

# 《微机原理与接口技术》

## 实验报告

实验题目：数字电子时钟

院系：信息科学技术学院

实验人员：孙越 PB16061434

陈晓冬 PB16120385

史张翔 PB16051056

# 目录

引言 .....	3
一、实验目的 .....	4
二、实验设计 .....	4
1、内容 .....	4
2、要求 .....	4
3、设计环境 .....	5
4、设计所用设备 .....	5
5、接线实现 .....	5
三、硬件芯片 .....	6
1、8254 芯片的内部结构及引脚 .....	6
2、8255 芯片的内部结构及引脚 .....	7
3、8259A 芯片的内部结构及引脚 .....	8
四、具体程序实现 .....	12
1、概述 .....	12
2、程序流程图 .....	12
3、主程序模块 .....	13
4、定时模块 .....	15
5、中断模块 .....	16
(1)、中断子程序主部分 .....	16
(2)、时间转换子程序模块 .....	17
(3)、时间递增子程序模块 .....	18
6、显示子程序模块 .....	19
五、汇编代码 .....	19
六、实验总结与收获 .....	26

# 引言

计算机的产生加快了人类改造世界的步伐，但是体积大，随着人们的生活越来越离不开计算机。微型处理器在这种情况下应运而生。纵观各个领域，从导弹的导航装置，到飞机上各种仪表的控制，从计算机的网络通讯与数据传输，到工业自动化过程的实时控制和数据处理，以及生活中使用的各种智能 IC 卡、电子宠物等，这些都离不开微型计算机。微机即是集 CPU ,RAM ,ROM ,定时，计数和多种接口于一体的微控制器。它体积小，成本低，功能强，广泛应用于智能产业和工业自动化上。

这次课程设计的题目是：数字电子时钟的设计。计时精确的电子钟在我们生活中能处处能见到。钟表的数字化给人们生产生活带来了极大的方便，而且大大地扩展了钟表原先的报时功能。诸如定时自动报警、按时自动打铃、时间程序自动控制、定时广播、定时启闭电路、定时开关烘箱、通断动力设备，甚至各种定时电气的自动启用等，所有这些，都是以钟表数字化为基础的。因此，研究数字钟及扩大其应用，有着非常现实的意义。通过对一个学期的微机原理与接口技术的学习，我们可以利用我们所学过的知识来实现数字电子钟的工作过程。我们选择数码管显示，8253 定时功能、8255 基本输入输出的功能、8259 的中断功能来实现数字电子钟的设计。其中附有电路原理图，电路接线图和源程序。因水平有限，难免有疏落不足之处，敬请指导老师批评指正。

## 一、实验目的

通过本次课程设计使学生更进一步掌握微机原理及应用课程的有关知识,灵活运用课本及所学知识,提高应用微机解决问题的能力;加深对所学的相关芯片的原理、内部结构、使用方法等有更加深刻的理解,学会利用课本知识联系实际应用及编程。通过查阅资料,结合所学知识进行软、硬件的设计,使学生初步掌握应用微机解决问题的步骤及方法。为以后学生结合专业从事微机应用设计奠定基础。

## 二、实验设计

### 1、内容

利用实验台上提供的可编程定时/计数器 8254、可编程中断控制器 8259A 和可编程并行接口芯片 8255 以及七段数码管显示电路设计一个电子时钟,并编制程序使得该电子时钟能够正常运行。

### 2、要求

本实验要求设计一个实时显示装置,用六个数码管显示时间,精确到秒。电子时钟的显示格式 HH:MM:SS,由左到右分别为时、分、秒,最大显示时间为 23:59:59,若超过则时分秒位清零从 00:00:00 重新开始

基本工作原理:利用 8254 可编程定时/计数器,通过 DOS 系统功能调用,对其送入一个计数初值,然后将输出的脉冲送到 8259 的 IR1 产生中断,在 8259 中断处理程序中,对时、分、秒进行计数,在等待中断的循环中,采用 8255 芯片,利用七段数码管显示时间,将 8255 的两个输出端口 A 口和 B 口实现其位选与段选。在主程序中要分别对 8254、8255、8259A 进行初始化编程,8254 的计数器 0 可在方式 3 下工作。时、分、秒分别对应 6 个存储单元,分别存放时、分、

秒的个位与十位。在中断来时，将秒数的个位加 1，判断是否到 10,如到了则十位加 1，个位清零；再判断十位是否到了 6,如到了则十位清零，分的个位加 1，同理对分、时做相应处理。七段数码管显示作为子程序，将时、分、秒对应存储单元的内容分别取出并查表转换为相应的代码，利用缓冲区和延时子程序进行显示。

3、设计环境

PC 机一台、WindowsXP 系统、实验箱、导线若干

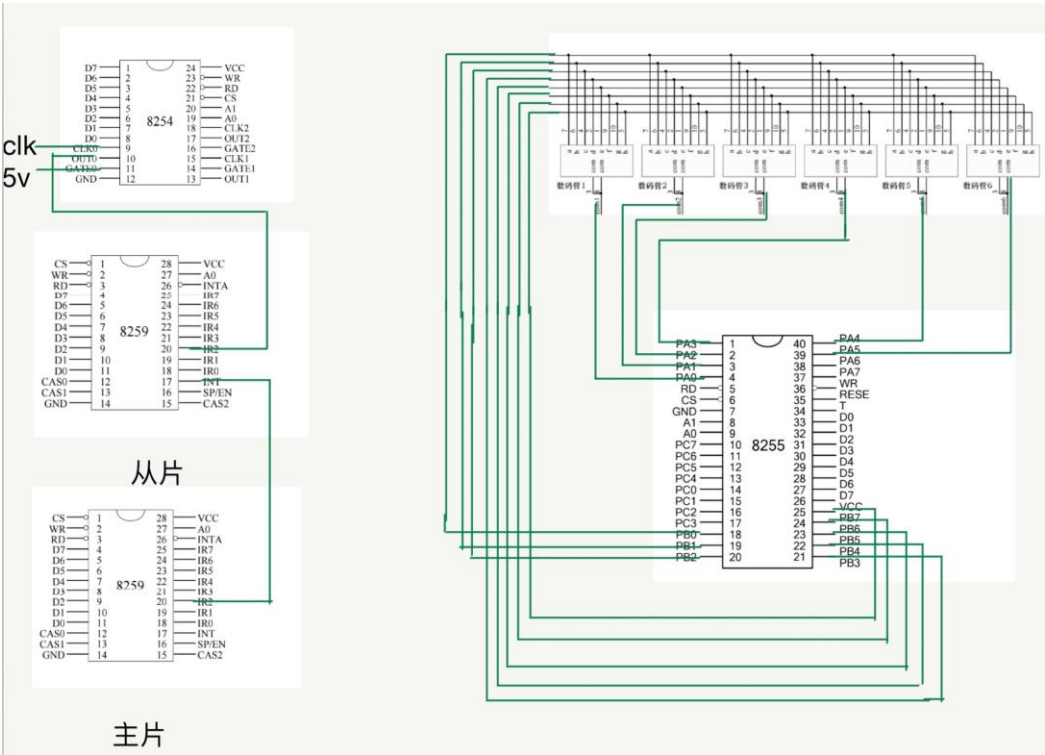
4、设计所用设备

8254 定时器：用于产生秒脉冲，其输出脉冲信号可作为中断请示信号送到 IR1

8255 并行接口：用作接口芯片，和数码管相连接以达到控制数码管显示实时时间

8259A 可编程中断控制器：用于产生每隔一秒的中断，以达到实时计时

5、接线实现



### 三、硬件芯片

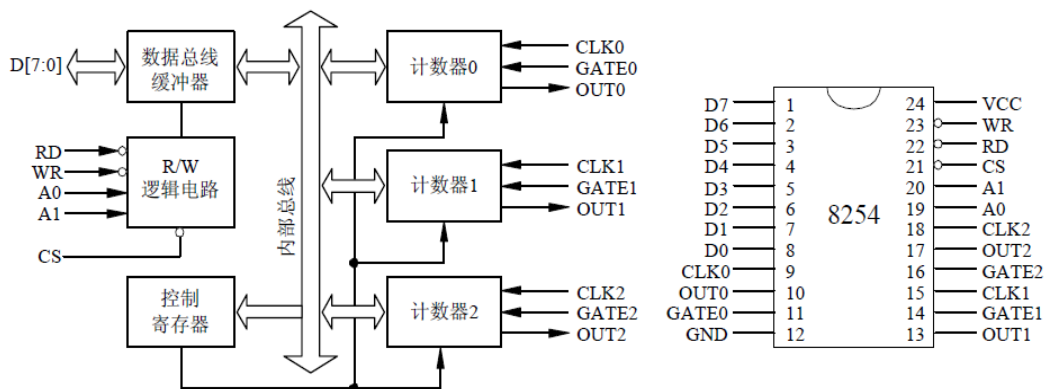
#### 1、8254 芯片的内部结构及引脚

8254 是 Intel 公司生产的可编程间隔定时器,是 8253 的改进型,比 8253 具有更优良的性能。8254 具有以下基本功能:

- (1) 有 3 个独立的 16 位计数器。
- (2) 每个计数器可按二进制或十进制 (BCD) 计数。
- (3) 每个计数器可编程工作于 6 种不同工作方式。
- (4) 8254 每个计数器允许的最高计数频率为 10MHz (8253 为 2MHz)。
- (5) 8254 有读回命令 (8253 没有),除了可以读出当前计数单元的内容外,还可以读出状态寄存器的内容。

(6) 计数脉冲可以是有规律的时钟信号,也可以是随机信号.计数初值公式为:  
$$n = f_{CLKi} \div f_{OUTi}$$
 其中  $f_{CLKi}$  是输入时钟脉冲的频率,  $f_{OUTi}$  是输出波形的频率。

8254 由与 CPU 的接口,内部控制电路和三个计数器组成.下图为内部接口与引脚



8254 的工作方式如下述:

- (1) 方式 0: 计数到 0 结束输出正跃变信号方式。
- (2) 方式 1: 硬件可重触发单稳方式。
- (3) 方式 2: 频率发生器方式 (自动重装初值)。

(4) 方式 3: 方波发生器 (自动重装初值) 。

(5) 方式 4: 软件触发选通方式。

(6) 方式 5: 硬件触发选通方式 (自动重装初值) 。

8254 的控制字有两个: 一个用来设置计数器的工作方式, 称为方式控制字; 另一个用来设置读回命令, 称为读回控制字。这两个控制字共用一个地址, 由标识位来区分。控制字格式如下表所示。

D7	D6	D5	D4	D3	D2	D1	D0
计数器选择		读/写格式选择		工作方式选择			计数码制选择
00—计数器 0		00—锁存计数值		000—方式 0			0—二进制数
01—计数器 1		01—读/写低 8 位		001—方式 1			1—十进制数
10—计数器 2		10—读/写高 8 位		010—方式 2			
11—读出控制 字标志		11—先读/写低 8 位 再读/写高 8 位		011—方式 3			
				100—方式 4			
				101—方式 5			

端口地址:

CS2 EQU 3040H ;片选 CS2 对应的 8254 端口始地址  
MY8254\_COUNT0 EQU CS2+00H ;8254 计数器 0 端口地址  
MY8254\_MODE EQU CS2+03H ;8254 控制寄存器端口地址

## 2、8255 芯片的内部结构及引脚

并行接口是以数据的字节为单位与 I/O 设备或被控制对象之间传递信息。

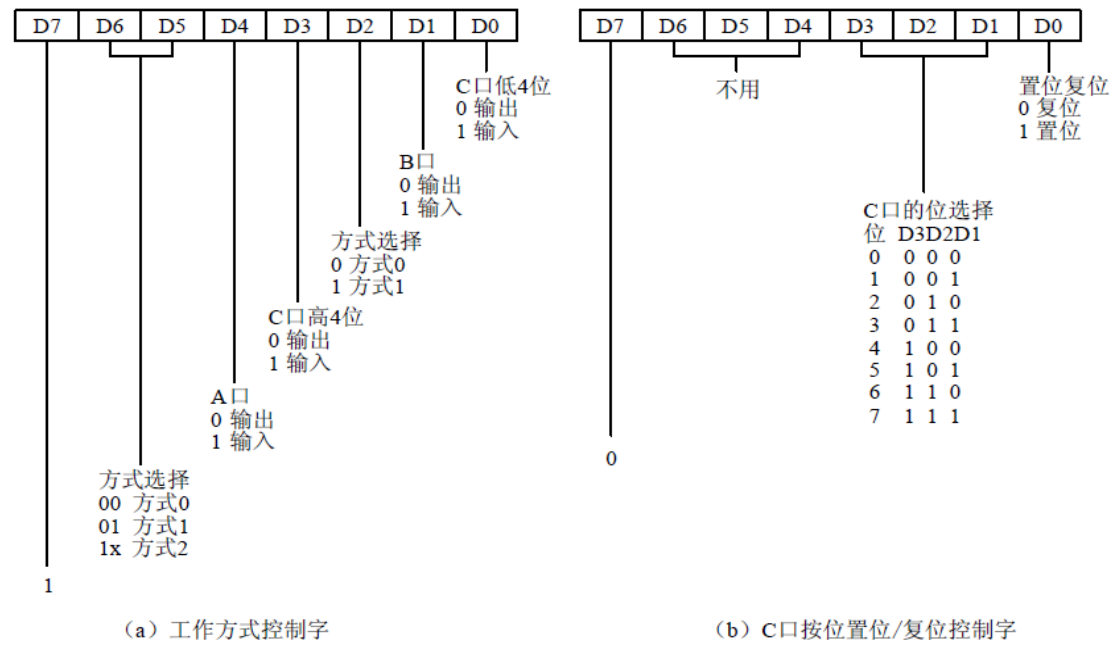
CPU 与接口之间的数据传送总是并行的, 即可以同时传递 8 位、16 位或 32 位等。8255 可编程外围接口芯片是 Intel 公司生产的通用并行 I/O 接口芯片, 它具有 A、B、C 三个并行接口, 用+5V 单电源供电, 能在以下三种方式下工作:

方式 0--基本输入/输出方式、

方式 1--选通输入/输出方式、

方式 2--双向选通工作方式。

8255 工作方式控制字和 C 口按位置位/复位控制字格式下图所示。



端口地址：

CS0 EQU 3000H ;片选 CS0 对应的端口始地址  
MY8255\_A EQU CS0+00H ;8255 的 A 口地址  
MY8255\_B EQU CS0+01H ;8255 的 B 口地址  
MY8255\_C EQU CS0+02H ;8255 的 C 口地址  
MY8255\_MODE EQU CS0+03H ;8255 的控制寄存器地址

### 3、8259A 芯片的内部结构及引脚

中断控制器 8259 是 Intel 公司专为控制优先级中断而设计开发的芯片。它将中断源优先级排队、辨别中断源以及提供中断矢量的电路集于一片中，因此无需附加任何电路，只需对 8259 进行编程，就可以管理 8 级中断，并选择优先模式和中断请求方式，即中断结构可以由用户编程来设定。同时，在不需增加其他电路的情况下，通过多片 8259 的级连，能构成多达 64 级的矢量中断系统。

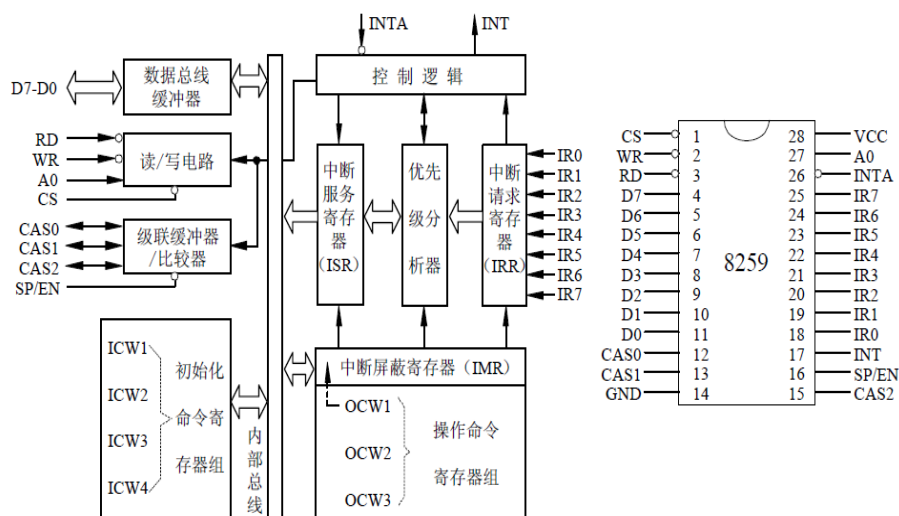
它的管理功能包括：

- 1) 记录各级中断源请求
- 2) 判别优先级，确定是否响应和响应哪一级中断



3) 响应中断时，向 CPU 传送中断类型号

8259 的内部结构和引脚如下图所示。



8259 的命令共有 7 个，一类是初始化命令字，另一类是操作命令。8259 的编程就是根据应用需要将初始化命令字 ICW1-ICW4 和操作命令字 OCW1-OCW3 分别写入初始化命令寄存器组和操作命令寄存器组。

ICW1-ICW4 各命令字格式如下图所示

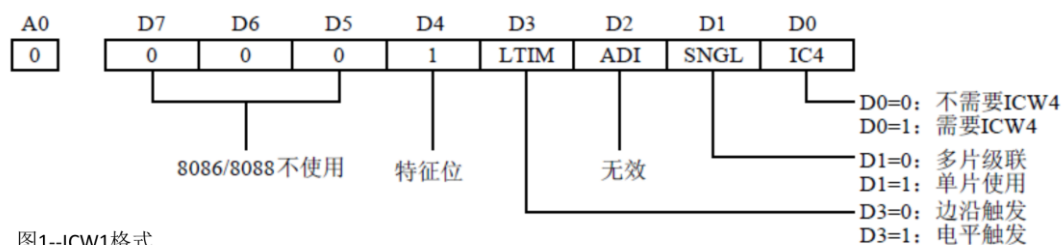


图1--ICW1格式

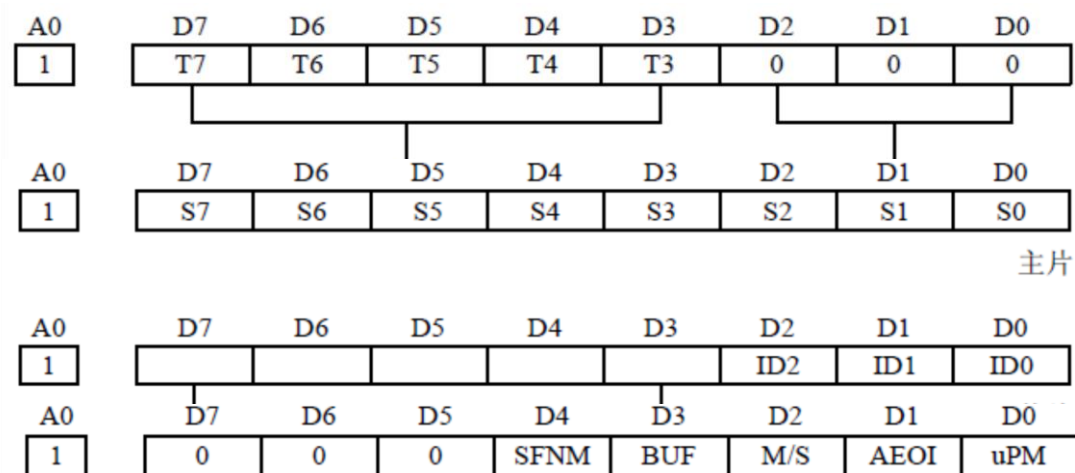


图4—ICW4格式

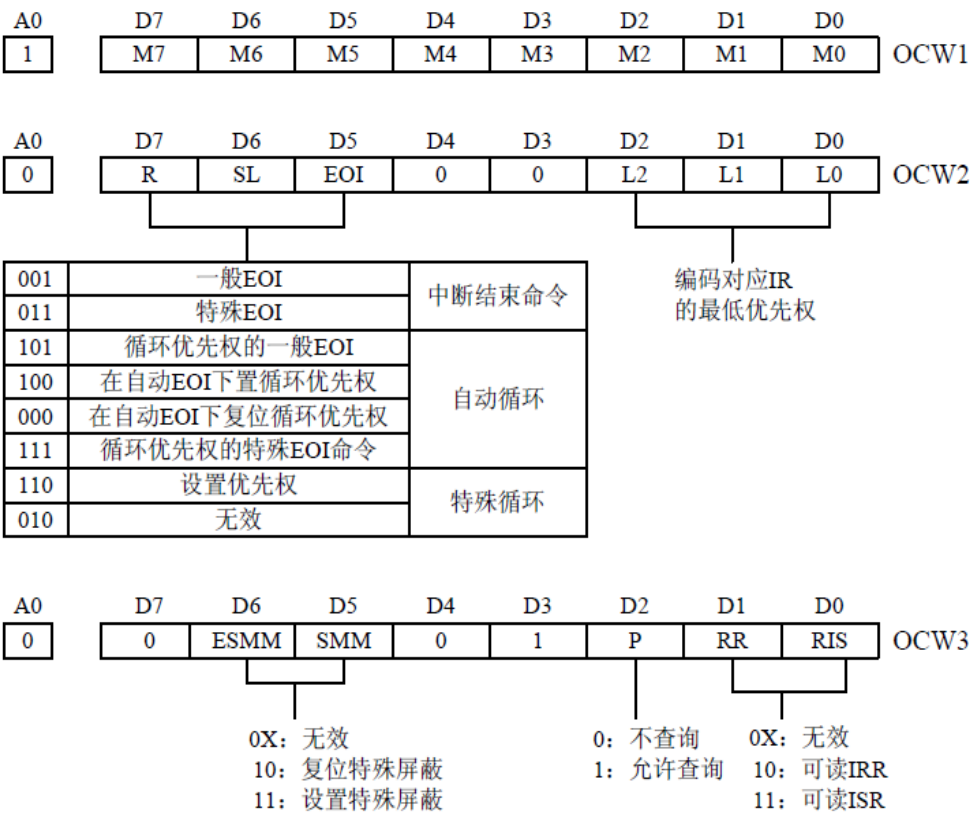
OCW1-OCW3 各命令字格式如下图所示，其中：

OCW1 用于设置中断屏蔽操作字

OCW2 用于设置优先级循环方式和中断结束方式的操作命令字

OCW3 用于设置和撤销特殊屏蔽方式、设置中断查询方式以及设置对 8259

内部寄存器的读出命令。



在硬件系统中，8259 仅占用两个外设接口地址，在片选有效的情况下，利用 A0 来寻址不同的寄存器和命令字。对寄存器和命令的访问控制如表所示。

A0	D4	D3	读信号	写信号	片选	操作
0			0	1	0	读出 ISR,IRR 的内容
1			0	1	0	读出 IMR 的内容
0	0	0	1	0	0	写入 OCW2
0	0	1	1	0	0	写入 OCW3
0	1	×	1	0	0	写入 ICW1
1	×	×	1	0	0	写入 OCW1, ICW2, ICW3, ICW4

在 80x86 系列 EPC 微机系统中，系统中包含了两片 8259 中断控制器，经级连可以管理 15 级硬件中断，但其中部分中断号已经被系统硬件占用，具体情况如表 2-1-2 示。两片 8259 的端口地址为：主片 8259 使用 020H 和 021H 两个端口；从片使用 0A0H 和 0A1H 两个端口。系统初始化两片 8259 的中断请求信号均采用上升沿触发，采用全嵌套方式，优先级的排列次序为 0 级最高，依次为 1 级、8 级~15 级，然后是 3 级~7 级。在实验平台上系统总线单元的 IR1,IR2 信号对应的中断线就是系统 8259 中断控制器的 IRQ10,IRQ7。

EPC 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	实验平台 IR2	0FH	003CH~003FH
从 8259 IRQ8	实时钟	70H	01C0H~01C3H
从 8259 IRQ9	保留	71H	01C4H~01C7H
从 8259 IRQ10	实验平台 IR1	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

IRQ\_IVADD EQU 01C8H ;IRQ10 对应的中断矢量地址01C8H

IRQ\_OCW1 EQU 0a1H ;IRQ10 对应PC 机内部8259 的OCW1 地址

IRQ\_OCW2 EQU 0a0H ;IRQ10 对应PC 机内部8259 的OCW2 地址

IRQ\_IM EQU 0FBH ;IRQ10 对应的中断屏蔽字0FBH

```
IRQ2_IVADD EQU    003CH    ;IRQ7 对应的中断矢量地址003CH
IRQ2_OCW1   EQU    021H    ;IRQ7 对应PC 机内部8259 的OCW1 地址
IRQ2_OCW2   EQU    020H    ;IRQ7 对应PC 机内部8259 的OCW2 地址
IRQ2_IM     EQU    07FH    ;IRQ7 对应的中断屏蔽字07FH
```

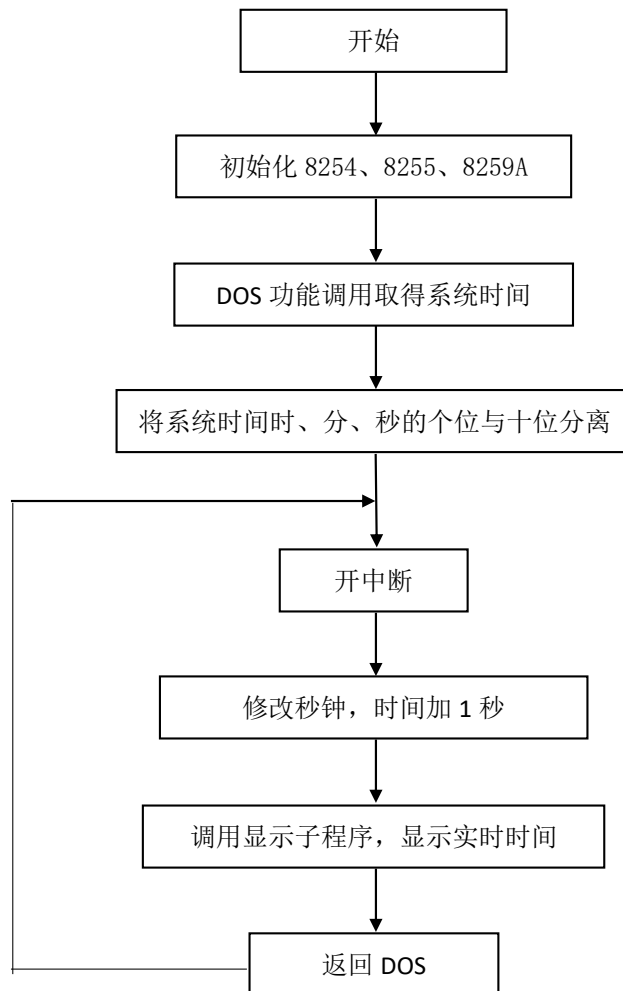
## 四、具体程序实现

### 1、概述

本系统设计的电子时钟以 8088 微处理器作为 CPU，用 8254 作为定时计数器产生时钟频率，8255 做可编程并行接口显示时钟，8259A 做中断控制器产生边缘中断。在此系统中，我们选择通过一次 DOS 调用之后，通过循环计数器 8254 以达到计时功能；故 8254 的功能是定时，接入 8254 的 CLK 信号作为 1.8432MHz 的周期性时钟信号。8254 采用计数器 0，工作于方式 3，将计数初值设为 4800H，使 8254 的 OUT0 端输出一个周期为 1s 的方波信号，此信号接 8259A 的 IR1 (IRQ10)，在接触到 8254 输入方波信号的上升沿的时候即进入中断中去执行自动修改秒钟及显示时间的功能，使数码管上的六位显示时间发生变化，所以将 8259A 的 IR1 产生中断，其他未屏蔽，主从片 8259，上升沿中断，中断号为 72H，工作于 8088/8086 方式。8255 芯片，将 8255A 的 A、B 口均工作于工作方式 0 下，且为输出控制 LED 的位选与片选，而 C 口用于修改时间，同样工作于方式 0 低四位输出。

系统共由以下四个模块组成：主程序初始化模块、定时模块、中断模块、显示模块

### 2、程序流程图



### 3、主程序模块

主程序模块是系统的核心，对可编程定时/计数器 8254、可编程中断控制器 8259A 和可编程并行接口芯片 8255 进行初始化、设置中断向量、DOS 系统调用获得系统时间等相应处理。主要由软件实现，主要功能是调用其他模块对系统工作进行协调.以达到实时计时功能。

在此系统中，我们选择通过一次 DOS 调用之后，通过循环计数器 8254 以达到计时功能；故 DOS 系统功能调用程序如下

```

;DOS 功能调用，从系统取得时间；CH:CL=时:分,DH:DL=秒:1/100 秒
MOV    AH,2CH
INT     21H
MOV     HOUR,CH      ;将时分秒分别存入数据段相应变量中保存

```

```
MOV    MINUTE, CL
MOV    SECOND, DH
```

故 8254 的功能是定时，接入 8254 的 CLK 信号作为 1.8432MHz 的周期性时钟信号。8254 采用计数器 0，工作于方式 3，将计数初值设为 4800H，使 8254 的 OUT0 端输出一个周期为 1s 的方波信号，故对 8254 的初始化程序如下

;8254 初始化程序, 中断产生设置

;选择通道 0, 先读/写计数器低字节, 后读/写高字节, 方式 3, 二进制计数

```
MOV    AL, 00110110B
MOV    DX, MY8254_MODE
OUT    DX, AL
MOV    AL, 00H                ;写入计数初值 4800H, 先写低字节, 再写高字节
MOV    DX, MY8254_COUNT0
OUT    DX, AL
MOV    AL, 48H
OUT    DX, AL
;8254 初始化程序结束
```

8255 芯片，将 8255A 的 A、B 口均工作于工作方式 0 下，且为输出控制七段数码管的位选与片选，而 C 口用于修改时间，同样工作于方式 0 低四位输出，故对 8255 的初始化程序如下

;8255 初始化程序

;A 口方式 0 输出, B 口方式 0 输出, C 口高四位输出, 低四位输入

```
MOV    AL, 10000001B
MOV    DX, MY8255_MODE
OUT    DX, AL
;8255 初始化程序结束
```

此信号接 8259A 的 IR1 (IRQ10)，在接触到 8254 输入方波信号的上升沿的时候即进入中断中去执行自动修改秒钟及显示时间的功能，使数码管上的六位显

示时间发生变化, 所以将 8259A 的 IR1 产生中断, 其他未屏蔽, 主从片 8259, 上升沿中断, 中断号为 72H, 工作于 8088/8086 方式。故对 8259A 的初始化程序如下

;8259 初始化程序

```
CLI
MOV     AL, 72H           ;保护中断
MOV     AH, 35H
INT     21H               ;取出中断类型为 72H 的中断向量, 存放到 ES:BX
PUSH    ES                ;将原中断向量送到堆栈保存
PUSH    BX
PUSH    DS                ;保存 DS
MOV     DX, OFFSET GETTIME ;DX 存放用户新中断向量偏移量
MOV     AX, SEG GETTIME
MOV     DS, AX            ;DS 存放用户新中断向量段基址量
MOV     AL, 72H
MOV     AH, 25H
INT     21H               ;设置新中断向量
POP     DS                ;恢复 DS

MOV     AL, 11111011B     ;设置 IRQ10 对应 PC 机内部的 8259 的 OCW1
OUT     21H, AL
MOV     AL, 11111011B     ;设置 IRQ7 对应 PC 机内部的 8259 的 OCW1
OUT     0A1H, AL
;8259 初始化程序结束
```

#### 4、定时模块

定时模块是为 8259A 提供中断请求信号的。由一片 8254 实现, 选用计数器 0, 工作在方式 3, 由时钟功能可知中断应该每秒开启一次以达到准确计时, 故输出值 OUT0 应为 1s, 其输出信号可作为 8259 的中断请求信号。

为使输出 1s 的连续方波, 时间常数  $N_0 = 1.8432\text{MHz} / 1\text{Hz} = 4800\text{H}$

定时模块程序代码如下:

```
MOV     AL, 00H           ;写入计数初值 4800H, 先写低字节, 再写高字节
MOV     DX, MY8254_COUNT0
```

```

OUT    DX, AL
MOV    AL, 48H
OUT    DX, AL

```

## 5、中断模块

中断模块实现动态显示的，硬件为一片 8259，由于中断请求信号为每秒一次，中断程序该为时间按秒增加，并显示，只要开中断，便可实现每秒显示时间增加一秒，从而达到动态显示的效果

### (1)、中断子程序主部分

中断子程序主部分主要是对时间进行处理，达到动态显示的效果，并写入使得中断结束的 EOI 命令，中断子程序主部分如下：

```

GETTIME    PROC    NEAR    ;中断服务子程序
            PUSH    DS      ;对寄存器进行堆栈保护
            PUSH    DX
            PUSH    AX
            PUSH    CX
            PUSH    BX
            MOV     AX, DATA
            MOV     DS, AX
            CALL    ADDTIME    ;调用时间递增子程序，以达到实时计时功能
            CALL    SETTIME    ;调用时间转换子程序
            MOV     CX, 40H
            ;通过重复调用 DIS 子程序使得数码管六位数字能够一起稳定输出
G01:
            CALL    DIS        ;调用显示子程序，使实时时间稳定输出
            LOOP    G01
EXIT:
            MOV     AL, 20H     ;用 EOI 命令结束 IR2 中断
            OUT     0A0H, AL
            MOV     AL, 20H     ;用 EOI 命令结束 IR1 中断
            OUT     20H, AL

```



```

POP     BX                ;恢复现场
POP     CX
POP     AX
POP     DX
POP     DS
IRET                    ;返回 DOS
GETTIME  ENDP

```

## (2)、时间转换子程序模块

将从 DOS 系统功能调用得到的时分秒的个位与十位通过除十操作进行分离

```

SETTIME  PROC  NEAR      ;时间转换子程序，对时间时分秒进行分离
    PUSH  AX              ;对寄存器 AX, CX 进行堆栈保护
    PUSH  CX
    MOV   AX, DATA
    MOV  DS, AX
    MOV   CH, 0           ;将 CX 高位设置为 0 方便做除数
    ;将已从系统获得的小时数进行十位和各位分离，方便显示输出
    MOV   AX, 00H
    MOV   AL, HOUR
    MOV   CL, 10
    DIV   CL
    MOV   HOUR1, AL        ;AL 为 AX/CL 的商，即小时数的十位
    MOV   HOUR2, AH        ;AH 为 AX/CL 的余数，即小时数的个位
    ;将已从系统获得的分钟数进行十位和各位分离，方便显示输出
    MOV   AX, 00H
    MOV   AL, MINUTE
    MOV   CL, 10
    DIV   CL
    MOV   MINUTE1, AL      ;AL 为 AX/CL 的商，即分钟数的十位
    MOV   MINUTE2, AH      ;AH 为 AX/CL 的余数，即分钟数的个位
    ;将已从系统获得的秒数进行十位和各位分离，方便显示输出
    MOV   AX, 00H
    MOV   AL, SECOND
    MOV   CL, 10
    DIV   CL
    MOV   SECOND1, AL      ;AL 为 AX/CL 的商，即秒数的十位

```

```

MOV     SECOND2, AH           ;AH 为 AX/CL 的余数, 即秒数的个位

POP     CX                   ;恢复现场
POP     AX
RET                                           ;返回 DOS
SETTIME ENDP

```

### (3)、时间递增子程序模块

在每次方波上升沿进入中断的时候产生秒数的递增, 并按时、分、秒的进制进行调整, 并存入相应的缓冲区之中, 使得动态显示不出现紊乱

```

ADDTIME PROC NEAR           ;时间递增子程序, 以达到实时计时功能
    PUSH    AX              ;对寄存器进行堆栈保护
    PUSH    DX

    INC     SECOND          ;秒数加一
    MOV     AL, SECOND
    CMP     AL, 60          ;秒数与 60 进行比较, 若等于 60 则产生进位且秒数归零
    JNE     EXIT0           ;若秒数小于 60 则退出子程序
    MOV     SECOND, 0       ;若秒数计时等于 60 则秒数归零
    INC     MINUTE          ;秒数计时等于 60 则产生分钟数进位
    MOV     AL, MINUTE
    CMP     AL, 60          ;分钟数与 60 进行比较, 若等于 60 则进位且分钟数归零
    JNE     EXIT0           ;若分钟数小于 60 则退出子程序
    MOV     MINUTE, 0       ;若分钟数计时等于 60 则分钟数归零
    INC     HOUR            ;分钟数计时等于 60 则产生小时数进位
    MOV     AL, HOUR
    CMP     AL, 24          ;小时数与 24 进行比较, 若小时数等于 24 则小时数归零
    JNE     EXIT0           ;若小时数小于 24 则退出子程序
    MOV     HOUR, 0         ;若小时数计时等于 24 则小时数归零

EXIT0:    NOP

    POP     DX              ;恢复现场
    POP     AX
    RET                                           ;返回 DOS

```

ADDTIME     ENDP

## 6、显示子程序模块

显示模块是将显示缓冲区的数值送到七段数码管显示的功能模块,显示缓冲区是由六个字节构成,分别存储时、分、秒的个位和十位。硬件上由六个数码管构成,软件上由扫描显示的方法实现

```
MOV     SI, OFFSET HOUR1    ;使 SI 指向时分秒的个位或十位的数据段
MOV     DL, 0DFH            ;打开第一个数码管
MOV     AL, DL
AGAIN:
PUSH    DX
MOV     DX, MY8255_A        ;A 口选择数码管六个显示管相应管
OUT     DX, AL              ;设置 X1 ~ X4, 选通一个数码管

MOV     AL, [SI]            ;分别取出数据段中存放时分秒的个位或十位
MOV     BX, OFFSET WTABLE;BX 存取七段代码表首地址
AND     AX, 00FFH          ;取低 8 位
ADD     BX, AX              ;通过查表得到时分秒的对应数码管显示数字的首地址
MOV     AL, [BX]            ;将键值作为偏移和键值基地址相加得到相应的键值
MOV     DX, MY8255_B        ;B 口选择数码管相应段
OUT     DX, AL              ;写入数码管 A ~ Dp
CALL    DALLY
INC     SI                  ;取下一个键值
POP     DX
MOV     AL, DL
TEST    AL, 01H            ;判断是否显示完?
JZ      OUT1                ;显示完, 返回
ROR     AL, 1               ;开启下一个数码管
MOV     DL, AL
JMP     AGAIN               ;未显示完, 跳回继续
```

## 五、汇编代码

;8254 计时器端口地址

CS2                    EQU     3040H   ;8254 计时器口地址

```
MY8254_COUNT0 EQU CS2+00H ;计数器 0 对应端口
MY8254_MODE EQU CS2+03H ;8254 控制字对应端口
```

;8255 并行接口端口地址

```
CS0 EQU 3000H ;8255 并行接口端口地址
MY8255_A EQU CS0+00H ;A 口端口地址
MY8255_B EQU CS0+01H ;B 口端口地址
MY8255_C EQU CS0+02H ;C 口端口地址
MY8255_MODE EQU CS0+03H ;8255 控制字对应端口
```

;8259 中断

```
IRQ_IVADD EQU 01C8H ;IRQ10 对应的中断矢量地址
IRQ_OCW1 EQU 0A1H ;IRQ10 对应 PC 机内部的 8259 的 OCW1 地址
IRQ_OCW2 EQU 0A0H ;IRQ10 对应 PC 机内部的 8259 的 OCW2 地址
IRQ_IM EQU 0FBH ;IRQ10 对应的中断屏蔽字
```

```
IRQ2_IVADD EQU 003CH ;IRQ7 对应的中断矢量地址 003CH
IRQ2_OCW1 EQU 021H ;IRQ7 对应 PC 机内部 8259 的 OCW1 地址
IRQ2_OCW2 EQU 020H ;IRQ7 对应 PC 机内部 8259 的 OCW2 地址
IRQ2_IM EQU 07FH ;IRQ7 对应的中断屏蔽字 07FH
```

```
DATA SEGMENT ;0~F 对应数码管的段位值
WTABLE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H,
          7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H
HOUR DB 00H ;DOS 功能调用读取时间的小时值
MINUTE DB 00H ;DOS 功能调用读取时间的分钟值
SECOND DB 00H ;DOS 功能调用读取时间的秒数值
```

```
HOUR1 DB 00H ;小时值的十位
HOUR2 DB 00H ;小时值的个位
MINUTE1 DB 00H ;分钟值的十位
MINUTE2 DB 00H ;分钟值的个位
SECOND1 DB 00H ;秒数值的十位
SECOND2 DB 00H ;秒数值的个位
```

```
DATA ENDS
```

```
CODE SEGMENT
```

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA

MOV DS, AX

;8255 初始化程序

; A 口方式 0 输出, B 口方式 0 输出, C 口高四位输出, 低四位输入

MOV AL, 10000001B

MOV DX, MY8255\_MODE

OUT DX, AL

;8255 初始化程序结束

;8254 初始化程序, 中断产生设置

;选择通道 0, 先读/写计数器低字节, 后读/写高字节, 方式 3, 二进制计数

MOV AL, 00110110B

MOV DX, MY8254\_MODE

OUT DX, AL

MOV AL, 00H ;写入计数初值 4800H, 先写低字节, 再写高字节

MOV DX, MY8254\_COUNT0

OUT DX, AL

MOV AL, 48H

OUT DX, AL

;8254 初始化程序结束

;8259 初始化程序

CLI

MOV AL, 72H ;保护中断

MOV AH, 35H

INT 21H ;取出中断类型为 72H 的中断向量, 存放到 ES:BX

PUSH ES ;将原中断向量送到堆栈保存

PUSH BX

PUSH DS ;保存 DS

MOV DX, OFFSET GETTIME ;DX 存放用户新中断向量偏移量

MOV AX, SEG GETTIME

MOV DS, AX ;DS 存放用户新中断向量段基址量

MOV AL, 72H

MOV AH, 25H

INT 21H ;设置新中断向量

```

POP    DS                ;恢复 DS

MOV     AL, 11111011B    ;设置 IRQ10 对应 PC 机内部的 8259 的 OCW1
OUT     21H, AL
MOV     AL, 11111011B    ;设置 IRQ7 对应 PC 机内部的 8259 的 OCW1
OUT     0A1H, AL
;8259 初始化程序结束

;DOS 功能调用, 从系统取得时间; CH:CL=时:分, DH:DL=秒:1/100 秒
MOV     AH, 2CH
INT     21H
MOV     HOUR, CH          ;将时分秒分别存入数据段相应变量中保存
MOV     MINUTE, CL
MOV     SECOND, DH
CALL    SETTIME           ;调用时间转换子程序对时间时分秒进行分离
STI                     ;开启中断
;主程序利用中断获得的当前时间, 输出到数码管
GO:
    JMP     GO            ;开中断
HALT:
    CLI                     ;关中断
    POP     DX            ;恢复原中断向量
    POP     DS
    MOV     AL, 72H        ;将中断类型号为 72H 的中断向量(DS:DX)置入中断
向量表中
    MOV     AH, 25H
    INT     21H
    STI
    MOV     AX, 4C00H      ;返回 DOS
    INT     21H

GETTIME PROC    NEAR    ;中断服务子程序
    PUSH    DS            ;对寄存器进行堆栈保护
    PUSH    DX
    PUSH    AX
    PUSH    CX
    PUSH    BX
    MOV     AX, DATA

```

```

MOV     DS, AX
CALL    ADDTIME      ;调用时间递增子程序, 以达到实时计时功能
CALL    SETTIME      ;调用时间转换子程序
MOV     CX, 40H
;通过重复调用 DIS 子程序使得数码管六位数字能够一起稳定输出
G01:
CALL    DIS          ;调用显示子程序, 使实时时间稳定输出
LOOP    G01
EXIT:
MOV     AL, 20H      ;用 EOI 命令结束 IR2 中断
OUT     0A0H, AL
MOV     AL, 20H      ;用 EOI 命令结束 IR1 中断
OUT     20H, AL
POP     BX           ;恢复现场
POP     CX
POP     AX
POP     DX
POP     DS
IRET                ;返回 DOS
GETTIME    ENDP

DALLY  PROC    NEAR    ;软件延时子程序
    PUSH    CX        ;对寄存器 AX, CX 进行堆栈保护
    PUSH    AX
    MOV     CX, 1000H  ;设置大循环数为 1000H
D1: MOV     AX, 100H    ;设置小循环数为 100H
D2: DEC     AX          ;进入小循环, 对 AX 进行递减直到 0
    JNZ     D2
    LOOP    D1         ;返回大循环
    POP     AX         ;恢复现场
    POP     CX
    RET                ;返回 DOS
DALLY  ENDP

SETTIME  PROC    NEAR    ;时间转换子程序, 对时间时分秒进行分离
    PUSH    AX        ;对寄存器 AX, CX 进行堆栈保护
    PUSH    CX
    MOV     AX, DATA

```

```

MOV DS, AX
MOV     CH, 0           ;将 CX 高位设置为 0 方便做除数
;将已从系统获得的小时数进行十位和各位分离，方便显示输出
MOV     AX, 00H
MOV     AL, HOUR
MOV     CL, 10
DIV     CL
MOV     HOUR1, AL       ;AL 为 AX/CL 的商，即小时数的十位
MOV     HOUR2, AH       ;AH 为 AX/CL 的余数，即小时数的个位
;将已从系统获得的分钟数进行十位和各位分离，方便显示输出
MOV     AX, 00H
MOV     AL, MINUTE
MOV     CL, 10
DIV     CL
MOV     MINUTE1, AL     ;AL 为 AX/CL 的商，即分钟数的十位
MOV     MINUTE2, AH     ;AH 为 AX/CL 的余数，即分钟数的个位
;将已从系统获得的秒数进行十位和各位分离，方便显示输出
MOV     AX, 00H
MOV     AL, SECOND
MOV     CL, 10
DIV     CL
MOV     SECOND1, AL     ;AL 为 AX/CL 的商，即秒数的十位
MOV     SECOND2, AH     ;AH 为 AX/CL 的余数，即秒数的个位

POP     CX              ;恢复现场
POP     AX
RET                                ;返回 DOS
SETTIME ENDP

```

;显示时间子程序，以缓冲区存放的键值为键值表偏移找到键值并显示

```

DIS     PROC     NEAR
        PUSH     AX           ;对寄存器进行堆栈保护
        PUSH     BX
        PUSH     CX
        PUSH     DX
        MOV     AX, DATA
        MOV     DS, AX

```



```

MOV     SI, OFFSET HOUR1    ;使 SI 指向时分秒的个位或十位的数据段
MOV     DL, 0DFH            ;打开第一个数码管
MOV     AL, DL

AGAIN:
PUSH    DX
MOV     DX, MY8255_A        ;A 口选择数码管六个显示管相应管
OUT     DX, AL              ;设置 X1 ~ X4, 选通一个数码管

MOV     AL, [SI]            ;分别取出数据段中存放时分秒的个位或十位
MOV     BX, OFFSET WTABLE;BX 存取七段代码表首地址
AND     AX, 00FFH          ;取低 8 位
ADD     BX, AX              ;通过查表得到时分秒的对应数码管显示数字的首地址
MOV     AL, [BX]            ;将键值作为偏移和键值基地址相加得到相应的键值
MOV     DX, MY8255_B        ;B 口选择数码管相应段
OUT     DX, AL              ;写入数码管 A ~ Dp
CALL    DALLY
INC     SI                  ;取下一个键值
POP     DX
MOV     AL, DL
TEST    AL, 01H            ;判断是否显示完?
JZ      OUT1                ;显示完, 返回
ROR     AL, 1               ;开启下一个数码管
MOV     DL, AL
JMP     AGAIN               ;未显示完, 跳回继续

OUT1:
POP     DX                  ;恢复现场
POP     CX
POP     BX
POP     AX
RET                          ;返回 DOS

DIS     ENDP

ADDTIME PROC NEAR          ;时间递增子程序, 以达到实时计时功能
PUSH    AX                  ;对寄存器进行堆栈保护
PUSH    DX

INC     SECOND              ;秒数加一
MOV     AL, SECOND

```

```

    CMP     AL, 60      ;秒数与 60 进行比较, 若等于 60 则产生进位且秒数归零
    JNE     EXIT0      ;若秒数小于 60 则退出子程序
    MOV     SECOND, 0   ;若秒数计时等于 60 则秒数归零
    INC     MINUTE      ;秒数计时等于 60 则产生分钟数进位
    MOV     AL, MINUTE
    CMP     AL, 60      ;分钟数与 60 进行比较, 若等于 60 则进位且分钟数归零
    JNE     EXIT0      ;若分钟数小于 60 则退出子程序
    MOV     MINUTE, 0   ;若分钟数计时等于 60 则分钟数归零
    INC     HOUR        ;分钟数计时等于 60 则产生小时数进位
    MOV     AL, HOUR
    CMP     AL, 24      ;小时数与 24 进行比较, 若小时数等于 24 则小时数归零
    JNE     EXIT0      ;若小时数小于 24 则退出子程序
    MOV     HOUR, 0     ;若小时数计时等于 24 则小时数归零

EXIT0:    NOP

    POP     DX          ;恢复现场
    POP     AX
    RET              ;返回 DOS
ADDTIME   ENDP

CODE     ENDS
END       START

```

## 六、实验总结与收获

1、在实践中熟悉了可编程定时/计数器 8254、可编程中断控制器 8259A 和可编程并行接口芯片 8255 的工作原理、内部结构，对微机原理理论课上学到的理论知识有了进一步的理解。

2、知道了可编程定时/计数器 8254、可编程中断控制器 8259A 和可编程并行接口芯片 8255 的实现功能及使用方法，灵活运用可编程定时/计数器 8254、可编程中断控制器 8259A 和可编程并行接口芯片 8255 及实验箱实现相应功能。成功实现数字电子时钟电路搭建、软件编译运行，对其原理有了进一步的认知。

3、通过查阅资料，结合所学知识进行软、硬件的设计，使学生初步掌握应用微机解决问题的步骤及方法。为以后结合专业从事微机应用设计奠定基础。

4、动手能力得到了锻炼，并学会了分析实验中遇到的 bug 及进行解决，同时进行反思总结。