

# 第 4 章 基于 Proteus 仿真的课程设计

## 4.1 多功能波形发生器设计

### 1. 设计要求

利用 DAC 芯片和并行接口芯片, 设计一个能输出矩形波、三角波、锯齿波和梯形波的多功能波形发生器, 要求:

- (1) 通过按键选择, 使 DAC 输出矩形波、三角波、锯齿波或梯形波;
- (2) 通过按键可以改变输出波形的幅值, 例如, 利用 1~5 这 5 个数字键改变其输出波形的幅值, 当按下 1~5 数字键时, 使 D/A 输出幅值从 1V 增加到 5V;
- (3) 用示波器观察输出波形。

### 2. 设计分析

DAC 芯片选用 DAC0832, 构成多功能波形发生器的模拟输出通道, 假设其工作在单缓冲方式, 输出单极性电压, 则输出电压  $V_{OUT}$  与输入数字量  $D$  之间的关系为:

$$V_{OUT} = -\frac{D}{256}V_{REF}$$

$D=0\sim 255$ , 基准电压  $V_{REF}=-5V$ 。

当  $D=0FFH=255$  时, 最大输出电压  $V_{max}=(255/-256)\times(-5V)=4.98V$ ; 当  $D=0$  时, 最小输出电压  $V_{min}=(0/-256)\times(-5V)=0V$ 。

DAC 输出模拟量与输入数字量成正比, 通过程序向 DAC 输出随时间规律变化的数字量, 在 DAC 输出端便可以得到规则波形。因此通过控制输出最大电压即可得到不同的输出幅值, 把 0~255 分成 5 个挡:

- 1 挡的最大幅值为 1V, 因此对应的  $D_1=51$ ;
- 2 挡的最大幅值为 2V, 因此对应的  $D_2=102$ ;
- 3 挡的最大幅值为 3V, 因此对应的  $D_3=153$ ;
- 4 挡的最大幅值为 4V, 因此对应的  $D_4=204$ ;
- 5 挡的最大幅值为 5V, 因此对应的  $D_5=255$ 。

按下数字键  $i$  时 ( $i=1\sim 5$ ), 设置不同的  $D_i$  值, 就可以得到对应的输出幅值。

此外还需要通过按键来选择输出 4 种不同的波形, 按键至少需要 10 个(包含 1 个停止键), 为此采用 4×4 矩阵式键盘。假设用 8255 来控制按键, 用 PC0~PC3 作为行线扫描输出, 用 PC4~PC7 作为列线状态输入。16 个按键中, 1~5 号数字键作为幅值选择, A-矩形波, B-三角波, C-锯齿波, D-梯形波, F-停止。为了调试方便, 在 8255 的 PA 口连接了一个共阴极七段数码管, 显示按下的按键键值, 用于观察按键与波形变化的关系。由此可知 8255 工作在方式 0, PC4~PC7 输入, PC0~PC3 输出, PA 输出, 因此控制字为 10001000B。

多功能波形发生器原理图见图 4.1.1。主程序流程图见图 4.1.2, 其中键盘扫描子程序可参考 3.8.3 节。

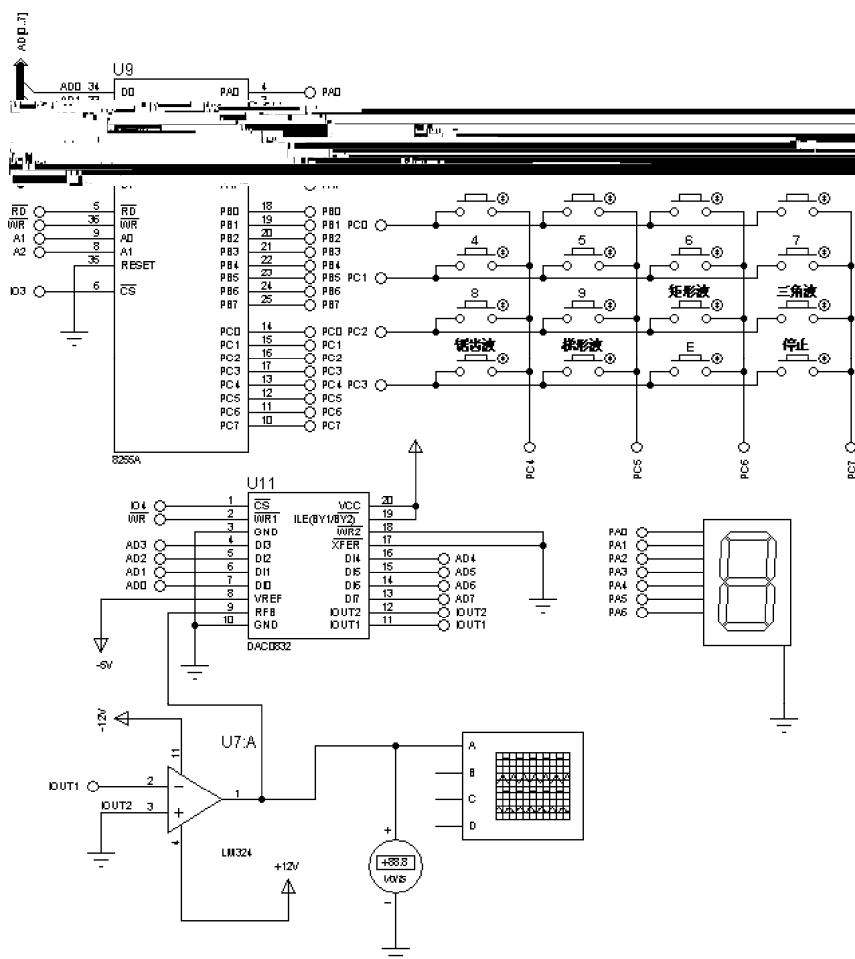


图 4.1.1 多功能波形发生器原理图

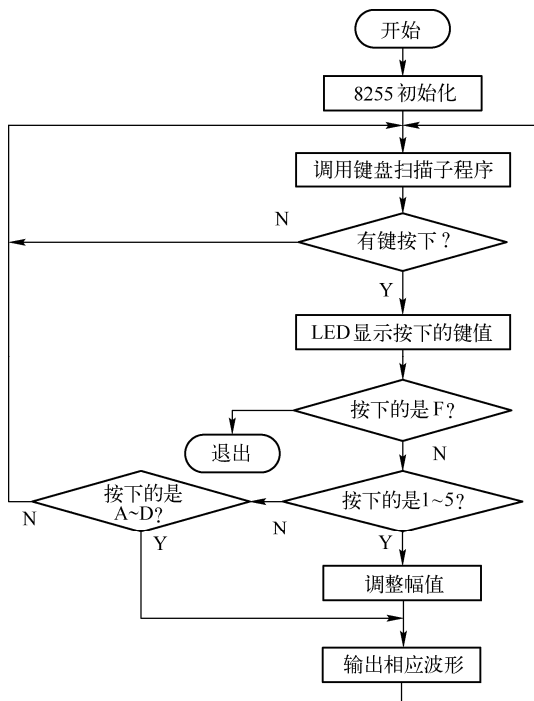


图 4.1.2 多功能波形发生器主程序流程图

## 参考程序 p4.1.1:

```

P8255_BASE EQU 02B0H
P8255_A     EQU P8255_BASE+0
P8255_B     EQU P8255_BASE+2
P8255_C     EQU P8255_BASE+4
P8255_CON  EQU P8255_BASE+6
P0832      EQU 02C0H

DATA        SEGMENT
    ORG     1000H
    TABLE DB 3FH, 06H, 5BH, 4FH, 66H      ;共阴极数码管段码 0,1,2,3,4
           DB 6DH, 7DH, 07H, 7FH, 6FH      ;5,6,7,8,9
           DB 77H, 7CH, 39H, 5EH, 79H, 71H ;A~F
    VAL     DB 51                          ;幅值, 初值对应 1 档
    CVAL    DB 0                            ;记录当前应该输出的波形
    KEYVAL  DB 0FFH

DATA        ENDS
CODE        SEGMENT
    ASSUME  CS:CODE, DS:DATA
START:     MOV     AX, DATA
           MOV     DS, AX
           MOV     DX, P8255_CON           ;初始化 8255
           MOV     AL, 10001000B          ;方式 0, PC4~PC7 输入, PC0~PC3 输出, PA 输出
           OUT    DX, AL
           MOV     BL, CVAL                ;BL 记录波形状态, 初始无输出
AGAIN:     MOV     BH, VAL                 ;BH 记录波形幅值
           CALL   SCAN                    ;调用键盘扫描子程序
           CMP    KEYVAL, 0FFH           ;无按键按下时, 输出原波形
           JZ     OUTPUTC
           ;-----LED 输出键值-----
           PUSH   BX
           MOV    AL, KEYVAL
           MOV    BX, OFFSET TABLE      ;取共阴极数码管段码表首地址
           XLAT                                ;查表取按键对应的段码
           MOV    DX, P8255_A            ;送段码至数码管所在端口
           OUT    DX, AL
           PUSH   BX
           ;-----键值判断-----
           MOV    AL, KEYVAL              ;AL 中记录键值, 方便判断
           CMP    AL, 0FH                 ;停止键按下
           JZ     EXITP                   ;退出
           MOV    BL, AL
KEY1:      CMP    BL, 01H                 ;按下数字键 1 吗?
           JNZ   KEY2                     ;否, 判断数字键 2
           MOV    VAL, 51                 ;设置幅值 1
           JMP    OUTPUTC                 ;继续输出上次的波形
KEY2:      CMP    BL, 02H                 ;按下数字键 2 吗?
           JNZ   KEY3                     ;否, 判断数字键 3
           MOV    VAL, 102                ;设置幅值 2
           JMP    OUTPUTC                 ;继续输出上次的波形

```

```

KEY3:  CMP    BL, 03H      ;按下数字键 3 吗?
       JNZ    KEY4        ;否, 判断数字键 4
       MOV    VAL, 153    ;设置幅值 3
       JMP    OUTPUTC     ;继续输出上次的波形
KEY4:  CMP    BL, 04H      ;按下数字键 4 吗?
       JNZ    KEY5        ;否, 判断数字键 5
       MOV    VAL, 204    ;设置幅值 4
       JMP    OUTPUTC     ;继续输出上次的波形
KEY5:  CMP    BL, 05H      ;按下数字键 5 吗?
       JNZ    KEYA        ;否, 判断是否是功能键
       MOV    VAL, 255    ;设置幅值 5
       JMP    OUTPUTC     ;继续输出上次的波形

KEYA:  CMP    BL, 0AH      ;判断按下的是否是 A~D 功能键
       JB     AGAIN
       CMP    BL, 0DH
       JA     AGAIN        ;不是则重新检测
       MOV    CVAL, BL    ;如果是则修改 CVAL 值为对应的波形键值
       JMP    OUTPUTD
EXITP: JMP    EXIT        ;退出跳转的中转点
       ;-----波形输出-----
OUTPUTC:MOV    BH, VAL    ;BH 记录最新的波形幅值
OUTPUTD:        ;设置波形
           CMP    CVAL, 0AH ;按下键 A(输出矩形波)吗?
           JZ     OUT1      ;是, 转矩形波输出
           CMP    CVAL, 0BH ;按下键 B(输出三角波)吗?
           JZ     OUT2      ;是, 转三角波输出
           CMP    CVAL, 0CH ;按下键 C(输出锯齿波)吗?
           JZ     OUT3      ;是, 转锯齿波输出
           CMP    CVAL, 0DH ;按下键 D(输出梯形波)吗?
           JZ     OUT4      ;是, 转梯形波输出
           JMP    AGAIN     ;没有调整波形输出, 继续扫描按键
OUT1:  MOV    DX, P0832    ;输出矩形波
       MOV    AL, 00H
       OUT    DX, AL
       CALL   DELAY
       MOV    AL, BH      ;设置波形极值
       OUT    DX, AL
       CALL   DELAY
       JMP    AGAIN
OUT2:  MOV    DX, P0832    ;输出三角波
       MOV    AL, 00H
UP2:   OUT    DX, AL
       INC    AL
       CMP    AL, BH      ;判断是否到最大值
       JNZ    UP2
DOWN2: DEC    AL
       OUT    DX, AL
       JNZ    DOWN2
       JMP    AGAIN

```

```

OUT3:  MOV    DX, P0832          ;输出锯齿波
        MOV    AL, 00H
UP3:   OUT    DX, AL
        INC    AL
        CMP    AL, BH          ;判断是否到最大值
        JNZ   UP3
        JMP   AGAIN

OUT4:  MOV    DX, P0832          ;输出梯形波
        MOV    AL, 00H
UP4:   OUT    DX, AL
        INC    AL
        CMP    AL, BH          ;判断是否到最大值
        JNZ   UP4
        CALL  DELAY
DOWN4: DEC    AL
        OUT    DX, AL
        JNZ   DOWN4
        JMP   AGAIN

EXIT:  JMP    $

SCAN   PROC                    ;键盘扫描子程序
        PUSH  DX
        PUSH  BX
        PUSH  CX
        PUSH  AX
KEY:   MOV    DX, P8255_C        ;取 C 口地址
        MOV    AL, 0
        OUT   DX, AL           ;输出 PC0~PC3 为 0 的扫描信号
        IN    AL, DX           ;读列状态, 其中有 1 位为 0 表示有键按下
        NOT   AL               ;取反后, 相应位为 1 表示有键按下
        TEST  AL, 0F0H         ;检测是否有键按下
        JZ    L1               ;如果没有键按下 AL=0, 跳转设置返回值
        CALL  DELAY10          ;软件延时去抖
;-----确定有键按下后, 判断具体哪一行有键按下-----
        MOV   CH, 0FEH         ;取初始扫描信号, 从第 0 行开始
        MOV   BL, 0            ;BL 记录行号
ROW:   MOV    AL, CH            ;取当前行进行扫描
        OUT   DX, AL           ;输出行扫描信号
        IN    AL, DX           ;读当前行按键状态
        NOT   AL               ;列反后, 相应位为 1 表示有键按下
        AND   AL, 0F0H         ;判断当前行是否有键按下?
        JNZ   NEXT            ;有键按下, 跳转至 NEXT
        INC   BL               ;无键按下, 行号加 1, 扫描下一行
        CMP   BL, 4            ;判断是否扫描完 4 行?
        JZ    L1               ;扫描完, 没有键按下 AL=0, 跳转设置返回值
        ROL   CH, 1            ;没有扫描完, 行扫描信号指向下一扫描行
        JMP   ROW              ;继续行扫描
NEXT:  MOV    BH, 3            ;计算被按键所在列号, 初值为 3

```

```

        MOV     CX, 4
COLUMN: TEST  AL, 80H           ;当前列有键按下?
        JNZ   EXIT1        ;有键按下, BH 为列号
        SHL   AL, 1        ;指向下一列
        DEC   BH           ;列号减 1
        LOOP  COLUMN       ;检测下一列
L1:     MOV   AL, 0FFH     ;没有按键按下
        JMP   QUIT
EXIT1:  MOV   AL, BL       ;计算按键序号, BL*4+BH->AL
        MOV   CL, 2
        SHL   AL, CL
        ADD   AL, BH
QUIT:   MOV   KEYVAL, AL  ;将键值保存在变量 KEYVAL 中
        POP   AX
        POP   CX
        POP   BX
        POP   DX
        RET
SCAN   ENDP

DELAY10 PROC           ;延时子程序 1
        PUSH  CX
        MOV   CX, 882
        LOOP  $
        POP   CX
        RET
DELAY10 ENDP

DELAY   PROC           ;延时子程序 2
        PUSH  CX
        MOV   CX, 0FFH
DELAY0: NOP
        LOOP  DELAY0
        POP   CX
        RET
DELAY   ENDP
CODE   ENDS
END     START

```

## 4.2 数据采集与监视系统设计

### 1. 设计要求

设计一个数据采集与监视系统:

- (1) 从外部输入一个连续变化的电压信号;
- (2) 在数码管上实时显示连续变化的输入电压值, 小数点后保留 2 位;
- (3) 将微机处理后的数据经 DAC 转换为模拟信号送示波器, 使示波器得到一个连续变

化的输出电压。

## 2. 设计分析

可以利用电位器输入连续变化的电压信号；利用 8253 构成定时通道，与 ADC0809 一起构成硬件延时等待式 A/D 转换输入通道，电路原理图如图 4.2.1 所示。

在图 4.2.1 中，ADC0809 各模拟通道采用地址线选择，IN0~IN7 对应的启动地址分别为 02A0H~02A7H，读取转换结果地址为 02A0H~02AFH 中的任意一个偶地址。DAC0832 采用单缓冲方式，端口地址为 02C0H。

由于 ADC0809 的等待时间为 128 $\mu$ s，而要确保 A/D 转换后到读取转换结果之间的等待时间大于 A/D 转换时间，这里取等待时间为 500 $\mu$ s。由 8253 产生周期为 500 $\mu$ s 的定时脉冲，定时时间到，则读取转换结果，进行数据处理；然后将转换的结果显示到数码管上，同时将数据送 DAC0832 进行 D/A 转换，再将转换结果送到示波器上显示。8253 的基地址为 02B0H，通道 0 工作于方式 3，CLK0 加 10kHz 的时钟信号。ADC0809 的 CLOCK 加 500kHz 的时钟信号。

多位数码管的段选和位选地址分别为 0280H 和 0290H。本设计中用到 3 位数码管分别显示电压值的 1 位整数(包括小数点)，2 位小数，靠右显示，因此只需用到图 4.2.1 中的 6、7、8 三位数码管。

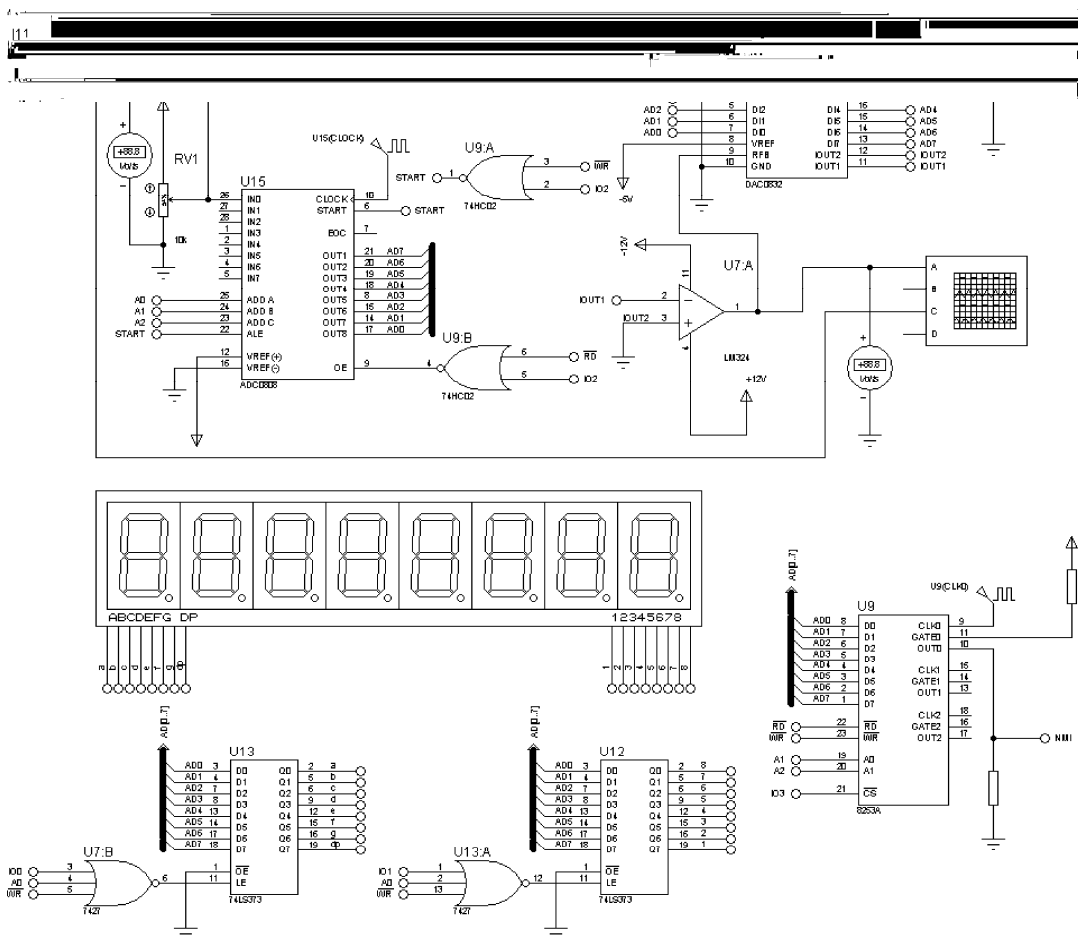


图 4.2.1 数据采集与监视系统原理图

完成本实验的软件包括主程序和中断服务程序两部分。主程序主要完成中断向量初始化、定时/计数器的初始化和首次启动 A/D 转换等，流程图如图 4.2.2 所示。定时中断服务程序主要完成读取转换结果、启动 D/A 转换、进行数据处理和显示结果以及再次启动 A/D 转换等功能；而在定时中断服务程序中，由于设计要求将 A/D 转换的结果显示到数码管上，而显示的电压数值只能以数码管段码的形式输出，这就要求计算机首先将从 ADC 采集而来的二进制数据经标度变换，转换为正确的 BCD 码，再将 BCD 码转换成数码管段码输出到数码管上。定时中断服务程序流程图如图 4.2.3 所示。

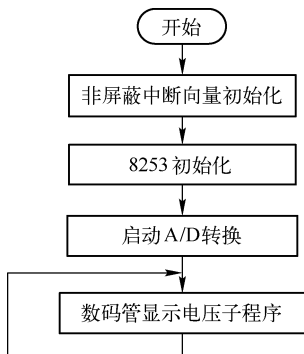


图 4.2.2 数据采集与监视系统主程序流程图

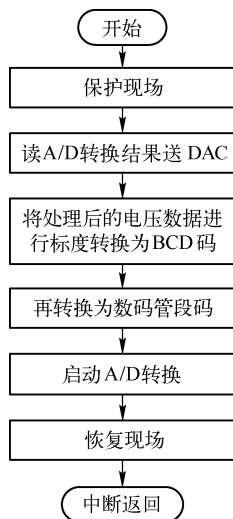


图 4.2.3 定时中断服务程序流程图

### 参考程序 p4.2.1:

```

DUAN373 EQU 0280H
WEI373 EQU 0290H
P0809 EQU 02A0H
P8253_BASE EQU 02B0H
P8253_0 EQU P8253_BASE+0
P8253_1 EQU P8253_BASE+2
P8253_2 EQU P8253_BASE+4
P8253_CON EQU P8253_BASE+6
P0832 EQU 02C0H

DATA SEGMENT
    ORG 1000H
    TABLE DB 3FH, 06H, 5BH, 4FH, 66H ;共阴极数码管段码 0,1,2,3,4
            DB 6DH, 7DH, 07H, 7FH, 6FH ;5,6,7,8,9
    VAL DB 5
    OUTV DB 4 DUP(?)
DATA ENDS

CODE SEGMENT
    ASSUMECS:CODE, DS:DATA
START: MOV AX, DATA
  
```



```

MOV     DS, AX
PUSH   ES                                ;中断向量初始化
XOR    AX, AX
MOV    ES, AX
MOV    AL, 2
XOR    AH, AH
SHL    AL, 1
SHL    AL, 1
MOV    SI, AX
MOV    AX, OFFSET NMI_SERVICE
MOV    ES:[SI], AX
MOV    BX, CS
MOV    ES:[SI+2], BX
POP    ES
MOV    AL, 16H                            ;8253 初始化
MOV    DX, P8253_CON                      ;写控制字, 计数器 0 方式 3, 二进制计数
OUT    DX, AL
MOV    AL, 5                              ;写计数器 0 计数初值, 定时 500us
MOV    DX, P8253_0                        ;只写低 8 位
OUT    DX, AL
MOV    DX, P0809                          ;启动 ADC0809
OUT    DX, AL
AGAIN:  NOP
        CALL  DISP
        JMP   AGAIN
NMI_SERVICE PROC
        PUSH  AX
        PUSH  BX
        PUSH  DX
        PUSH  CX
        PUSH  SI
        MOV   DX, P0809                    ;读 A/D 转换结果
        IN   AL, DX
        MOV   DX, P0832                    ;数据送 D/A 转换
        OUT  DX, AL
        LEA  SI, OUTV                      ;进行数据处理
        MOV  BL, VAL                       ;转换结果*5/255 得到整数部分(标度转换)
        MUL  BL
        MOV  BL, 255
        DIV  BL
        MOV  BX, OFFSET TABLE
        XLAT                                ;转换为数码管段码
        OR   AL, 80H                       ;为整数部分加上小数点
        MOV  [SI], AL
        MOV  CX, 2                          ;小数部分为 2 位
ZH1:   INC  SI                              ;保存小数部分
        MOV  AL, AH                        ;余数送 AL 中
        MOV  BL, 10

```

```

        MUL     BL                ;余数*10/255 得到小数部分
        MOV     BL, 255
        DIV     BL
        MOV     BX, OFFSET TABLE
        XLAT                    ;转换为数码管段码
        MOV     [SI], AL
        LOOP    ZH1
        MOV     DX, P0809        ;启动 0809A/D 转换
        OUT     DX, AL
        POP     SI
        POP     CX
        POP     DX
        POP     BX
        POP     AX
EXIT:    IRET
NMI_SERVICE ENDP

DISP    PROC                    ;显示子程序
        PUSH   BX
        PUSH   SI
        PUSH   CX
        PUSH   DX
        PUSH   AX
        MOV    BL, 11111011B    ;位码初值, 从 6 位开始显示
        MOV    SI, OFFSET OUTV
NEXT:    MOV    DX, WEI373
        MOV    AL, 0FFH
        OUT    DX, AL
        MOV    DX, DUAN373
        MOV    AL, [SI]
        OUT    DX, AL
        MOV    DX, WEI373
        MOV    AL, BL
        OUT    DX, AL
        CALL   DELAY
        ROR    BL, 1
        INC    SI
        LOOP   NEXT
        POP    AX
        POP    DX
        POP    CX
        POP    SI
        POP    BX
        RET
DISP    ENDP

DELAY   PROC                    ;延时子程序

```

```

PUSH    BX
PUSH    CX
MOV     BX, 1
LP1:    MOV     CX, 100
LP2:    LOOP   LP2
        DEC    BX
        JNZ   LP1
        POP   CX
        POP   BX
        RET
DELAY  ENDP
CODE   ENDS
END    START

```

如图 4.2.4 所示，当调节滑动变阻器使得电压表 1 读数为 4.5V 时，运算放大器输出的电压读数为 4.47V，相应地数码管显示的 A/D 转换结果为 4.49V，见图 4.2.5。数码管输出与电压表 1 的输出不同，是 A/D 转换及标度转换带来的误差造成的。而电压表 2 的读数与其他两者不同，是因为 D/A 转换进一步引入了误差。可以验证，虚拟示波器输入的 2 路信号基本是同步变化的。

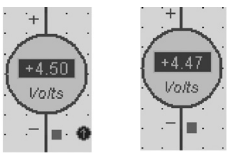


图 4.2.4 电压表读数示例



图 4.2.5 数码管显示转换结果示例

## 4.3 用 LED 显示的电子时钟设计

### 1. 设计要求

设计一个基于 LED 显示器显示的电子时钟。要求：

- (1) 在 LED 显示器上以 HH-MM-SS(时-分-秒)的形式显示时间；
- (2) 时间每秒更新一次；
- (3) 当前时间可以调整。

### 2. 设计分析

硬件电路主要分为秒定时信号产生电路、LED 显示器接口、按键接口。假设秒定时信号产生电路用 8253 构成；LED 显示器采用多位数码管动态扫描、分时循环显示原理，用 8255 的 PA 口和 PB 口分别作为段选和位选；8255 的 PC 口接 3 个独立式按键分别用作时、分、秒的调整，见图 4.3.1。

假设 8253 的 CLK1 输入的时钟信号频率为 100kHz，为了实现 1s 的定时，需要 2 级计数通道级联，计数器 1 的计数初值设为 1000，工作在方式 3，OUT1 输出的是 1kHz 的方波信号，同时作为 CLK2 的输入，计数器 2 工作在方式 2，计数初值为 100，输出周期为 1s 的负脉冲，每隔 1s 向 8086 发一次中断请求。这里采用的是非屏蔽中断，也可以采用可屏蔽中

断，并利用 8259 来管理。

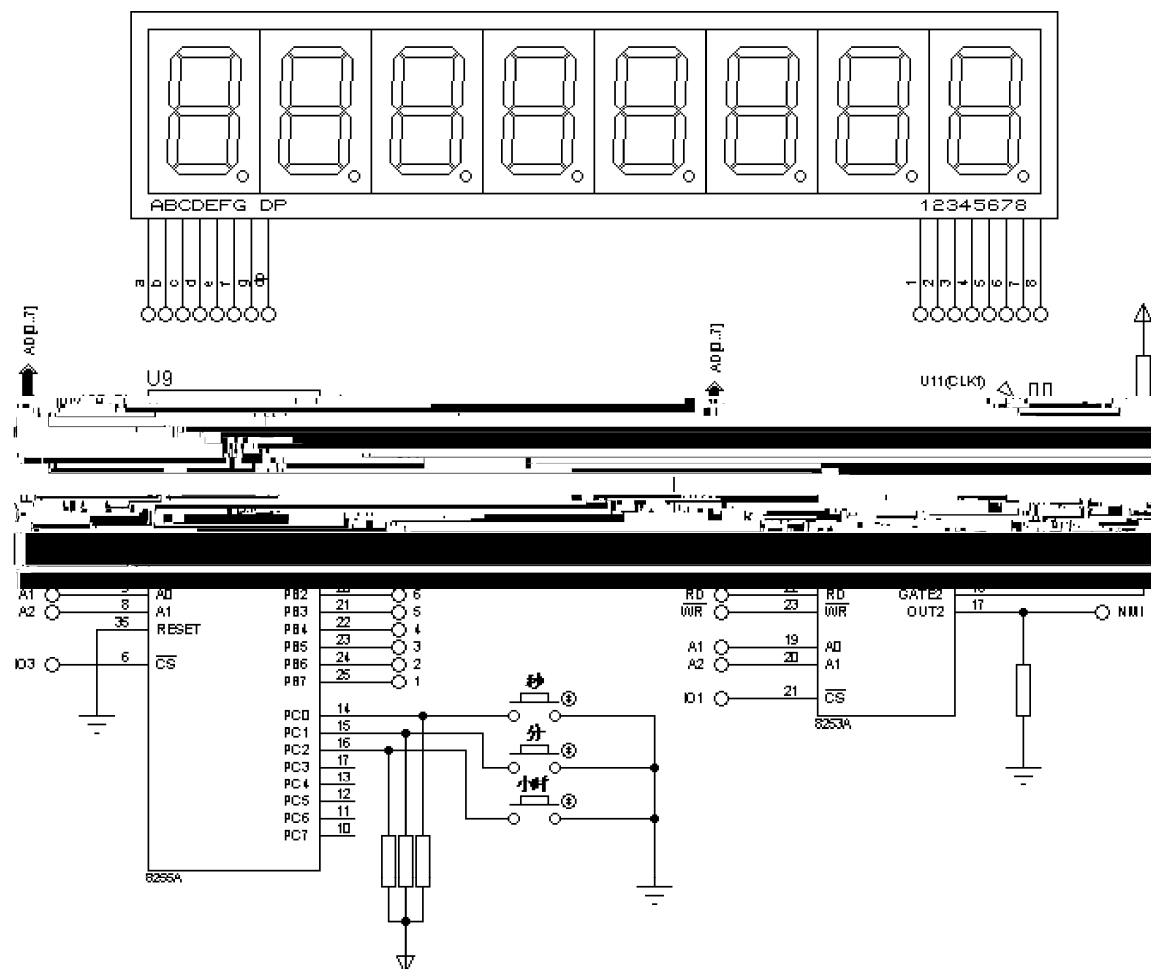


图 4.3.1 用 LED 显示的电子时钟实验原理图

在此基础上，实现电子时钟的原理是：利用秒定时产生电路产生周期性的定时中断信号，每隔 1s 向微机发送一次 NMI 中断请求；CPU 响应 NMI 中断时，在中断服务程序中通过软计数来获得秒、分、时的值。时、分、秒各对应一个变量，在中断服务程序按 BCD 码规则进行加 1 运算。

程序主要包括：

(1) 主程序：主要包括一些初始化操作(设置时间初值、初始化中断向量表、初始化 8253 等)，调用 LED 显示程序和按键调整子程序，相应的流程图见图 4.3.2。

(2) 中断服务子程序：用于时、分、秒的计数处理，相应的流程图见图 4.3.3。

(3) LED 显示子程序：从时、分、秒的计数变量中取出时间数据(BCD 码)，分离成时、分、秒的个位、十位共 6 个字节，转换成数码管的段码，以及时与分、分与秒之间的间隔符“-”的段码依次送 LED 显示器显示。

(4) 按键调整子程序：检测“小时”、“分”和“秒”等按键是否被按下，按下则相应的变量按 BCD 码规则进行加 1 处理。按键检测时需要注意去抖和重复计数等问题的处理。相

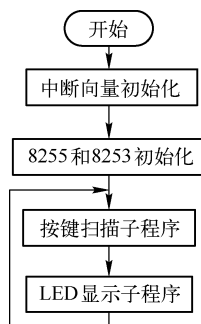


图 4.3.2 主程序流程图

应的流程图见图 4.3.4。

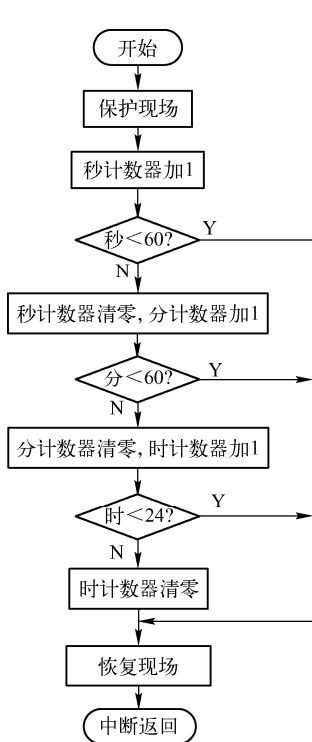


图 4.3.3 定时中断处理程序流程图

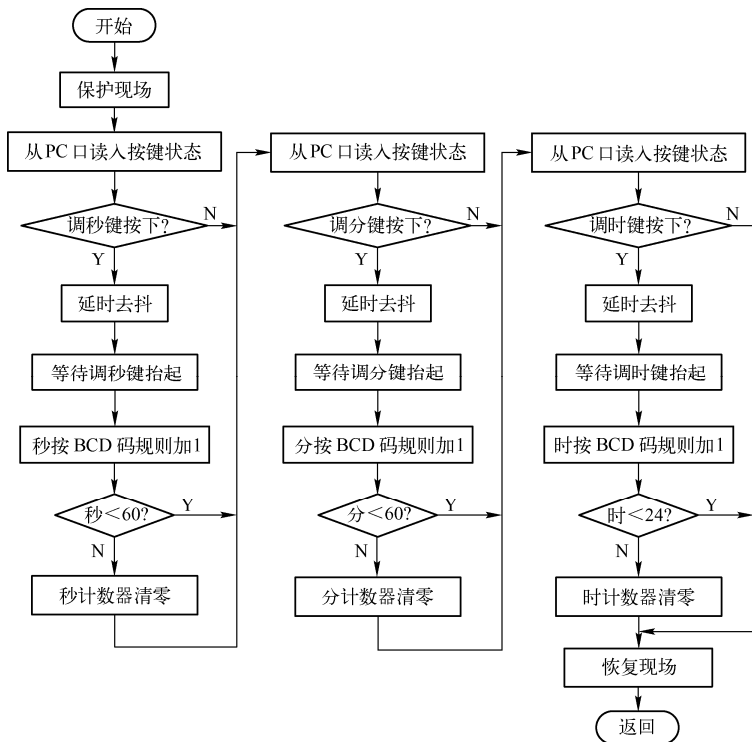


图 4.3.4 按键调整子程序流程图

### 参考程序 p4.3.1:

```

IOCON EQU 02B6H ;8255 端口定义
IOA EQU 02B0H
IOB EQU 02B2H
IOC EQU 02B4H
TCONTRO EQU 0296H ;8253 端口定义
TCON0 EQU 0290H
TCON1 EQU 0292H
TCON2 EQU 0294H
DATA SEGMENT
    ORG 1000H
    TABLE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH
    ;0~9 的共阴极数码管段码
    TABLE_END = $
    SEC DB 58H
    MIN DB 59H
    HOU DB 23H
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA
        MOV DS, AX
NMI_INIT: ;NMI 中断向量初始化
        PUSH ES
  
```

```

XOR     AX, AX                ;AX 清 0
MOV     ES, AX
MOV     AL, 2                 ;NMI 中断类型码为 2
XOR     AH, AH                ;AH 清空
SHL     AL, 1                 ;2*4
SHL     AL, 1
MOV     SI, AX                ;SI 指向中断向量在中断向量表中的起始地址
MOV     AX, OFFSET NMI_SERVICE ;取中断服务程序的偏移地址
MOV     ES:[SI], AX           ;送中断向量表
MOV     BX, CS                ;取中断服务程序的段地址
MOV     ES:[SI+2], BX        ;送中断向量表
POP     ES

MOV     AL, 10000001B         ;8255 初始化
MOV     DX, IOCON             ;方式 0, PA 口 PB 口输出, PC 口低 4 位输入
OUT     DX, AL

MOV     AL, 10110101B         ;8253 初始化
MOV     DX, TCONTRO           ;T2 读写高低字节工作方式 2 BCD 码
OUT     DX, AL                ;写入控制字
MOV     DX, TCON2             ;T2 地址
MOV     AL, 00H               ;写入计数初值低字节
OUT     DX, AL
MOV     AL, 01H               ;写入计数初值高字节
OUT     DX, AL
MOV     AL, 01110111B         ;T1 读写高低字节工作方式 3 BCD 码
MOV     DX, TCONTRO           ;写入控制字
OUT     DX, AL
MOV     DX, TCON1             ;T1 地址
MOV     AL, 00H               ;写入计数初值低字节
OUT     DX, AL
MOV     AL, 10H               ;写入计数初值高字节
OUT     DX, AL
MAIN:   CALL    KEY
        CALL    DISP
        JMP     MAIN

NMI_SERVICE:                    ;中断服务子程序
        PUSH   AX
        MOV    AL, SEC           ;每中断一次, 秒按 BCD 码加 1
        ADD    AL, 1
        DAA
        MOV    SEC, AL
        CMP    SEC, 60H
        JB    EXIT
        MOV    SEC, 0
        MOV    AL, MIN
        ADD    AL, 1
        DAA
        MOV    MIN, AL
        CMP    MIN, 60H

```

```

        JB      EXIT
        MOV     MIN, 0
        MOV     AL, HOU
        ADD     AL, 1
        DAA
        MOV     HOU, AL
        CMP     HOU, 24H
        JB      EXIT
        MOV     HOU, 0
EXIT:    POP     AX
        IRET

DISP    PROC                                ;显示子程序
        MOV     AL, 0FFH
        MOV     DX, IOB                    ;位选
        OUT     DX, AL
        ;-----秒-----
        MOV     BX, OFFSET TABLE        ;取秒的个位
        MOV     AL, SEC
        AND     AL, 0FH
        XLAT
        MOV     DX, IOA                    ;送段码
        OUT     DX, AL
        MOV     AL, 11111110B            ;选通秒个位
        MOV     DX, IOB
        OUT     DX, AL
        CALL    DELAY
        MOV     AL, 0FFH
        MOV     DX, IOB                    ;消影
        OUT     DX, AL
        MOV     AL, SEC
        AND     AL, 0F0H                    ;分离秒的十位
        MOV     CL, 4
        SHR     AL, CL
        MOV     BX, OFFSET TABLE        ;取秒的个位
        XLAT
        MOV     DX, IOA                    ;送段码
        OUT     DX, AL
        MOV     AL, 11111101B            ;选通秒十位
        MOV     DX, IOB
        OUT     DX, AL
        CALL    DELAY
        MOV     AL, 0FFH
        MOV     DX, IOB                    ;消影
        OUT     DX, AL
        ;-----"-"-----
        MOV     AL, 40H                    ;"-"的段码
        MOV     DX, IOA                    ;送段码
        OUT     DX, AL
        MOV     AL, 11111011B            ;选通"-"位
        MOV     DX, IOB

```

```

OUT    DX, AL
CALL   DELAY
MOV    AL, 0FFH
MOV    DX, IOB          ;消影
OUT    DX, AL
; -----分-----
MOV    BX, OFFSET TABLE ;取分的个位
MOV    AL, MIN
AND    AL, 0FH
XLAT
MOV    DX, IOA          ;送段码
OUT    DX, AL
MOV    AL, 11110111B    ;选通分个位
MOV    DX, IOB
OUT    DX, AL
CALL   DELAY
MOV    AL, 0FFH
MOV    DX, IOB          ;消影
OUT    DX, AL
MOV    AL, MIN
AND    AL, 0F0H        ;分离分的十位
MOV    CL, 4
SHR    AL, CL
MOV    BX, OFFSET TABLE ;取分的个位
XLAT
MOV    DX, IOA          ;送段码
OUT    DX, AL
MOV    AL, 11101111B    ;选通分十位
MOV    DX, IOB
OUT    DX, AL
CALL   DELAY
MOV    AL, 0FFH
MOV    DX, IOB          ;消影
OUT    DX, AL
; -----"-"------
MOV    AL, 40H          ;"-"-的段码
MOV    DX, IOA          ;送段码
OUT    DX, AL
MOV    AL, 11011111B    ;选通"-"-位
MOV    DX, IOB
OUT    DX, AL
CALL   DELAY
MOV    AL, 0FFH
MOV    DX, IOB          ;消影
OUT    DX, AL
; -----时-----
MOV    BX, OFFSET TABLE ;取时的个位
MOV    AL, HOU
AND    AL, 0FH
XLAT

```



```

MOV     DX, IOA           ;送段码
OUT     DX, AL
MOV     AL, 10111111B    ;选通时个位
MOV     DX, IOB
OUT     DX, AL
CALL    DELAY
MOV     AL, 0FFH
MOV     DX, IOB           ;消影
OUT     DX, AL
MOV     AL, HOU
AND     AL, 0F0H         ;分离时的十位
MOV     CL, 4
SHR     AL, CL
MOV     BX, OFFSET TABLE ;取时的个位
XLAT
MOV     DX, IOA           ;送段码
OUT     DX, AL
MOV     AL, 01111111B    ;选通时十位
MOV     DX, IOB
OUT     DX, AL
CALL    DELAY
RET
DISP   ENDP

KEY    PROC               ;按键调整子程序
PUSH    AX
PUSH    DX
MOV     DX, IOC
IN      AL, DX
TEST    AL, 01H          ;调秒键是否按下
JNZ     NEXTMIN          ;未按下则检测调分按键
CALL    DELAY
MOV     DX, IOC           ;消抖
IN      AL, DX
TEST    AL, 01H
JNZ     NEXTMIN
L1:    MOV     DX, IOC     ;等待调秒按键抬起
IN      AL, DX
TEST    AL, 01H
JZ      L1
MOV     AL, SEC           ;调秒
ADD     AL, 1
DAA
MOV     SEC, AL
CMP     SEC, 60H
JB      NEXTMIN
MOV     SEC, 0
NEXTMIN:MOV    DX, IOC
IN      AL, DX
TEST    AL, 02H          ;调分键是否按下

```

```

        JNZ     NEXTHOU           ;未按下则检测调时按键
        CALL   DELAY
        MOV    DX, IOC           ;消抖
        IN     AL, DX
        TEST   AL, 02H
        JNZ     NEXTHOU
L2:     MOV    DX, IOC           ;等待调分按键抬起
        IN     AL, DX
        TEST   AL, 02H
        JZ     L2
        MOV    AL, MIN           ;调分
        ADD    AL, 1
        DAA
        MOV    MIN, AL
        CMP    MIN, 60H
        JB     NEXTHOU
        MOV    MIN, 0
NEXTHOU: MOV    DX, IOC
        IN     AL, DX
        TEST   AL, 04H           ;调时键是否按下
        JNZ     EXITKEY         ;未按下则退出
        CALL   DELAY
        MOV    DX, IOC           ;消抖
        IN     AL, DX
        TEST   AL, 04H
        JNZ     EXITKEY
L3:     MOV    DX, IOC           ;等待调时按键抬起
        IN     AL, DX
        TEST   AL, 04H
        JZ     L3
        MOV    AL, HOU           ;调时
        ADD    AL, 1
        DAA
        MOV    HOU, AL
        CMP    HOU, 24H
        JB     EXITKEY
        MOV    HOU, 0
EXITKEY: POP    DX
        POP    AX
        RET
KEY     ENDP

DELAY   PROC                   ;延时子程序
        PUSH  BX
        PUSH  CX
        MOV   BX, 1
LP1:    MOV    CX, 469
LP2:    LOOP   LP2
        DEC   BX
        JNZ   LP1

```