Подсистема прерываний

Intel 8259 PIC

- 8 линий прерываний (IRQ0-IRQ7)
- Каскадный режим (Slave подключается к IRQ2 Master’a)
- Каждому прерыванию назначен приоритет

Intel 8259 PIC

Номер	Приоритет	Устройство
IRQ0	0	Таймер
IRQ1	1	Клавиатура
IRQ2		Каскадное подключение второго контроллера
IRQ3	10	Последовательный порт COM2
IRQ4	11	Последовательный порт COM1
IRQ5	12	Параллельный порт LPT2
IRQ6	13	Контроллер Floppy-дисководов
IRQ7	14	Параллельный порт LPT1

Intel 8259 PIC

Номер	Приоритет	Устройство
IRQ8	2	Часы реального времени
IRQ9	3	Свободен
IRQ10	4	Контроллер видеоадаптера
IRQ11	5	Свободен
IRQ12	6	Мышь PS/2
IRQ13	7	Математический сопроцессор
IRQ14	8	Первый контроллер жесткого диска
IRQ15	9	Второй контроллер жесткого диска

Порты

- Первый контроллер: 0x20h (команды) и 0x21h (данные)
- Второй контроллер: 0xA0h (команды) и 0xA1h (данные)
- В datasheet признак A0 различает порт команд (A0 = 0) и порт данных (A0 = 1)

Команды

- ICW - Initialization command word, ICW1-ICW4 (ICW1-ICW3) отправляются последовательно
- ICW1 отправляется в командный порт (A0 = 0), затем ICW2-ICW4 - в порт данных (A0 = 1)
- OCW - Operation command word
- OCW1 отправляется в порт данных (A0 = 1), затем OCW2 и OCW3 - в командный порт (A0 = 0)

ICW1

Бит	7	6	5	4	3	2	1	0
Описание	0	0	0	1	Триггер	Размер	Режим	ICW4
Значение	0	0	0	1	0	0	0	1

- Начальная команда
- Бит 0 - необходимость ICW4 (1 - команда будет вызвана)
- Бит 1 - использование ведомого контроллера (0 - используется)
- Бит 2 - размер вектора прерывания (1 - 4 байта, 0 - 8 байт)
- Бит 3 - режим срабатывания триггера (1 - фронт, 0 - уровень)

ICW2

Бит	7	6	5	4	3	2	1	0
Описание	T7	T6	T5	T4	T3	0	0	0
Значение	0/1	0/1	0/1	0/1	1/0	0	0	0

- Установка смещения набора векторов прерываний. Для ведущего по умолчанию - 0x08h (IRQ0 - 0x08h, IRQ7 - 0x0Fh), для ведомого по умолчанию - 0x70h (IRQ8 - 0x70h, IRQ15 - 0x77h).

- Смещение должно быть кратно 8, так как последние три бита добавляет контроллер в качестве номера прерывания

ICW3 - ведущий

Бит	7	6	5	4	3	2	1	0
Описание	S - IRQ7	S - IRQ6	S - IRQ5	S - IRQ4	S - IRQ3	S - IRQ2	S - IRQ1	S - IRQ0
Значение	0	0	0	0	0	1	0	0

- Управление каскадностью
- Для ведущего нужно установить 1 на той линии, к которой подключен ведомый (эта линия - IRQ2)

ICW3 - ведомый

Бит	7	6	5	4	3	2	1	0
Описание	0	0	0	0	0	MIRQ2	MIRQ1	MIRQ0
Значение	0	0	0	0	0	0	1	0

- Управление каскадностью
- Для ведомого используются только биты 0-2. Указывается номер входа ведущего контроллера, к которому подключен ведомый (2-й вход)

ICW4

Бит	7	6	5	4	3	2	1	0
Описание	0	0	0	SFNM	BUF	M/S	AEOI	CPU
Значение	0	0	0	0	0	0	0	1

- Бит 0 - тип процессора, 8086 - 1, 8085 - 0
- Бит 1 - режим автоматического окончания прерывания (1 - включен)
- Бит 2 - определяет ведущего (1) или ведомого (0) контроллера при включенном буферизированном режиме
- Бит 3 - определяет, включен ли буферизированный режим (1 - включен)
- Бит 4 - определяет, включен ли специальный вложенный режим

OCW1

Бит	7	6	5	4	3	2	1	0
Описание	M7	M6	M5	M4	M3	M2	M1	M0

- M0-M7 - биты маски прерываний
- Если бит равен 0, прерывание разрешено, иначе - запрещено (замаскировано)
- После инициализации маска сбрасывается, поэтому может появиться необходимость запомнить маску и выставить ее заново после инициализации

OCW2

Бит	7	6	5	4	3	2	1	0
Описание	R	SL	EOI	0	0	L2	L1	L0
Значение	0	0	1	0	0	0	0	0

- Бит 5 - End of Interrupt, отправляется 1, когда обработка прерывания завершена

- Если прерывание пришло из ведущего контроллера, то EOI отправляется только в ведущий, иначе в оба

OCW3

Бит	7	6	5	4	3	2	1	0
Описание	0	ESMM	SMM	0	1	P	RR	RIS
Значение	0	0	0	0	1	0	1	0/1

- Бит 1 - считывание регистров (1 - считать)
- Бит 0 - выбор считываемого регистра (0 - IRR, 1 - ISR)
- Значение считывается из командного порта

Регистры ISR и IRR

- IRR - interrupt request register, хранит запросы на прерывание

- ISR - interrupt service register, хранит обслуживаемые в данный момент прерывания

Дополнительная информация для выполнения лабораторной работы

Получение и установка векторов функций обработки прерываний в DOS

- Функция `getvect(int intr_num)` считывает значение вектора со смещением `intr_num`, которое является адресом функции-обработчика прерывания

- Функция `void setvect(int intr_num, void interrupt(*isr)())` устанавливает адрес функции-обработчика для вектора прерывания со смещением `intr_num`

Взаимодействие с портами

- Функция `inp(int port_address)` возвращает байт, считанный из порта `port_address`
- Функция `outp(int port_address, int value)` записывает младший байт `value` в порт `port_address`

Запись в видеопамять

- В DOS есть возможность записать символы напрямую в видеопамять
- Один из способов - записать последовательность символов, начиная с адреса 0xb800:0000, где первый байт - это ASCII-кол символа, а второй - цвет символа и фона

Пример записи в видеопамять

- Следующий код записывает символ ‘0’ в левый верхний угол экрана

```
struct VIDEO
{
 unsigned char symbol;
 unsigned char attribute;
};

// ...

VIDEO far* screen = (VIDEO far *)MK_FP(0xB800, 0);
screen->symbol='0';
screen->attribute=0x5E;
```

Резидентный режим

- В DOS есть возможность сделать программу резидентной, то есть передать управление ОС, но остаться в оперативной памяти (TSR - Terminate and Stay Resident)

- При этом программа может реагировать на прерывания

Резидентный режим

- В реализации резидентного режима много нюансов (определение размера резидентной части, перезапускаемость, закрываемость и т.д.), поэтому в целях упрощения можно воспользоваться следующим кодом, который не предусматривает их все

```
FP_SEG (fp) = _psp;
FP_OFF (fp) = 0x2c;
_dos_freemem(*fp);

_dos_keep(0,(_DS -_CS)+(_SP/16)+1);
```

Ссылки

- https://wiki.osdev.org/8259_PIC
- https://habr.com/en/post/446312/
- В. Несвижский "Программирование аппаратных средств в Windows", с. 495, с. 122
- http://www.shikadi.net/moddingwiki/B800_Text