча EGA- и VGA-
IRQ 10	72h	Свободно
IRQ11	73h	Свободно
IRQ12	74h	Свободно
IRQ13	75h	Сопроцессор математики с плавающей точкой
Линия запроса	Номер прерывания	Обычное использование
IRQ 14	76h	Контроллер накопителя на жестком диске
IRQ 15	77h	Свободно

5.3. Немаскируемые прерывания

Процессор, кроме входа INTR, использует еще один вход -вход немаскируемого прерывания, или NMI (NonMaskable Interrupt). Название входа говорит о том, что программное обеспечение не может блокировать восприятие сигнала. Когда на входе NMI появляется сигнал, процессор без помощи PIC генерирует байт номера прерывания, равный двум. В отличие от входа INTR, NMI является "чувствительным к фронту сигнала" (edge sensitive). Генерацию прерывания 2 вы-

зывает изменение состояния линии: с логического нуля на логическую единицу. После того, как прерывание сгенерировано, высокий потенциал линии не способен вызвать очередную генерацию прерываний. Только возврат сигнала в нуль, а затем - в единицу заставит процессор генерировать очередное немаскируемое прерывание.

Сигнал на входе NMI имеет более высокий приоритет, чем INTR, и используется для организации реакции процессора на критические для системы ситуации: обнаружение ошибки четности в данных, хранимых в памяти, выключение питания и т.п.

5.4. Программные прерывания

Когда в программе встречается инструкция INT, процессор исполняет действия, рассмотренные ранее для аппаратного прерывания. Отличие состоит в том, что байт номера прерывания задается самой инструкцией. В этой связи не требуется выполнение циклов INTA. Инструкция INT имеет более высокий приоритет, чем аппаратные и немаскируемые прерывания: если процессор начинает исполнение инструкции INT, он не прерывается сигналами на линиях NMI и INTR. Многие из программных прерываний используются для доступа к ISR BIOSa, операционной системы или инсталлируемых драйверов. Кратко правила взаимодействия с ISR (номер прерывания, описание функции, значения регистров на входе в ISR и после ее завершения, индикация ошибок и т.п.) называют интерфейсом прикладной программы или API (Application Program Interface).

5.5. Исключительные ситуации

Исключительные ситуации - это генерация внутренних прерываний процессором при возникновении необычных условий во время исполнения машинных инструкций. Примером таких ситуаций для микропроцессора Intel 8086/88 является "деление на нуль" (генерируется прерывание 0) и "пошаговое исполнение" (генерация прерывания 1 после завершения текущей инструкции). Число исключительных ситуаций, генерируемых процессорами 80286 и 80386, значительно больше. Для них используются прерывания с номерами 05h и больше (например, для 80386 от 05h до 10h включительно). Многие из этих исключительных ситуаций могут генерироваться только при переключении в защищенный режим работы и связаны с нарушением защиты памяти. Для того чтобы избежать "столкновения" прерываний с одинаковыми номерами, закрепленных за аппаратными прерываниями и исключительными ситуациями защищенного режима, операционная система может выполнить перепрограммирование контроллера прерываний.

5.6. Базовая система ввода-вывода BIOS. Прерывания BIOS. Области данных и таблицы BIOS

Первые 20 прерываний с номерами от 00H до 1Fh закреплены за прерываниями, генерируемыми аппаратными средствами, либо предназначенными для управления аппаратурой персонального компьютера. ISR этих прерываний вместе с некоторыми данными образуют так называемую базовую систему ввода-вывода или BIOS (Base Input-Output System). Все ISR и данные BIOSa записаны в ПЗУ. ISR, входящие в BIOS, представляют собой самый нижний уровень иерархической структуры программного обеспечения (ПО) управления аппаратными средствами компьютера. Они взаимодействуют с аппаратурой на уровне физических сигналов, портов, заданных адресов и в этой связи являются немобильной частью ПО. При появлении новых аппаратных средств приходится перерабатывать BIOS. Поэтому принято различать версии BIOS по дате разработки. Кроме того, для облегчения дополнений BIOSa новые периферийные устройства снабжаются своей секцией ПЗУ, а основной блок BIOS, при загрузке системы проверяет наличие дополнительных секций и "переключает" на них соответствующие прерывания.

Важной особенностью BIOSa является стандартный интерфейс с программой практически для всех персональных компьютеров на базе микропроцессоров семейства Intel. Другими словами, BIOS выполняет роль "экрана" между программами (в частности, программами MS-DOS) и большим разнообразием конкретных аппаратных средств. Например, для вывода символа на экран дисплея независимо от типа дисплея и используемого адаптера необходимо выполнить инструкцию INT 10h с теми же самыми значениями во внутренних регистрах. Все детали интерфейса программы с ВЮ5ом описываются в техническом справочнике BIOS.

При выполнении ISR BIOS для хранения данных используется зарезервированная область памяти, называемая областью данных BIOSa. Она начинается с адреса 40:00h и занимает 256 байт до адреса 40:FFh. Здесь располагается ряд таблиц, копируемых из ПЗУ при начальной загрузке системы и уточняемых по результатам тестирования узлов компьютера. При выполнении функций BIOS многие параметры изменяются. Например, корректируется адрес позиции курсора на экране, номер установленного режима адаптера дисплея и т.п. Другими словами, таблицы в области данных BIOSa отражают текущие параметры и состояние аппаратных средств компьютера.

5.7. Функции библиотеки C++ для доступа к обработчикам прерывания

Библиотечные функции C++, как правило, в конечном итоге обращаются к

ISR BIOS или MS-DOS. В тех случаях, когда необходимо непосредственное обращение к BIOS или MS-DOS, используются специальные функции, описываемые далее.

```
int int86(int intno, union REGS *inregs, union REGS *outregs)
```

Функция загружает внутренние регистры микропроцессора значениями, записанными в объединении по шаблону union REGS, на начало которого указывает inregs, и выполняет прерывание с номером intno. Значения внутренних регистров на выходе из прерывания записываются в объединении по шаблону union REGS, на начало которого указывает outregs. Описание объединений выполняет точка вызова функции. Шаблон union REGS описан в заголовочном файле <dos.h> и представляет собой объединение двух структур:

```
struct WORDREGS
 {
    unsigned int ax, bx, ex, dx, si, di, cflag, flags;
    };

struct BYTEREGS
  {
    unsigned char al, ah, bl, bh, cl, ch, dl, dh;
  };

union REGS
 {
   struct WORDREGS x;
   struct BYTEREGS h;
 };
```

Структура WORDREGS используется для доступа к регистрам как двухбайтовым единицам. Структура BYTEREGS позволяет осуществлять доступ к отдельным байтам РОН. Поле структуры flags позволяет перед вызовом задать, а после вызова прочесть значение регистра флагов. Так как многие функции MS-DOS используют флаг переноса для сигнализации об ошибках в программе-обработчике прерывания, в структуре WORDREGS специально выделено поле cflag для значения флага переноса.

Все функции int...() возвращают значение регистра AX на выходе из ISR. Недостатком функции int86() является возможность доступа лишь к ограниченному числу регистров. При выполнении некоторых функций MS-DOS значения задают-

ся и в сегментных регистрах. В таких (правда, достаточно редких) случаях следует использовать более общую функцию int86x():

```
int int86x(int intno, union REGS *inregs, union REGS *outregs, struct SREGS *segregs)
```

В отличие от int86() перед выполнением прерывания intno дополнительно устанавливаются сегментные регистры из структурной переменной по шаблону SREGS. В функцию передается указатель на эту структурную переменную. По возвращении из ISR в структурную переменную по шаблону SREGS дополнительно копируются значения всех сегментных регистров. Если необходимо выполнить обращение к функции MS-DOS (т. е. прерывание 21h с заданным значением AH), можно использовать функцию intdos(), всегда обращающуюся к прерыванию 21h.

```
int intdos(union REGS *inregs, union REGS *outregs)
```

В отличие от ранее рассмотренных функций данной функции не передается номер генерируемого прерывания, так как всегда генерируется прерывание 21h.

5.8. Предварительная подготовка к работе

1. Ознакомиться с аппаратными средствами системы прерывания.
2. Ознакомиться с программными средствами системы прерывания.

5.9. Порядок выполнения работы

Реализовать программу, которая позволяет вызывать звук системного динамика нажатием клавиши F1. Использовать собственную подпрограмму обработки прерывания в цепочке с подпрограммой обработки прерывания от клавиатуры

5.10. Содержание отчета

Отчет по лабораторной работе должен содержать:

- титульный лист;
- задание на лабораторную работу;
- блок-схему алгоритма с пояснениями;
-текст программы;
- примеры запуска программы;
- структурная схема аппаратных средств, используемых при выполнении