西安电子科技大学

微型计算机原理及接口技术 课程实验报告

实验名称 微型计算机原理及接口技术

计算机科学与技术学院 1903015 班

实验报告成绩

姓名 吕思勤 学号 19030130448

实验批次 **第六批**

实验开始日期 2021 年 11 月 27 日

实验室门号 **EII 312**

|  |
| --- |
| 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） |

实验一 汇编语言编程实验

一、实验目的

1. 掌握汇编语言的编程方法；

2. 掌握DOS功能调用的使用方法；

3. 掌握汇编语言程序的调试运行过程。

二、实验内容

1.在屏幕上显示自己的学号姓名信息。

2.将指定数据区的字符串数据以ASCII码形式显示在屏幕上，并通过DOS功能调用完成必要提示信息的显示。

3. 循环从键盘读入字符并回显在屏幕上，然后显示出对应字符的ASCII码，直到输入”Q”或“q”时结束。

4. 自主设计输入显示信息，完成编程与调试，演示实验结果。

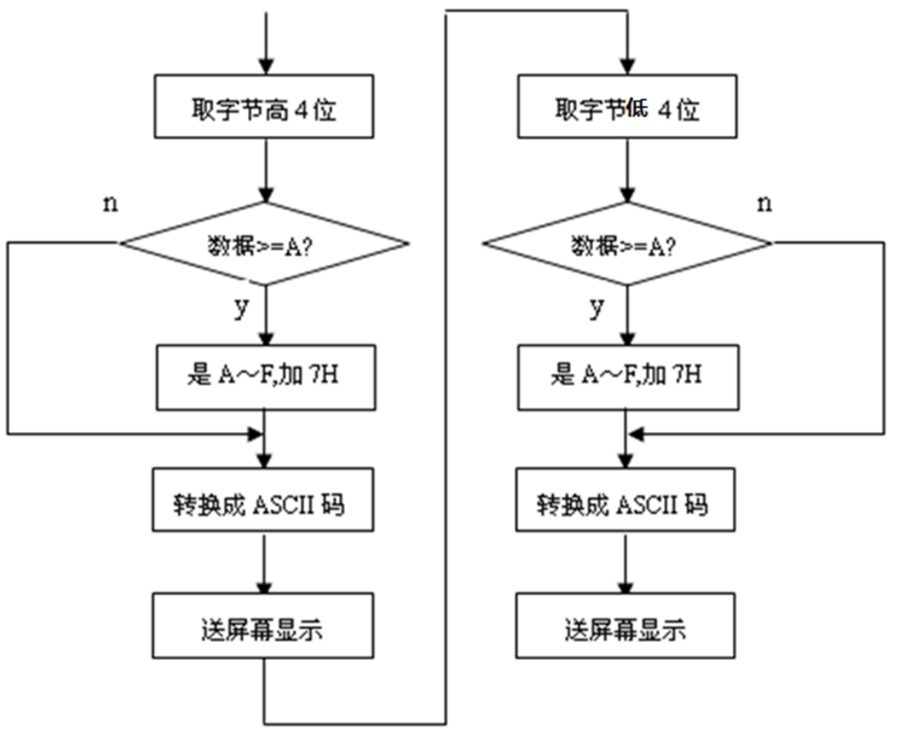
三、实验步骤

1. 运行星研软件，根据实验内容，参考程序流程图编写程序。

2. 选择“项目”菜单中的 “编译连接”对实验程序进行编译连接。

3.使用“运行”菜单中的“进行调试”命令进入调试状态，执行单步调试或全速运行，观察调试过程中寄存器窗、观察光和变量窗等信息窗内各寄存器及数据区的内容，在信息窗的DOS窗口可显示和输入相应信息和数据。

字符转换为ASCII码流程图



四、实验程序

MYSTACK SEGMENT STACK

DB 10 DUP (?)

MYSTACK ENDS

DATA SEGMENT

BUFFER1 DB 'The ASCII of the sting "ZXCVBNM" is:','$'

STRING DB 'ZXCVBNM','$'

BUFFER2 DB 0AH,0DH,'$'

BUFFER3 DB 'LVSIQIN19030130448','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV DX,OFFSET BUFFER1;

MOV AH,09H

INT 21H

MOV SI,OFFSET STRING;

PGM1:

CMP BYTE PTR [SI],'$'

JE PGM3;

MOV AL,DS:[SI]

AND AL,0F0H;

MOV CL,4

SHR AL,CL;

CMP AL,0AH;

JB C2;

ADD AL,07H

C2:

ADD AL,30H

MOV DL,AL;

MOV AH,02H

INT 21H

MOV AL,DS:[SI]

AND AL,0FH;

CMP AL,0AH

JB C3

ADD AL,07H

C3:

ADD AL,30H

MOV DL,AL;

MOV AH,02H

INT 21H

INC SI

JMP PGM1

PGM2:

MOV AH,09H

MOV DX,OFFSET BUFFER2

INT 21H

MOV AH,01H

MOV AL,00H

INT 21H

CMP AL,'q'

JE PGM4

CMP AL,'Q'

JE PGM4

MOV BL,AL

AND AL,0F0H

MOV CL,4

SHR AL,CL;

CMP AL,0AH;

JB C4;

ADD AL,07H

C4:

ADD AL,30H

MOV DL,AL;

MOV AH,02H

INT 21H

MOV AL,BL

AND AL,0FH;

CMP AL,0AH

JB C5

ADD AL,07H

C5:

ADD AL,30H

MOV DL,AL;

MOV AH,02H

INT 21H

INC SI

JMP PGM2

PGM3:

MOV AH,09H

MOV DX,OFFSET BUFFER2

INT 21H

MOV DX,OFFSET BUFFER3

MOV AH,09H

INT 21H

JMP PGM2

PGM4:

MOV AH,4CH

INT 21H

CODE ENDS

END START

实验二 数码转换实验

一、实验目的

1. 掌握不同进制数及编码相互转换的程序设计方法。

2. 掌握运算类指令编程及调试方法。

3. 掌握循环程序的设计方法。

二、实验内容

1. 重复从键盘输入不超过5位的十进制数，按回车键结束输入；

2. 将该十进制数转换成二进制数；结果以2进制数的形式显示在屏幕上；

3. 如果输入非数字字符，则报告出错信息，重新输入；

4. 直到输入“Q”或‘q’时程序运行结束。

5. 键盘输入一字符串，以空格结束，统计其中数字字符的个数，在屏幕显示

三、实验原理

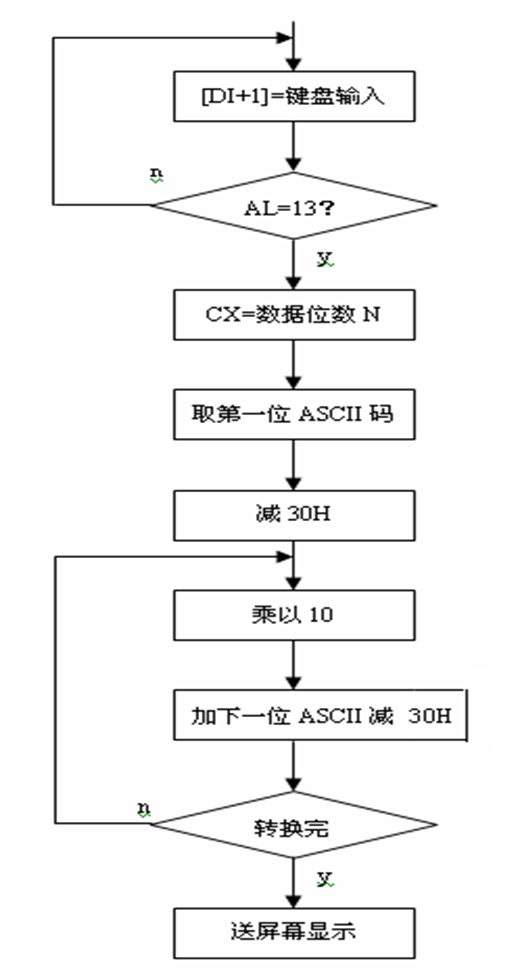
十进制数可以表示为：Dn\*10n+Dn-1\*10n-1+…+D0\*100=Di\*10i

其中Di代表十进制数1、2、3、…、9、0。

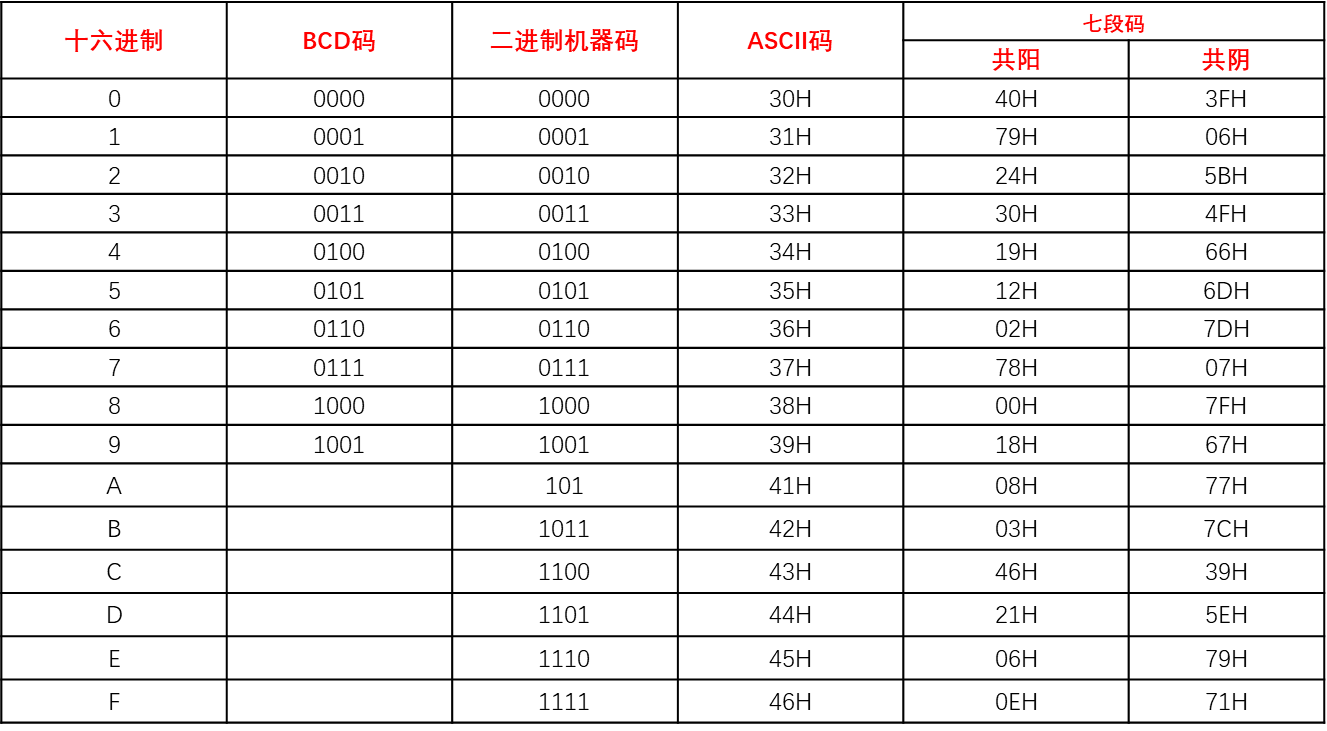
上式可以转换为：Di\*10i=（（（Dn\*10+Dn-1）\*10+ Dn-2）\*10+…+ D1）\*10+ D0

由上式可归纳出十进制数转换为二进制数的方法：从十进制数的最高位Dn开始做乘10加次位的操作，依此类推，则可求出二进制数结果。

十进制ASCII码转换为二进制数流程图



数码转换对应关系表



四、实验程序

MYSTACK SEGMENT STACK

DB 10 DUP (0)

MYSTACK ENDS

DATA SEGMENT

PROMPT DB 0DH,0AH,'Input no more than 5 digits(Less than 65536, press Q/q to quit):',0DH,0AH,'$'

ERROR DB 0DH,0AH,"Error : Illegal input! please input a number",0DH,0AH,'$'

ENT DB 0DH,0AH,'$'

DATA ENDS

CODE SEGMENT

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

START:

MOV AX,DATA

MOV DS,AX

MOV AX,0

MOV BX,0

MOV DX,0

LEA DX,PROMPT

MOV AH,09H

INT 21H

INPUT:

MOV AH,01H

MOV AL,00H

INT 21H

CMP AL,'Q'

JE EXIT

CMP AL,'q'

JE EXIT

CMP AL,00H;

JE CHANGE

CMP AL,30H;

JL ERRORS

CMP AL,39H

JG ERRORS

SUB AL,30H;BCD

PUSH AX;

MOV AX,BX

MOV DX,0AH

MUL DX;

MOV BX,AX

POP AX

ADD BL,AL

CMP BX,0H

JE C1

JMP INPUT

ERRORS:

LEA DX,ERROR

MOV AH,09H

INT 21H

MOV BX,0H;

JMP INPUT

EXIT:

MOV AH,4CH

INT 21H

CHANGE:

CMP BX,0H

JE ERRORS

C1:

MOV CL,15

PUSH DX;

PUSH AX;

LEA DX,ENT

MOV AH,09H

INT 21H

POP AX

POP DX

LOOPS:

MOV DX,BX

CMP CL,0

SHR DX,CL

AND DX,0001H

ADD DX,30H

MOV AH,02H

INT 21H

DEC CL

CMP CL,0

JNS LOOPS;

PUSH DX

PUSH AX

LEA DX,ENT

MOV AH,09H

INT 21H

POP AX

POP DX

JMP INPUT

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.当开关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。

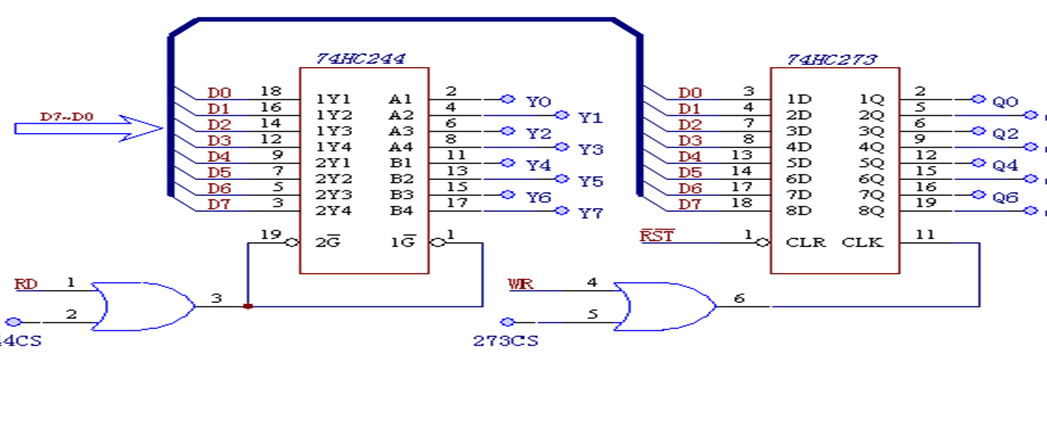
3.当开关Yi全为低电平时，发光二极管Qi从右至左轮流点亮。

4.自主设计控制及显示模式，完成编程调试，演示实验结果。

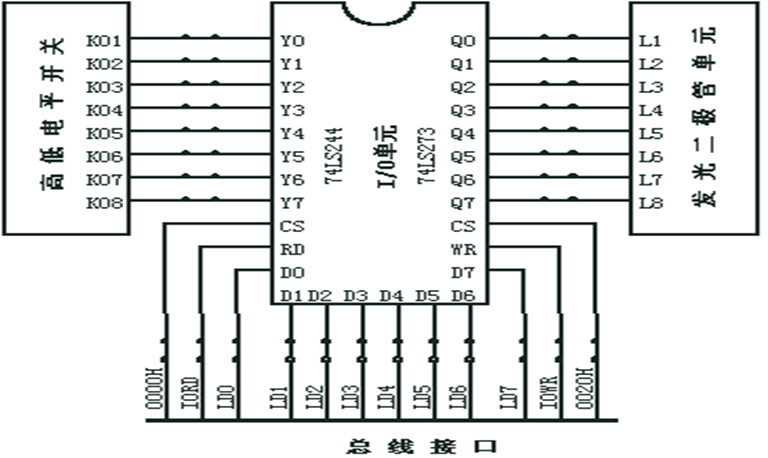
三、实验原理

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。



74LS244与74LS273扩展I/O口原理图



实验连线图

四、实验步骤

1. 按照实验连线图连接：

B4模块244的CS244——总线模块A3的CS1，输入数据信号IN0—IN7（JP52）——开关（JP27）K1—K8；

B4模块273的CS273——总线模块A3的CS2，输出数据信号O0—O7（JP51）——发光二极管（JP18）L1—L8；

B4模块的WR、RD分别连到总线模块A3的IOW、IOR；

B4模块的数据信号D0—D7（JP57）连到总线模块的数据信号D0—D7（JP42）。

2. 编写实验程序，编译连接，运行程序；

3. 拨动开关，观察发光二极管的变化。

五、实验程序

IO244 EQU 0230H

IO273 EQU 0230H

\_STACK SEGMENT STACK

DW 100 DUP(?)

\_STACK ENDS

\_DATA SEGMENT WORD PUBLIC 'DATA'

\_DATA ENDS

CODE SEGMENT

START PROC NEAR

ASSUME CS:CODE, DS:\_DATA, SS:\_STACK

MOV AX,\_DATA

MOV DS,AX

MOV DX, IO244

IN AX, DX

MOV DX, IO273

OUT DX, AX

CALL Delay

CALL Delay

CALL Delay

CMP AL,0FFH

JZ Func1

CMP AL,00H

JZ Func2

JMP START

Func1: CALL LEFT

JMP START

Func2: CALL RIGHT

JMP START

Delay PROC NEAR

Delay1: XOR CX,CX

LOOP $

RET

Delay ENDP

RIGHT PROC NEAR

MOV AX,0FFFEH

LP1: OUT DX,AX

CALL Delay

TEST AX,8000H

JZ BAC1

ROL AX,1

JMP LP1

BAC1:

RET

RIGHT ENDP

LEFT PROC NEAR

MOV AX,07FFFH

LP2: OUT DX,AX

CALL Delay

TEST AX,0001H

JZ BAC2

ROR AX,1

JMP LP2

BAC2:

RET

LEFT ENDP

START ENDP

CODE ENDS

END START

实验四 8255可编程并行接口实验

一、实验目的

1.了解8255 芯片工作原理，掌握其初始化、工作方式及应用。

2.编程使用8255 芯片实现简单的输入输出控制。

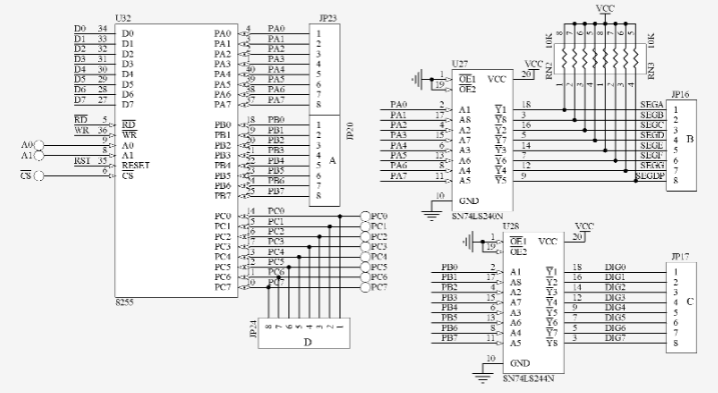
二、实验内容

1.利用8255 的A口读取拨动开关状态，利用8255 的B口将对应位状态用发光二极管显示。拨动开关为低电平时，对应的发光二极管亮；拨动开关为高电平时，对应的发光二极管灭。

2.利用8255 的A口读取拨动开关状态。拨动开关为全低电平时，发光二 极管从左至右流水灯点亮；拨动开关为全高电平时，发光二极管从右至左流水 灯点亮。

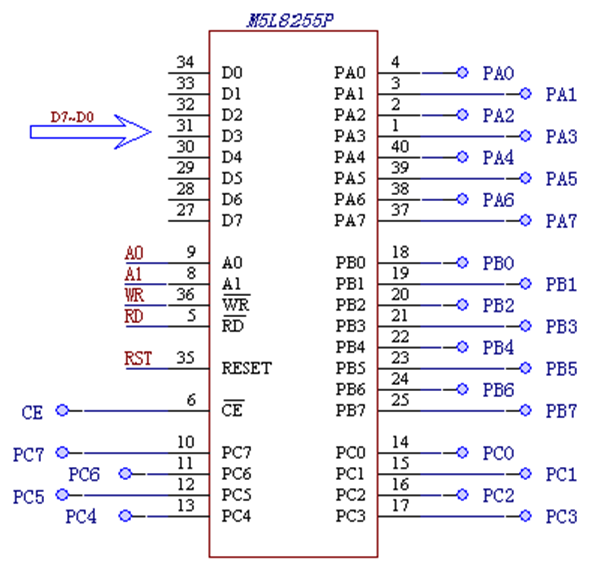
3.自主发挥设计控制及显示模式。

三、实验原理

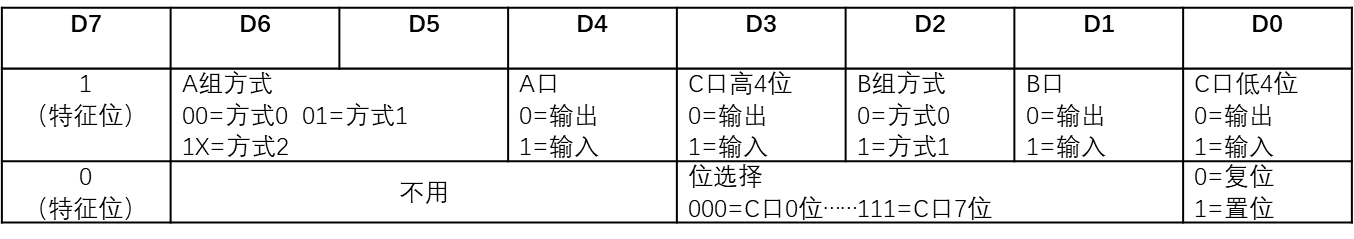


8255是一个通用可编程并行接口电路。它具有A、B、C三个8位并行口。其中C口也可用作A、B口的联络信号及中断申请信号。通过编程，它可以被设置为基本输入输出、选通输入输出以及双向传送方式。对于C口还具有按位置0、1的功能。

可编程并行接口8255芯片接口电路



8255控制字

  
四、实验步骤

1.硬件连接：连接片选；连接8255 端口选择信号；连接相关数据信号。

2.编程调试：编写源程序；运行调试。

五、实验程序

1. 基本输入输出实验

IOY0 EQU 0C000H ;8255的CS引脚连接IOY0对应的端口始地址

MY8255\_A EQU IOY0+00H\*2 ; ;8255的端口A地址

MY8255\_B EQU IOY0+01H\*2 ;8255的端口B地址

MY8255\_C EQU IOY0+02H\*2 ;8255的端口C地址

MY8255\_MODE EQU IOY0+03H\*2 ;8255的控制寄存器地址

CODE SEGMENT ;定义代码段

ASSUME CS:CODE

START: MOV DX,MY8255\_MODE ;8255控制端口地址为3006H

MOV AL,82H ;设置8255方式控制字,A、B口工作方式0，A口输出,B口输入

OUT DX,AL

LOOP1: MOV DX,MY8255\_B ;读B口发出的起始数据，B口输入

IN AL,DX

MOV DX,MY8255\_A ;写A口发出的起始数据，A口输出

OUT DX,AL

MOV AH,1 ;判断是否有按键按下

INT 16H

JZ LOOP1 ;无按键则跳回继续循环，有则退出

QUIT: MOV AX,4C00H ;结束程序退出

INT 21H

CODE ENDS

END START

2. 流水灯显示实验

IOY0 EQU 0C000H ;8255的CS引脚连接IOY0对应的端口始地址

MY8255\_A EQU IOY0+00H\*2 ;8255的端口A地址

MY8255\_B EQU IOY0+01H\*2 ;8255的端口B地址

MY8255\_C EQU IOY0+02H\*2 ;8255的端口C地址

MY8255\_MODE EQU IOY0+03H\*2 ;8255的控制寄存器地址

STACK1 SEGMENT STACK ;定义堆栈段

DW 256 DUP(?)

STACK1 ENDS

DATA SEGMENT ;定义数据段 LA DB ?

LB DB ?

DATA ENDS

CODE SEGMENT ;定义代码段

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV DX,MY8255\_MODE ;8255控制端口地址为3006H

MOV AL,80H ;设置8255方式控制字,A、B口工作方式0，A、B口输出

OUT DX,AL

MOV DX,MY8255\_A ;写A口发出的起始数据

MOV AL,80H

OUT DX,AL ;点亮数据灯D7

MOV LA,AL

MOV DX,MY8255\_B ;写B口发出的起始数据

MOV AL,01H

OUT DX,AL ;点亮数据灯D8

MOV LB,AL

LOOP1: CALL DALLY

MOV AL,LA ;将A口起始数据右移再写入A口

ROR AL,1

MOV LA,AL

MOV DX,MY8255\_A

OUT DX,AL

MOV AL,LB ;将B口起始数据左移再写入B口

ROL AL,1

MOV LB,AL

MOV DX,MY8255\_B

OUT DX,AL

MOV AH,1 ;判断是否有按键按下

INT 16H

JZ LOOP1 ;无按键则跳回继续循环，有则退出

DALLY PROC NEAR ;软件延时子程序

PUSH CX

PUSH AX

MOV CX,0FFFH

D1: MOV AX,0FFFFH

D2: DEC AX

JNZ D2

LOOP D1

POP AX

POP CX

RET

DALLY ENDP

CODE ENDS

END START

3. 控制8段数码管显示实验

由8255的A口输出控制一个8段LED数码管，由B口PB3~PB0接入4位开关，

编写一段程序，用数码管显示开关状态。从题目可以得到：A口作为方式0输出，B口低端用作输入，C口没有使用，可以做为输出。则方式控制字为： 10000 010 B。开关状态：4位——16种，用LED数码管显示：0～9，A~F十六进制数显示。

IOY0 EQU 0C000H ;8255的CS引脚连接IOY0对应的端口始地址

MY8255\_A EQU IOY0+00H\*2 ;8255的端口A地址

MY8255\_B EQU IOY0+01H\*2 ;8255的端口B地址

MY8255\_C EQU IOY0+02H\*2 ;8255的端口C地址

MY8255\_MODE EQU IOY0+03H\*2 ;8255的控制寄存器地址

DATA SEGMENT ;定义LED段码表

TAB\_LED DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H DB 7FH,6FH,77H,7CH,39H,5EH,79H,71H

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA, CS:CODE

START: MOV AX,DATA

MOV DS,AX

MOV AL,82H ;1000 0010 A端口方式0输出，B方式0输入

MOV DX,MY8255\_MODE

OUT DX,AL

LOOP1: MOV DX,MY8255\_B

IN AL,DX ;读入开关状态

AND AL,0FH ;屏蔽高四位

MOV BX,OFFSET TAB\_LED ;取LED段码表首地址

XLAT ;查出AL对应的LED段控制码

MOV DX,MY8255\_A ;8255的A端口是LED段码输出口

OUT DX,AL ;输出段控制信号

CALL DALLY

JZ LOOP1

DALLY PROC NEAR ;软件延时子程序 MOV CX,0070H

D1: MOV AX,000FH

D2: DEC AX JNZ D2

LOOP D1

RET

DALLY ENDP

CODE ENDS END START