**西安电子科技大学**

**微机原理与系统设计 课程实验报告**

**实验名称 可编程并行接口实验**

计算机 学院 1803011 班

成 绩

姓名 张帅豪 学号 18030100101

同作者

实验日期 2020 年 12 月 12 日

实验地点 **E-II 312** 实验批次 **第四批**

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验所用仪器（或实验环境）  三、实验基本原理及步骤（或方案设计及理论计算）  四、实验数据记录（或仿真及软件设计）  五、实验结果分析及回答问题（或测试环境及测试结果） |

**实验四 可编程并行接口8255实验**  
一、实验目的  
了解可编程并行接口芯片8255的内部结构、工作方式、初始化编程及应用。  
二、实验内容  
1.流水灯实验：利用8255的A口、B口循环点亮发光二极管。

2.交通灯实验：利用8255的A口模拟交通信号灯。

3.I/O输入输出实验：利用8255的A口读取开关状态，8255的B口把状态送发光二极管显示。

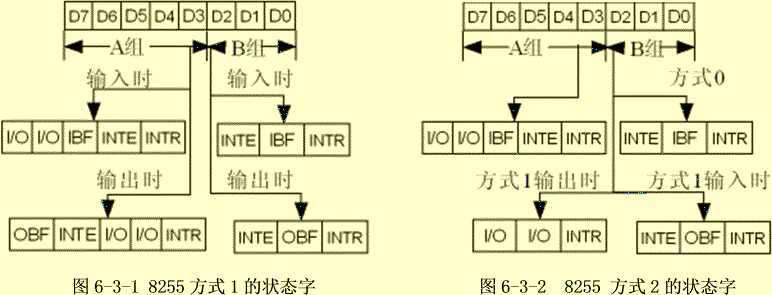
4.在完成(1)基础上，增加通过读取开关控制流水灯的循环方向和循环方式。

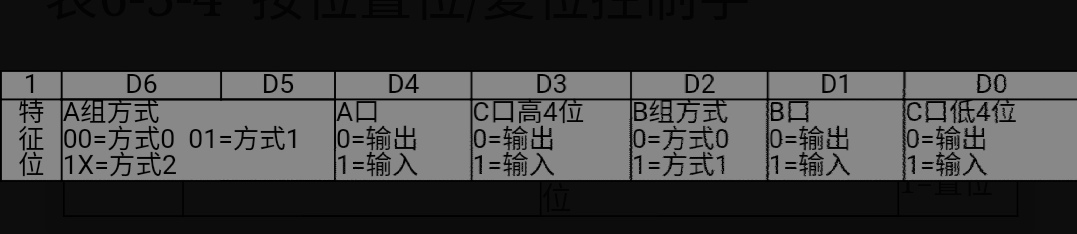
5.在完成(2)基础上，增加通过读取开关控制交通红绿灯的亮灭时间。

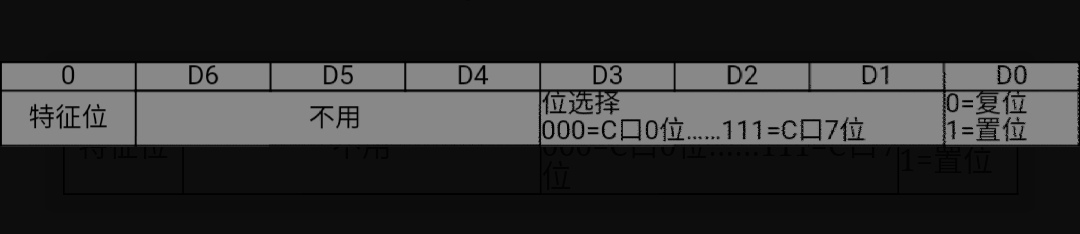
三、实验说明  
1、8255A的内部结构  
（1）数据总线缓冲器：这是一个双向三态的8位数据缓冲器，它是8255A与微机系统数据总线的接口。输入输出的数据、CPU输出的控制字以及CPU输入的状态信息都是通过这个缓冲器传送的。  
（2）三个端口A，B和C：A端口包含一个8位数据输出锁存器和缓冲器，一个8位数据输入锁存器。B端口包含一个8位数据输入/输出锁存器和缓冲器，一个8位数据输入缓冲器。C端口包含一个8位数据输出锁存器及缓冲器，一个8位数据输入缓冲器（输入没有锁存器）。

(3）A组和B组控制电路：这是两组根据CPU输出的控制字控制8255工作方式的电路，它们对于CPU而言，共用一个端口地址相同的控制字寄存器，接收CPU输出的一字节方式控制字或对C口按位复位字命令。方式控制字的高5位决定A组工作方式，低3位决定B组的工作方式。对C口按位复位命令字可对C口的每一位实现置位或复位。A组控制电路控制A口和C口上半部，B组控制电路控制B口和C口下半部。  
（4）读写控制逻辑：用来控制把CPU输出的控制字或数据送至相应端口，也由它来控制把状态信息或输入数据通过相应的端口送到CPU。

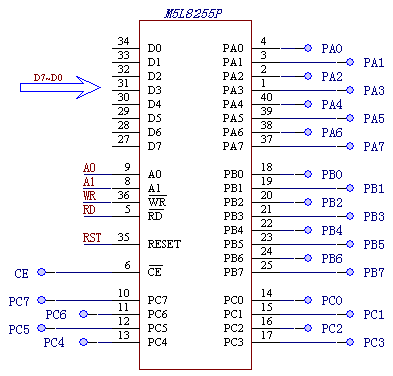
、8255A的工作方式  
方式0—基本输入输出方式；方式1—选通输入输出方式；方式2—双向选通输入输出方式

  
3、8255A的状态字

  
4、8255A的控制字



四、实验原理图



五、实验步骤

1.流水灯实验

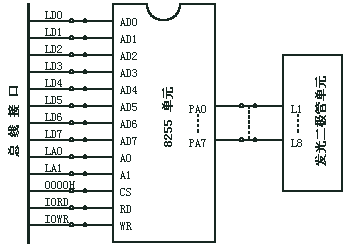
模块的WR、RD分别连到ISA总线接口模块的IOWR、IORD。

模块的数据（AD0～AD7）、地址线（A0～A7）分别连到ISA总线接口模块的数据（LD0～LD7）、地址线（LA0～LA7）。

8255模块选通线CE连到ISA总线接口模块的0000H。

8255的PA0～PA7连到发光二极管的L0～L7；8255的PB0～PB7连到发光二极管的L8～L15。

运行程序，观察发光二极管。



MY\_STACK SEGMENT PARA 'STACK'   
 DB 100 DUP(?)   
MY\_STACK ENDS   
;   
MY\_DATA SEGMENT PARA 'DATA'   
IO\_9054base\_address DB 4 DUP(0) ;PCI卡9054芯片I/O基地址暂存空间   
IO\_base\_address DB 4 DUP(0) ;PCI卡I/O基地址暂存空间   
pcicardnotfind DB 0DH,0AH,'pci card not find or address/interrupt error !!!',0DH,0AH,'$'   
GOOD DB 0DH,0AH,'The Program is Executing !',0DH,0AH,'$'   
LA DB ?   
LB DB ?   
P8255\_A DW 0000H   
P8255\_B DW 0001H   
P8255\_C DW 0002H   
P8255\_MODE DW 0003H   
   
DELAY\_SET EQU 0FFFH   
MES2 DB ' PCI CONFIG READ ERROR! $'   
MY\_DATA ENDs   
   
MY\_CODE SEGMENT PARA 'CODE'   
   
MY\_PROC PROC FAR   
 ASSUME CS:MY\_CODE, DS:MY\_DATA, SS:MY\_STACK   
   
START:   
.386 ;386模式编译   
 MOV AX,MY\_DATA   
 MOV DS,AX   
 MOV ES,AX   
 MOV AX,MY\_STACK   
 MOV SS,AX   
 CALL FINDPCI ;自动查找PCI卡资源及IO口基址   
 MOV CX,word ptr IO\_base\_address   
; MOV CX,0E800H ;直接加入(E800:本机PCI卡IO口基址)   
   
 ADD P8255\_A,CX ;PCI卡IO基址+偏移   
 ADD P8255\_B,CX   
 ADD P8255\_C,CX   
 ADD P8255\_MODE,CX   
   
 MOV DX,P8255\_MODE ;8255初始化，三个口全为输出   
 MOV AL,80H   
 OUT DX,AL   
   
 MOV DX,P8255\_A ;PA口赋初值   
 MOV AL,7FH   
 OUT DX,AL   
 MOV LA,AL   
   
 MOV DX,P8255\_B ;PB口赋初值   
 MOV AL,0FEH   
 OUT DX,AL   
 MOV LB,AL   
 CALL DELAY   
   
A1: MOV AL,LA ;PA 口数据右移一位   
 ROR AL,1   
 MOV LA,AL   
 MOV DX,P8255\_A   
 OUT DX,AL   
 MOV AL,LB   
 ROL AL,1   
 MOV LB,AL   
 MOV DX,P8255\_B ;PB口数据右移一位   
 OUT DX,AL   
 CALL DELAY ;延时   
 CALL BREAK ;按任意键退出   
 JMP A1   
MY\_PROC ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 延时子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
DELAY PROC NEAR ;延时程序   
 PUSHF   
 PUSH DX   
 PUSH CX   
 MOV DX,DELAY\_SET   
D1: MOV CX,-1   
D2: DEC CX   
 JNZ D2   
 DEC DX   
 JNZ D1   
 POP CX   
 POP DX   
 POPF   
 RET   
DELAY ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 按任意键退出子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
BREAK PROC NEAR   
 MOV AH,06H   
 MOV DL,0FFH   
 INT 21H   
 JE RETURN   
 MOV AX,4C00H   
 INT 21H   
RETURN: RET   
BREAK ENDP   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 找卡子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
IO\_port\_addre EQU 0CF8H ;32位配置地址端口   
IO\_port\_data EQU 0CFCH ;32位配置数据端口   
IO\_PLX\_ID EQU 200810B5H ;PCI卡设备及厂商ID   
BADR0 = 10H ;基地址寄存器0   
BADR1 = 14H ;基地址寄存器1   
BADR2 = 18H ;基地址寄存器2   
BADR3 = 1CH ;基地址寄存器3   
   
FINDPCI PROC NEAR ;查找PCI卡资源并显示   
 PUSHAD   
 PUSHFD   
 MOV EBX,080000000H   
FINDPCI\_next:   
 ADD EBX,100H   
 CMP EBX,081000000H   
 JNZ findpci\_continue   
 MOV DX,offset pcicardnotfind ;显示未找到PCI卡提示信息   
 MOV AH,09H   
 INT 21H   
 MOV AH,4CH   
 INT 21H ;退出   
findpci\_continue:   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 CMP EAX,IO\_PLX\_ID   
 JNZ findpci\_next ;检查是否发现PCI卡   
   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 ADD EAX,BADR1   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 MOV dword ptr IO\_9054base\_address,EAX   
 AND EAX,1   
 JZ findPCI\_next ;检查是否为i/o基址信息   
 MOV EAX,dword ptr IO\_9054base\_address   
 AND EAX,0fffffffeh   
 MOV dword ptr IO\_9054base\_address,EAX ;去除i/o指示位并保存   
   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 ADD EAX,BADR2   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 MOV dword ptr IO\_base\_address,EAX   
 AND EAX,1   
 JZ findPCI\_next ;检查是否为i/o基址信息   
 MOV EAX,dword ptr IO\_base\_address   
 AND EAX,0fffffffeh   
 MOV dword ptr IO\_base\_address,EAX ;去除i/o指示位并保存   
 MOV DX,offset good ;显示开始执行程序信息   
 MOV AH,09H   
 INT 21H   
 POPfd   
 POPad   
 RET   
findPCI ENDP   
   
MY\_CODE ENDS   
   
 END START

2.交通灯实验

交通灯实验连线图

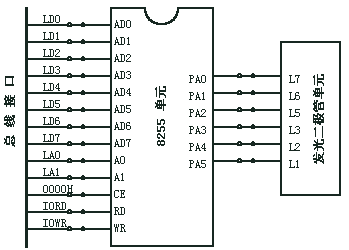
模块的WR、RD分别连到ISA总线接口模块的IOWR、IORD。

模块的数据（AD0～AD7）、地址线（A0～A7）分别连到ISA总线接口模块的数据（LD0～LD7）、地址线（LA0～LA7）。

8255模块选通线CE连到ISA总线接口模块的0000H。

8255的PA0-L7、PA1-L6、PA2-L5、PA3-L3、PA4-L2、PA5-L1。

运行程序，观察发光二极管。



;交通灯   
   
MY\_STACK SEGMENT PARA 'STACK'   
 DB 100 DUP(?)   
MY\_STACK ENDS   
   
MY\_DATA SEGMENT PARA 'DATA'   
IO\_9054base\_address DB 4 DUP(0) ;PCI卡9054芯片I/O基地址暂存空间   
IO\_base\_address DB 4 DUP(0) ;PCI卡I/O基地址暂存空间   
pcicardnotfind DB 0DH,0AH,'pci card not find or address/interrupt error !!!',0DH,0AH,'$'   
GOOD DB 0DH,0AH,'The Program is Executing !',0DH,0AH,'$'   
   
P8255\_A DW 0000H   
P8255\_B DW 0001H   
P8255\_C DW 0002H   
P8255\_MODE DW 0003H   
DELAY\_SET EQU 1fffH   
DELAY\_SET1 EQU 08FfH   
   
MES2 DB ' PCI CONFIG READ ERROR! $'   
   
MY\_DATA ENDs   
   
MY\_CODE SEGMENT PARA 'CODE'   
   
MY\_PROC PROC FAR   
 ASSUME CS:MY\_CODE, DS:MY\_DATA, SS:MY\_STACK   
   
START:   
.386 ;386模式编译   
 MOV AX,MY\_DATA   
 MOV DS,AX   
 MOV ES,AX   
 MOV AX,MY\_STACK   
 MOV SS,AX   
 CALL FINDPCI ;自动查找PCI卡资源及IO口基址   
 MOV CX,word ptr IO\_base\_address   
; MOV CX,0E800H ;直接加入(E800:本机PCI卡IO口基址)   
   
 ADD P8255\_A,CX ;PCI卡IO基址+偏移   
 ADD P8255\_B,CX   
 ADD P8255\_C,CX   
 ADD P8255\_MODE,CX   
   
 MOV DX,P8255\_MODE   
 MOV AL,80H ;3个口全部为输出   
 OUT DX,AL   
   
 MOV DX,P8255\_A   
 CALL ST0 ;全为红灯   
TRAFFIC1: CALL ST1 ;南北为绿灯，东西为红灯   
 CALL ST2 ;南北黄灯闪烁，东西为红灯   
 CALL ST3 ;南北为红灯，东西为绿灯   
 CALL ST4 ;南北为红灯，东西黄灯闪烁   
 CALL BREAK   
 JMP TRAFFIC1   
MY\_PROC ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\*初始状态全为红灯\*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
ST0 PROC NEAR   
 MOV AL,1BH;0001 1011   
 OUT DX,AL   
 CALL DELAY   
 RET   
ST0 ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\*南北为绿灯，东西为红灯子程序\*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
ST1 PROC NEAR   
 MOV DX,P8255\_A   
 MOV AL,2bH ;0011 0011 ;00 101 011   
 OUT DX,AL   
 CALL DELAY   
 call delay   
 call delay   
 call delay   
 call delay   
 RET   
ST1 ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\*南北红灯闪烁，东西为红灯子程序\*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
ST2 PROC NEAR   
 MOV CX,2H   
ST20: MOV AL,33H ;0010 1011 ;00 110 011   
 OUT DX,AL   
 CALL DELAY   
 MOV AL,3BH ;0011 1011   
 OUT DX,Al   
 CALL DELAY   
 LOOP ST20   
 RET   
ST2 ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\*南北为红灯，东西为绿灯子程序\*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
ST3 PROC NEAR   
 MOV AL,1dH ;0001 1110 ;00 011 101   
 OUT DX,AL   
 CALL DELAY   
 call delay   
 call delay   
 call delay   
 call delay   
 RET   
ST3 ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\*南北为红灯，东西黄灯闪烁子程序\*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
ST4 PROC NEAR   
 MOV CX,2H   
ST40: MOV AL,1eH;0001 1101 ;00 011 110   
 OUT DX,AL   
 CALL DELAY   
 MOV AL,1FH;0001 1111 ; 00 011 111   
 OUT DX,AL   
 CALL DELAY   
 LOOP ST40   
 RET   
ST4 ENDp   
   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 延时子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
DELAY PROC NEAR ;延时程序   
 PUSHF   
 PUSH DX   
 PUSH CX   
 MOV DX,DELAY\_SET   
D1: MOV CX,-1   
D2: DEC CX   
 JNZ D2   
 DEC DX   
 JNZ D1   
 POP CX   
 POP DX   
 POPF   
 RET   
DELAY ENDp   
   
DELAY1 PROC NEAR ;延时程序   
 PUSHF   
 PUSH DX   
 PUSH CX   
 MOV DX,DELAY\_SET1   
D3: MOV CX,-1   
D4: DEC CX   
 JNZ D4   
 DEC DX   
 JNZ D3   
 POP CX   
 POP DX   
 POPF   
 RET   
DELAY1 ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 按任意键退出子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
BREAK PROC NEAR ;按任意键退出   
 PUSHF   
 PUSH AX   
 PUSH DX   
 MOV AH,06H   
 MOV DL,0FFH   
 INT 21H   
 JE RETURN   
 MOV AX,4C00H   
 INT 21H   
RETURN: POP DX   
 POP AX   
 POPF   
 RET   
BREAK ENDP   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 找卡子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
;FUNCTION CODE   
IO\_port\_addre EQU 0CF8H ;32位配置地址端口   
IO\_port\_data EQU 0CFCH ;32位配置数据端口   
IO\_PLX\_ID EQU 200810B5H ;PCI卡设备及厂商ID   
BADR0 = 10H ;基地址寄存器0   
BADR1 = 14H ;基地址寄存器1   
BADR2 = 18H ;基地址寄存器2   
BADR3 = 1CH ;基地址寄存器3   
   
FINDPCI PROC NEAR ;查找PCI卡资源并显示   
 PUSHAD   
 PUSHFD   
 MOV EBX,080000000H   
FINDPCI\_next:   
 ADD EBX,100H   
 CMP EBX,081000000H   
 JNZ findpci\_continue   
 MOV DX,offset pcicardnotfind ;显示未找到PCI卡提示信息   
 MOV AH,09H   
 INT 21H   
 MOV AH,4CH   
 INT 21H ;退出   
findpci\_continue:   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 CMP EAX,IO\_PLX\_ID   
 JNZ findpci\_next ;检查是否发现PCI卡   
   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 ADD EAX,BADR1   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 MOV dword ptr IO\_9054base\_address,EAX   
 AND EAX,1   
 JZ findPCI\_next ;检查是否为i/o基址信息   
 MOV EAX,dword ptr IO\_9054base\_address   
 AND EAX,0fffffffeh   
 MOV dword ptr IO\_9054base\_address,EAX ;去除i/o指示位并保存   
   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 ADD EAX,BADR2   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 MOV dword ptr IO\_base\_address,EAX   
 AND EAX,1   
 JZ findPCI\_next ;检查是否为i/o基址信息   
 MOV EAX,dword ptr IO\_base\_address   
 AND EAX,0fffffffeh   
 MOV dword ptr IO\_base\_address,EAX ;去除i/o指示位并保存   
 MOV DX,offset good ;显示开始执行程序信息   
 MOV AH,09H   
 INT 21H   
 POPfd   
 POPad   
 RET   
findPCI ENDP   
   
MY\_CODE ENDS   
   
 END START

3.I/O输入输出实验

I/O输入输出实验连线图

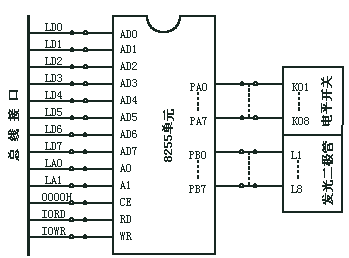
该模块的WR、RD分别连到ISA总线接口模块的IOWR、IORD。

该模块的数据（AD0～AD7）、地址线（A0～A7）分别连到ISA总线接口模块的数据（LD0～LD7）、地址线（LA0～LA7）。

8255模块选通线CE连到ISA总线接口模块的0000H。

8255的PA0～PA7接开关K0～K7，8255的PB0～PB7接发光二极管L0～L7。

运行程序，拨动开关,观察发光二极管。



;I/O输入输出   
   
MY\_STACK SEGMENT PARA 'STACK'   
 DB 100 DUP(?)   
MY\_STACK ENDS   
;   
MY\_DATA SEGMENT PARA 'DATA'   
IO\_9054base\_address DB 4 DUP(0) ;PCI卡9054芯片I/O基地址暂存空间   
IO\_base\_address DB 4 DUP(0) ;PCI卡I/O基地址暂存空间   
pcicardnotfind DB 0DH,0AH,'pci card not find or address/interrupt error !!!',0DH,0AH,'$'   
GOOD DB 0DH,0AH,'The Program is Executing !',0DH,0AH,'$'   
LA DB ?   
LB DB ?   
P8255\_A DW 0000H   
P8255\_B DW 0001H   
P8255\_C DW 0002H   
P8255\_MODE DW 0003H   
   
DELAY\_SET EQU 01FFH   
MES2 DB ' PCI CONFIG READ ERROR! $'   
MY\_DATA END   
   
MY\_CODE SEGMENT PARA 'CODE'   
   
MY\_PROC PROC FAR   
 ASSUME CS:MY\_CODE, DS:MY\_DATA, SS:MY\_STACK   
   
START:   
.386 ;386模式编译   
 MOV AX,MY\_DATA   
 MOV DS,AX   
 MOV ES,AX   
 MOV AX,MY\_STACK   
 MOV SS,AX   
 CALL FINDPCI ;自动查找PCI卡资源及IO口基址   
 MOV CX,word ptr IO\_base\_address   
; MOV CX,0E800H ;直接加入(E800:本机PCI卡IO口基址)   
   
 ADD P8255\_A,CX ;PCI卡IO基址+偏移   
 ADD P8255\_B,CX   
 ADD P8255\_C,CX   
 ADD P8255\_MODE,CX   
   
 MOV DX,P8255\_MODE ;8255初始化   
 MOV AL,90H   
 OUT DX,AL   
   
E: MOV DX,P8255\_A   
 IN AL,DX   
 MOV DX,P8255\_B   
 OUT DX,AL   
 CALL DELAY   
 JMP E   
   
MY\_PROC ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 延时子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
DELAY PROC NEAR ;延时程序   
 PUSHF   
 PUSH DX   
 PUSH CX   
 MOV DX,DELAY\_SET   
D1: MOV CX,-1   
D2: DEC CX   
 JNZ D2   
 DEC DX   
 JNZ D1   
 POP CX   
 POP DX   
 POPF   
 RET   
DELAY ENDp   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 按任意键退出子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
BREAK PROC NEAR   
 MOV AH,06H   
 MOV DL,0FFH   
 INT 21H   
 JE RETURN   
 MOV AX,4C00H   
 INT 21H   
RETURN: RET   
BREAK ENDP   
;   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
; /\* 找卡子程序 \*/   
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
;   
IO\_port\_addre EQU 0CF8H ;32位配置地址端口   
IO\_port\_data EQU 0CFCH ;32位配置数据端口   
IO\_PLX\_ID EQU 200810B5H ;PCI卡设备及厂商ID   
BADR0 = 10H ;基地址寄存器0   
BADR1 = 14H ;基地址寄存器1   
BADR2 = 18H ;基地址寄存器2   
BADR3 = 1CH ;基地址寄存器3   
   
FINDPCI PROC NEAR ;查找PCI卡资源并显示   
 PUSHAD   
 PUSHFD   
 MOV EBX,080000000H   
FINDPCI\_next:   
 ADD EBX,100H   
 CMP EBX,081000000H   
 JNZ findpci\_continue   
 MOV DX,offset pcicardnotfind ;显示未找到PCI卡提示信息   
 MOV AH,09H   
 INT 21H   
 MOV AH,4CH   
 INT 21H ;退出   
findpci\_continue:   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 CMP EAX,IO\_PLX\_ID   
 JNZ findpci\_next ;检查是否发现PCI卡   
   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 ADD EAX,BADR1   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 MOV dword ptr IO\_9054base\_address,EAX   
 AND EAX,1   
 JZ findPCI\_next ;检查是否为i/o基址信息   
 MOV EAX,dword ptr IO\_9054base\_address   
 AND EAX,0fffffffeh   
 MOV dword ptr IO\_9054base\_address,EAX ;去除i/o指示位并保存   
   
 MOV DX,IO\_port\_addre   
 MOV EAX,EBX   
 ADD EAX,BADR2   
 OUT DX,EAX ;写地址口   
 MOV DX,IO\_port\_data   
 IN EAX,DX ;读数据口   
 MOV dword ptr IO\_base\_address,EAX   
 AND EAX,1   
 JZ findPCI\_next ;检查是否为i/o基址信息   
 MOV EAX,dword ptr IO\_base\_address   
 AND EAX,0fffffffeh   
 MOV dword ptr IO\_base\_address,EAX ;去除i/o指示位并保存   
 MOV DX,offset good ;显示开始执行程序信息   
 MOV AH,09H   
 INT 21H   
 POPfd   
 POPad   
 RET   
findPCI ENDP   
   
MY\_CODE ENDS   
   
 END START