北京交通大学汇编与接口技术实验报告

姓名：程维森

学号：21231264

实验三、利用 8255 触发 8259A 的中断实验

一、实验目的

该实验能使学生掌握 8259A 矢量中断方式的硬件连接和软件编程的方法，同时使同学掌握中断和其它接口芯片配合来完成某一特定任务的方法。

二、实验内容

脉冲开关/PLUS 作为 8255 的 PC6 的赋值来源，然后 PC6 作为 8259A 的中断源，每按动一次/PLUS 开关，就向 8259A 发出一个中断请求，就会让某个 LED 指示灯交替点亮和熄灭（亮的灭掉，灭掉的点亮）。选择完成在数码管上显示“8259-A”。

表 1：主从片 8259A 的端口地址如下所示：

用户中断输入引脚　　对应 8259 引脚输入　　中断屏蔽字　中断类型号　8259 端口地址

IRQ5（MIR5） 主片 IR5 11011111B 35H 20H,21H
IRQ8（SIR0）从片 IR0 11111110B 70H 0A0H，0A1H
表 2：32 位微机主 8259A 的中断类型号与中断源的对应关系是：
中断源 IR0 IR1 IR2 IR3 IR4 IR5 IR6 IR7
主片中断向量 30H 31H 32H 33H 34H 35H 36H 37H
从片中断向量 70H 71H 72H 73H 74H 75H 76H 77H
三、实验编程提示
1、 MIR5 是接到 PC 的主 8259A 中断控制器的 IR5 端，因此不需要对 8259A 初始化（ICW1-ICW4），但要进行设置中断矢量和打开中断等操作；注意：微机中采用的是非自动结束，则需要在中断结束前（中断服务程序的最后）发中断结束命令。另外、写入中断屏蔽字应采用“读—修改—写”过程，如下所述：

```
IN  AL，21H
AND AL，0DFH
OUT 21H，AL
```

2、 中断服务程序的主要功能是交替点亮和熄灭某个 LED 指示灯（即第 1 次进中断点亮某个 LED，第 2 次进中断则熄灭该 LED）。当然 8255A 能使用前需要在主程序中初始化。
3、 主程序可以采样死等待的方式，如果要结束程序，可以按动实验平台的 Reset 键。也可以采用计数的方法，即进中断 5 次后结束程序。
4、 8255 的 PC6 作为中断源，应采用上升沿触发方式（由低到高的变化），为了能够进行下次中断，PC6 必须变为低点平。
四、实验步骤
1、 根据原理图自行连接实验线路（需要连接红线）。
2、 正确理解实验原理。
3、 编写实验程序，并上机调试，观察实验结果。
五、思考题
如何理解采用中断方式进行实时控制，请举一些可能的应用例子。

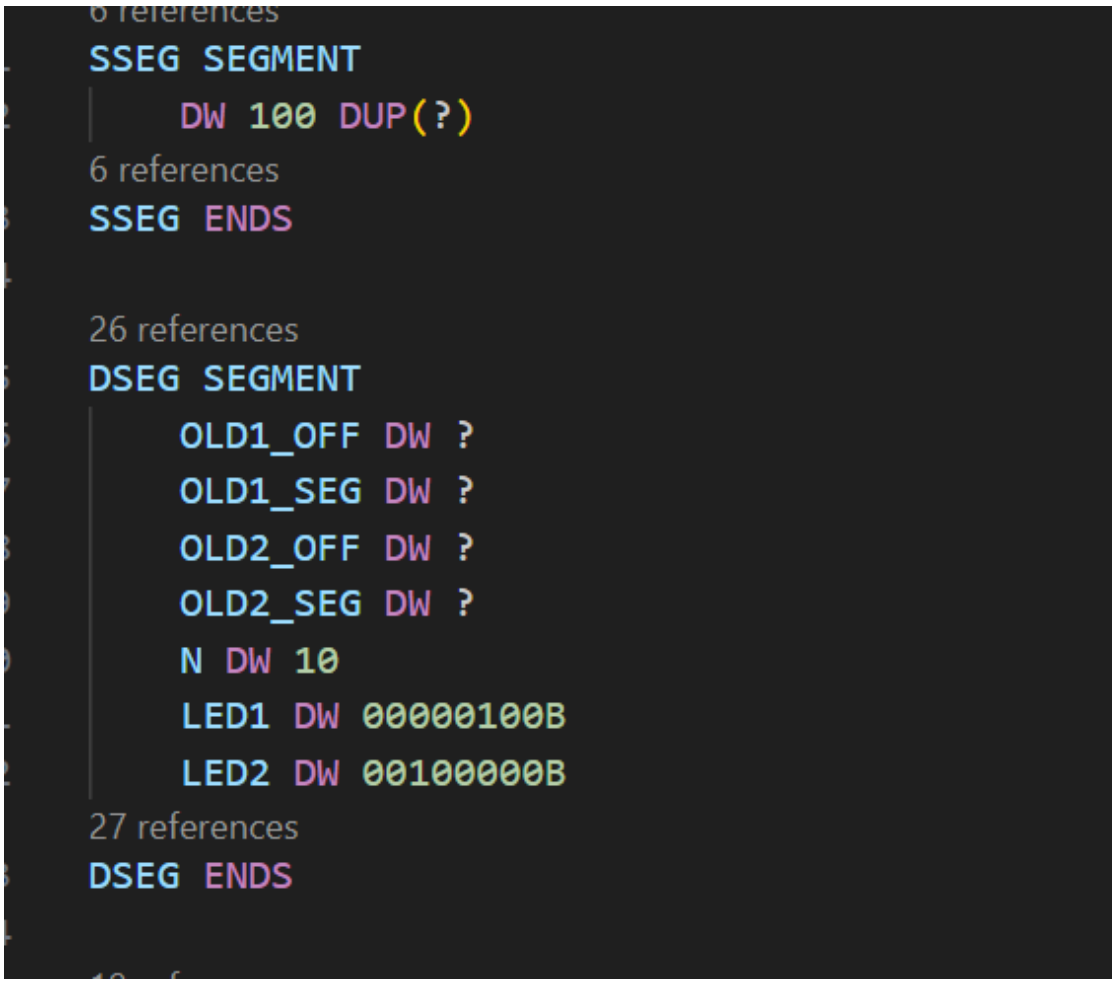
实验内容：

实验步骤：

初始化段和变量：

设置数据段（DSEG）和代码段（CSEG）。

定义存储器段（SSEG）和其中的变量。



I/O 端口操作：

初始化 I/O 端口，清空中断屏蔽位。

调用 SET_TABLE 过程设置中断向量表。

使用 OUT 指令向特定 I/O 端口发送数据，控制 LED 或外部设备的状态。

运用 IN 指令从 I/O 端口读取数据。

```
14 references
START:
    MOV AX,DSEG
    MOV DS,AX

    CLI
    IN AL,21H
    AND AL,0F3H
    OUT 21H,AL

    IN AL,0A1H
    AND AL,0FBH
    OUT 0A1H,AL

    CALL SET_TABLE
    MOV DX,283H
    MOV AL,90H
    OUT DX,AL
    XOR CX,CX
```

中断处理程序：

设置中断处理程序，并在需要时调用它们。

SW1_INT 和 SW2_INT 是两个中断服务程序，它们根据特定条件执行对应的操作，如更改 LED 状态。

```
3 references
S1:
    STI
    HLT
    CMP CX,N
    JNZ S1
    CLI
    CALL RECOVER_TABLE
    IN AL,21H
    OR AL,0CH
    OUT 21H,AL
    IN AL,0A1H
    OR AL,04H
    OUT 0A1H,AL
    STI
    MOV AH,4CH
    INT 21H

8 references
SET_TABLE PROC
    PUSH DS
    PUSH DI
    PUSH BX

    MOV AX,0;
    MOV DS,AX
    MOV DI,4*0BH
    MOV BX,[DI]
    MOV OLD1_OFF,BX
    MOV BX,[DI+2]
    MOV OLD1_SEG,BX
    MOV BX,OFFSET SW1_INT
    MOV [DI],BX
    MOV BX,SEG SW1_INT
    MOV [DI+2],BX

    MOV DI,4*72H
    MOV BX,[DI]
    MOV OLD2_OFF,BX
    MOV BX,[DI+2]
    MOV OLD2_SEG,BX
    MOV BX,OFFSET SW2_INT
    MOV [DI],BX
    MOV BX,SEG SW2_INT
    MOV [DI+2],BX

    POP BX
    POP DI
    POP DS
    RET
```

主程序：

在 START 标签开始执行主程序。

执行一系列指令，包括数据移动、比较、跳转、循环和中断处理程序的调用。

```
SW1_INT PROC
    PUSH AX
    PUSH DX
    CLI
    INC CX
    MOV AX,LED1
    MOV DX,281H
    OUT DX,AX
    XOR AX,00000100B
    MOV LED1,AX
    MOV AL,20H
    OUT 20H,AL
    POP DX
    POP AX
    STI
    IRET
7 references
SW1_INT ENDP

7 references
SW2_INT PROC
    PUSH AX
    PUSH DX
    CLI
    INC CX
    MOV AX,LED2
    MOV DX,281H
    OUT DX,AX
    XOR AX,00100000B
    MOV LED2,AX
    MOV AL,20H
    OUT 20H,AL
    MOV AL,20H
    OUT 0A0H,AL
    POP DX
    POP AX
    STI
    IRET
7 references
SW2_INT ENDP

2 references
CODE SEGMENT
7 references
ASSUME CS:CODE
14 references
START:
214 references
MOV DX,293H
214 references
MOV AL,10001001B
47 references
OUT DX,AL
8 references
XOR AX,AX
```

程序结束：

结束主程序。

输出 8259-A

```
    OUT DX,AL
    MOV DX,290H
    MOV AL,7FH
    OUT DX,AL
    MOV DX,291H
    MOV AL,00000001B
    OUT DX,AL

    MOV DX,291H
    MOV AL,00000000B
    OUT DX,AL
    MOV DX,290H
    MOV AL,5BH
    OUT DX,AL
    MOV DX,291H
    MOV AL,00000010B
    OUT DX,AL

    MOV DX,291H
    MOV AL,00000000B
    OUT DX,AL
    MOV DX,290H
    MOV AL,6DH
    OUT DX,AL
    MOV DX,291H
    MOV AL,00000100B
    OUT DX,AL

    MOV DX,291H
    MOV AL,00000000B
    OUT DX,AL
    MOV DX,290H
    MOV AL,6FH
    OUT DX,AL
    MOV DX,291H
    MOV AL,00001000B
    OUT DX,AL

    MOV DX,291H
    MOV AL,00000000B
    OUT DX,AL
    MOV DX,290H
    MOV AL,40H
    OUT DX,AL
    MOV DX,291H
    MOV AL,00010000B
    OUT DX,AL

    MOV DX,291H
    MOV AL,00000000B
    OUT DX,AL
    MOV DX,290H
    MOV AL,77H
    OUT DX,AL
    MOV DX,291H
```

结束代码段（CODE）。

代码详解：

SET_TABLE 过程

SET_TABLE 过程用于设置中断向量表，将两个中断服务程序的地址写入中断向量表中的特定位置。

首先保存寄存器内容，然后将中断服务程序 SW1_INT 和 SW2_INT 的偏移和段地址写入预先定义的中断向量表位置。

这里使用 MOV 和 OUT 指令来将地址写入中断向量表。

RECOVER_TABLE 过程

RECOVER_TABLE 过程用于恢复中断向量表，将原来保存的中断向量恢复到原始状态。

与 SET_TABLE 相反，这个过程会将之前保存的中断服务程序的地址恢复到中断向量表中。

SW1_INT 和 SW2_INT 过程

SW1_INT 和 SW2_INT 是两个中断服务程序。

SW1_INT 用于处理与 LED1 相关的中断，SW2_INT 用于处理与 LED2 相关的中断。

这两个中断服务程序会进行特定的操作，包括更改 LED1 和 LED2 的状态，并将对应的中断标志位置反。

START 主程序

主程序开始于标签 START。

首先将数据段地址加载到 DS 寄存器中，然后禁止中断。

接着进行端口操作，清除与 8259A 中断控制器相关的屏蔽位。

调用 SET_TABLE 过程设置中断向量表，然后初始化一个计数器 CX。

通过循环（S1 标签），执行一系列操作，期间会触发 SW1_INT 和 SW2_INT 中断服务程序。

循环结束后，恢复中断向量表，最后结束程序并返回 DOS。

其他部分

代码中还包括了初始化数据段、LED 变量、中断服务程序和一些输入输出指令等。

实验结果图片：

TPC-ZK-II综合开放式微机原理实验系统
HUAWEI Mate60 Pro | XMAGE

显示了 8259-A

五、思考题

如何理解采用中断方式进行实时控制，请举一些可能的应用例子。

采用中断方式进行实时控制意味着系统能够在出现特定事件或条件时立即进行响应和处理。这种方式常用于需要及时响应外部事件或需要周期性执行任务的场景。下面是一些可能的应用例子：

实时数据采集与处理：传感器数据的实时采集，例如温度、湿度、压力等。当传感器数据超过阈值时，中断可用于立即处理或记录数据。

实时通信：在串行或并行通信中，中断可用于实时处理接收到的数据。例如，串口通信时，接收到新数据时触发中断，立即进行数据接收或处理。

定时器和计时器：使用定时器中断来实现时间相关的任务，例如定时执行任务、产生精确的时间延迟等。

实时控制系统：用于控制和监测系统的状态并做出实时响应。例如，嵌入式系统中的自动化控制系统，根据传感器输入实时调整控制设备。

多任务处理：操作系统中的任务调度和切换。当某个任务需要立即执行时，操作系统可以利用中断实现任务间的快速切换。

用户交互：例如，键盘输入的中断处理，使计算机能够实时响应用户的按键操作。

实时监控与报警：监控系统中，中断可用于实时检测异常情况并发出警报。例如，监控摄像头捕捉异常活动或监测设备状态。