

# *АРХИТЕКТУРА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ*

Лабораторная работа №3

## **ПОДСИСТЕМА ПРЕРЫВАНИЙ**

К теме:

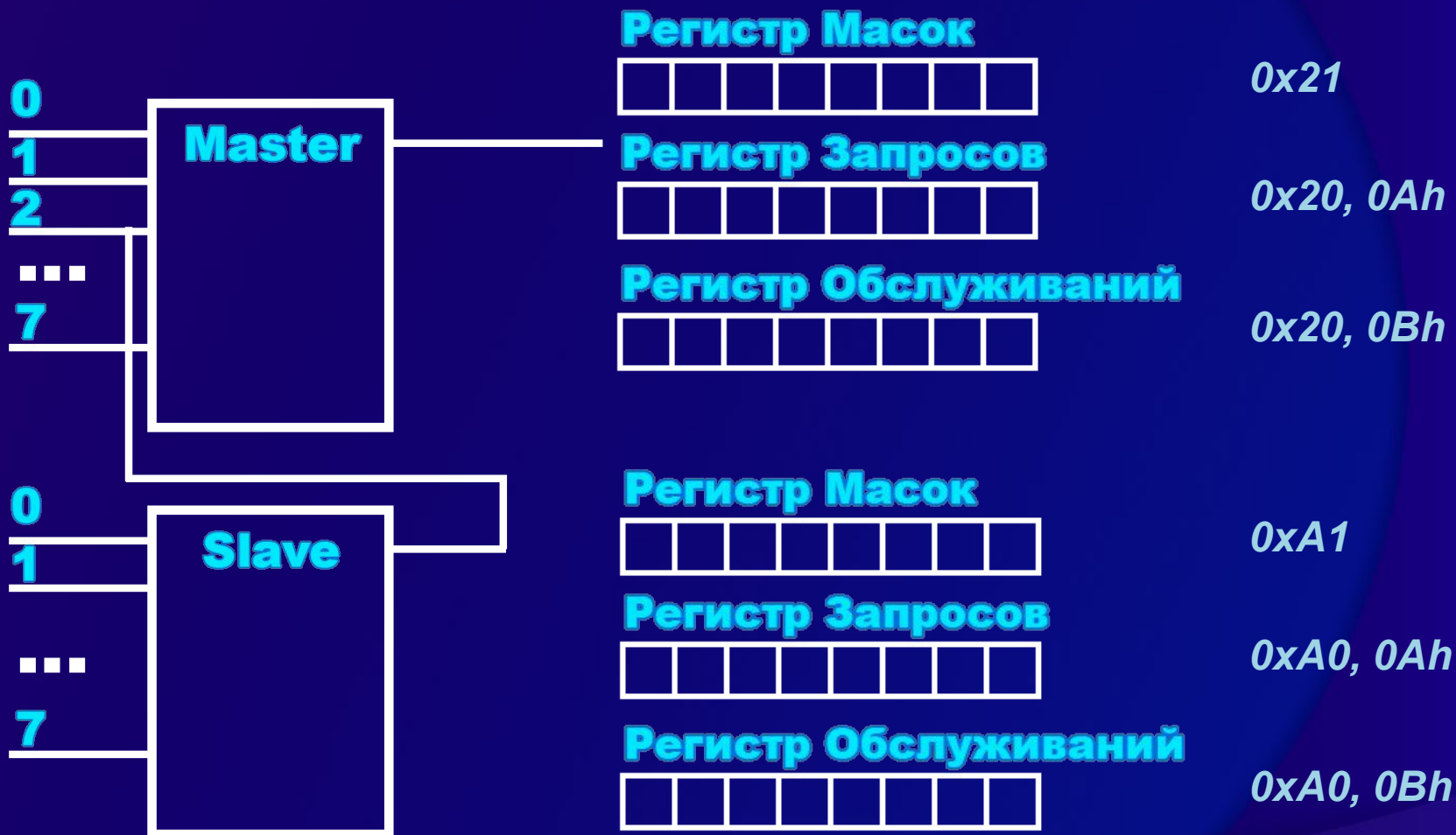
Организация подсистемы прерываний.  
Контроллер прерываний.

## **ЦЕЛЬ РАБОТЫ**

Изучить организацию прерываний в IBM PC с использованием контроллера прерываний.

Продолжительность работы - 4 часа.

# Контроллер прерываний



# Задание

Под MS DOS написать программу, которая:

- 1) выполняет инициализацию контроллера прерываний;

# Задание

Под MS DOS написать программу, которая:

- ⦿ 1) выполняет инициализацию контроллера прерываний;
- ⦿ 2) выводит на экран содержимое регистров запросов, обслуживания и масок для ведущего и ведомого контроллеров (через видеобuffer).

# Задание

Под MS DOS написать программу, которая:

- ◎ 1) выполняет инициализацию контроллера прерываний;
- ◎ 2) выводит на экран содержимое регистров запросов, обслуживания и масок для ведущего и ведомого контроллеров (через видеобuffer).
- ◎ \* При нажатии на клавиши меняется цвет или фон выводимой информации.

*Программа должна быть резидентной. Все векторы прерываний переопределяются, новый базовый адрес выбирается в соответствии с вариантом.*

# Пример (test3.cpp)

```
int main()
{
    unsigned far *fp; //объявляем указатель
    init();

    FP_SEG (fp) = _psp; // получаем сегмент
    FP_OFF (fp) = 0x2c; // и смещение сегмента данных
                        //с переменными среды,
    _dos_freemem(*fp); //чтобы освободить память

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

    return 0;
}
```

# Пример (test3.cpp)

```
void init()
{
    intOld = getvect(0x__) ; //старые обработчики 08-0F, 70-77
    ...

    setvect(0x__, intNew) ; // свои обработчики
    ...

    _disable() ; // запретить прерывания (cli)

    outp(0x20, 0x11) ; // ICW1
    outp(0x21, 0x70) ; // ICW2
    outp(0x21, 0x04) ; // ICW3
    outp(0x21, 0x01) ; // ICW4

    ...
    _enable() ; // разрешить прерывания (sti)
}
```



# Пример (test3.cpp)

```
void interrupt  intNew(...) { get_reg(); intOld(); }

void get_reg() // получить состояние регистров
{
    outp(0x20, 0x0B); //PO для ведущего
    print(inp(0x20));
    ...
}

void print(int val) // распечатать состояние регистров
{
    int i;
    VIDEO far* screen = (VIDEO far *)MK_FP(0xB800,
0);

    for(i=7;i>=0;i--) //поделить по битам
    {
        screen->symb= val%2 + '0'; //заполнить экран
        screen->attr=0x5E;
        screen++;
        val/=2;
    }
}
```