# 实验一 8086汇编编程实验

### 一、实验目的

1. 掌握汇编语言的编程方法

2. 掌握汇编运行环境的使用方法

3. 掌握汇编语言程序的调试运行过程

### 二、实验内容

1. 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。

2. 在屏幕上显示自己的学号姓名信息。

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

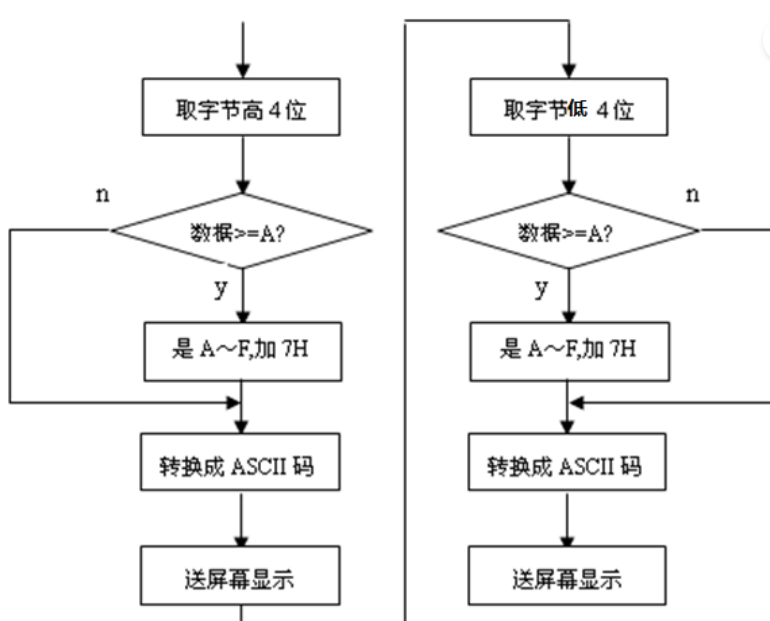
1. 根据实验内容，参考程序流程图编写程序。

2. 选择“项目”菜单中的“编译”或“编译连接”对实验程序进行编译连接。

1. 选择“调试”菜单中的“进行调试”，进入Debug调试，观察调试过程中传输指令执行后各寄存器及数据区的内容。按F9连续运行。

### 四、实验原理

字符转换为ASCII码流程图

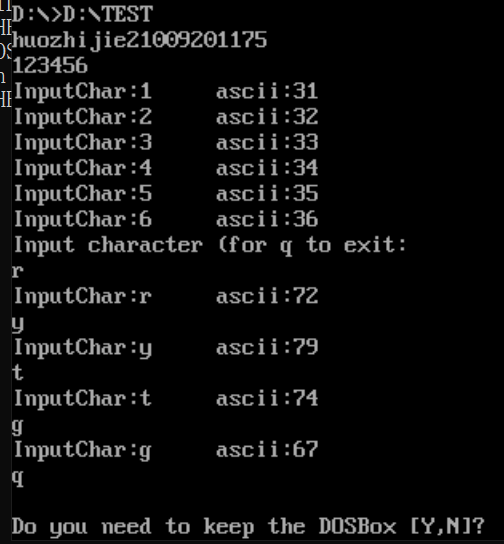


### 五、实验结果

**关键实验源码：**

1. cmp    bl, 9
2. ja     display\_ascii\_alph1
3. display\_ascii\_numb1:
4. add    bl, '0'
5. mov    dl, bl
6. mov    ah, 2
7. int    21h
8. jmp    display\_ascii\_low4
9. display\_ascii\_alph1:
10. add    bl, 37h
11. mov    dl, bl
12. mov    ah, 2
13. int    21h
14. display\_ascii\_low4:
15. and    bh, 0Fh
16. cmp    bh, 9
17. ja     display\_ascii\_alph2
18. display\_ascii\_numb2:
19. add    bh, '0'
20. mov    dl, bh
21. mov    ah, 2
22. int    21h
23. jmp    display\_ascii\_end
24. display\_ascii\_alph2:
25. add    bh, 37h
26. mov    dl, bh
27. mov    ah, 2
28. int    21h

**运行结果：**



# 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。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **十六进制** | **BCD码** | **二进制机器码** | **ASCII码** | **七段码** | |
| **共阳** | **共阴** |
| 0 | 0000 | 0000 | 30H | 40H | 3FH |
| 1 | 0001 | 0001 | 31H | 79H | 06H |
| 2 | 0010 | 0010 | 32H | 24H | 5BH |
| 3 | 0011 | 0011 | 33H | 30H | 4FH |
| 4 | 0100 | 0100 | 34H | 19H | 66H |
| 5 | 0101 | 0101 | 35H | 12H | 6DH |
| 6 | 0110 | 0110 | 36H | 02H | 7DH |
| 7 | 0111 | 0111 | 37H | 78H | 07H |
| 8 | 1000 | 1000 | 38H | 00H | 7FH |
| 9 | 1001 | 1001 | 39H | 18H | 67H |
| A |  | 101 | 41H | 08H | 77H |
| B |  | 1011 | 42H | 03H | 7CH |
| C |  | 1100 | 43H | 46H | 39H |
| D |  | 1101 | 44H | 21H | 5EH |
| E |  | 1110 | 45H | 06H | 79H |
| F |  | 1111 | 46H | 0EH | 71H |

### 四、实验结果

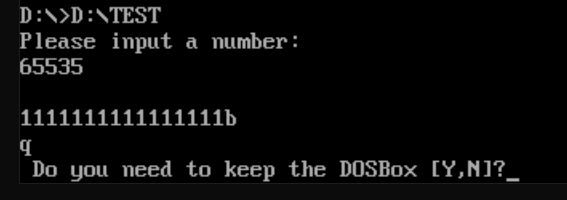
**关键代码（1，2，3）：**

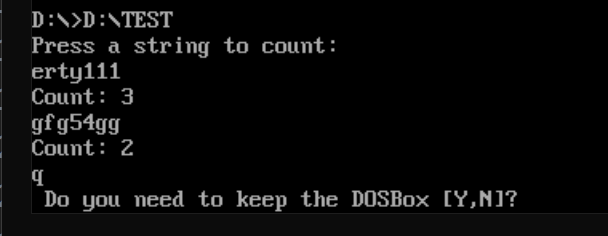
1. ; x10
2. sub al, '0'
3. mov bl, al
4. mov ax, 0Ah
5. mul decimal
6. mov decimal, ax
7. xor bh, bh
8. add decimal, bx
9. jmp read
10. output:
11. mov bx, decimal
12. mov ah, 2
13. mov dl, 0Ah
14. int 21h
15. mov dl, 0Dh
16. int 21h
17. ;循环次数
18. mov cx, 10h
19. bit:
20. test bx, 8000h
21. jz set0

**（4）**

1. read:
2. mov ah, 1
3. int 21h
4. cmp al, 'q'
5. je quit
6. cmp al, 'Q'
7. je quit
8. cmp al, ' '
9. je count
10. cmp al, '0'
11. jb nonNumber
12. cmp al, '9'
13. ja nonNumber
14. inc cx
15. nonNumber:
16. jmp read
17. count:
18. mov ah, 2
19. mov dl, 0Ah
20. int 21h
21. mov dl, 0Dh
22. int 21h
23. mov ah, 9
24. lea dx, counttext
25. int 21h
26. mov ah, 2
27. mov dl, cl
28. add dl, '0'
29. int 21h
30. xor cx, cx ; 清零cx寄存器
31. mov dl, 0Ah
32. int 21h
33. mov dl, 0Dh
34. int 21h
35. jmp read

**运行结果：**





# 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。

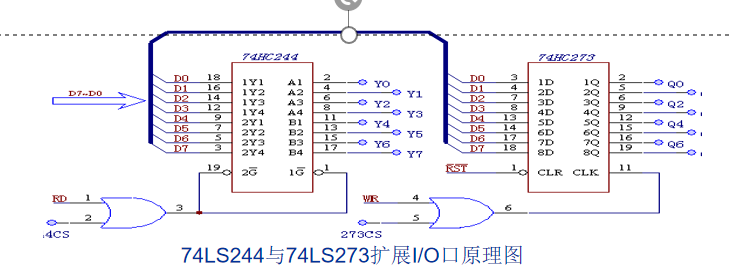
3.当开关Yi全为低电平时，发光二极管Qi从右至左轮流点亮。

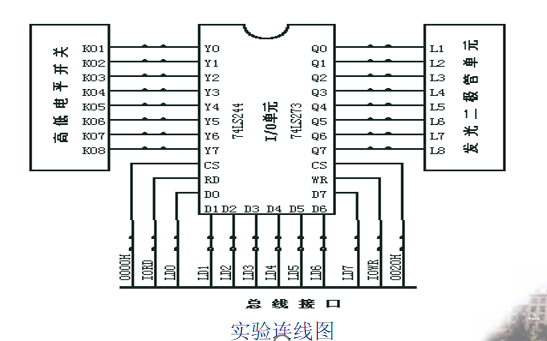
4.自主设计控制及显示模式，完成编程调试，演示实验结果。

### 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。

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：

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。

### 五、实验结果

关键代码：

1. INPUT:
2. MOV DX, IO244
3. IN AX, DX
4. CMP AX, 0FFFFH ;若开关全为低电平
5. JZ Q1   ;从右往左依次点亮
6. CMP AX, 0  ;若开关全为高电平
7. JZ Q2   ;从左往右依次点亮
8. MOV DX, IO273
9. NOT AX   ;AX取非送给273,点亮对应的二极管
10. OUT DX, AX
11. JMP INPUT
12. Q1:
13. MOV AX, 7FFFH
14. MOV DX, IO273
15. R2L:
16. CALL  DELAY  ;延时
17. OUT DX, AX  ;送给273,点亮对应的二极管
18. ROL AX, 1
19. CMP AX, 7FFFH
20. JNE R2L   ;若相等,说明一轮从右往左已经完成,若不等,则继续循环
21. JMP INPUT
23. Q2:
24. MOV AX, 0FFFEH
25. MOV DX, IO273
26. L2R:
27. CALL  DELAY
28. OUT DX, AX
29. ROR AX, 1
30. CMP AX, 0FFFEH
31. JNE L2R   ;若相等,说明一轮从左往右已经完成,若不等,则继续循环
32. JMP INPUT  ;继续读入

# 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。

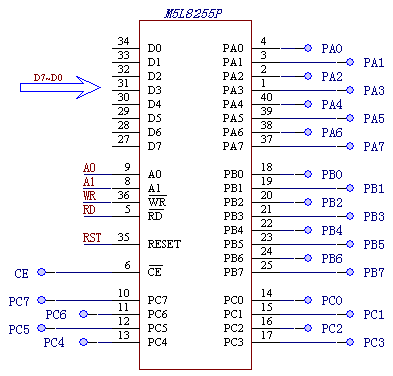
3.I/O输入输出实验：利用8255的A口读取开关状态，8255的B口把状态送发光二极管显示。

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

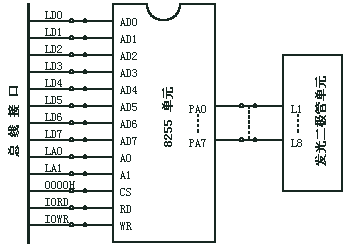
### 三、实验原理

  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。



### 四、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。

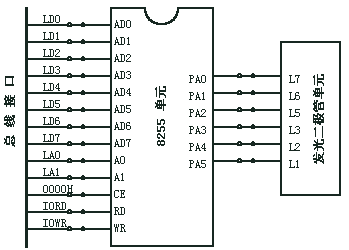
模块的数据（AD0～AD7）、地址线（A0～A7）分别连到ISA总线接口模块的数据（LD0～LD7）、地址线（LA0～LA7）。

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。

2.**交通灯实验**



模块的WR、RD分别连到ISA总线接口模块的IOWR、IORD。

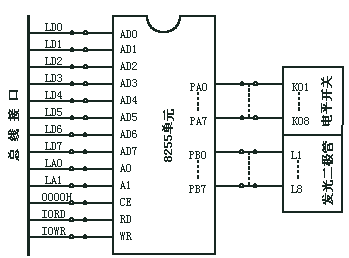
模块的数据（AD0～AD7）、地址线（A0～A7）分别连到ISA总线接口模块的数据（LD0～LD7）、地址线（LA0～LA7）。

8255模块选通线CE连到ISA总线接口模块的0000H。

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

运行程序，观察发光二极管。

**3.I/O输入输出实验**

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。

### 实验结果

**关键代码：**

1. IN  AX, DX
2. CMP AX, 0FFFFH
3. JZ Q1
4. CMP AX, 0FE00H
5. JZ Q2
7. MOV DX,PA\_ADD
8. ;MOV DX, IO273
9. NOT AX
10. OUT DX, AX
11. JMP INPUT
12. Q1:
13. MOV AX, 7FFFH
14. MOV DX,PA\_ADD
15. ;MOV DX, IO273
16. R2L:
17. CALL Light500ms
18. OUT DX, AX
19. ROL AX, 1
20. CMP AX, 7FFFH
21. JNE R2L
22. JMP INPUT
24. Q2:
25. MOV AX, 0FFFEH
26. MOV DX,PA\_ADD
27. ;MOV DX, IO273
28. L2R:
29. CALL Light500ms
30. OUT DX, AX
31. ROR AX, 1
32. CMP AX, 0FFFEH
33. JNE L2R
34. JMP INPUT