

# 中断

## 一、实验目的

- 1、掌握 PC 机中断处理系统的基本原理。
- 2、学会编写中断服务程序。

## 二、实验原理与内容

### 1、实验原理

PC 机用户可使用的硬件中断只有可屏蔽中断，由 8259 中断控制器管理。中断控制器用于接收外部的中断请求信号，经过优先级判别等处理后向 CPU 发出可屏蔽中断请求。IBMPC、PC/XT 机内有一片 8259 中断控制器对外可以提供 8 个中断源：

中断源	中断类型号	中断功能
IRQ0	08H	时钟
IRQ1	09H	键盘
IRQ2	0AH	保留
IRQ3	0BH	串行口 2
IRQ4	0CH	串行口 1
IRQ5	0DH	硬盘
IRQ6	0EH	软盘
IRQ7	0FH	并行打印机

8 个中断源的中断请求信号线 IRQ0~IRQ7 在主机 62 线 ISA 总线插座中可以引出，系统已设定中断请求信号为“边沿触发”，普通结束方式。对于 PC/AT 及 286 以上微机内又扩展了一片 8259 中断控制，IRQ2 用于两片 8259 之间级连，对外可以提供 16 个中断源：

中断源	中断类型号	中断功能
IRQ8	070H	实时时钟
IRQ9	071H	用户中断
IRQ10	072H	保留
IRQ11	073H	保留
IRQ12	074H	保留
IRQ13	075H	协处理器
IRQ14	076H	硬盘
IRQ15	077H	保留

TPC-USB 实验板上，固定的接到了 3 号中断 IRQ3 上，即进行中断实验时，所用中断型号为 0BH。

2、实验内容

实验电路如图 1，直接用手动产单脉冲作为中断请求信号(只需连接一根导线)。要求每按一次开关产生一次中断，在屏幕上显示一次“TPCA Interrupt!”，中断 10 次后程序退出。

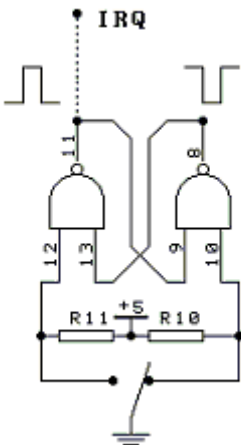
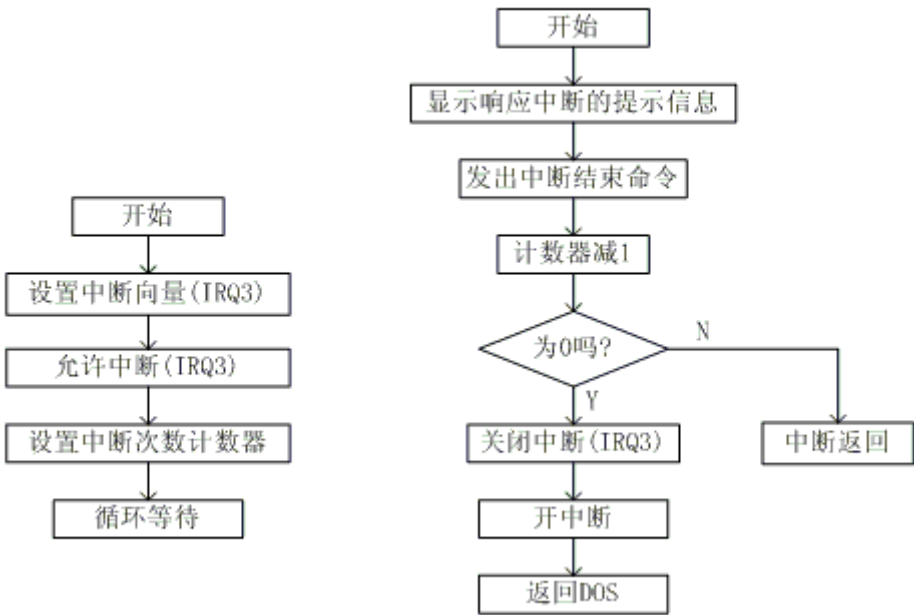


图 1

流程图



### 三、参考程序：

data segment

    mess db 'TPCA interrupt!','0dh,0ah','\$'

data ends

code segment

assume cs:code, ds:data

start:

**mov ax, cs**

**mov ds, ax**

**mov dx, offset int3**

    mov ax, 250bh

    int 21h ;设置 IRQ3 的中断矢量

**in al, 21h** ;读中断屏蔽寄存器

    and al, 0f7h ;开放 IRQ3 中断

    out 21h, al

    mov cx, 10 ;开放 IRQ3 中断

    sti ;置中断标志位

L1: jmp L1

int3: ;中断服务程序

**mov ax, data**

**mov ds, ax**

**mov dx, offset mess**

    mov ah, 09

    int 21h ;显示每次中断的提示信息

    mov al, 20h

    out 20h, al ;发出 EOI 结束中断

**loop next**

**in al, 21h** ;读中断屏蔽寄存器

**or al, 08h** ;关闭 IRQ3 中断

    out 21h, al

    sti ;置中断标志位

    mov ah, 4ch

    int 21h

**next: iret**

code ends

end start

# 交通灯控制实验

## 一、实验目的

通过并行接口 8255 实现十字路口交通灯的模拟控制,进一步掌握对并行口的使用。

## 二、实验内容

如下图 1 所示, L7、L6、L5 作为南北路口的交通灯与 PC7、PC6、PC5 相连, L2、L1、L0 作为东西路口的交通灯与 PC2、PC1、PC0 相连。编程使六个灯按交通灯变化规律亮灭。

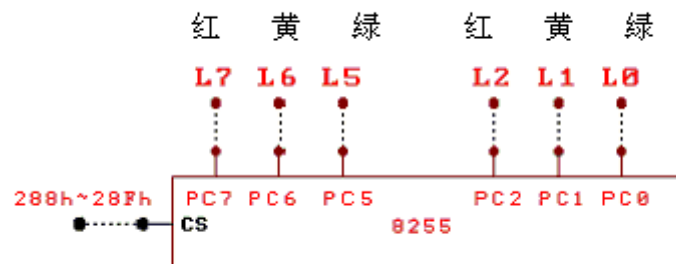


图 1

## 三、编程提示:

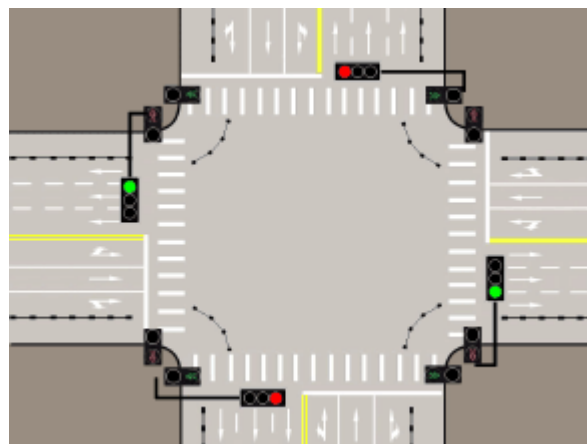


图 2 交通信号灯

十字路口交通灯的变化规律要求：

- (1) 南北路口的绿灯、东西路口的红灯同时亮 30 秒左右。
- (2) 南北路口的黄灯闪烁若干次，同时东西路口的红灯继续亮。
- (3) 南北路口的红灯、东西路口的绿灯同时亮 30 秒左右。
- (4) 南北路口的红灯继续亮、同时东西路口的黄灯亮闪烁若干次。
- (5) 转(1)重复。

算法流程图：

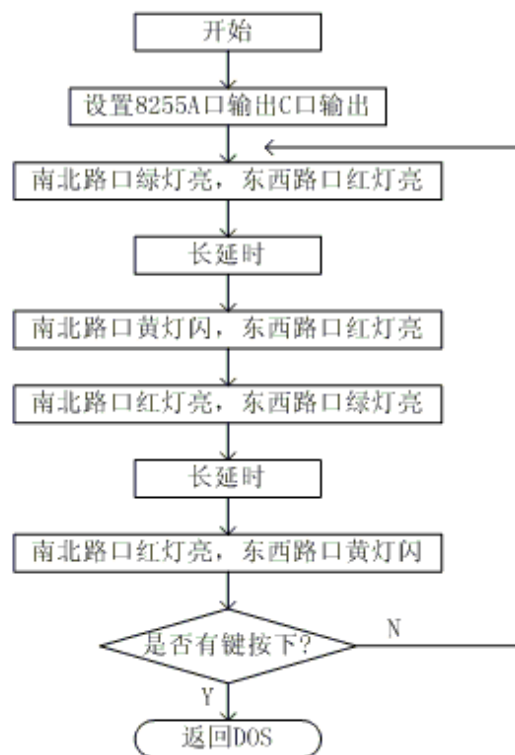


图 3 交通信号灯控制流程图

说明：六个灯状态转换图

1 红 1 黄 **1 绿** 0 0 **2 红** 2 黄 2 绿      灯亮长延时

1 红 **1 黄** 1 绿 0 0 **2 红** 2 黄 2 绿

1 红 1 黄 1 绿 0 0 **2 红** 2 黄 2 绿

1 红 **1 黄** 1 绿 0 0 **2 红** 2 黄 2 绿

1 红 1 黄 1 绿 0 0 **2 红** 2 黄 2 绿

1 红 **1 黄** 1 绿 0 0 **2 红** 2 黄 2 绿

1 红 1 黄 1 绿 0 0 **2 红** 2 黄 2 绿

**1 红** 1 黄 1 绿 0 0 2 红 2 黄 **2 绿**      灯亮长延时

**1 红** 1 黄 1 绿 0 0 2 红 **2 黄** 2 绿

**1 红** 1 黄 1 绿 0 0 2 红 2 黄 2 绿

**1 红** 1 黄 1 绿 0 0 2 红 **2 黄** 2 绿

**1 红** 1 黄 1 绿 0 0 2 红 2 黄 2 绿

**1 红** 1 黄 1 绿 0 0 2 红 **2 黄** 2 绿

**1 红** 1 黄 1 绿 0 0 2 红 2 黄 2 绿

```
portc1  db  24h, 44h, 04h, 44h, 04h, 44h, 04h    ;六个灯可能
          db  81h, 82h, 80h, 82h, 80h, 82h, 80h    ;的状态数据
```

#### 四、参考程序

```
*****
;
;*      十字路口红绿灯模拟演示程序
;*  端口各灯的设置:    1 红 1 黄 1 绿 0 0 2 红 2 黄 2 绿    *;
*****
```

```

data segment
io8255a      equ 28ah
io8255b      equ 28bh
portc1  db  24h, 44h, 04h, 44h, 04h, 44h, 04h  ;六个灯可能
          db  81h, 82h, 80h, 82h, 80h, 82h, 80h  ;的状态数据
          db  0ffh                                ;结束标志
data ends
code  segment
      assume  cs:code, ds:data
start:
      mov     ax, data
      mov     ds, ax
      mov     dx, io8255b
      mov     al, 90h
      out     dx, al          ;设置 8255 为 C 口输出
      mov     dx, io8255a
re_on:
      mov     bx, 0
on:
      mov     al, portc1[bx]
      cmp     al, 0ffh
      jz      re_on
      out     dx, al          ;点亮相应的灯
      inc     bx
      mov     cx, 200         ;参数赋初值
      test    al, 21h         ;是否有绿灯亮
      jz      L0              ;没有,短延时
      mov     cx, 2000        ;有,长延时
L0:
      mov     di, 9000        ;di 赋初值 9000
L1:
      dec     di              ;减 1 计数
      jnz     L1              ;di 不为 0
      loop    L0

      push    dx
      mov     ah, 06h
      mov     dl, 0ffh
      int     21h
      pop     dx              ;判断是否有按键

```

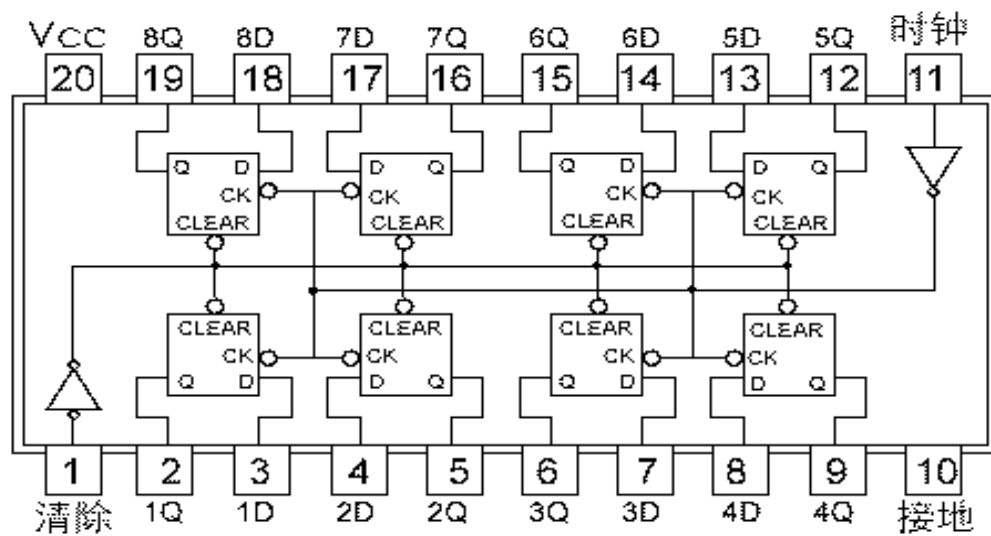
```

        jz  on                ;没有,转到 on
exit:
        mov  ah,4ch           ;返回
        int  21h
code ends
end start

```

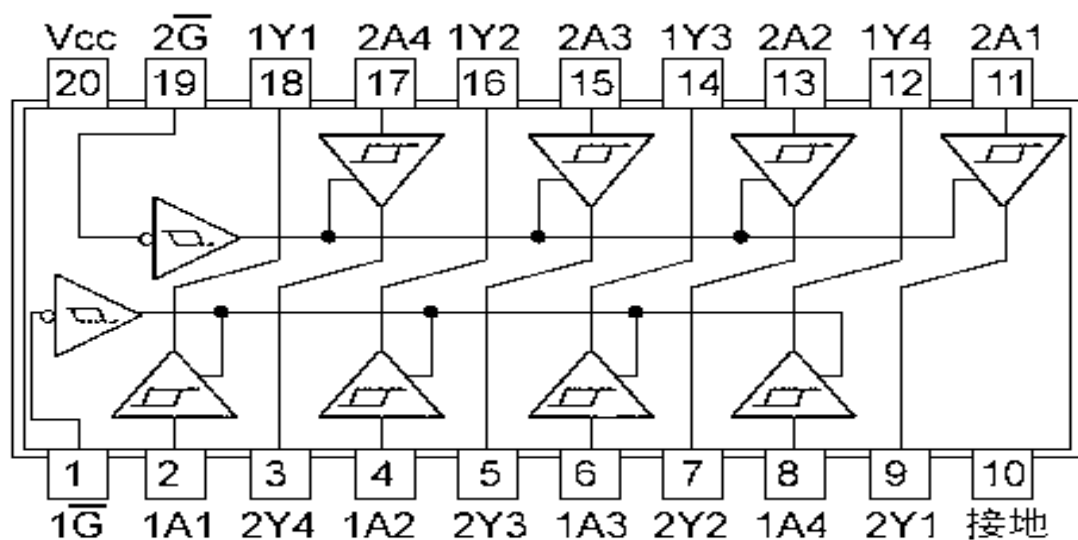
## 芯片使用示例

- 74LS273 是 8 位数据/地址锁存器
- 74LS273 是一种带清除功能的 8D 触发器
  - 1D~8D 为数据输入端，1Q~8Q 为数据输出端，
  - 正脉冲触发，低电平清除，常用作 8 位地址锁存器





**74244 八缓冲器/线驱动器/线收发器**  $Y=A$



此芯片共分两组各4位进行缓冲， $1\overline{G}$ 和 $2\overline{G}$ 分别为两组缓冲的门控端，低电平有效，高电平时输出为高阻态。