# 《微机系统》知识点练习

1． 说明 CISC、RISC 技术的特点及其主要区别？

CISC 是指复杂指令系统计算机，RISC 是指精简指令系统计算机。

他们的区别在于不同的 CPU 设计理念和方法。RISC 指令系统仅包含哪些必要的经常使用的指令，不经常使用的功能，往往通过基本指令组合来完成。完成特殊功能时效率比较低。CISC 的指令系统比较丰富，一些特殊功能都有相应的指令。处理特殊任务效率较高。

RISC 对存储器操作相对简单，使对存储器访问的控制简化；而 CISC 机器的存储器操作指令较多，对存储器的访问有更多的指令直接操作，要求的控制逻辑比较复杂。RISC 在一条指令执行的适当地方可以响应中断；而 CISC 机器是在一条指令执行结束后响应中断。

RISC CPU 的电路构成比 CISC CPU 简单，因此面积小、功耗也更低；CISC 电路　CPU 电路复杂，同水平比 RISC CPU 面积大、功耗大。RISC CPU 结构比较简单，布局紧凑规整，设计周期较短，比较容易采用一些并行计算的最新技术；CISC CPU 结构复杂，设计周期长，技术更新难度大。从使用角度看，RISC 微处理器结构简单，指令规整，性能容易把握，易学易用；CISC 微处理器结构复杂，功能强大，实现特殊功能容易。

2． 80x86 标志寄存器中 ZF 位等于 1，说明(　　)。 A

A．运算结果等于 0　　B. 运算结果大于 0

C. 运算结果不等于 0　　D.运算结果溢出

3． 8086CPU 由哪两大部分组成？简述它们的主要功能。

总线接口部件 BIU 跟执行部件 EU。

总线接口部件（BIU）是联系微处理器内部与外部的重要通道，其主要功能是负责微处理器内部与外部的信息传递。主要任务：（1）取指令（2）形成物理地址（3）传送数据

EU 完成控制器的功能，它负责执行指令并对相应的硬件部分进行控制，它的主要功能就是完成全部指令的执行。EU 完成以下主要任务：（1）指令译码（2）执行指令（3）向 BIU 传送偏移地址信息（4)管理通用寄存器和标志寄存器。

4． 80x86 上电后第一条执行的指令的地址是(　　)。B

A. 0000H:0000H　　B.0FFFFH:0000H

C. 0FFFFH:0100H　　C.0000H:0100H

5． 说明 80X86 中，使用对齐数据与非对齐数据对数据的访问速度有什么影响？

当访问的数据是一个对准数据时，一个总线周期可以完成读写，如果访问的数据不是对准的数据时，要通过两个总线周期完成读写过程，因此编程时应当应尽量使数据对准存放。

6． 8086 系统通过(　　)控制线来区分是存贮器访问还是 I/O 访问的。D

A. $ALE$　　B. $\overline{RD}$　　C. $\overline{WR}$　　D. $M/\overline{IO}$

7． 80x86 执行 in al, dx 时，控制信号是(　　)。A

A. $\overline{RD}$ 为低电平，$\overline{WR}$ 为高电平，$M/\overline{IO}$ 为低电平

B. $\overline{RD}$ 为高电平，$\overline{WR}$ 为低电平，$M/\overline{IO}$ 为高电平

C. $\overline{RD}$ 为高电平，$\overline{WR}$ 为低电平，$M/\overline{IO}$ 为低电平

D. $\overline{RD}$ 为低电平，$\overline{WR}$ 为高电平，$M/\overline{IO}$ 为高电平

8． 在 80x86 实模式下，mov ax, [bp + si]的源操作数物理地址为(　　)。C

A. 16 × (ds) + (bp) + (si)　　B. (ds) + (bp) + (si)

C. 16 × (ss) + (bp) + (si)　　D. 16 × (cs) + (bp) + (si)

9． 8086CPU 在最小模式下，执行 MOV AL, [1000H]时，控制信号是(　　)。D

A. $\overline{RD}$ 为低电平，$\overline{WR}$ 为高电平，$M/\overline{IO}$ 为低电平；

B. $\overline{RD}$ 为高电平，$\overline{WR}$ 为低电平，$M/\overline{IO}$ 为高电平；

C. $\overline{RD}$ 为高电平，$\overline{WR}$ 为低电平，$M/\overline{IO}$ 为低电平；

D. $\overline{RD}$ 为低电平，$\overline{WR}$ 为高电平，$M/\overline{IO}$ 为高电平。

10． 将十进制 35 以压缩 BCD 码格式传送到 AL 中，正确的指令是(　　)。D

A. MOV AX, 0305H　　B. MOV AX, 0035

C. MOV AX, 0305　　D. MOV AX, 0035H

11． 以下软中断中，哪个是系统功能调用？(　　)B

A. INT 10H　　B. INT 21H

C. INT 16H　　D. INT 13H

12． 给定字符串“Do not lose faith,　as long as the unremittingly,　you will get some fruits.”试以 EXE 程序结构编写只统计其中英文字母个数的程序。

```
        DSEG    SEGMENT 'DATA'
      X DB “Do not lose faith,  as long as the unremittingly,  you will get some fruits.$”
      N       DW 0
DSEG    ENDS
SSEG    SEGMENT STACK   'STACK'
        DW      100H    DUP(?)
SSEG    ENDS
CSEG    SEGMENT 'CODE'
Main    PROC    FAR
      PUSH    DS
      MOV     AX, 0
      PUSH    AX
      MOV     AX, DSEG
      MOV     DS, AX
      MOV     ES, AX
      MOV CX, 0 ; 保存计数值
MOV BX, offset X
DEC BX
Nt: INC BX
MOV AL, [BX] ; 取一个字符
CMP AL, ‘$’
JE  Exit
CMP AL, ‘,’
JE  Nt
CMP AL, ‘ ’
JE Nt
CMP AL, ‘.’
JE Nt
INC CX
JMP Nt
Exit:MOV N, CX
      RET
START   ENDP
CSEG    ENDS
        END     Main
```

13．试编写由键盘输入一个以回车（0x0D）作为结束的字符串，将其按 ASCII 码由大到小的顺序输出到显示器上的源程序。

```
          DSEG       SEGMENT 'DATA'
       RESULT DB 100 DUP(1)
       N          DB 0
DSEG       ENDS
SSEG       SEGMENT STACK     'STACK'
            DW          100H       DUP(?)
SSEG       ENDS
CSEG       SEGMENT 'CODE'
START      PROC       FAR
; STORE RETURN ADDRESS TO OS:
       PUSH       DS
       MOV        AX, 0
       PUSH       AX
; SET SEGMENT REGISTERS:
       MOV        AX, DSEG
       MOV        DS, AX
       MOV        ES, AX
;1.输入保存-----------------------
       MOV CL, 0
       LEA SI, RESULT
input_next:
:       MOV AH, 1
       INT 21H
       CMP AL, 0DH ; 回车为 ODH
       JZ toSort
       MOV [SI],AL
       INC SI
       INC CL
       JMP input_next
toSort:
       MOV N, CL ;输入的 ASCII 码个数保存到 N 中
;2.排序--------------------
       MOV DH, 0
       MOV    DL, N
       DEC    DX
next1:       LEA    BX, RESULT
       MOV    CX,DX
next2:       MOV    AL,[BX]
       CMP    AL,[BX+1]
       JNC noX
       XCHG AL,[BX+1]
       MOV [BX],AL
noX:       INC BX
       LOOP next2
```

```
        DEC DX
        JNZ next1
    ;3.显示----------------
        MOV CH, 0
        MOV CL,N
        LEA BX, RESULT
    displayNext:
        MOV DL,[BX]
        MOV AH,2
        INT 21H
        INC BX
        LOOP displayNext
        RET
    START    ENDP
    CSEG     ENDS
             END     START
```

14．　一个程序有 7 个任务，则 GDT 和 LDT 的个数分别是(　　)。A

A. 1, 7　　B. 6, 1

C. 1，1　　D. 1，6

15．　一般 GDT 和 LDT 的个数是(　　)。C

A. 与任务数相同的 GDT，与任务数相同的 LDT　　B. 任务数相同的 GDT， 1 个 LDT

C. 1 个 GDT，与任务数相同的 LDT　　D. 无限制，无限制

16．　说明 Pentium 实地址模式的特点，并说明 8086 工作模式、Pentium 实地址模式、Pentium 虚拟 8086 模式之间有什么异同?

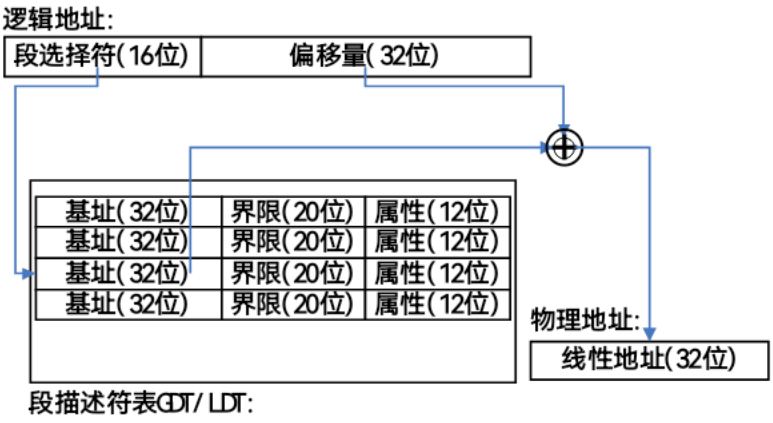
**答：**Pentium 实地址模式特点：能有效地使用 8086 所没有的寻址方式、32 位寄存器和大部分指令。实地址方式，Pentium 与 8086 兼容，基本体系结构相同。

虚拟 8086 方式与实地址方式的不同：　1）虚拟 8086 方式是一个程序的运行方式。

　　2）实地址方式是处理器的工作方式。

17．　画图详细说明，在 Pentium 处理器保护方式下，在采用分段存贮器管理时，数据访问逻辑地址转换为物理地址的过程。

答：Pentium微处理器的分段存储管理机制允许将46位虚拟地址映射到硬件所需的32位物理地址。如图，首先由虚拟地址(逻辑地址)段选择符部分的13位索引字段确定段描述符在段描述符表中的位置，然后取出段描述符中的32位基地址并与逻辑地址中的32位偏移量相加，得到32位的线性地址。若无分页功能，则线性地址就直接是物理地址。



18． 在查询方式工作时，接口状态信息是通过( )传送给 CPU 的。 B

A. 控制总线 B. 数据总线 C. 地址总线 D. 中断

19． 下列几种芯片中能接管总线、控制 I/O 接口与存储器直接进行数据传送的是( )。A

A. 8237A B. 8259A C. 8255A D. 8253A

20． CPU 与外设之间的数据传送有哪几种控制方式？并分别做简要说明。

（1）程序查询方式：CPU 通过查询 I/O 设备的状态，断定哪个设备需要服务，然后转入相应的服务程序。

（2）程序中断方式：当 I/O 设备需要 CPU 为其服务时，可以发生中断请求信号 INTR，CPU 接到请求信号后，中断正在执行的程序，转去为该设备服务，服务完毕，返回原来 被中断的程序并继续执行。

（3）直接存储器存取（DMA）方式：采用这种方式时，在 DMA 控制器的管理下，I/O 设备和存储器直接交换信息，而不需要 CPU 介入。

（4）I/O 处理机方式：引入 I/O 处理机，全部的输入/输出操作由 I/O 处理机独立承担。

21． 8259A 中断控制器工作在正常结束全嵌套方式时，清零中断服务寄存器 ISR 相应位的操作是在( )。C

A. 进入中断服务程序时 B. 进入中断服务程序之前

C. 中断服务程序返回之前 D.在计算中断向量地址时

22． 80x86 在响应可屏蔽中断时，在第二个中断响应总线周期的 T3 下降沿读取数据总线的内容当做( )。A

A. 中断类型码 B.中断服务程序入口地址 C. 状态数据 D. 控制数据

23． 8086CPU 中，置 1 标志寄存器 IF 位之后，( )。 B

A．不响应软件中断 B. 可以响应可屏蔽中断

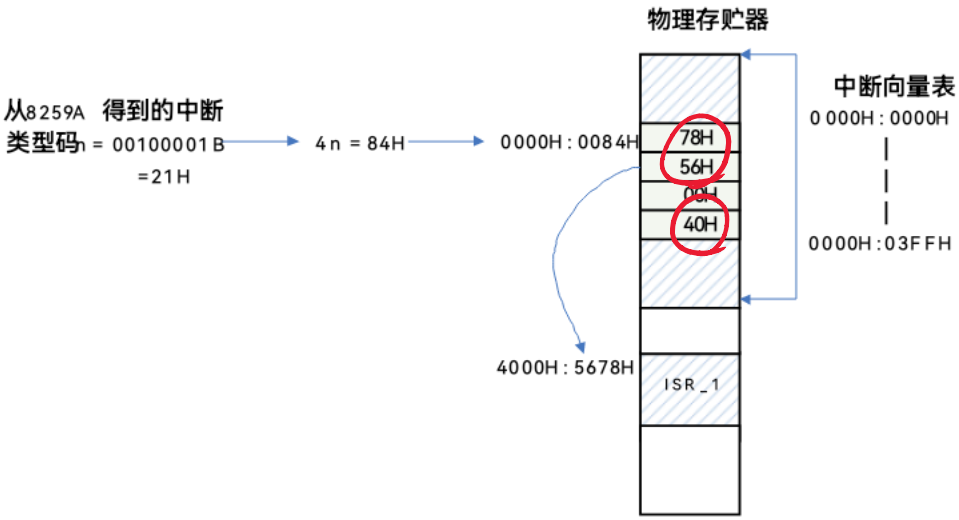
C. 不响应可屏蔽中断 D. 只响应非屏蔽中断

24． 若用 8 片 8259A 级联，则最多可管理( )个中断源。C

A. 32 B.56 C. 57 D. 64

25． 8259A 中断类型码初始化控制字 ICW2 写入的是 20H，在 8259A 中断请求引脚 IR1 连接一个中断源，其对应的中断服务程序名称为 ISR_1（ISR_1 从 4000H:5678H 开始分配内存地址）。请画图并说明，从中断源，到中断向量表，再到中断服务程序的对应关系。

答：因为 8259A 的 IR1 连接有一个中断源，当有中断时，自动填充中断类型码的低三位为 001B,则对应的中断类型码为 0010 0001B =21H，CPU 在中断响应周期时取到中断类型码 n = 21H，4n = 84H,到 0000H 段偏移为 84H 的位置取出中断服务程序 ISR_1 的入口地址 4000H：5678H，跳到中断服务程序 ISR_1 去执行。




26． 8259A 响应中断过程中会连续执行两个 INTA 中断响应周期，说明每个周期的功能是什么？

**第一个 INTA 中断响应周期**：使 IRR 的锁存功能失效；使当前中断服务寄存器 ISR 中的相应位置 1；使 IRR 寄存器中的相应位（即（2）中设置 ISR 为 1 所对应的 IRR 中的位）清 0。

**第二个 INTA 中断响应周期**：将中断类型码寄存器 ICW2 中的内容送到数据总线的 D7-D0，即为 CPU 提供中断类型码；如果 ICW4（方式控制字）中的中断自动结束位为 1，那么，在第二个 INTA 负脉冲结束时，8259A 会将第一个 INTA 负脉冲到来时设置的当前中断服务寄存器 ISR 的相应位清 0。

27．PCI 总线叫（　　）。D

A．控制局域网络总线　　B. 工业标准微机总线

C. 通用串行总线　　D. 外围部件互联总线

28．为什么对 8253A 写入计数值 0 是最大的计数值？在二进制计数方式下计数值相当于多少？如果 8253A 的时钟 CLK 频率为 1.193MHz，工作在周期工作方式时，计算最大可以产生每秒多少次的周期信号（保留一位小数）？

因为 8253A 的 OUT 引脚是的计数器从 1 减少到 0 是有变化，设置计数值为 0 时，第一时钟到来时数值减 1 后变为 65535，OUT 引脚并不变化，则设置计数值为 0 则比 65535 还多一个，相当于 65536.

取最大计数值 65536， 则时钟为 1.193MHz 时， 在周期工作方式时，周期 = 1.193MHz/65536 = 18.2Hz

29．若要用 8253A 产生 10ms 周期的信号，做为 8259A 中断请求输入，其中 8253A 的时钟 CLK 频率为 1MHz，问 8253A 计数器需要工作在哪两个工作方式之一？并计算出 8253A 的计数初值是多少？

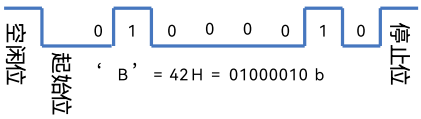
**方式 2，方式 3，时间常数= 10000**

30．8251A 在接收数据时可检测到几种错误？每一种错误是如何产生的？

接收数据位奇偶校验不对，则标志有奇偶错误，在状态寄存器中 PE 会置 1；CPU 还没在把上一个接收的数据取走，下一个数据已经到来，则产生溢出错误，在状态寄存器中 OE 会置 1；当没在检测到停止位时，产生帧错误，状态寄存器中的 FE 会置 1。

31．在 8251A 异步方式时，接收时钟 RxC 和发送时钟 TxC 都等于 115200Hz，波特率因子为 1，通信格式为 115200、8、N、1，即波特率为 115200、8 个数据位、无奇偶校验位和一个停止位。计算每秒可以传送多少字节？ 并画出传送字符‘B’的帧格式。

115200/10 = 11520 字节/s




32．ADC0809 的输入电压范围为 0 ~ 5V、参考电压 Vref+为 5V， Vref-为 0V。当 CPU 接收到的 ADC0809 数字量为 32 时，外部输入到 ADC0809 的电压是多少伏?

5/256 = V/32， V = 32*5/256 = 5/8 = 0.125V

33．DAC0832 的参考电压 Vref 为 3V，经过运算放大器后的输出电压范围为 0 ~ 3V，若送到 DAC0832 的数字量是 128，DAC0832 输出电压是多少伏?

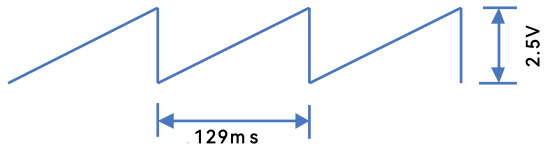
3/256 = V/128， V = 128*3/256 = 1.5V

34．DAC 采用单极性输出，参考电压 Vref 为 5V，输出电压范围为 0V ~ +5V。假设运行如下程序，试画出 DAC 输出的电压波型，并在图上标出最大峰值电压和周期值。

```
    org 100h
    mov dx, PORT_DAC ; DAC 端口地址
st:mov al, 0
nxt:out dx, al
    inc al
    call delay_1ms ; 延时 1 毫秒
    cmp al, 80h
    jbe nxt
    jmp st
ret
```

答:

峰值电压= 5V*80H/256 = 5V*128/256 = 2.5V
每个周期输出 0 到 80H，共 129 个值，每个周期约等于 1ms，则周期为 129*1ms = 129ms




35． DAC0832 采用单极输出，参考电压 Vref 为 5V，输出电压范围为 0V ~ +5V。假设要输出如下图所示的方波，需要交替写入到 DAC0832 的数字量是哪两个数字量？交替写入 DAC0832 两个数字量之间的软件延时子程序的延时是什么毫秒？


**方波频率**f = 100Hz

**交替输出 0 和 160（0A0H），间隔 = (1/100)(1/2)秒 = 5ms**

36． 什么是键盘的行扫描法和线反转法？其实现过程有哪些区别？

行扫描法是步进扫描方式，行输出，列输入。每次输出的值中只有一位是 0，其它都是 1，每次输出的 0 位置是移动的，以低电平扫描行输出。

每扫描输出一行，同时通过检查列线的输入， 如果输入是全 1，说明本行没有按键，则输出下一个行扫描值， 再进行检查列输入值，如果输入值不是全 1， 则当前行扫描值对应的非 0 的行有按键，当前列值非 0 对应列有按键，由此时的行值和列值可以定出按键的位置。

线反转法是首先行输出列输入，之后反转为行输入列输出，行列输入输出要有可以改变方向的功能。首先行输出全 0，检查输入列值，如果是全 1，则没有按键，如果不是全 1，则保存读入的列值，之后反转行为输入列为输出，把上次读入的列值从列中输出，检查读入的行值并保存。由读入列值和反转后读入的行值可以确定按键的位置。

行扫描法只要求行输出，列输入，接口简单，但是软件复杂；线反转法要求接口具有方向可变功能，硬件复杂，而软件实现简单。

37． 选用如下图给出的元器件设计一个恒温箱温度采集控制系统。该系统有两个状态：设置状态和控制状态。在设置状态时，通过键盘可以修改恒温箱的**设定温度**；在控制状态时，用开关量输出进行简单控制。**检测温度**与**设定温度**进行比较，当**检测温度**小于**设定温度**时，控制继电器加热；当**检测温度**大于**设定温度**时，关闭加热。当有按键时，发出 **1kHz 声音**，用于按键提示。

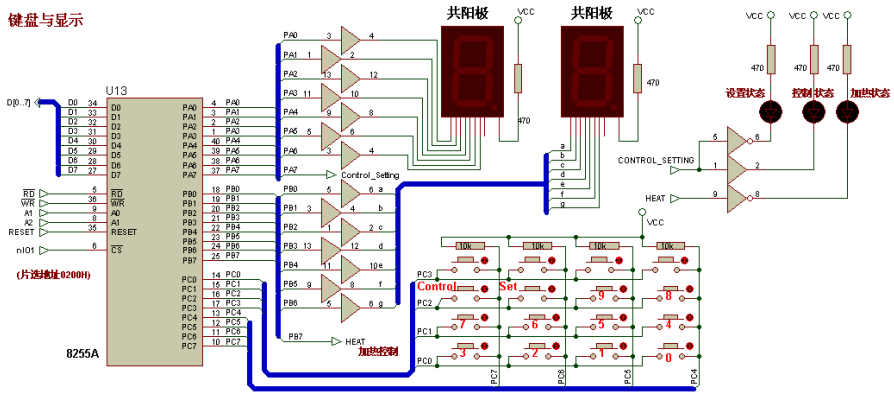
系统有两位七段数码管显示温度值（0~99℃）。在设置状态时，系统显示设定温度；在控制状态时，系统显示当前检测温度。系统通过 4x4 键盘输入设定温度值和启动控制，键盘有 0~9 键、Setting 键和 Control 键共 12 个键可用。

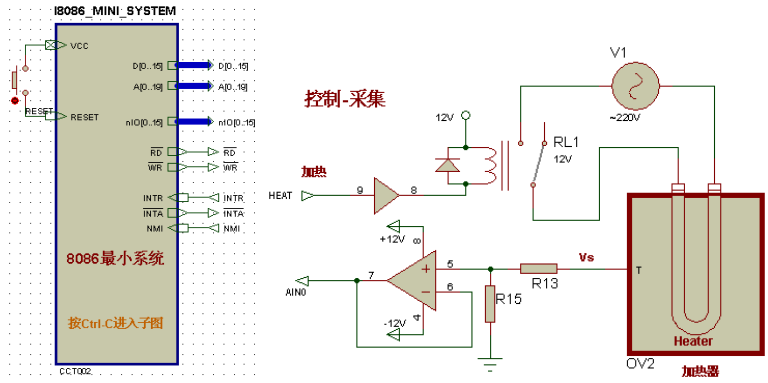
画出系统的硬件连接原理图，并标明分配给各元器件的端口地址。

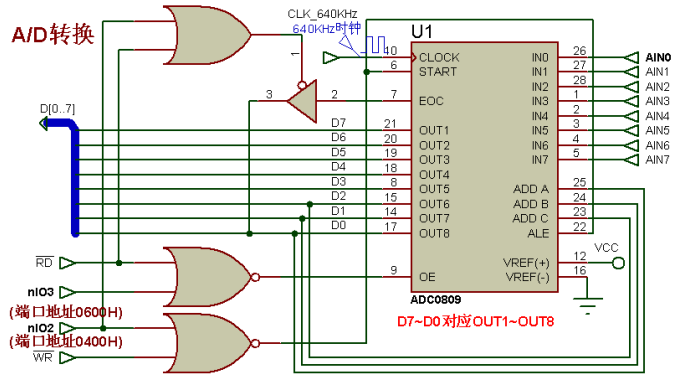
写出“0”对应的七段数码管译码值；编写 8255、8253 初始化程序；

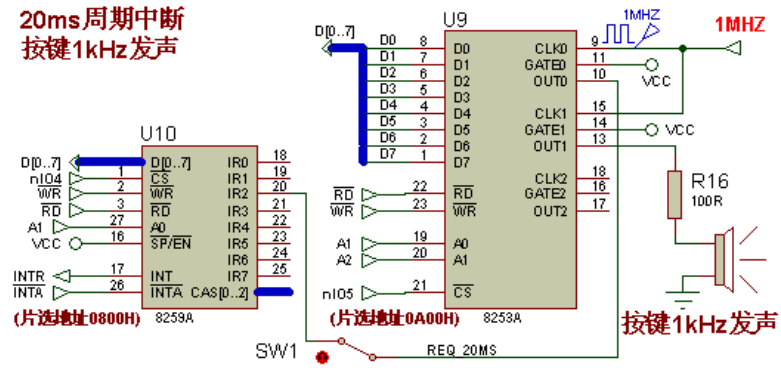
（3）编写 AD 转换子程序（adc）、显示子程序（display）、按键识别子程序（key）和主程序（main）。

（1）硬件原理图如下




（2）写出“0”对应的七段数码管译码值；编写 8255、8253 初始化程序；

“0”对应的七段数码管译码值是：0C0H

```
;8255 初始化
init_8255 proc near
      mov dx, PORT_CTR_8255
```

```
    mov al, 10001000b   ;初始化 8255 控制字
    out dx, al
    ret
init_8255 endp

;8253_计数器 0 初始化---20ms 中断请求
init_8253 proc near
    mov dx, PORT_CTR_8253
    mov al, 00110100b   ;初始化 8253 控制字,计数 0，r/w 低 8 位 ,高 8 位，方式 2，二进制计数
    out dx, al
    mov dx, PORT_COUNTER0_8253
    mov ax, 20000       ;计数常数   ; 20ms/(1/1Mhz) =20000
    out dx, al ;写低 8 位
    mov al, ah
    out dx, al ;写高 8 位
    ret
init_8253 endp

;开始发声，8253 计数器 1，方式 3，  1kHz 方波----
startBeep proc near
    mov dx, PORT_CTR_8253
    mov al, 01110110b ; 初始化 8253 控制字,计数 1，r/w 低 8 位 ,高 8 位，方式 3，二进制计数
    out dx, al
    mov dx, PORT_COUNTER1_8253
    mov ax, 1000 ; 计数常数   ; 1mhz/1khz =1000
    out dx, al ; 写低 8 位
    mov al, ah ; 写 8 位
    out dx, al
    ret
startBeep endp
;----5.2 停止发声，方式 0----
stopBeep proc near
    mov dx, PORT_CTR_8253
    mov al, 01110000b ; 初始化 8253 控制字，计数 1,方式 0
    out dx, al
    ret
stopBeep endp
;----5.3 发声 100ms----
beep_200ms proc near
    push dx
    call startBeep
    delay 0ffh
    call stopBeep
    pop dx
    ret
beep_200ms endp
```

（3）编写 AD 转换子程序（adc）、显示子程序（display）、按键识别子程序（key）和主程序（main）。

```
Adc proc near
    ; 启动 AD 转换
      mov dx, PORT_START_0809
      mov al, 0
      out dx, al    ;选择通道 0
      mov is_adc_started, 1; 下次是查询状态
  sample_ch0:
    ; 查询转换结束引脚 eoc，非阻塞方式
      mov dx, PORT_EOC_0809
      in al, dx
      test al, 00000001b ;eoc 引脚为高电平?
      jz exit_isr
      ;eoc 变高到此，输入 adc 转换结束数值
      mov dx, PORT_DATA_0809
      in al, dx
      mov [sample_ch0_val], al; 保存 AIN0 采集数字量
      mov is_adc_started, 0 ; 下次是启动 AD 状态
      call calc_ad2tmp; 计算当前温度值
  exit_isr:
ret
Adc endp

display proc far
      push ax
      push bx
      push cx
      mov bx, offset led_table
      cmp is_in_control_state, 1
      je    get_cur_tmp_val
      mov cl, setting_tmp_val
      jmp start_display
 get_cur_tmp_val:
      mov cl, sample_tmp_val
 start_display:
      mov al, 0
to_w10:
      cmp cl, 10
      jb    display_w10
      inc   al
      sub cl, 10
      jmp to_w10
    …
      cmp is_on_heatting_state, 1; 在加热时
      je on_heatting
```

```
        and al, 01111111b
        jmp out_w1
    on_heatting:
        or al, 10000000b
    out_w1:
        mov dx, PORT_B_8255
        out dx, al ;显示个位
        pop cx
        pop bx
        pop ax
        ret
display endp


key_identify proc near
        ;(1)判断是否有键按下
        mov dx, PORT_C_8255
        mov al,0
        out dx, al; PC3~0 行输出全 0
        nop
        in al, dx ;读入列值 PC7~4
        and al, 11110000b
        cmp al, 11110000b
        jne re_confident
        mov is_new_key, 0;   无新键值
        jmp error_exit
        ;(2)有键按下，软延时，再次判断
    re_confident:
        delay 0ffh
        in al, dx ;读入列值 PC7~4
        and al, 11110000b
        cmp al, 11110000b
        jne has_key
        mov is_new_key, 0;   无新键值
        jmp error_exit
        ;(3)识别按键。确实有键按下，开始扫描，
  has_key:
        mov ah, 11111110b
        mov cx,4
    scan_next_row:
        mov al, ah
        out dx, al
        nop
        in al, dx
        and al, 11110000b
        cmp al, 11110000b
```

```
        jne find_key ; 行值 ah, 列值 al
        rol ah, 1
        loop scan_next_row
        jmp error_exit
  find_key:
        and ah, 00001111b
        mov cl,4
        shr al, cl
        ; 计算键位值，行值 ah, 列值 al
        ;计算行计数值
      …
        mov ah, bl ; 保存行计数值 ah
        ;计算列计数值
        mov bl, -1 ;计数 0 位置
        mov cx, 4
    next_column:
        inc bl; 列号计数
        shr al, 1
        jnc find_column
        loop next_column
        mov is_new_key, 0;   无新键值
        jmp error_exit
  find_column:
        mov al, bl;保存列计数值 al
        shl ah, 1
        shl ah, 1; x4
        add al, ah ;al 键位置值
        mov cur_key, al;   保存当前键值
        mov is_new_key, 1;   ***有新键值
        call beep_200ms;发声
      ;(4)判断是否键释放，行输出全 0
no_release_wait:
        mov al, 0
        out dx, al ; PC7~4 列输入，PC3~0 行输出
        nop
        in   al, dx; 读入行值
        and al, 11110000b
        cmp al, 11110000b
        jne   no_release_wait
        delay 0ffh   ;键释放，软延时
    error_exit:
        ret
key_identify endp

main:
        ;----(0)初始化
```

```
    call init_vct_table ;中断时用
    call init_8255   ;8255A 初始化;!!!初始化 8255 时，端口数据消失
    call init_8253 ;中断时不用
    call init_8259 ;仿真时不用
    call display
    ; --->转到两个状态
main_loop:
    cmp is_in_control_state, 1
    …
 ;----(1)设置状态---------------
loop_in_setting_state:
    call key_identify;识别按键
    cmp is_new_key, 1
    jne loop_in_setting_state
    cmp cur_key, 0Bh; 0Bh,进入控制状态
    je change2control_state
    …
    jmp main_loop

;----(2)控制状态-----------
loop_in_control_state:
    call key_identify
    cmp cur_key, 0AH; 0AH,设置按键
    je   change2setting_state;  转到设置状态
   …
    call display
    jmp main_loop
    ret
```

38．　选择使用如下元器件，设计一个多路传感器电压采集显示系统。

设计要求:

（1）由两位共阳极七段数码管组成静态显示器，由八个按键组成线性键盘，8 通道的 ADC 连接 8 个传感器（由可变电阻代替），一路 DAC 的输出连接一个电压表， 并由 8259 和 8253 组成定时中断请求和按键提示音。

（2）每 20 毫秒由 8253 产生一次中断请求。设 8253 时钟 CLK 等于 1MHZ，8253 的 OUT0 连接到 8259 的 IR2 端，已知写入 8259 的 ICW2 是 08H，每次中断时，中断服务程序对 ADC 的 8 个通道传感器电压值采集一遍，中断服务程序函数名称为 isr_20ms_adc。

（3）当有按键按下时，发出 1kHz 提示声音，两位七段数码管显示与按键对应的 ADC 通道电压值（保留小数点后 1 位，单位为 V），K0 键对应第一个 ADC 通道，...,K7 对应第 8 个 ADC 通道，并用 DAC0832 输出这个通道传感器电压的值，用电压表可以测量。

（4）当无按键按下时， 不发提示音，用两位七段数码管显示八个传感器平均电压值（电压范围为 0.0V~5.0V，保留小数点后 1 位数，单位为 V），并用 DAC0832 输出八个传感器电压的平均值，用电压表可以测量。

试完成如下设计:

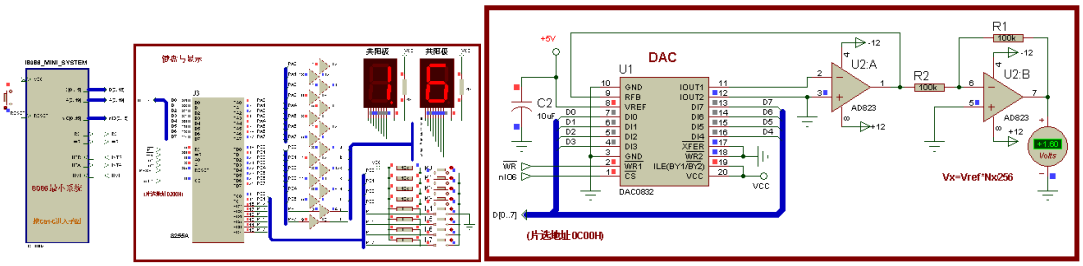
1. 画出与 8086 连接的硬件原理图。

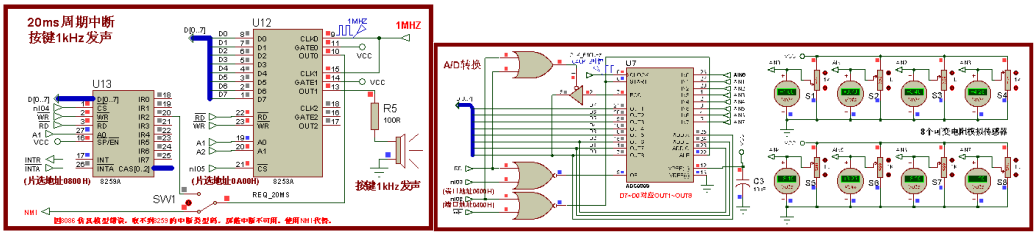
2. 编写 8255、8253 和中断向量表的初始化程序；写出七段数码管段码译码表；并编写多路传感器电压采

集显示系统的程序。

答:

（1）




（2）

```
; 由 MASM32 编译，并开始使用.IF .REPEAT .WHILE .UNTIL 等宏，可以不用低层的分支结构，
; 结构化更好。
; Author: hyp@jlu.edu.cn , QQ:hyper/790516
;----0.主程序----------------------------------------------------------------
main proc far
        ; (1)初始化
        call init_vct_table ;中断向量初始化
        call init_8255          ;8255A 初始化
        call init_8253          ;8253 初始化
        call init_8259          ; 8259 初始化
        enable_isr;  开中断
        ; (2)循环
        .WHILE 1
            call key_identify           ;判断和识别按键
            .IF key_is_pressed == 0 ;无键按下时
                call display_AV_Vx    ;显示 8 个通道平均电压
            .ELSE                               ;有键按下时
                call display_CH_Vx    ;显示当前按键对应 Kn 通道电压
            .ENDIF
        .ENDW
        ret
main endp

;1.中断向量表初始化
init_vct_table proc near
```

```
        ;mov bx, 4*12h ; n = 12H, IR2 有中断请求输入, ICW2=10H
        mov bx, 4*2       ; n = 2, NMI 非屏蔽中断
        mov ax, 0
        mov es, ax
        mov ax, offset isr_20ms_adc
        mov es:[bx], ax
        mov ax, seg isr_20ms_adc
        mov es:[bx+2], ax
        ret
init_vct_table endp
;--------------------------------------------------------------------------
;2.8255 初始化
init_8255 proc near
        out_port PORT_CTR_8255, 10001001b    ;初始化 8255 控制字
        ret
init_8255 endp
;--------------------------------------------------------------------------
;3.8253_计数器 0 初始化---20ms 中断请求
init_8253 proc near
        out_port PORT_COUNTER_CTR, 00110100b    ;初始化 8253 控制字,计数 0，r/w 低 8 位 ,高 8 位，
方式 2，二进制计数
        mov ax, 20000 ;计数常数    ; 20ms/(1/1Mhz) =20000, ; 40ms/(1/1Mhz) =40000
        out_port PORT_COUNTER0, al ;写低 8 位
        out_port PORT_COUNTER0, ah ;写高 8 位
        ret
init_8253 endp
;--------------------------------------------------------------------------
;4.8259 初始化
init_8259 proc near
        out_port PORT_8259_0, 00010010b    ;初始化 8259 控制 ICW1
        out_port PORT_8259_1, 08H; ICW2
        out_port PORT_8259_1, 01H; ICW4
        ret
init_8259 endp
;--------------------------------------------------------------------------
;5.发声 xxms----
beep_start proc near
        push ax
        out_port PORT_COUNTER_CTR, 01110110b ; 开始发声：初始化 8253 控制字,计数 1，r/w 低 8 位 ，
高 8 位，方式 3，二进制计数
        mov ax, 1000 ; 计数常数    ; 1mhz/1khz =1000
        out_port PORT_COUNTER1, al ; 写低 8 位
        out_port PORT_COUNTER1, ah ; 写 8 位
        pop ax
        ret
beep_start endp
```

```
beep_stop proc near
    push ax
    out_port PORT_COUNTER_CTR, 01110000b ; 停止发声：初始化 8253 控制字，计数 1,方式 0,
    pop ax
    ret
beep_stop endp
;6.中断服务程序--- 双状态，非阻塞方式
; 进两次采集一次 AD 值，40ms 采集一次，循环采集 8 个通道 AD 值,并计算 8 个 Vx_x10 电压值
isr_20ms_adc proc far
    .IF ( I_Flag == 1)
        push ax
        push bx
        push cx
        push dx
        push si
        pushf
        .IF is_adc_started == 0 ; 没有启动，则启动
            out_port PORT_ADC_START, cur_sensor_channel ;启动当前通道
            mov is_adc_started, 1; 标志已经启动
        .ELSE
            ; 查询转换结束引脚 eoc，非阻塞
            in_port PORT_ADC_EOC ; 输入值在 al 中
            test al, 00000001b     ;eoc 引脚为高电平?
            .IF !zero?  ; AD 转换结束?
                in_port PORT_ADC_DATA ; 输入值在 al 中
                mov si, offset sensor_Nx
                mov bh, 0
                mov bl, cur_sensor_channel
                add si, bx
                mov [si], al ; 保存当前通道 ADC 数字量
                ;计算 Vx_x100 电压，并保存。sensor_Vx_x100 = 100*5*Nx/256
                mov si, offset sensor_Vx_x100
                shl bx, 1 ; x2
                add si, bx
                mov ah, 0
                mov cx, 500
                mul cx ; dxax = ax *cx
                mov cx, 256
                div cx ; ax = dxax/cx
                mov [si], ax
                ; 下次采集下一个通道
                inc cur_sensor_channel
                .IF cur_sensor_channel == 8
                    mov cur_sensor_channel, 0
                .ENDIF
                mov is_adc_started, 0 ; 下次是启动 AD 状态
```

```
                .ENDIF
            .ENDIF
            popf
            pop si
            pop dx
            pop cx
            pop bx
            pop ax
        .ENDIF
        iret
isr_20ms_adc endp
;--------------------------------------------------------------------------
;7.识别键盘---8 个线性按键，非阻塞。出口：保存按键位置值到当前按键变量 cur_key
key_identify proc near
        in_port PORT_C_8255 ; ;读入列值 PC7~0
        ;(1)判断是否有键按下
        .IF al == 11111111b
            mov key_is_pressed, 0 ;无按键
            call beep_stop              ;停止发声
        .ELSE
        ;(2)有键按下，软延时，再次判断
            ;delay 0ffh
            in_port PORT_C_8255 ;读入列值 PC7~0
            .IF al == 11111111b ;再次判断
                mov key_is_pressed, 0 ;无按键,是干扰
                call beep_stop              ;停止发声
            ;(3)确实有键按下，识别具体按键。
            .ELSE
                mov ch, 8 ;共 8 个按键
                mov cl, 0 ;按键序号
                .WHILE ch != 0
                    shr al, 1
                    .IF carry? ;C=1?
                        inc cl
                    .ELSE
                        .BREAK ;C=0?
                    .ENDIF
                    dec ch
                .ENDW
                mov cur_key, cl ;保存键序号值到当前按键变量
                mov key_is_pressed, 1 ;有按键
                call beep_start ;发声
            .ENDIF
        .ENDIF
        ret
key_identify endp
```

```
;--------------------------------------------------------------------------
;8.Kn 按键按下时，显示 ADC n 通道的电压值，并输出这个电压值到 DAC
display_CH_Vx proc near
    mov si, offset sensor_Vx_x100
    mov bh, 0
    mov bl, cur_key
    shl bx, 1 ; x2
    add si, bx
    disable_isr   ;关中断
    mov ax, [si] ;取某通道的 sensor_Vx_x100
    enable_isr    ;开中断
    ;计算百位和十位数电压数值
    mov bx, offset led_table
    mov cl, 100
    div cl ; ahal = ax/cl
    xlat      ;al <-- [bx + al]
    and al,01111111b ;加小数点
    out_port PORT_A_8255, al ;显示带小数点的百位数
    mov al, ah
    mov ah, 0
    mov cl, 10
    div cl ; ahal = ax/cl
    xlat      ;al <-- [bx + al]
    out_port PORT_B_8255, al ;显示十位数
    ; DAC 输出当前通道的电压
    mov si, offset sensor_Nx
    mov bh, 0
    mov bl, cur_key
    add si, bx
    disable_isr   ;关中断
    mov al, [si] ;取当前通道 Nx 值
    enable_isr    ;开中断
    out_port PORT_DAC_DATA, al
    ret
display_CH_Vx endp
;--------------------------------------------------------------------------
;9.显示 8 个通道平均电压
display_AV_Vx proc near
    ; 计算 8 个通道的平均 Nx 值
    mov bx, offset sensor_Nx
    mov cl, 8
    mov ax, 0
    mov dx, 0
    clc
    .WHILE (cl != 0)
        disable_isr   ;关中断
```

```
            mov dl, [bx]
            enable_isr      ;开中断
            adc ax, dx
            inc bx
            dec cl
        .ENDW
        mov cl, 8
        div cl; al = ax/cl
        mov sensor_Nx_AV, al
        out_port PORT_DAC_DATA, al; DAC 输出平均电压
        ; Vx_x100 = 500*Nx/256
        mov ah, 0
        mov cx, 500
        mul cx ; dxax = ax*cx
        mov cx, 256
        div cx ; ax = dxax/cx, sensor_Vx_AV_x100
        ; 计算百位和十位数电压数值
        mov cl, 100
        div cl ; ah(余)al(商) = ax/cl
        mov bx, offset led_table
        xlat        ;al <-- [bx + al]
        and al,01111111b;加小数点
        out_port PORT_A_8255, al ;显示百位
        mov al, ah ; ah(余)
        mov ah, 0
        mov cl, 10
        div cl ; ahal=ax/cl
        xlat        ;al <-- [bx + al]
        out_port PORT_B_8255, al ;显示十位数
        ret
display_AV_Vx endp
;-------------------------------------------------------------------------------
end
```