

XI`AN TECHNOLOGICAL UNIVERSITY

课程设计报告

课程名称 汇编程序设计与微机接口

专 业： 计算机科学与技术

班 级： 16060104

姓 名： 杨旭

学 号： 16060104117

指导教师： 马静

成 绩：

2019 年 1 月 9 日­­

**课程设计（论文）任务书**

**2018-2019学年第1学期**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | **汇编语言与微机接口** | | | | | | | |
| **课设题目** | **竞赛抢答器** | | | | | | | 项目B |
| **设计任务** | 设计一个6路的智力竞赛抢答器。在主持人侧，设置启动和复位开关以及抢答指示电路。选手侧各设置1个抢答按钮和指示灯。能够点亮选手灯。能够显示选手编号。 | | | | | | | |
| **设计要求** | 1.硬件测试通过。8259能够实现中断。8254能够实现呼吸灯。8255能够控制L8—L15的亮与灭。  2. 启动与复位。主持人按START键表示表示启动一次抢答，绿色发光二极管亮，作允许抢答指示。此时选手侧可以按动各自的抢答按钮。主持人按RESET表示复位。红绿指示灯全灭，选手灯全灭。主持人可以再次按START键，启动下一次抢答。  3. 抢答。使用实验仪的逻辑电平开关K0-K5来代表6个抢答按钮。当开关向上拨时表示按下按钮。收到第1个抢答信号后，主持人侧红色发光二极管亮（同时，绿色发光二极管还亮）作已经抢答的指示。  4. 显示选手灯。在单个LED灯上显示抢先一组的组别。  5. 在数码管上显示选手编号。  6、同时考虑抢答允许时间,10秒内有效。10秒之后无人抢答，本次抢答作废。 | | | | | | | |
| **成果要求** | 1. 设计出完整的硬件线路图。  2、编写出全部控制程序。  3、在实验箱上调试通过。  4、参加答辩或小测验。  5、写出完整、规范的课程设计报告。在指定的时间内提交指导教师。 | | | | | | | |
| **其他事项** | 课程设计的封面、内容要求以及格式要求等相关资料，请参看附件。格式必须按照要求编辑、打印、装订。如不符合要求的将按不合格处理。在课程设计中严禁出现相互抄袭、雷同的情况。如有发现，将按照零分处理。 | | | | | | | |
| **指导教师签名** | |  | | | **日期** | 2019．1.9 | | |
| **团队成员** | 杨 旭（16060104117）  赵泽鑫（16060103121） | | | | **同组学生** | | 16060103121赵泽鑫 | |
| **专业** | 计算机科学与技术 | | **班级** | 16060104 | **学号** | 16060104117 | | |

学生签名： 2019年1月9日

摘要

中断技术在计算机中应用极为广泛，它不仅可用于数据传输，提高数据传输过程中CPU的利用效率，还可以用来处理一些需要实时响应的事件，例如异常，时钟，掉电，特殊状况等。

本文选择的芯片8259、8255和8254。用8259芯片，实现中断控制。结合8254实现定时中断，使用8255芯片作为接口。在Masm for Windows 集成实验环境下，编写汇编程序，成功采用硬件中断实现了竞赛抢答器

目录

[摘要 3](#_Toc534890514)

[一、设计目的 4](#_Toc534890515)

[二、设计任务与要求 4](#_Toc534890516)

[三、设计方案 5](#_Toc534890517)

[四、硬件线路设计 7](#_Toc534890518)

[五、软件设计 8](#_Toc534890519)

[1、程序流程图 8](#_Toc534890520)

[2、程序说明 9](#_Toc534890521)

[六、源程序 11](#_Toc534890522)

[七、调试过程 20](#_Toc534890523)

[八、总结 21](#_Toc534890524)

[九、参考文献 21](#_Toc534890525)

# 一、设计目的

1. 采用硬件中断解决竞赛抢答器问题。

2. 进一步巩固8259、8254以及8255的应用。掌握综合设计能力。

# 二、设计任务与要求

* **设计任务**

设计一个6路的智力竞赛抢答器。在主持人侧，设置启动和复位开关以及抢答指示电路。选手侧各设置1个抢答按钮和指示灯。能够点亮选手灯。能够显示选手编号。

* **设计要求**

1.硬件测试通过。8259能够实现中断。8254能够实现呼吸灯。8255能够控制L8—L15的亮与灭。

2. 启动与复位。主持人按START键表示表示启动一次抢答，绿色发光二极管亮，作允许抢答指示。此时选手侧可以按动各自的抢答按钮。主持人按RESET表示复位。红绿指示灯全灭，选手灯全灭。主持人可以再次按START键，启动下一次抢答。

3. 抢答。使用实验仪的逻辑电平开关K0-K5来代表6个抢答按钮。当开关向上拨时表示按下按钮。收到第1个抢答信号后，主持人侧红色发光二极管亮（同时，绿色发光二极管还亮）作已经抢答的指示。

4. 显示选手灯。在单个LED灯上显示抢先一组的组别。

5. 在数码管上显示选手编号。

6、同时考虑抢答允许时间,10秒内有效。10秒之后无人抢答，锁死。本次抢答作废。

* **成果要求**

1. 设计出完整的硬件线路图。

2、编写出全部控制程序。

3、在实验箱上调试通过。

4、参加答辩或小测验。

5、写出完整、规范的课程设计报告。在指定的时间内提交指导教师。

# 三、设计方案

**一、 需求分析**

**问题要求**设计一个6路的智力竞赛抢答器。在主持人侧，设置启动和复位开关以及抢答指示电路。选手侧各设置1个抢答按钮和指示灯。能够点亮选手灯。能够显示选手编号。

本系统要达到的主要目标。深入了解和分析被控对象的控制要求。控制要求主要指控制的基本方式、应完成的动作、自动工作循环的组成。对较复杂的控制系统，还可将控制任务分成几个独立部分。这样可以化繁为简，有利于编程和调试。

还有输入(采样)和输出信号(控制)需求、人机交换需求、工作环境需求和经济指标等。

**二、确定硬件I/O设备**

本课题需要的输入设备有：8个开关K0～K7。一个按键，KK1+。

本课题需要的输出设备有：8个LED灯L0～L7。一位数码管显示器。

**三、选择合适的接口芯片**

选择的芯片为**8259、8255和8254。**

1、8259芯片，实现中断控制。结合8254实现定时中断。启动和停止也由相应的中断服务程序实现。有三个中断源开放，MIR6、MIR7和SIR1。MIR6接OUT0，MIR7接KK1+，SIR1接KK2+。

2、8255芯片，有PA、PB和PC三个并行口。PA口接8个开关的输入，让PB口去控制选手LED灯L0～L7。PC口自然去接数码管。

3、8254芯片，实现硬件定时。定时10s。有两个通道CNT0和CNT1可用。选用CNT0和CNT1使用。

**四、整理各个接口芯片与系统的接线图**

根据试验箱上提供的资源，参照做过的硬件实验，整理各个接口芯片与系统的接线图。合理分配端口地址。

1. 端口地址

表2 实验箱8259端口地址说明

|  |  |  |  |
| --- | --- | --- | --- |
| 8259端口地址 | | | |
| 主片 | | 从片 | |
| 偶地址 | 奇地址 | 偶地址 | 奇地址 |
| 20H | 21H | 0A0H | 0A1H |

表3 实验箱IO端口地址说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试验箱IO端口地址 | | | | |
|  | A1A0=00 | A1A0=01 | A1A0=10 | A1A0=11 |
| IOY0 | 0600H | 0602H | 0604H | 0606H |
| IOY1 | 0640H | 0642H | 0644H | 0646H |
| IOY2 | 0680H | 0682H | 0684H | 0686H |
| IOY3 | 06C0H | 06C2H | 06C4H | 06C6H |

1. 各个芯片与系统总线的连接

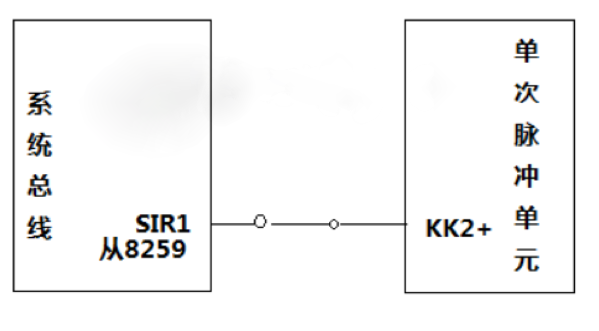
表4 芯片与系统总线的连接

|  |  |  |
| --- | --- | --- |
| 系统总线 | 8255 | 8354 |
| XD7～XD0 | D7～D0 | D7～D0 |
| XA2 | A1 | A1 |
| XA1 | A0 | A0 |
|  |  |  |
|  |  |  |
| IOY0 |  |  |
| IOY1 |  |  |

# 四、硬件线路设计

1、硬件接线图

8259芯片与脉冲开关相接接线图如下：

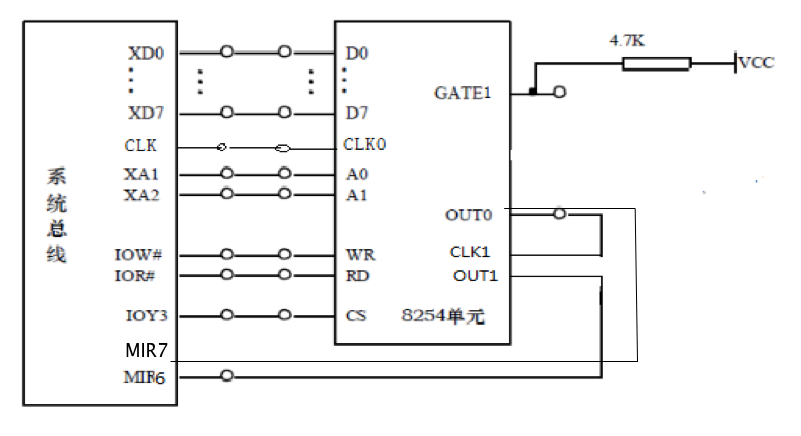


单次脉冲输出与主片 8259 的 IR7 相连，每按动一次单次脉冲，产生一

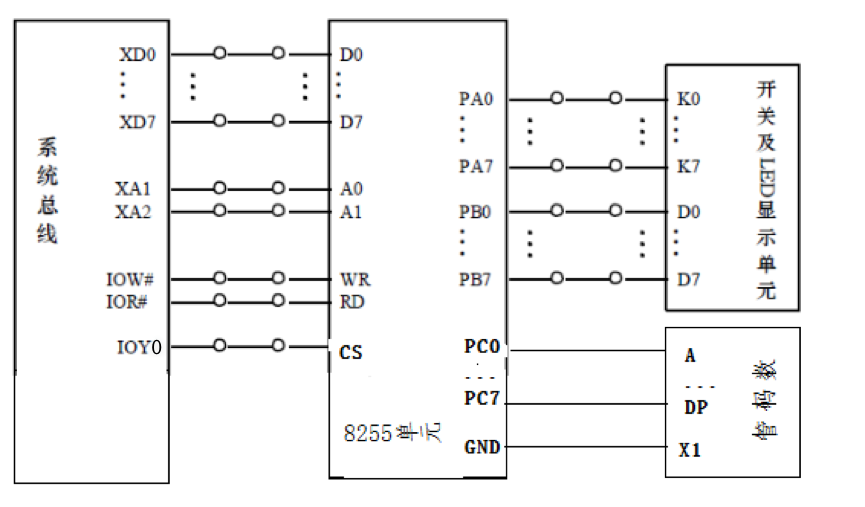
次外部中断，在显示屏上输出一个字符”1”。在对 8259 进行编程时，首先必须进行初始化。一般先使用 CLI 指令将所有的可屏蔽中断禁止，然后写入初始化命令字。8259 有一个状态机控制对寄存器的访问，不正确的初始化顺序会造成异常初始化。在初始化主片 8259 时，写入初始化命令字的顺序是：ICW1、ICW2、ICW3、然后是 ICW4，初始化从片 8259 的顺序与初始化主片 8259 的顺序是相同的。系统启动时，主片 8259 已被初始化，且 4 号中断源（IR4）提供给与 PC 联机的串口通信使用，其它中断源被屏蔽.

8254接线图如下：

应用 8254 的定时功能，产生一个10s一个产生中断。



8255接线图如下：



并行接口是以数据的字节为单位与 I/O 设备或被控制对象之间传递信息。CPU 和接口之间的数据传送总是并行的，即可以同时传递 8 位、16 位或 32 位等。8255 可编程外围接口芯片是 Intel 公司生产的通用并行 I/O 接口芯片，它具有 A、B、C 三个并行接口，用+5V 单电源供电，能在以下三种方式下工作：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--双向选通工作方式

2、端口地址说明

表2-1 实验箱IO端口地址说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试验箱IO端口地址 | | | | |
|  | A1A0=00 | A1A0=01 | A1A0=10 | A1A0=11 |
| IOY0 | 0600H | 0602H | 0604H | 0606H |
| IOY1 | 0640H | 0642H | 0644H | 0646H |
| IOY2 | 0680H | 0682H | 0684H | 0686H |
| IOY3 | 06C0H | 06C2H | 06C4H | 06C6H |

# 五、软件设计

## 1、程序流程图

开始

设置LED字形码初值、标志位初值

对8259A、8255A、8253进行初始化

设置抢答按键中断向量

并使IF置1开放中断

关闭LED显示器

检测开始键是否按下

检测开始键是否弹起

点亮绿灯

置标志位值为FFH

使IF=1开放中断

检测复位键是否按下

检测复位键是否弹起

关闭LED显示器、指标灯

将标志位清零，使IF=1开入中断

结束

Y

Y

Y

Y

N

N

N

N

先从82H 口中读一个字

先从82H 口中读一个字

先从82H 口中读一个字

先从82H 口中读一个字

# 2、程序说明

;--------------------------------------------------------------- 8255初始化

INITA:

PUSH AX

PUSH DX

PUSH SI

MOV DX,P8255M ;8255

MOV AL,90H

OUT DX,AL

MOV DX,P8255B

MOV AL,00H

OUT DX,AL

MOV DX,P8255C

MOV AL,0FFH

OUT DX,AL

POP SI

POP DX

POP AX

;--------------------------------------------------------------- 中断芯片

INITB: PUSH DS

MOV AX,0000H

MOV DS,AX

MOV AX,OFFSET MIR7

MOV SI,003CH

MOV [SI],AX

MOV AX,CS

MOV SI,003EH

MOV [SI],AX

MOV AX,OFFSET MIR6

MOV SI,0038H

MOV [SI],AX

MOV AX,CS

MOV SI,003AH

MOV [SI],AX

MOV AX, OFFSET SIR1

MOV SI, 00C4H

MOV [SI], AX

MOV AX, CS

MOV SI, 00C6H

MOV [SI], AX

CLI

POP DS

;--------------------------------------------------------------- 主

MOV AL,11H

OUT 20H,AL ;ICW1

MOV AL,08H

OUT 21H,AL ;ICW2

MOV AL,04H

OUT 21H,AL ;ICW3

MOV AL,01H

OUT 21H,AL ;ICW4

;---------------------------------------------------------------从片8259

MOV AL, 11H

OUT 0A0H, AL ;ICW1

MOV AL, 30H

OUT 0A1H, AL ;ICW2

MOV AL, 02H

OUT 0A1H, AL ;ICW3

MOV AL, 01H

OUT 0A1H, AL ;ICW4

;---------------------------------------------------------------

MOV AL,2BH ;OCW1 = 00101011B

OUT 21H,AL

MOV AL, 0FDH ;OCW1 = 1111 1101B

OUT 0A1H,AL

STI

;AA1: NOP

; JMP AA1

;--------------------------------------------------------------- 定时器芯片

INITC: MOV DX, TIMERM

MOV AL,36H

OUT DX,AL

MOV DX, TIMER0 ;计数器0

MOV AL,0E8H

OUT DX,AL

MOV AL,03H

OUT DX,AL

MOV DX, TIMERM

MOV AL,76H

OUT DX,AL

MOV DX, TIMER1 ;计数器1

MOV AL,0E8H

OUT DX,AL

MOV AL,03H

OUT DX,AL

INITD: NOP

# 六、源程序

1、参考程序

P8255A EQU 0600H

P8255B EQU 0602H

P8255C EQU 0604H

P8255M EQU 0606H

T8254A EQU 06C0H

T8254B EQU 06C2H

T8254C EQU 06C4H

T8254M EQU 06C6H

DATA SEGMENT

LEDCODE DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H

DB 7FH,6FH,77H,7CH,39H,5EH,79H,71H

LCODE DB ? ;选手灯代码

DCODE DB 3FH ;显示代码缓冲

DATA ENDS

SSTACK SEGMENT STACK

DW 256 DUP(?)

SSTACK ENDS

CODE SEGMENT

ASSUME CS:CODE, SS:SSTACK, DS:DATA

START: MOV AX,DATA

MOV DS,AX

MAIN: MOV DX,P8255M

MOV AL,XXH ;写8255控制字。

OUT DX,AL ;PA输入、PB输出、PC输出。MODE0

MOV DX,P8255B

MOV AL,00H ;

OUT DX,AL

MOV DX,P8255C

OUT DX,AL

;------------------------------------------------------------------------

NEW: MOV DX,P8255A

IN AL,DX

TEST AL,80H

JNZ SECOND

FIRST: MOV DX,P8255C ;K7=0

MOV AL,00H ;

OUT DX,AL

JMP NEW

;---------------------------------------=-----------------------------

SECOND: NOP

MOV DX,P8255A

IN AL, DX

TEST AL,7FH ;K0---K6=全0?

JNZ THIRD

MOV DX,P8255M

MOV AL,0EH

OUT DX,AL ;PC7=0

MOV AL,0DH ;PC6=1

OUT DX,AL

JMP NEW

THIRD: NOP ; K0---K6=非全0

MOV DX,P8255M

MOV AL,0FH ;PC7=1

OUT DX,AL

MOV AL,0CH ;PC6=0

OUT DX,AL

;------------------------------------------------------------------

FORTH: NOP

FIFTH: NOP

CALL SWITCH

DONE: NOP

CALL LEDLAMP

;CALL DISPLY

JMP NEW

;------------------------------------------------------------------------

SWITCH PROC NEAR ;

PUSH AX

PUSH DX

K0: NOP

K1: NOP

K2: NOP

K3: NOP

K4: NOP

K5: NOP

POP DX

POP AX

RET

SWITCH ENDP

LEDLAMP PROC NEAR ;

PUSH AX

PUSH DX

NOP

POP DX

POP AX

RET

LEDLAMP ENDP

DISPLY PROC NEAR ;

PUSH AX

PUSH DX

NOP

POP DX

POP AX

RET

DISPLY ENDP

CODE ENDS

END START

2、最终程序清单

P8255A EQU 0600H

P8255B EQU 0602H

P8255C EQU 0604H

P8255M EQU 0606H

TIMER0 EQU 06C0H

TIMER1 EQU 06C2H

TIMER2 EQU 06C4H

TIMERM EQU 06C6H

DATA SEGMENT

DISCODE DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H

DB 7FH,6FH,77H,7CH,39H,5EH,79H,71H

STATUS DB 00H ;程序状态

LEDCODE DB ? ;选手灯代码

DIGBUFF DB 3FH ;数码管显示代码

TIME DW 10000

KK DB 0

DATA ENDS

SSTACK SEGMENT PARA STACK 'STACK'

DW 32 DUP(?)

SSTACK ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA,SS: SSTACK

START: MOV AX,DATA

MOV DS,AX

;------------------------------------------------------------- 8255初始化

INITA:

PUSH AX

PUSH DX

PUSH SI

MOV DX,P8255M ;8255

MOV AL,90H

OUT DX,AL

MOV DX,P8255B

MOV AL,00H

OUT DX,AL

MOV DX,P8255C

MOV AL,0FFH

OUT DX,AL

POP SI

POP DX

POP AX

;--------------------------------------------------------------- 中断芯片

INITB: PUSH DS

MOV AX,0000H

MOV DS,AX

MOV AX,OFFSET MIR7

MOV SI,003CH

MOV [SI],AX

MOV AX,CS

MOV SI,003EH

MOV [SI],AX

MOV AX,OFFSET MIR6

MOV SI,0038H

MOV [SI],AX

MOV AX,CS

MOV SI,003AH

MOV [SI],AX

MOV AX, OFFSET SIR1

MOV SI, 00C4H

MOV [SI], AX

MOV AX, CS

MOV SI, 00C6H

MOV [SI], AX

CLI

POP DS

;------------------------------------------------------------- 主

MOV AL,11H

OUT 20H,AL ;ICW1

MOV AL,08H

OUT 21H,AL ;ICW2

MOV AL,04H

OUT 21H,AL ;ICW3

MOV AL,01H

OUT 21H,AL ;ICW4

;---------------------------------------------------------------从片8259

MOV AL, 11H

OUT 0A0H, AL ;ICW1

MOV AL, 30H

OUT 0A1H, AL ;ICW2

MOV AL, 02H

OUT 0A1H, AL ;ICW3

MOV AL, 01H

OUT 0A1H, AL ;ICW4

;---------------------------------------------------------------

MOV AL,2BH ;OCW1 = 00101011B

OUT 21H,AL

MOV AL, 0FDH ;OCW1 = 1111 1101B

OUT 0A1H,AL

STI

;AA1: NOP

; JMP AA1

;--------------------------------------------------------------- 定时器芯片

INITC: MOV DX, TIMERM

MOV AL,36H

OUT DX,AL

MOV DX, TIMER0 ;计数器0

MOV AL,0E8H

OUT DX,AL

MOV AL,03H

OUT DX,AL

MOV DX, TIMERM

MOV AL,76H

OUT DX,AL

MOV DX, TIMER1 ;计数器1

MOV AL,0E8H

OUT DX,AL

MOV AL,03H

OUT DX,AL

INITD: NOP

;---------------------------------------------------------------

BEGIN: NOP

AA2: JMP BEGIN

;---------------------------------------------------------------

MIR7 PROC NEAR

PUSH AX

PUSH DX

;MOV AX,0137H

;INT 10H

;MOV AX,0120H

;INT 10H

MOV AL,STATUS

CMP AL,01H

JNE DONE

DEC TIME

MOV BX,TIME

CMP BX,0

JNE DONE

MOV STATUS,00000011B

MOV DX,P8255C

MOV AL,10000000B

OUT DX,AL

DONE: MOV AL,20H

OUT 20H,AL

POP DX

POP AX

IRET

MIR7 ENDP

;---------------------------------------------------------------

LINKT PROC

PUSH DX

PUSH AX

MOV DX,P8255C

MOV AL,40H

OUT DX,AL

MOV STATUS,01H

POP AX

POP DX

RET

LINKT ENDP

;---------------------------------------------------------------

MIR6 PROC NEAR

PUSH AX

PUSH DX

MOV AX,0136H

INT 10H

MOV AX,0120H

INT 10H

CALL FIRST

MOV AL,20H

OUT 20H,AL

POP DX

POP AX

IRET

MIR6 ENDP

;------------------

FIRST PROC

PUSH AX

PUSH DX

PUSH BX

MOV AL,STATUS

CMP AL,01H

JNZ DONE1

MOV DX,P8255A

IN AL,DX

TEST AL,01H

JNZ K1

TEST AL,02H

JNZ K2

TEST AL,04H

JNZ K3

TEST AL,08H

JNZ K4

TEST AL,10H

JNZ K5

TEST AL,20H

JNZ K6

JMP DONE1

K1:

MOV AL,06H

MOV BL,11000001B

JMP QUIT

K2:

MOV AL,5BH

MOV BL,11000010B

JMP QUIT

K3:

MOV AL,4FH

MOV BL,11000100B

JMP QUIT

K4:

MOV AL,66H

MOV BL,11001000B

JMP QUIT

K5:

MOV AL,6DH

MOV BL,11010000B

JMP QUIT

K6:

MOV AL,7DH

MOV BL,11100000B

JMP QUIT

QUIT:

MOV DX,P8255B

OUT DX,AL

MOV DX,P8255C

MOV AL,BL

OUT DX,AL

MOV STATUS,00000011B

JMP DONE1

DONE1:

POP AX

POP BX

POP DX

RET

POP BX

POP DX

POP AX

IRET

FIRST ENDP

;------------------

SIR1 PROC NEAR

PUSH AX

PUSH DX

MOV AX,0131H

INT 10H

MOV AX,0120H

INT 10H

MOV AL,KK

CMP AL,0

JE RE

MOV KK,0

CALL LINKT

JMP DONE2

RE: MOV KK,1

CALL RESET

JMP DONE2

DONE2: MOV AL, 20H

OUT 0A0H, AL

OUT 20H, AL

POP DX

POP AX

IRET

SIR1 ENDP

RESET PROC NEAR

PUSH DX

PUSH AX

MOV LEDCODE,00H

MOV DX,P8255B

MOV AL,00H

OUT DX,AL

MOV DX,P8255C

MOV STATUS,00H

MOV TIME,10000

OUT DX,AL

POP AX

POP DX

RET

RESET ENDP

CODE ENDS

END START

# 七、调试过程

（1）、硬件测试都顺利通过。8254能够实现呼吸灯。8255能够控制L0—L7的亮与灭。8259能够实现中断。硬件连线是参照以前做的实验的接线图连的，硬件连线测试很快顺利的通过了。在后期的整体连线中，因为粗心接错线了，检查过程中都改过来了。

（2）、进行抢答设计。使用实验仪的逻辑电平开关K0-K6来代表6个抢答按钮。当开关向上拨时表示按下按钮。收到第1个抢答信号后，主持人侧红色发光二极管亮（同时，绿色发光二极管灭）作抢答指示。

（3）、显示选手灯，在老师的提醒下，用了一个转换单元，经过代码改进，测试，实现了在单个LED灯上显示抢先一组的组别。

（4）、在数码管上显示选手编号，开始编号一直按8，4，2，1的方式显示，并不是预期的结果，经过查阅资料，最终实现了按选手的编号显示结果。

（5）、同时考虑抢答允许时间,10秒内有效。10秒之后无人抢答，本次抢答作废。在这个问题上，应该是此次实验遇到的最大问题，既要实现超时后跳转到复位功能模块，同时还要满足有人抢答了，计时器立即停止，并保存选手的编号，从数码管上显示出来。经过与同学讨论，以及初步设想，重新编写了代码，在实验平台上的运行，多次的调试、改进、查阅资料等等，最终解决了这一大难题。

# 八、总结

通过本次实验:选择的芯片8259、8255和8254处理竞赛抢答器问题,用8259芯片，实现中断控制。结合8254实现定时中断，使用8255芯片作为接口。在Masm for Windows 集成实验环境下，编写汇编程序，成功采用硬件中断实现了竞赛抢答器。

本次实验让我掌握了可编程并行接口8255、可编程中断控制器8259、可编程定时/计数器8254三个芯片的基本使用方法，让我对微型计算机原理与接口技术有了更深的理解。

# 九、参考文献

[1]吴宁,乔亚男.微型计算机与汇编技术.北京:清华大学出版社,2016四版

[2] 马静.微机原理与接口技术

<https://www.icourse163.org/spoc/course/XATU1001957019?tid=1003424007&_trace_c_p_k2_=aff3c0f4d1534042aa31f039027f67db>,2018