**西安电子科技大学**

**微机原理与系统 课程实验报告**

**实验名称 微机原理与系统设计**

计算机 学院 1503012 班

成绩

姓名秦龙 学号15030120044

同作者秦龙

实验日期 2017 年 11 月25 日晚

实验地点**E-II-311**实验批次**第4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） |

汇编语言上机器实验（一）

题目：从键盘上输入一串字符，当输入字符是回车时,显示出输入的所有字符(必做)

DATAS SEGMENT

hello DB 100,?,100 dup('$')

DATAS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS

START:

MOV AX,DATAS

MOV DS,AX

LEA DX,hello

mov ah,10

int 21h

mov dl,0ah

mov ah,2

int 21h

mov dl,0dh

mov ah,2

int 21h

lea dx,hello[2]

MOV AH,9

INT 21H

CALL CRLF

JMP START

crlf proc near

mov dl,0dh

mov ah,02h

int 21h

mov dl,0ah

mov ah,02h

int 21h

ret

crlf endp

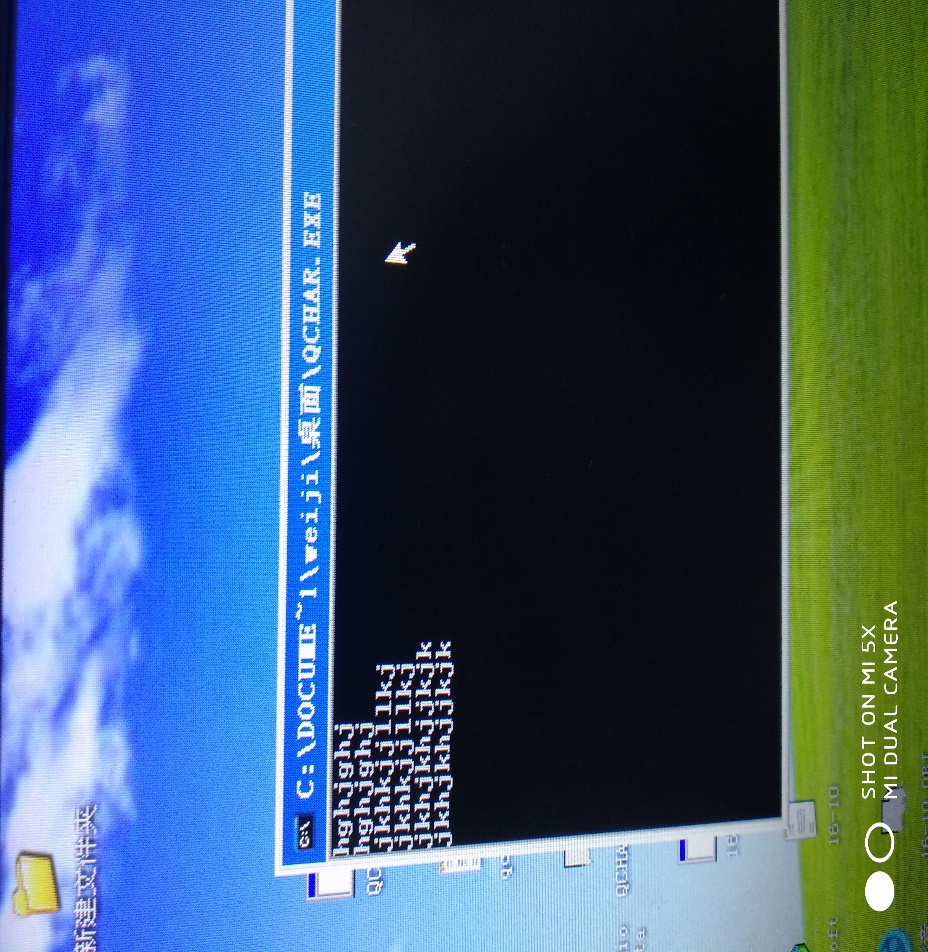
MOV AH,4CH

INT 21H

CODES ENDS

END START

实验结果：



汇编语言上机器实验（二）

题目：键盘输入十六进制数转化为十进制输出

assume cs:code

code segment

start:

call readHex

call print

call crlf

jmp start

mov ah,4ch

int 21h

print proc near

xor cx, cx

mov bx, 10

Q0:

xor dx, dx

div bx

xor dx, 0e30H

push dx

inc cx

cmp ax, 0

jnz Q0

Q1:

pop ax

int 10h

loop Q1

ret

print endp

crlf proc near

mov dl,0dh

mov ah,02h

int 21h

mov dl,0ah

mov ah,02h

int 21h

ret

crlf endp

;read a Hex number (16bit)

;输入数据存储在ax中返回

;不做输入数据有效性检查

readHex proc near

mov cx,0404h

mov bx,0

mov ah,1

L0:

int 21h

cmp al,'F'

ja L1

cmp al,'9'

ja L2

jmp L3

L1:

sub al,20H

L2:

sub al,7H

L3:

sub al,30H

shl bx,cl

or bl,al

dec ch

jnz L0

;

mov ah,0eh

mov al,0dh

int 10h

mov al,0ah

int 10h

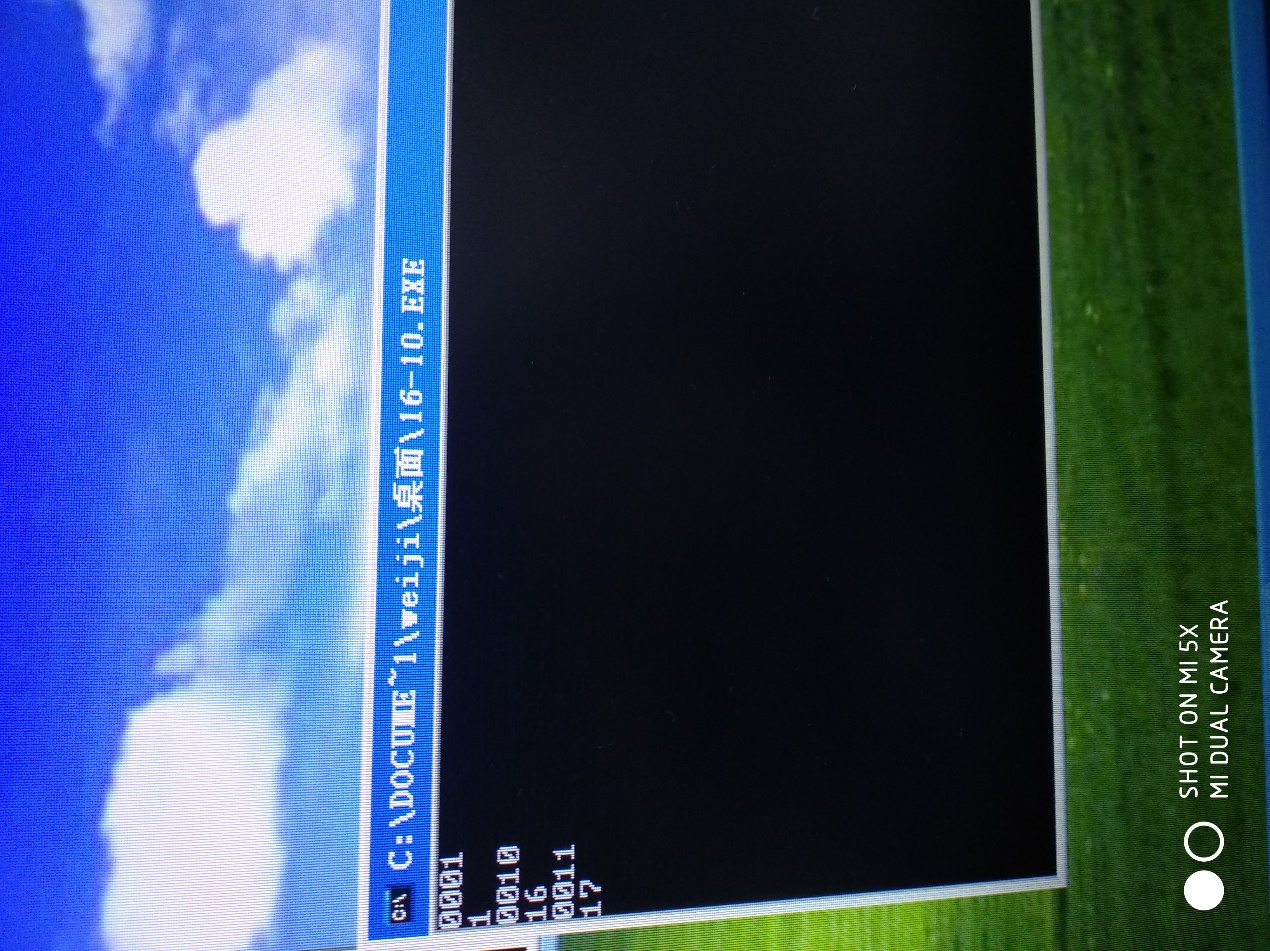
mov ax, bx

ret

readHex endp

code ends

end start



实验三 基本IO口扩展实验

**一、实验目的**

1. 了解TTL芯片扩展简单I/O口的方法。

2. 掌握数据输入输出程序编制的方法。

**二、实验内容说明**

本实验要求用74LS244作为输入口，读取开关状态，并将此状态通过74LS273连到发光二极管显示。具体实验内容如下：

1.开关Yi为低电平时对应的发光二极管亮，Yi为高电平时对应的发光二极管灭。

2.当开关Yi全为高电平时，发光二极管Qi从左至右轮流点亮。

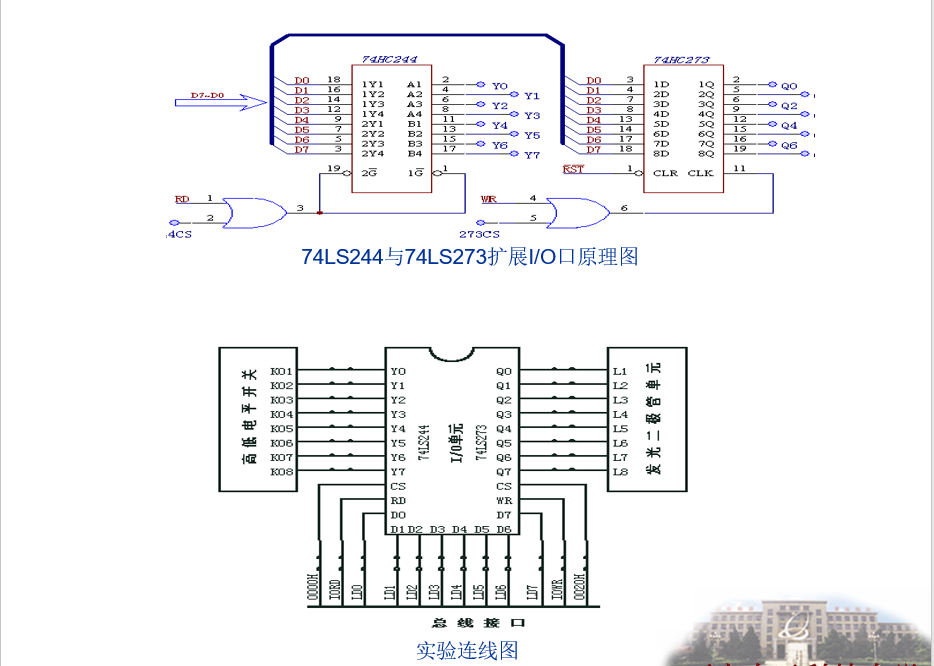
3.当开关Yi全为低电平时，发光二极管Qi从右至左轮流点亮。

4.自主设计控制及显示模式，完成编程调试，演示实验结果。

**三、实验原理**

**74LS244**是一种三态输出的8总线缓冲驱动器，无锁存功能，当G为低电平，Ai信号传送到Yi，当为高电平时，Yi处于禁止高阻状态;

**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。



**四、实验步骤**

1.按照实验连线图连接：

244的CS接到ISA总线接口模块的0000H，Y7—Y0——开关K1—K8。

273的CS接到ISA总线接口模块的0020H，Q7—Q0——发光二极管L1—L8。

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）连到ISA总线接口模块的数据（LD0～LD7）。

2.编写实验程序，编译链接，运行程序

3. 拨动开关，观察发光二极管的变化。

程序代码：

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,'$'

LS244 DW 00000H

LS273 DW 00020H

RA DB ?

LB DB ?

DELAY\_SET EQU 0FFFH ;延时常数

MY\_DATA ENDs

MY\_CODE SEGMENT PARA 'CODE'

MY\_PROC PROC FAR

ASSUME CS:MY\_CODE, DS:MY\_DATA, SS:MY\_STACK

MAIN:

.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 LS244,CX ;PCI卡IO基址+偏移

ADD LS273,CX

MOV RA,7FH

MOV LB,0FEH

READ1: MOV DX,LS244 ;读取开关状态

IN AL,DX

CMP AL,55H ;如果是55右移

JE READ2

CMP AL,0AAH ;如果是AA左移

JE READ3

NOT AL ;取反

JMP READ4

READ2: CALL RIGHT

JMP READ4

READ3: CALL LEFT

READ4: MOV DX,LS273

OUT DX,AL ;送LED显示

CALL DELAY

CALL BREAK

JMP READ1

MY\_PROC ENDp

;

RIGHT PROC NEAR

MOV AL,RA

ROR AL,1

MOV RA,AL

RET

RIGHT ENDP

LEFT PROC NEAR

MOV AL,LB

ROL AL,1

MOV LB,AL

RET

LEFT ENDP

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; /\*按任意键退出\*/

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

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

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; /\*延时程序\*/

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

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

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; /\* 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

实验四 可编程并行接口实验

一、实验目的

1. 了解可编程并行接口8255的内部结构，

2. 掌握工作方式、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。

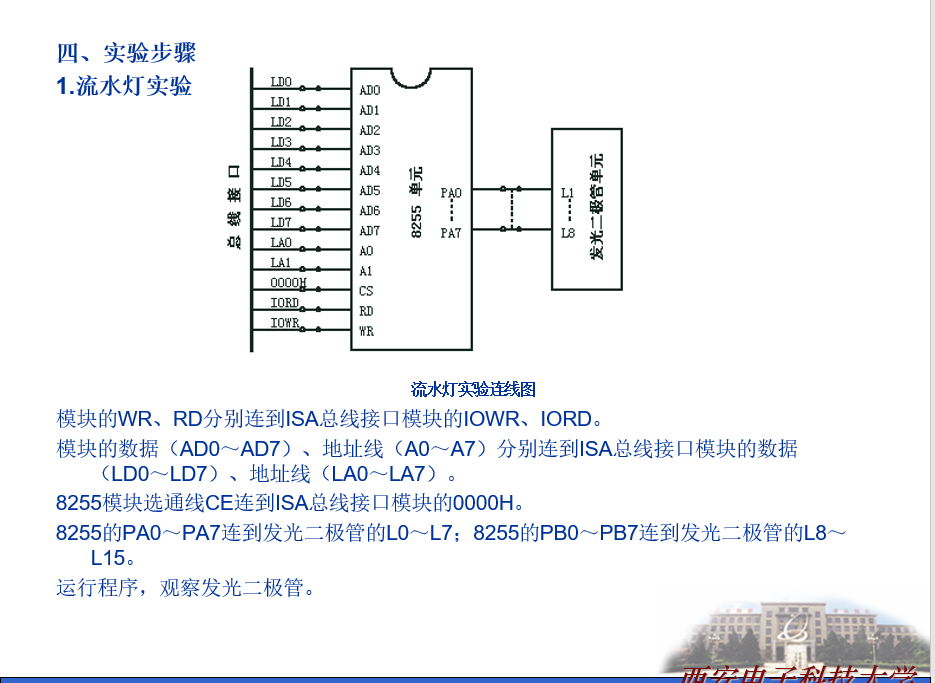
2.交通灯实验：利用8255的A口模拟交通信号灯。

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。

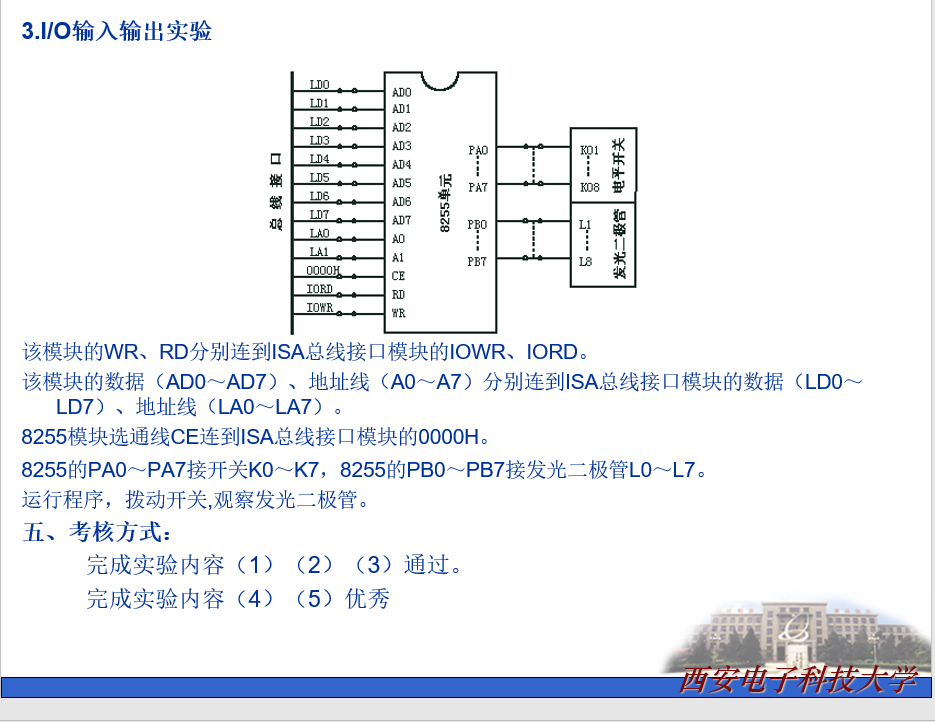
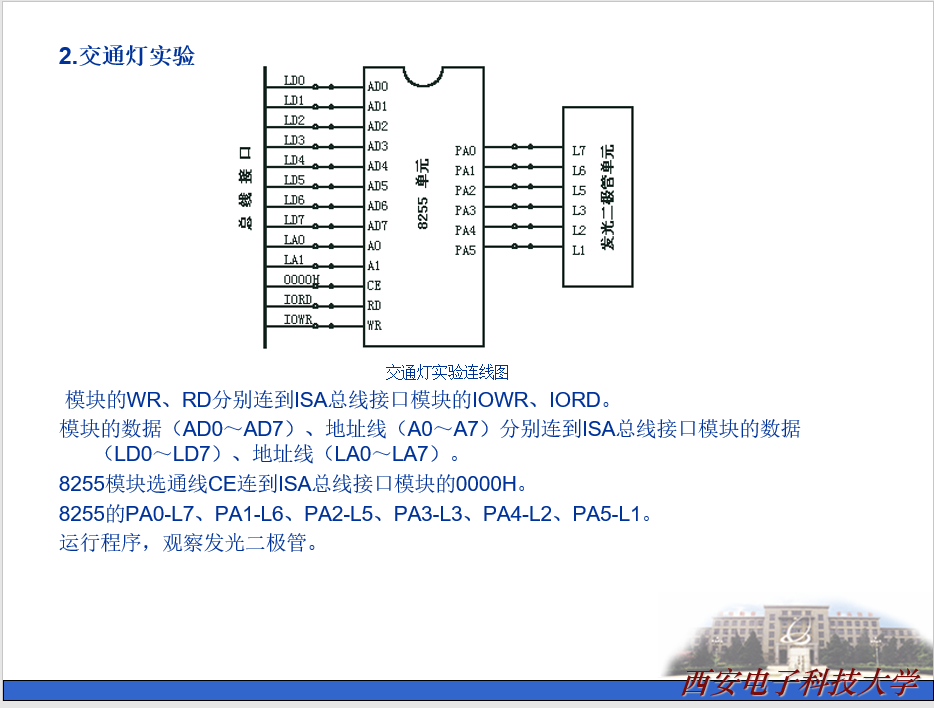
4.在完成(1)基础上，增加通过读取开关控制流水灯的循环方向和循环方式。

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、实验原理

8255是一个通用可编程并行接口电路。它具有A、B、C三个8位并行口。其中C口也可用作A、B口的联络信号及中断申请信号。通过编程，它可以被设置为基本输入输出、选通输入输出以及双向传送方式。对于C口还具有按位置

0、1的功能。



程序代码：

;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

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

MES4 DB 10,13,'The data error !Please enter again !' ,10,13,'$'

MES3 DB 10,13,'Please enter 0~9 to Play !' ,10,13

DB 'ENTER CR TO RETURN !',0DH,0AH,'$'

MES2 DB ' PCI CONFIG READ ERROR! $'

DDATA DB ?

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

st1: MOV DX,offset good ;显示开始执行程序信息

MOV AH,09H

INT 21H

CALL FINDPCI ;查找PCI卡资源并显示

MOV CX,word ptr IO\_base\_address

ADD P8255\_A,CX

ADD P8255\_B,CX

ADD P8255\_C,CX

ADD P8255\_MODE,CX

SSS:

MOV DX,P8255\_MODE

MOV AL,80H ;写8255控制字A、B输出，C输入

OUT DX,AL

mov al,0ffh

qq: MOV DX,P8255\_a

OUT DX,AL

MOV DX,P8255\_b

OUT DX,AL

MOV DX,P8255\_c

OUT DX,AL

; in al,dx

inc al

jmp qq

START1: MOV DX,OFFSET MES3

MOV AH,09H

INT 21H

ERROR: MOV DX,OFFSET MES2 ;显示读错误信息

MOV AH,09H

INT 21H

EXIT: MOV AX,4C00H

INT 21H

MY\_PROC ENDp

DELAY20 PROC NEAR ;延时程序

PUSHF

PUSH AX

PUSH DX

PUSH CX

MOV AH,86H

MOV CX,00

MOV DX,20

INT 15H

POP CX

POP DX

POP AX

POPF

RET

DELAY20 ENDp

DELAY500 PROC NEAR ;延时程序

PUSHF

PUSH AX

PUSH DX

PUSH CX

MOV AH,86H

MOV CX,07h

MOV DX,0d0H

INT 15H

POP CX

POP DX

POP AX

POPF

RET

DELAY500 ENDp

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

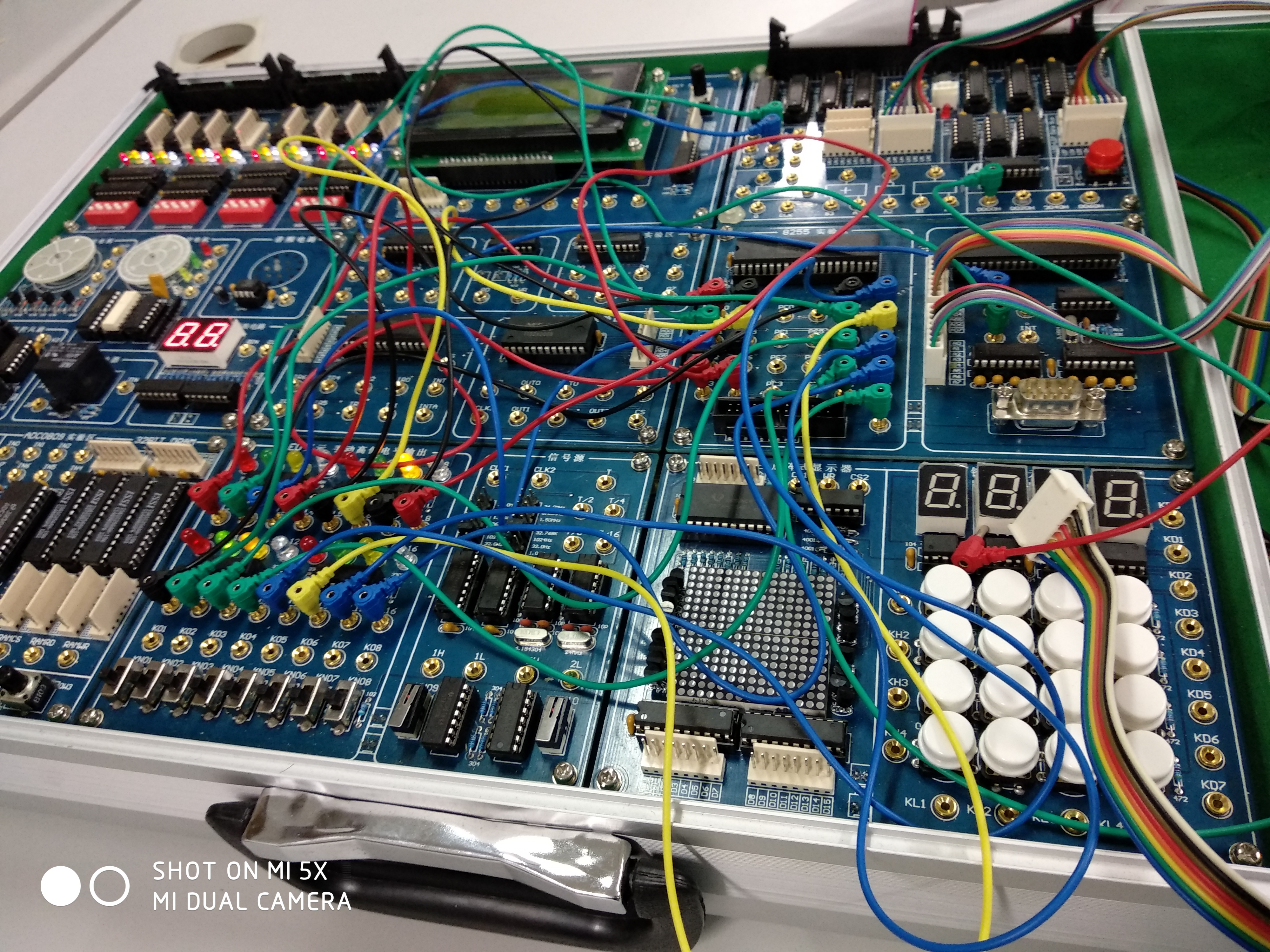
POPad

RET

findPCI ENDP

MY\_CODE ENDS

END START



实验心得：

此次微机原理与系统实验非常提高学生的综合素质，才开始对于汇编语言编程设计还是很陌生的，通过几次的试验下来对于汇编程序有了很大的进步。