

12.30微机实验考试

161710223 陈伟

一、四则运算

实验任务

完成32 位无符号数的加法、减法，16 位乘以16 位，32 位除以16 位除法的四则运算练习。

代码

```
DATA SEGMENT
A DW 1234H,5678H ;被加数
B DW 0FEDCH,123AH ;加数
C DW 2 DUP(0) ;预留和
AD DW 0FEDCH,0BA98H ;被减数
BD DW 1234H,789AH ;减数
CD DW 2 DUP(0) ;预留差
A1 DW 0D678H ;被乘数
B1 DW 0012H ;乘数
C1 DW 2 DUP(0) ;预留积
A2 DW 5678H,0234H ;被除数
B2 DW 0F234H ;除数
C2 DW 2 DUP(0) ;预留商,余数
DATA ENDS

STACK1 SEGMENT STACK
DB 100 DUP(0)
STACK1 ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA,SS:STACK1

START PROC FAR
    PUSH DS ;标准序
    MOV AX,0
    PUSH AX
    MOV AX,DATA
    MOV DS,AX

    MOV AX,A ;32位加32位
    ADD AX,B
    MOV C,AX
    MOV AX,A+2
    ADC AX,B+2 ;用ADC考虑到CF
    MOV C+2,AX

    MOV AX,AD ;32位减32位
    SUB AX,BD
    MOV CD,AX
    MOV AX,AD+2
    SBB AX,BD+2 ;用SBB考虑到CF
```

```

MOV    CD+2,AX
MOV    AX,A1
MUL    B1
MOV    C1,AX          ;将AX中保存的结果低16位存放到C1中
MOV    C1+2,DX        ;将DX中保存的结果高16位存放到C1+2中

MOV    DX,A2+2        ;DX存放高16位
MOV    AX,A2          ;AX存放低16位
DIV    B2
MOV    C2,AX          ;保存商
MOV    C2+2,DX        ;保存余数

RET
START ENDP

CODE   ENDS
      END    START

```

二、数据统计

实验任务

本实验要求通过求某数据区内负数的个数来表现循环程序的结构形式。要求实验程序在数据段中存放一组数据，分类统计数据中正数、负数和零的个数，并分别存入内存变量 Positive、Negative和 Zero 中。将所有数据累加求和，存入SUM 中。

原始代码

```

DATA SEGMENT
    NUM      DB 12H,88H,82H,89H,33H,90H,0H,10H,0BDH,01H
    Positive  DB DUP (0)
    Negative  DB DUP (0)
    Zero      DB DUP (0)
    SUM       DW 2 DUP (0)
DATA ENDS

STACK1 SEGMENT STACK
    DB 100 DUP(0)
STACK1 ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA,SS:STACK1

START PROC FAR
    PUSH DS
    MOV  AX, 0
    PUSH AX
    MOV  AX, DATA
    MOV  DS, AX

    MOV  CX,10          ;循环10次
    LEA  SI,NUM
    MOV  BX,0

LAB1:

```

```

MOV DL,[SI]
CMP DL,0           ;判断num是否为0
JG LAB2           ;num>0,跳转到LAB2
JL LAB3           ;num<0,跳转到LAB3
INC ZERO          ;num=0,ZERO加1
JMP LAB4

LAB2:
INC Positive      ;负数+1
JMP LAB4

LAB3:
INC Negative      ;正数+1

LAB4:
MOV AL,[SI]
CBW              ;符号拓展,8位拓展为16位
ADD SUM,AX        ;+sum
ADC [SUM+2],0
INC SI
LOOP LAB1         ;循环

RET

START ENDP
CODE ENDS
END START

```

显示数据核心代码

```

MOV DX,OFFSET result1 ;显示Positive
MOV AH,9
INT 21H
MOV BL,Positive
MOV AL,BL
SHR AL,4
CALL ToASCII         ;把高四位转换为对应的ASCII码
MOV [result],AL
MOV AL,BL
CALL ToASCII         ;把低四位转换为对应的ASCII码
MOV [result+1],AL
MOV DX,OFFSET result
MOV AH,9
INT 21H

ToASCII PROC
AND AL,0FH
ADD AL,'0'
CMP AL,'9'
JBE LAB5
ADD AL,7
LAB5:
RET
ToASCII ENDP

```

排序部分核心代码

```

XOR AX,AX           ;选择排序算法
MOV BX,OFFSET NUM   ;I=0

```

```

MOV SI,0
FORI:
MOV DI,SI
INC DI ;J=I+1
FORJ:
MOV AL,[BX+SI]
CMP AL,[BX+DI] ;A[i]与A[j]比较
JLE NEXTJ ;A[i]小于等于A[j]跳转
XCHG AL,[BX+DI] ;A[i]与A[j]交换
MOV [BX+SI],AL
NEXTJ:
INC DI ;J=J+1
CMP DI,10 ;J<10跳转
JB FORJ
NEXTI:
INC SI ;I=I+1
CMP SI,9
JB FORI ;I<9时跳转

```

三、代码转换

实验任务

从键盘输入小写字母最多 20 个, 以 '.'号作为结束标志 将其变换成应的大写字母输出在屏幕上。

代码

```

CRLF MACRO
MOV DL, 0DH
MOV AH, 02H
INT 21H
MOV DL, 0AH
MOV AH, 02H
INT 21H
ENDM

DATA SEGMENT
MES1 DB 'PLEASE INPUT THE SMALL LETTER, ENDED WITH ".":$'
MES2 DB 'THE CAPITAL LETTER IS:$'
SMALL DB 50 ;预留键盘输入缓冲区长度为 50 个
DB 0 ;预留实际键盘输入字符数的个数
DB 50 DUP(0)
CAPITAL DB 50 DUP(0) ;预留大写字母缓冲区长度为 50 个
DATA ENDS

STACK1 SEGMENT STACK
DB 100 DUP (0)
STACK1 ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:STACK1
START PROC FAR
PUSH DS
MOV AX, 0
PUSH AX

```

```

MOV     AX, DATA
MOV     DS, AX
MOV     AH, 9
MOV     DX, OFFSET MES1 ;输出提示信息MES1
INT     21H
CRLF

MOV     AH, 0AH
LEA     DX, SMALL;接收小写字符串
INT 21H
CRLF                                ;宏调用

LEA     BX, SMALL+2
LEA     DI, CAPITAL
MOV     CX, 20                    ;最多20个字符
LAB:    MOV AL, [BX]
        CMP AL, 2EH                ;是否遇到句号.
        JE     KE
        SUB     AL, 20H              ;转为大写,ASCLL-20H
        MOV     [DI], AL
        INC     BX
        INC     DI
        LOOP    LAB

KE:     MOV     AL, '$'              ;大写字符串后加"$"
        MOV     [DI], AL
        MOV     DX, OFFSET MES2 ; 输出提示信息MES2
        MOV     AH, 9
        INT     21H
        CRLF

        MOV     DX, OFFSET CAPITAL
        MOV     AH, 9 ; 输出大写字符串
        INT     21H
        RET

START ENDP
CODE ENDS
        END START

```

探究题(只让小写字母ASCII-20H)

```

LEA     BX, SMALL+2
LEA     DI, CAPITAL
MOV     CX, 20                    ;最多20个字符
LAB:    MOV     AL, [BX]
        CMP     AL, 2EH                ;是否遇到句号.
        JE     KE
        CMP     AL, 'a'                ;是否>=a
        JL     NEXT
        CMP     AL, 'z'                ;是否<=z
        JG     NEXT
        SUB     AL, 20H              ;转为大写,ASCLL-20H
NEXT:
MOV     [DI], AL
        INC     DI
INC     BX

```

实验中所使用**DOS 功能调用（INT 21H）说明如下****(1) 显示单个字符输出**

入口：AH=02H

调用参数：DL= 输出字符

(2) 显示字符串

入口：AH=09H

调用参数：DS:DX= 串地址，'\$'为结束字符

(3) 键盘输入

入口：AH=0AH

调用参数：DS:DX= 输入缓冲区地址，首字节为缓冲区字节长度，第二字节为实际输入的字符计数

(4) 返回 DOS 系统

入口：AH=4CH

调用参数：AL= 返回码

四、8254定时/计数器

实验任务

分别对8254芯片的3个定时/计数器编程，并选择合适的工作方式和初值，以达到如下的效果：

- 1) 定时/计数器0，计数脉冲频率为18.432KHz，OUT0分两路输出，一路外接2位LED，使其以亮0.5秒灭0.5秒循环闪亮，另一路作为计数器1的计数脉冲CLK1。
- 2) 定时/计数器1，OUT1的输出外接2位LED，使其以亮3秒灭1秒循环闪亮。
- 3) 定时/计数器2的计数脉冲来自单次脉冲单元，按压开关产生的脉冲作为计数器2的计数脉冲。OUT2外接2位LED，当按压开关到17次时LED长亮，并将按压开关剩余次数将在屏幕上显示。

原理图

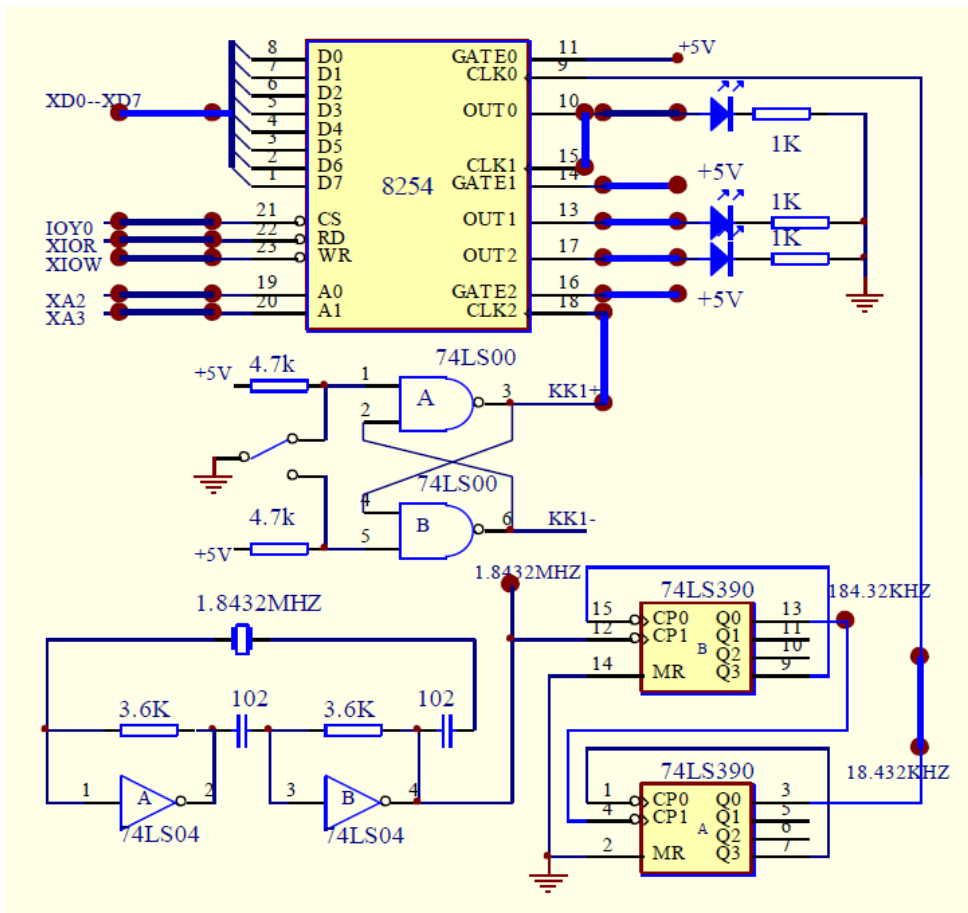


图 3-2-2 8254 定时/计数器电原理图

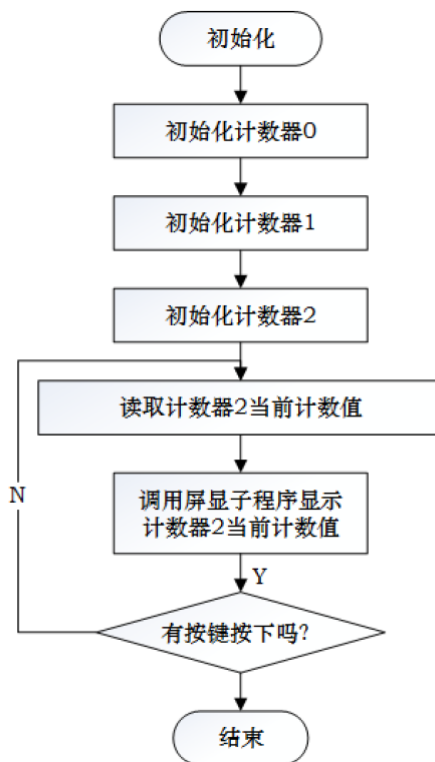


图 3-2-3 8254 定时/计数器参考程序流程图

代码

```

IOY0    EQU 0E000H
TIMER0  EQU IOY0+00H*4 ;8254计数器0端口地址
TIMER1  EQU IOY0+01H*4 ;8254计数器1端口地址

```

```

TIMER2 EQU IOY0+02H*4 ;8254计数器2端口地址
TCTL EQU IOY0+03H*4 ;8254控制寄存器端口地址

STACK1 SEGMENT STACK
        DW 256 DUP(?)
STACK1 ENDS

DATA SEGMENT
        MES0 DB 'Pressed: $'
        MES1 DB 'Press any key to exit!', 0DH, 0AH, '$'
        NUM DB ?
DATA ENDS

CODE SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:STACK1
START: MOV AX, DATA
        MOV DS, AX

        MOV DX, OFFSET MES1
        MOV AH, 9
        INT 21H

        MOV DX, TCTL ;控制字, 选用计数器0, 写16位值, 方式3, 二进制计数
        MOV AL, 00110110B
        OUT DX, AL

        MOV DX, TIMER0 ;写入4800H
        MOV AL, 00H
        OUT DX, AL
        MOV AL, 48H
        OUT DX, AL

        MOV DX, TCTL ;控制字, 选用计数器1, 写低字节, 方式2, 二进制计数
        MOV AL, 01010100B
        OUT DX, AL

        MOV DX, TIMER1 ;写入04H
        MOV AL, 04H
        OUT DX, AL

        MOV DX, TCTL ;控制字, 选用计数器2, 写低字节, 方式0, 二进制计数
        MOV AL, 10010000B
        OUT DX, AL

        MOV DX, TIMER2
        MOV AL, 0FH
        OUT DX, AL

L1: MOV DX, TIMER2 ;读入计数器2值保存
     IN AL, DX
     MOV NUM, AL
     CALL DISP

     MOV AL, NUM
     CMP AL, 0
     JZ QUIT ;计数至0时退出

```


| | | |
|-------|---------------------|-------------------------|
| | MOV DL, 0FFH | ;判主键盘有无键按下 |
| | MOV AH, 6 | |
| | INT 21H | |
| | JZ L1 | ;无键按下跳转 |
| QUIT: | MOV AX, 4C00H | ;结束程序退出 |
| | INT 21H | |
| DISP | PROC | ;显示子程序 |
| | MOV DX, OFFSET MES0 | ;显示MES0 |
| | MOV AH, 9 | |
| | INT 21H | |
| | MOV AL, NUM | |
| | CMP AL, 9 | ;判断是否<=9 |
| | JLE L2 | ;若是则为'0'-'9',ASCII码加30H |
| | ADD AL, 7 | ;否则为' A'-'F',ASCII码加37H |
| L2: | ADD AL, 30H | ;在显示器上显示按压开关的次数 |
| | MOV DL, AL | |
| | MOV AH, 2 | |
| | INT 21H | |
| | MOV DL, 0DH | |
| | INT 21H | |
| | RET | |
| DISP | ENDP | |
| CODE | ENDS | |
| | END | START |

五、8259中断控制器

实验任务

用单次脉冲单元的KK1+和KK2+产生单次脉冲，模拟两个中断源，在KK1+触发的中断服务程序中显示字符“0”，在KK2+触发的中断服务程序中显示字符“1”。

原理

表 3-3-2 PC 微机系统中的硬件中断

| 中断号 | 功能 | 中断向量号 | 中断向量地址 |
|--------------|-----------|-------|-------------|
| 主 8259 IRQ0 | 日时钟/计数器 0 | 08H | 0020H~0023H |
| 主 8259 IRQ1 | 键盘 | 09H | 0024H~0027H |
| 主 8259 IRQ2 | 接从片 8259 | 0AH | 0028H~002BH |
| 主 8259 IRQ3 | 串行口 2 | 0BH | 002CH~002FH |
| 主 8259 IRQ4 | 串行口 1 | 0CH | 0030H~0033H |
| 主 8259 IRQ5 | 并行口 2 | 0DH | 0034H~0037H |
| 主 8259 IRQ6 | 软盘 | 0EH | 0038H~003BH |
| 主 8259 IRQ7 | 并行口 1 | 0FH | 003CH~003FH |
| 从 8259 IRQ8 | 实时钟 | 70H | 01C0H~01C3H |
| 从 8259 IRQ9 | 保留 | 71H | 01C4H~01C7H |
| 从 8259 IRQ10 | 保留 | 72H | 01C8H~01CBH |
| 从 8259 IRQ11 | 保留 | 73H | 01CCH~01CFH |
| 从 8259 IRQ12 | 保留 | 74H | 01D0H~01D3H |
| 从 8259 IRQ13 | 协处理器中断 | 75H | 01D4H~01D7H |
| 从 8259 IRQ14 | 硬盘控制器 | 76H | 01D8H~01DBH |
| 从 8259 IRQ15 | 保留 | 77H | 01DCH~01DFH |

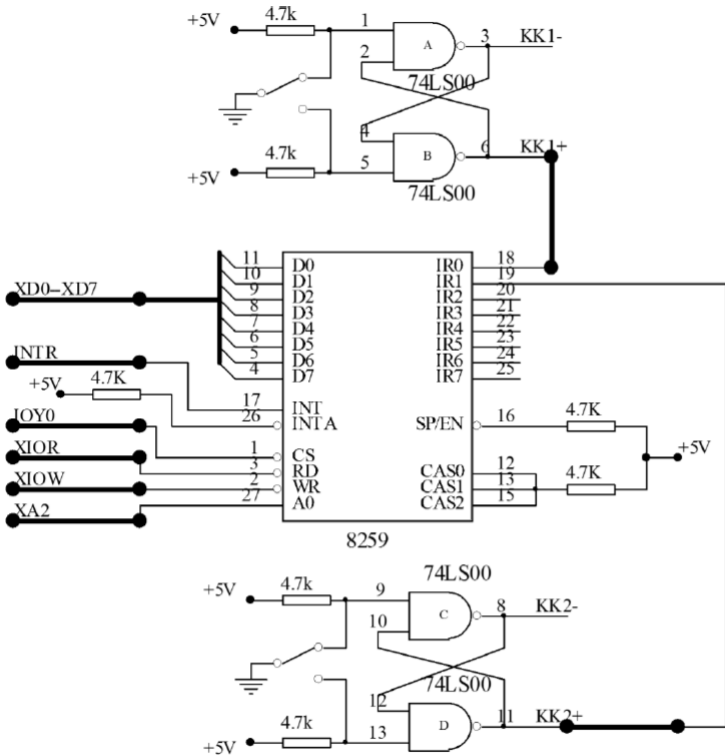


图 3-3-4 8259 中断控制器电原理图

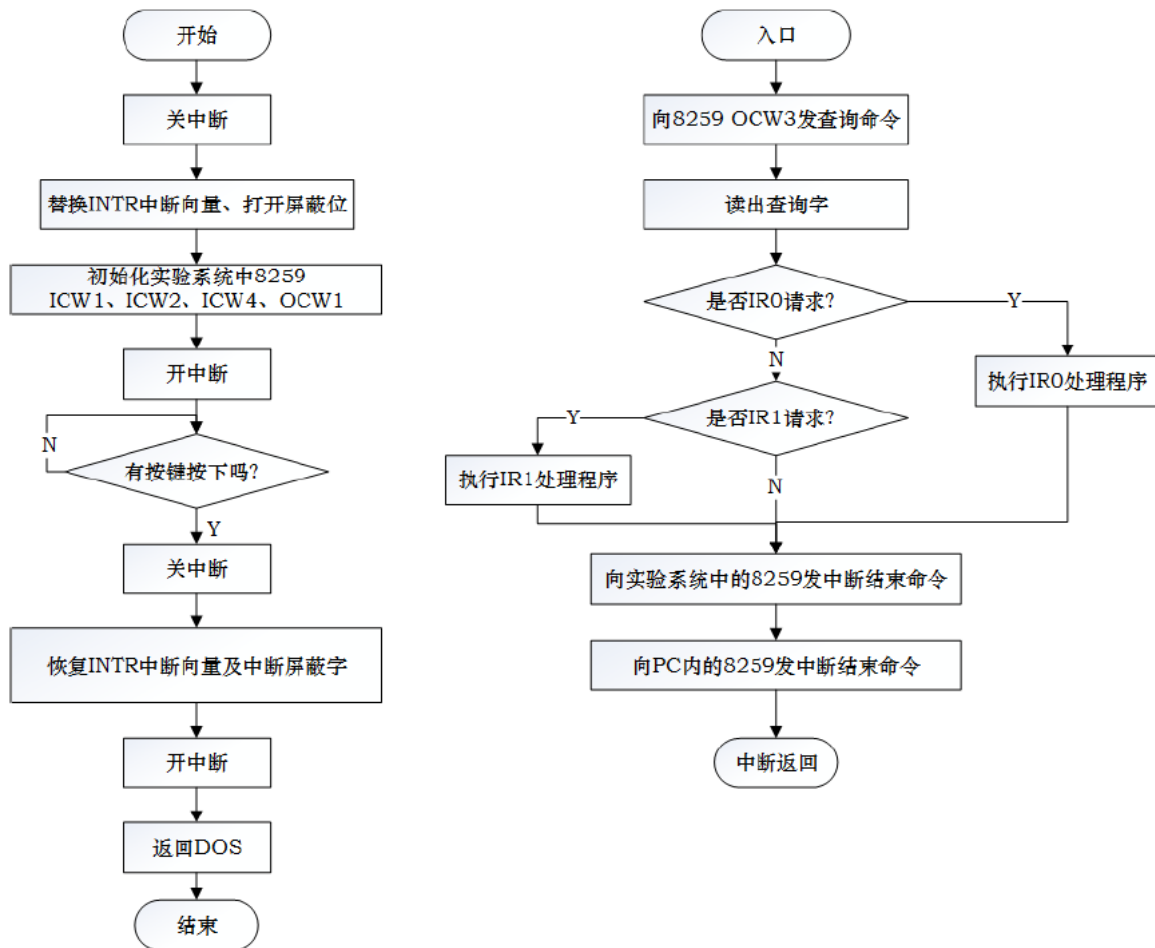


图 3-3-5 8259 中断控制器参考流程图

代码

```

INTR_IVADD EQU 003CH ;INTR对应的中断矢量地址
INTR_OCW1 EQU 021H ;INTR对应PC机内部8259的OCW1地址
INTR_OCW2 EQU 020H ;INTR对应PC机内部8259的OCW2地址
INTR_IM EQU 07FH ;INTR对应的中断屏蔽字
IOY0 EQU 0E000H ;片选IOY0对应的端口始地址

MY8259_ICW1 EQU IOY0+00H ;实验系统中8259的ICW1端口地址
MY8259_ICW2 EQU IOY0+04H ;实验系统中8259的ICW2端口地址
MY8259_ICW3 EQU IOY0+04H ;实验系统中8259的ICW3端口地址
MY8259_ICW4 EQU IOY0+04H ;实验系统中8259的ICW4端口地址
MY8259_OCW1 EQU IOY0+04H ;实验系统中8259的OCW1端口地址
MY8259_OCW2 EQU IOY0+00H ;实验系统中8259的OCW2端口地址
MY8259_OCW3 EQU IOY0+00H ;实验系统中8259的OCW3端口地址

STACK1 SEGMENT STACK
        DW 256 DUP(?)
STACK1 ENDS

DATA SEGMENT
        MES DB 'Press any key to exit!',0AH,0DH,0AH,0DH,'$'
        CS_BAK DW ? ;保存INTR原中断处理程序入口段地址的变量
        IP_BAK DW ? ;保存INTR原中断处理程序入口偏移地址的变量
        IM_BAK DB ? ;保存INTR原中断屏蔽字的变量
DATA ENDS
  
```

```

CODE                SEGMENT
                    ASSUME CS:CODE, DS:DATA, SS:STACK1

START:
    MOV AX, DATA
    MOV DS, AX
    MOV DX, OFFSET MES      ;显示退出提示
    MOV AH, 09H
    INT 21H

    CLI
    MOV AX, 0000H           ;替换INTR的中断矢量
    MOV ES, AX
    MOV DI, INTR_IVADD      ;保存INTR原中断处理程序入口偏移地址
    MOV AX, ES:[DI]
    MOV IP_BAK, AX
    MOV AX, OFFSET MYISR    ;设置当前中断处理程序入口偏移地址
    MOV ES:[DI], AX

    ADD DI, 2               ;保存INTR原中断处理程序入口段地址
    MOV AX, ES:[DI]
    MOV CS_BAK, AX
    MOV AX, SEG MYISR       ;设置当前中断处理程序入口段地址
    MOV ES:[DI], AX

    MOV DX, INTR_OCW1       ;设置中断屏蔽寄存器，打开INTR的屏蔽位
    IN  AL, DX              ;保存INTR原中断屏蔽字
    MOV IM_BAK, AL

    AND AL, INTR_IM         ;允许PC机内部8259的 IR7 中断
    OUT DX, AL

    MOV DX, MY8259_ICW1     ;初始化实验系统中8259的ICW1
    MOV AL, 13H            ;边沿触发、单片8259 、需要ICW4
    OUT DX, AL

    MOV DX, MY8259_ICW2     ;初始化实验系统中8259的ICW2
    MOV AL, 08H
    OUT DX, AL

    MOV DX, MY8259_ICW4     ;初始化实验系统中8259的ICW4
    MOV AL, 01H            ;非自动结束EOI
    OUT DX, AL

    MOV DX, MY8259_OCW3     ;向8259的OCW3发送读取IRR命令
    MOV AL, 0AH
    OUT DX, AL

    MOV DX, MY8259_OCW1     ;初始化实验系统中8259的OCW1
    MOV AL, 0FCH           ;打开IR0和IR1的屏蔽位
    OUT DX, AL
    STI

WAIT1:
    MOV AH, 1              ;判断是否有按键按下
    INT 16H                ;无按键则跳回继续等待，有则退出
    JZ WAIT1

```

```

QUIT:
    CLI
    MOV AX, 0000H          ;恢复INTR原中断矢量
    MOV ES, AX
    MOV DI, INTR_IVADD     ;恢复INTR原中断处理程序入口偏移地址
    MOV AX, IP_BAK
    MOV ES:[DI], AX

    ADD DI, 2
    MOV AX, CS_BAK         ;恢复INTR原中断处理程序入口段地址
    MOV ES:[DI], AX
    MOV DX, INTR_OCW1
    MOV AL, IM_BAK         ;恢复INTR原中断屏蔽寄存器的屏蔽字
    OUT DX, AL
    STI
    MOV AX, 4C00H          ;返回到 DOS
    INT 21H

MYISR PROC NEAR           ;中断处理程序MYISR
    PUSH AX
QUERY:                    ;向8259的OCW3发送读取IRR 命令
    MOV DX, MY8259_OCW3
    IN AL, DX              ;读出IRR寄存器值

    AND AL, 03H
    CMP AL, 01H
    JE IROISR              ;若为IRO请求, 跳到IRO处理程序
    JNE IR1ISR             ;若为IR1请求, 跳到IR1处理程序
    JMP OVER

IROISR:                   ;IRO处理, 显示字符串STR0
    MOV AL, 30H
    MOV AH, 0EH
    INT 10H
    MOV AL, 20H
    INT 10H
    JMP OVER

IR1ISR:                   ;IR1处理, 显示字符串STR1
    MOV AL, 31H
    MOV AH, 0EH
    INT 10H
    MOV AL, 20H
    INT 10H
    JMP OVER

OVER:
    MOV DX, INTR_OCW2     ;向PC机内部8259发送中断结束命令
    MOV AL, 20H
    OUT DX, AL
    MOV AL, 20H
    OUT 20H, AL
    POP AX

    IRET

```

MYISR ENDP

CODE ENDS

END START

六、8255并口控制器

实验任务

用8255实现键盘扫描与显示功能,当小键盘有键按下时,键值在数码管最右位的位置上显示,数码管上显示最新六位键值内容,当主键盘有键按下时,返回系统。

原理图

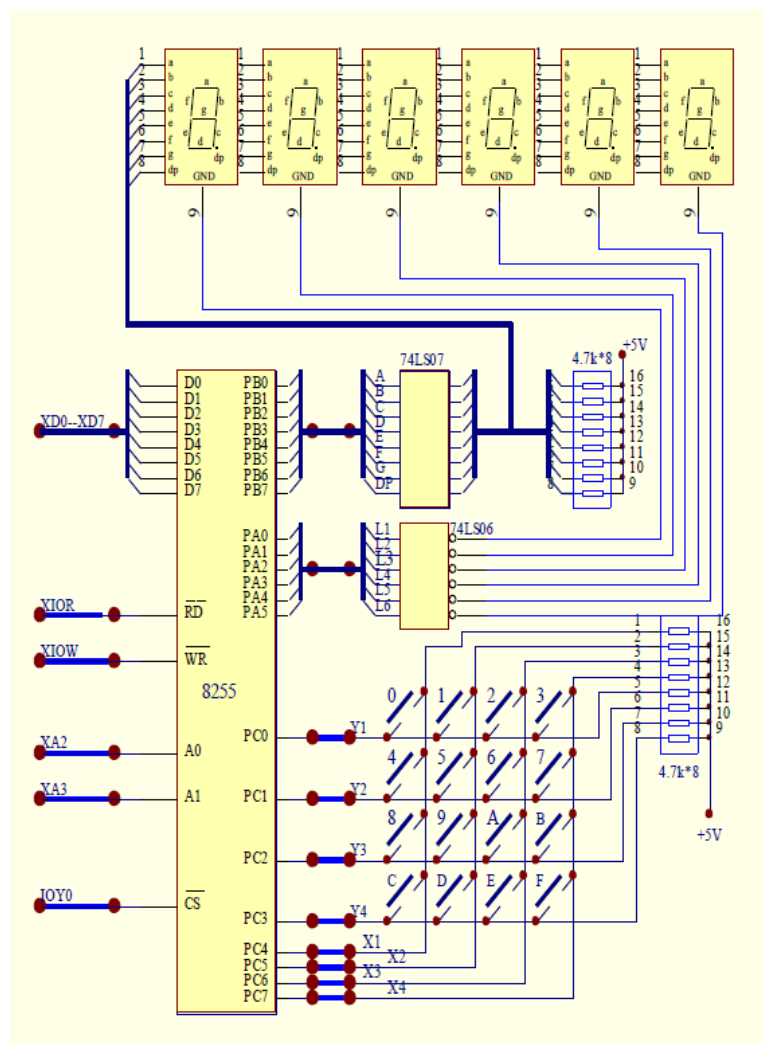
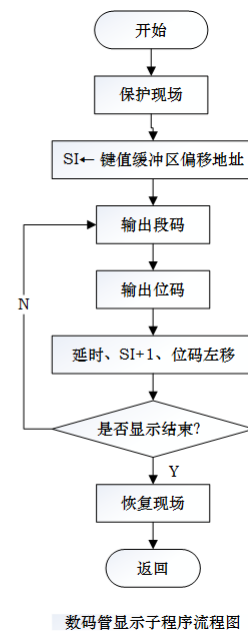
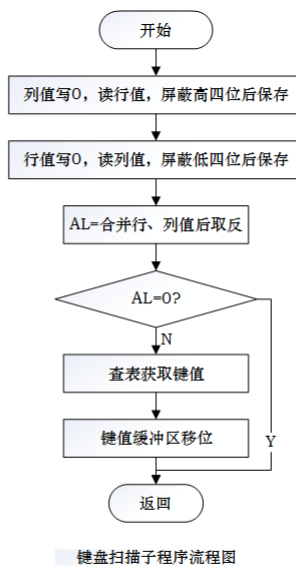
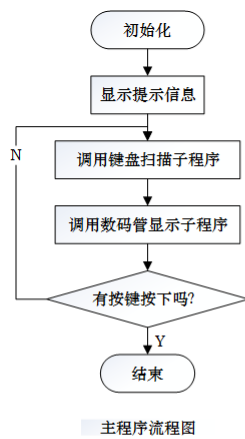


图 3-4-3 键盘扫描及显示实验电原理框图



原理

在本实验中, 键盘是由4 行×4 列共16 个按键组成, 其中PC0 ~ PC3 作为行选择信号Y1 ~ Y4, PC4 ~ PC7 作为列扫描信号X1 ~ X4。在键盘扫描中,常用的方法有逐行扫描和行反转法两种,本实验采用行反转法,其键盘扫描方式如下:

1. 写8255 工作方式控制字,使C4 ~ PC7 为0(列值为0),然后读入PC0 ~ PC3(行值),如果此时有键按下,则对应的行值为0。
2. 写8255 工作方式控制字, 使PC0 ~ PC3 为0(行值为0),再读入C4 ~ PC7(列值),因此时按键没有松开,故对应的列值为0;因此,当一个键按下时,可以读到一对唯一的行值和列值
3. 合并行值和列值,与每个键盘扫描码逐一比较,就可以确定是那一个键按下了。在软件设计上要注意消除按键抖动的处理以及数码管显示的刷新。数码管显示器是由6 位共阴的7 段数码管组成,在电路中它也接成扫描电路方式, 6 位共阴的7段数码共用段码A ~ G、Dp,当位码L1 ~ L6 某位0 为时,则该位数码管点亮,显示相应的字符。要在六位数码管上显示不同的内容,需一定频率循环地往六位数码管上输送段码与位码,在因此这种显示方式称为动态显示方式。

代码

```

IOY0    EQU    0E000H           ;片选IOY0对应的端口始地址
PA55    EQU    IOY0+00H*4       ;8255的A口地址
PB55    EQU    IOY0+01H*4       ;8255的B口地址
PC55    EQU    IOY0+02H*4       ;8255的C口地址
PCTL    EQU    IOY0+03H*4       ;8255的控制寄存器地址

DATA    SEGMENT
    BUFF    DB    6 DUP(10H)
    TABLE1 DB    11H,21H,41H,81H,12H,22H,42H,82H           ;取反后的键盘扫描码
            DB    14H,24H,44H,84H,18H,28H,48H,88H           ;数码管的段码表

    DCTBL    DB    3FH,06H,5BH,4fh,66h,6dh,7dh,07h,7fh,6fh
            DB    77h, 7ch,39h,5eh,79h,71h,00h

    MES      DB    'Press any key on the small keyboard!',0DH,0AH
            DB    'Press key to display on the led!',0dh,0ah,'$'
    MESS     DB    'Press main keyboard any key to exit!', 0dh, 0ah, 0dh, 0ah, '$'
    KEYC     DB    ?
    KEY      DB    ?
DATA    ENDS
  
```

```
STAC    SEGMENT PARA STACK
        DB 256 DUP(?)
STAC    ENDS
```

```
CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:STAC
```

```
START:
    MOV AX, DATA
    MOV DS, AX
    MOV DX, OFFSET MES
    MOV AH, 9
    INT 21H

    MOV DX, OFFSET MESS
    MOV AH, 9
    INT 21H
```

```
LOP1:
    CALL TESTKEY
    CALL DISP
    MOV DL, 0FFH
    MOV AH, 6
    INT 21H
    JZ LOP1
```

```
QUIT:
    MOV AX, 4C00H
    INT 21H
```

```
TESTKEY PROC
```

```
KEY0:
    MOV AL, 81H                ;8255控制字PC0-3入,PC4-7出
    MOV DX, PCTL
    OUT DX, AL

    MOV AL, 00                ;C口输出0
    MOV DX, PC55
    OUT DX, AL

    IN AL, DX                 ;读入行值,屏蔽列值后保存
    AND AL, 0FH
    MOV KEYC,AL
```

```
KEY1:
    MOV AL, 88H                ;8255控制字PC0-3出,PC4-7入
    MOV DX, PCTL
    OUT DX, AL

    MOV AL, 00                ;C口输出0
    MOV DX, PC55
    OUT DX, AL

    IN AL, DX                 ;读入列值,屏蔽行值后合并取反
    AND AL, 0F0H
    OR AL, KEYC
    NOT AL
```



```
CMP AL, 0          ;无键按下退出子程序
JZ KEYEND
```

```
MOV SI, OFFSET TABLE1    ;查找按键的值
MOV CX, 16
MOV DL, 00H
```

KEY2:

```
CMP AL, [SI]
JZ KEY3
INC SI
INC DL
DEC CX
JZ KEYEND
JMP KEY2
```

KEY3:

```
MOV KEY, DL
MOV SI, OFFSET BUFF+1
MOV DI, OFFSET BUFF
MOV CX, 5
```

KEY4: ;显示缓冲区内容向前移一位

```
MOV AL, [SI]
MOV [DI], AL
INC SI
INC DI
LOOP KEY4
```

```
MOV AL, KEY      ;当前键值存入BUF[5]单元
MOV [DI], AL
```

```
MOV AL, 88H      ;8255控制字,PC0-3出 ,PC4-7入
MOV DX, PCTL
OUT DX, AL
```

KEY5:

```
MOV AL, 00      ;判断按键是否释放
MOV DX, PC55
OUT DX, AL
IN AL, DX
AND AL, 0F0H
CMP AL, 0F0H
JNZ KEY5
```

KEYEND:

```
RET
```

TESTKEY ENDP

DISP PROC

```
PUSH DS
PUSH AX
MOV CL, 1
MOV SI, OFFSET BUFF
```

DIS2:

```
MOV AL, [SI]      ;输出段码
LEA BX, DCTBL
```

```

XLAT
MOV DX, PB55
OUT DX, AL
MOV DX, PA55          ;输出位码
MOV AL, CL
NOT AL
OUT DX, AL
CALL DELAY
INC SI                ;段码地址+1
ROL CL, 1             ;位码左移1位
CMP CL, 40H           ;位码是最后位码?
JNZ DIS2              ;不是转DIS2
POP AX                ;是返回
POP DS
RET
DISP ENDP

DELAY PROC NEAR
    PUSH CX
    PUSH BX
    MOV BX, 80H

DEL1:
    MOV CX, 0FFFFH
    LOOP $
    DEC BX
    JNZ DEL1
    POP BX
    POP CX
    RET
DELAY ENDP

CODE    ENDS
        END START

```

七、A/D与 D/A 转换实验

实验任务

从键盘输入两位十六进制值经 0832 转换成数字量后送 0809 的通道 0 再经 A/D 转换后在屏幕上显示此数字量。

原理图

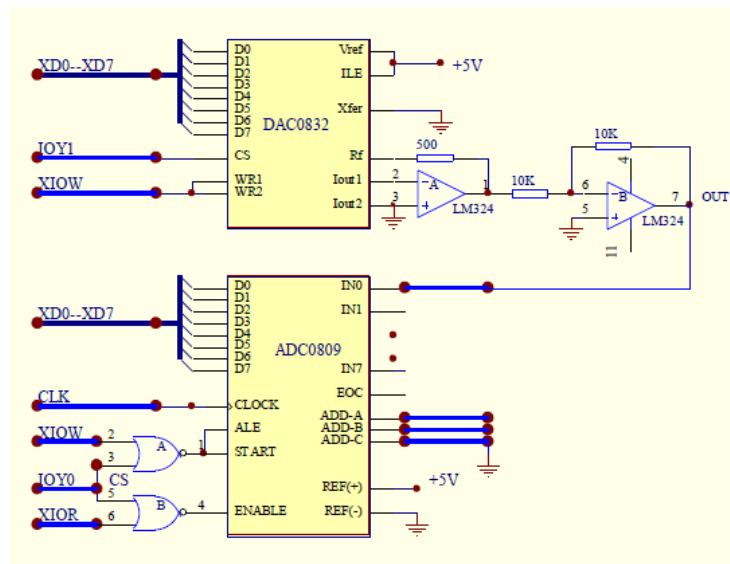
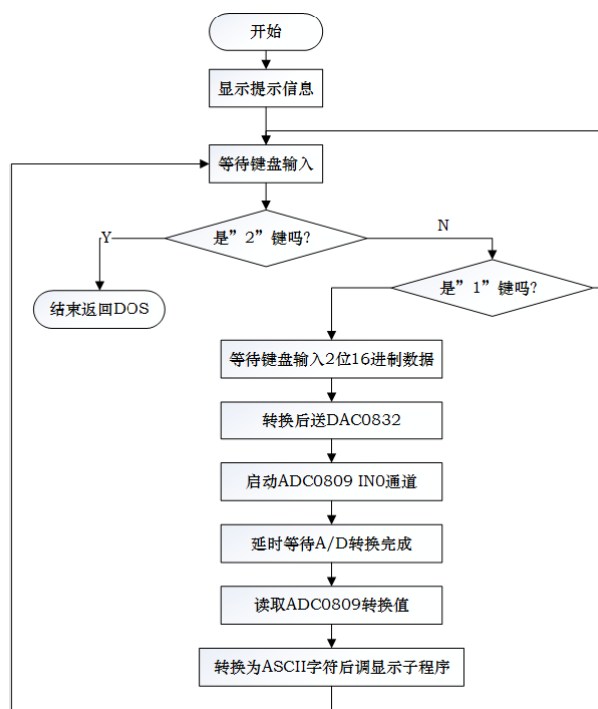


图 3-5-3 A/D 转换实验电原理框图



代码

```

CRLF MACRO                                ;宏定义了回车
    MOV DL, 0DH
    MOV AH, 02H
    INT 21H
    MOV DL, 0AH
    INT 21H
    ENDM

IOY0 EQU 0E000H                          ;片选 IOY0 对应的端口始地址
IOY1 EQU 0E040H                          ;片选 IOY1 对应的端口始地址
ADCS EQU IOY0                            ;AD0809 的端口地址
DACS EQU IOY1                            ;DAC0832 的端口地址

STAC SEGMENT PARA STACK
    DW 256 DUP(?)
STAC ENDS

```

DATA SEGMENT

```
MES0 DB 'PRESS 1 TO INPUT DATA!', 0DH, 0AH
      DB 'PRESS 2 TO QUIT!', 0DH, 0AH, 0DH, 0AH, '$'
MES1 DB '*****PLEASE INPUT DATA OF HEX!*****', 0DH, 0AH, '$'
MES2 DB '0832 OUTPUT DATA = $'
MES3 DB '0809 INPUT DATA = $'
BUF DB 2 DUP(?)
```

DATA ENDS

CODE SEGMENT

```
ASSUME CS:CODE, DS:DATA, SS:STAC
```

START:

```
MOV AX, DATA
MOV DS, AX

LEA DX, MES0
MOV AH, 9
INT 21H
```

LOP1:

```
MOV DL, 0FFH          ;检测键盘输入
MOV AH, 6
INT 21H
JZ LOP1

CMP AL, '1'
JZ DA
CMP AL, '2'
JZ EXIT0
JMP START
```

EXIT0: JMP EXIT

```
DA: LEA DX, MES1      ;显示MES1
MOV AH, 9
INT 21H
```

```
LEA DX, MES2          ;显示MES2
MOV AH, 9
INT 21H
```

;十六进制值存入BUF和BUF[1]

```
MOV AH, 1
INT 21H
MOV BUF, AL
INT 21H
MOV BUF[1], AL
MOV AH, 2
CRLF
```

;十六进制转换十进制

DA0:

```
MOV AL, BUF
SUB AL, 30H
CMP AL, 9
JBE A0
SUB AL, 7
```

A0: ;十六进制转换十进制

```
MOV BL, AL
MOV AL, BUF[1]
SUB AL, 30H
CMP AL, 9
JBE B0
SUB AL, 7
```

B0:

```
MOV CL, 4
ROL BL, CL
XOR AL, BL
MOV DX, DACS ;启动0832
OUT DX, AL
```

AD: ;启动0809 INO

```
MOV DX, ADCS
OUT DX, AL
CALL DELAY
LEA DX, MES3 ;显示MES3
MOV AH, 9
INT 21H

MOV DX, ADCS ;读入0809 INO值
IN AL, DX
MOV BL, AL
AND AL, 0F0H ;显示高位

MOV CL, 4
ROL AL, CL
CALL CRT1
MOV AL, BL ;显示低位
AND AL, 0FH
CALL CRT1
CRLF
INT 21H
JMP START
```

EXIT:

```
MOV AX, 4C00H
INT 21H
```

CRT1 PROC ;在屏幕上显示一位16进制字符

```
ADD AL, 30H
CMP AL, 39H
JBE D0
ADD AL, 7
```

D0: MOV DL, AL

```
MOV AH, 2
INT 21H
RET
```

CRT1 ENDP

DELAY PROC NEAR

```
PUSH CX
MOV CX, 0FFFFH
LOOP $
POP CX
```

```
RET
DELAY ENDP

CODE ENDS
END START
```

八、点阵LED显示设计实验

原理

8×8点阵LED相当于8×8个发光管组成的阵列，对于共阳极LED来说，其中每一行共用一个阳极（行控制），每一列共用一个阴极（列控制）。行控制和列控制满足正确的电平就可使相应行列的发光管点亮。实验平台上点阵LED的管脚及相应的行、列控制位如图4-13-1所示。

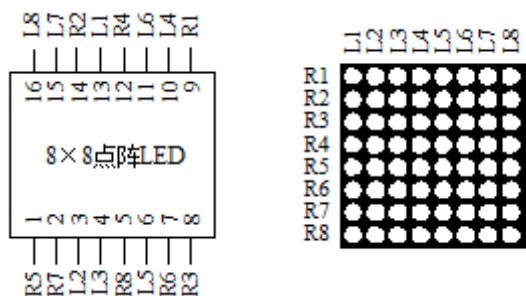


图4-13-1 点阵LED管脚图

共阳极和共阴极LED的内部结构分别如图4-13-2和4-13-3所示。

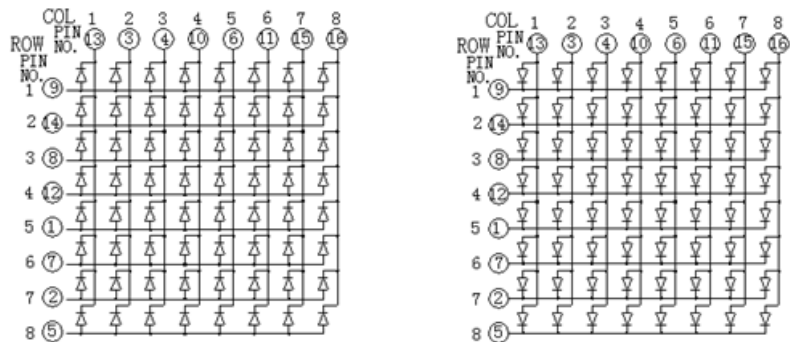


图4-13-2 共阳极LED内部结构图

图4-13-3 共阴极LED内部结构图

在TD-PIT++实验系统上的LED点阵单元采用了4片Φ1.9的共阴极LED点阵组成16×16的点阵。利用取字模软件得到汉字字符数组，设计程序，在点阵上滚动显示“南京航空航天大学”。实验参考接线如图4-13-5所示。

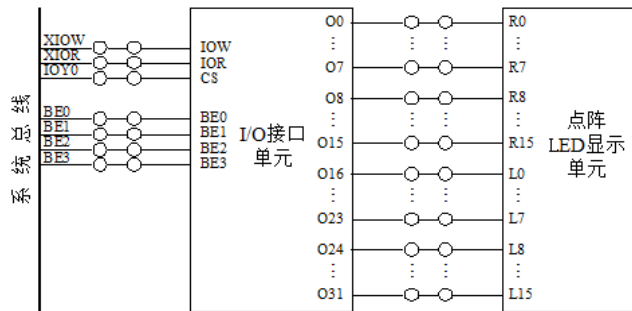


图4-13-4 16×16点阵汉字显示实验参考接线图Pit++)

代码

;Led-HZ.asm,32位LED点阵汉字显示实验

INCLUDE NUAA.inc

.386P

IOY0 EQU 0E000H ;片选IOY0对应的端口始地址

STACK1 SEGMENT STACK
DW 256 DUP(?)

STACK1 ENDS

DATA SEGMENT USE16

ADDR DW ?

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

A2: MOV ADDR,OFFSET HZDOT ;取汉字数组始地址

MOV SI,ADDR

A1: MOV CX,20H ;控制1屏显示时间

LOOP2: CALL DISPHZ

SUB SI,32

LOOP LOOP2

KEY: MOV AH,1 ;判断是否有按键按下?

INT 16H

JNZ QUIT

ADD SI,2

MOV AX,SI

SUB AX,ADDR

CMP AX,288 ;比较文字是否显示完毕

JNB A2

JMP A1

QUIT: MOV EAX,0 ;灭灯

MOV DX,IOY0

OUT DX,EAX

MOV AX,4C00H ;结束程序退出

INT 21H

DISPHZ PROC NEAR ;显示1屏汉字子程序

PUSH CX

MOV CX,16

MOV BX,0FFFEH

LOOP1: MOV AL,BYTE PTR[SI]

MOV AH,BYTE PTR[SI+1]

ROL EAX,16

MOV AX,BX

ADD SI,2

ROL BX,1

NOT EAX

MOV DX,IOY0

OUT DX,EAX

CALL DALLY

LOOP LOOP1

POP CX

RET

DISPHZ ENDP

DALLY PROC NEAR ;软件延时子程序

MOV AX,0FFFFH

D1: DEC AX

JNZ D1

RET

DALLY ENDP
CODE ENDS
END START