《微机系统》练习题

1．微处理器、微型计算机、微型计算机系统的区别是什么？

微处理器：一般也称中央处理器（CPU），是本 du 身具有运算能力和控制功能，是微型计算机的核心。

微型计算机：由 CPU、存储器、输入输出接口电路和系统总线构成。

微型计算机系统：以微型计算机为主体，配上系统软件和外部设备以后，就成为了计算机系统。

2．微型计算机有哪些基本部分构成？

中央处理器、总线、存储器、输入/输出接口

3． CISC、RISC 的技术特点。

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。

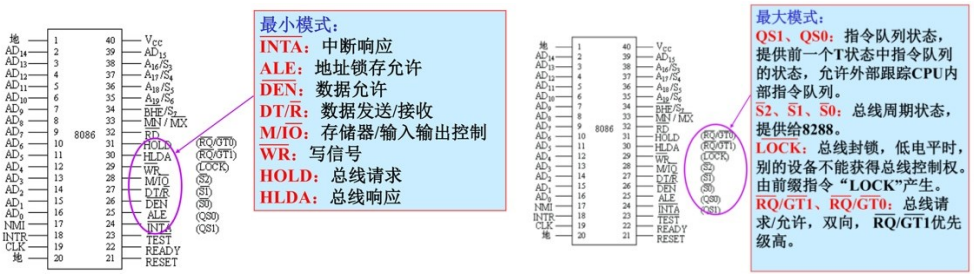
RISC CPU 的电路构成比 CISC CPU 简单，因此面积小、功耗也更低；CISC 电路 CPU 电路复杂，同水平比 RISC CPU 面积大、功耗大。RISC CPU 结构比较简单，布局紧凑规整，设计周期较短，比较容易采用一些并行计算的最新技术；CISC CPU 结构复杂，设计周期长，技术更新难度大。从使用角度看，**RISC 微处理器结构简单，指令规整，性能容易把握，易学易用；CISC 微处理器结构复杂，功能强大，实现特殊功能容易。**

4． 8086 的两种工作方式由什么决定？两种工作方式有什么差别?

由 MN/#MX 确定最小/最大模式；

不同：

1. 部分引脚功能不同：



2. 最小模式是单机系统，只有一个 8086CPU，所有控制信号由该 CPU 产生。系统中的控制电路可减到最小

最大模式是多机系统，有两个以上 CPU，一个主 8086CPU，其他为协处理器（如 8087、8089），控制信号由总线控制器 8288 产生。

5．8086CPU 的组成与功能。

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**

执行部件 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。**

6．8086 微处理器有那几个段寄存器？简述他们的主要功能。

4 个，CS 内容指出当前代码段，SS 内容指出当前程序所使用的堆栈段，DS 指出了当前程序使用的数据段，ES 指出了当前程序使用的附加段。

7．什么是物理地址、逻辑地址？逻辑地址 2000:2345H 对应的物理地址是多少？

物理地址：信息在存储器中**实际存放的地址**。地址通过地址线给出。

逻辑地址：编程中使用的地址

22345H

8．若代码段寄存器 CS＝3200H，指令指针 IP＝0FF00H，此时，指令存放的物理地址是什么？说明指向这一物理地址的 CS 和 IP 值是否唯一。

41F00H

不唯一，如 CS=3300H，指令指针 IP=0EF00H，指向的物理地址仍然是 41F00H。

9．对齐数据与非对齐数据的不同结构对数据的访问速度的影响。

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。

10．简述 Pentium 微处理器的主要性能特点。

采用超标量双流水线结构；

采用两个彼此独立的高速缓冲储存器：

采用全新设计的增强型浮点运算器；。

常用指令进行了固化及微代码改进，一些常用的指令用硬件实现。

11．Pentium 有什么主要技术特点？至少说出四点。

1. 超标量流水线，超标量为 2
2. 分立 Cache，指令数据各 8K
3. 重新设计的浮点运算部件，8 级流水
4. 动态转移预测
5. 内部结构 32 位，外部 DB64 位，属于 32 位机
6. 4 种工作方式：实地址、保护虚地址、虚拟 8086、系统管理

12．说明 Pentium 处理器引脚#Cache 和#KEN 的作用.

#Cache: **Cache 控制**，指示目前处于 Cache 周期

#KEN: **Cache 允许**，有效时，指示**进入突发读周期**，将外部数据复制到内部 Cache（Cache 的填充）。

13．说明 Pentium 处理器引脚＃BE7～＃BE0 的作用。

#BE7-#BE0: **字节允许**，A2-A0 在 CPU 中被编码，形成#BE7-#BE0 的输出（并不是 A2-A0 的简单译码），**用于字节选择**。

14．什么是指令周期？什么是总线周期？什么是时钟周期？说明三者的关系。

时钟周期：CPU 脉冲时间（CPU 频率的倒数）。

总线周期：CPU 经过总线执行信息的输入/输出过程，称为总线周期。

指令周期：完成一条指令的时间。

关系：一个总线周期包含多个指令周期，一个指令周期包含多个时钟周期。

15. 说明寄存器 EAX、AX、AH、AL 之间的关系。

累加器 EAX 既可以保存算数喝逻辑运算的操作数，也可以保存地址。他的名称来源于 8086 处理机的通用寄存器 AX，所以，EAX 的低 16 位可按原来的名字访问。16 位寄存器 AX 每个字节均有另一个名字，字节寄存器命名为 AH（高字节）和 AL（低字节），这些 8 位通用寄存器也可以按原来的名字访问。

16. Pentium 总线操作有几种时钟状态？T12、T2P、TD 与一般的时钟状态有什么区别？

Ti：空闲状态，**不执行总线周期，处于等待、就绪状态**，BOFF、RESET 信号进入 Ti。

T1：总线状态 1，地址、状态有效，ADS 有效，外部电路可以**利用 ADS 将地址和状态送入锁存器**，它是总线周期的第 1 个时钟周期。

T2：总线状态 2，数据有效，它是总线周期的第 2 个或更后一些的时钟周期，此时 **CPU 对 BRDY 采样，为低就绪，为高等待**。

T12：流水线状态 1，两个总线周期部分重叠，**M1 的 T2 与 M2 的 T1 重叠**。

T2P：流水线状态 2，两个总线周期部分重叠，**M1 的 T2 与 M2 的 T2 重叠**。若 T12 已经完成，而 M1 仍未结束，就会在 T12 之后形成 T2p，一般出现在存储器或外设较慢的情况。

TD：休止状态，读写转换的过渡状态，出现在 T12 之后，插入在读写操作之中，用于调整，此时，**数据无效，CPU 不对 BRDY 采样**。奔腾在流水线访问期间使用 TD，非流水线的读写转换通常用 Ti 描述。

17. IP/EIP 寄存器的用途是什么？

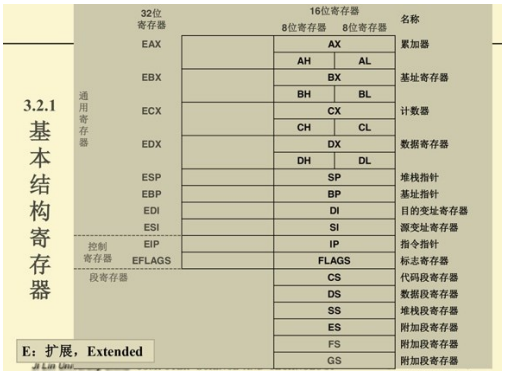
IP/EIP 内容为下一条要取入 CPU 的指令在内存中的偏移地址。CPU 复位后，IP/EIP 清零。每取一条指令，IP/EIP 自动增加取入 CPU 的字节数目。

18. Pentium 微处理机在实模式下操作时，段寄存器的用途是什么？

在实地址模式下，Pentium 的运行方式为 8086 方式，CPU 可以直接访问段寄存器，每个段寄存器定义一个 64KB 存储器段的起点，既给出相应的段基址。CS 内容指出当前代码段，SS 内容指出当前程序所使用的堆栈段，DS 指出了当前程序使用的数据段，ES 指出了当前程序使用的附加段。

19. Pentium 的寄存器组包括哪些类型的寄存器？简要说明基本结构寄存器、系统级寄存器的用途。

基本结构寄存器：16 个，其中通用 8、控制 2、段寄存器 6



系统级寄存器：9 个，其中系统 4、控制 5

四个系统地址寄存器（GDTR、IDTR、LDTR、TR）、五个控制寄存器（CR0-CR4，用于控制管理）

调试寄存器：8 个

模型专用寄存器：20 个

浮点寄存器：18 个

20. 在 Pentium 处理器总线周期中，＃NA 有效具有什么样的意义？

#NA: **下一地址有效**，用于支持地址流水线操作。该引脚有效的两个时钟周期后，CPU 可送出新地址（启动下一条指令）

21. CPU 与存储器连接时主要应考虑哪些问题？

（1）CPU 总线驱动能力。

（2）CPU 的时序与存储器存取速度之间的配合。

（3）数据线的连接。

22. 什么是高速缓存? 计算机中设置 Cache 的作用是什么?能不能把 Cache 的容量扩大，最后取代主存?

为弥补主存速度，在 CPU 与主存之间设置的高速、小容量的存储器，构成 Cache-主存存储层次，速度是 Cache 的，容量是主存的。

不能，Cache 容量越大，价格越高，且如果取消主存，当 CPU 访问 Cache 失败时，**需要将辅存的内容调入 Cache 再由 CPU 访问，造成 CPU 等待时间太长，损失更大**。

23. 说明伪指令 END、ENDS、ENDP、ENDM 的区别。

END 程序汇编结束

ENDS 段定义结束

ENDP 过程定义结束

ENDM 宏定义结束

24. 指出下列指令的错误：（1）MOV AH，BX（2）ADD 15H，BX（3）MOV CS，AX（4）MOV AX，[SI][DI] （5）MOV BYTE PTR [BX]，1000

不可以将 16 位数据存入 8 位寄存器

不可以将立即数作为目的操作数

不可以修改代码段寄存器

[SI][DI] 都是变址寄存器，没有变址+变址这样的寻址方式的，所以是错的，只有变址+基址寻址方式

1000 无法用 8 位表示

25. 设要在地址为 DAT1 的数据区中顺次存放以下数据：’A’，’B’，0，0，’C’，’D’，0，0，写出分别用命令 DB、DW 和 DD 实现的语句。

```
DAT1 SEGMENT
     DB  ‘AB’,0,0, ‘CD’,0,0
     DW  ‘BA’,0,0, ‘DC’,0,0
     DD  ‘’
```

26. 简述子程序与宏指令的区别。

1、在源程序中，通过**书写宏名来引用宏**，而**子程序是通过 CALL 指令来调用**；
2、汇编程序对宏**通过宏扩展来加入其定义体**，宏引用多少次，就相应扩展多少次，所以，引用宏**不会缩短**目标程序；而子程序代码在目标程序中只出现一次，调用子程序是执行同一程序段，因此，目标程序也得到相应的简化；
3、宏引用时，参数是通过“实参”替换“形参”的方式来实现传递的，参数形式灵活多样，而子程序调用时，参数是通过寄存器、堆栈或约定存储单元进行传递的；
4、宏引用语句扩展后，目标程序中就不再有宏引用语句，运行时，不会有额外的时间开销，而子程序的调用在目标程序中仍存在，子程序的调用和返回均需要时间。

27. 说明 RET，END，HLT 的用途和区别.

RET 子程序返回指令
END 汇编结束指令
HLT 暂停指令 停止软件的执行，暂停的三种方式：中断、硬件复位、DMA 操作。

28. 执行下面的指令序列后，写出标志寄存器 CF、PF、ZF、SF、OF 位的状态。

```
MOV AX，35E5H
MOV BX, 7832H
ADD AX, BX
```

答：CF=0、PF=1、ZF=0、SF=1、OF=1

29. 阅读下列程序段，将执行结果填入后面的空格内。

```
MOV AX，BX
     NOT AX
     ADD AX，BX
     INC AX
     AX=____，CF=____
```

答：AX=0，CF=0

30. 说明下列指令中源操作数的寻址方式。

(1) MOV CX，24[BX][SI]
(2) MOV BX，[2000H]
(3) INC BYTE PTR[BX]
(4) MOV AX，[BP+DI]

答：

(1) 相对基址变址
(2) 直接寻址
(3) 寄存器间接寻址
(4) 基址变址

存储器寻址方式举例

寻址方式	有效地址EA如何计算?	基址	变址	比例因子	位移量
1. 直接寻址	MOV BX, [1234H]				√
2. 寄存器间接寻址	MOV AL, [BX]	√*	√*		
3. 寄存器相对寻址	MOV BX, [SI+40H]	√*	√*		√
4. 基址变址寻址	MOV DX, [BX+SI]	√	√		
5. 相对基址变址寻址	MOV AX, [BP+DI+2]	√	√		√
6. 比例变址寻址	MOV EAX, [4×ECX]		√	√	
7. 相对比例变址寻址	MOV EDX, [4×ECX+5]		√	√	√
8. 基址比例变址寻址	MOV AL, [EBX+2×EDI]	√	√	√	
9. 相对基址比例变址寻址	MOV AL, [EBX+2×EDI-2]	√	√	√	√

31. 解释 PUSH DI 指令是怎样工作的

（1）SP←SP-1，调整 SP 指针。

（2）将 DI 内容的高 8 位压入堆栈指针 SP 所指的存储单元。

（3）SP←SP-1，调整 SP 指针。

（4）将 DI 内容的低 8 位压入堆栈指针 SP 所指的存储单元。

32. 对于下面的数据定义，各条 MOV 指令单独执行后，AX 寄存器的内容是什么？

```
    ORG 2000H
A1    DB    ?
A2    DW    20 DUP(? )
A3    DB    'ABCD'
A4    DW    10 DUP(? )
      (1)  MOV    AX, TYPE  A1
      (2)  MOV    AX, OFFSET  A2
      (3)  MOV    AX, LENGTH  A3
（4）MOV    AX, SIZE  A4
```

答：

(1) 1

(2) 2001H

(3) 1 （如果用 DUP，则得到元素个数，否则总为 1）

（4）20

33. 伪指令的作用是什么？

伪指令又称为操作，他们不像机器指令那样在程序运行期间由计算机来执行的，而是**在汇编程序对源程序汇编是由汇编程序处理的操作**。伪指令主要完成**处理器选择、定义程序模式、定义数据、分配存储区、指示程序结束**等功能。

34. 标号的作用是什么？

代码段中的名字字段叫做标号。它用来**表示一个指令语句的符号地址**，可以用该符号地址来访问该指令。

35. 假设 VAR12 和 VAR34 为字变量，LAB 为标号，试指出下列指令的错误之处。

（1） ADD VAR12 ，VAR34

ADD 指令的源操作数和目的操作数**不能同时为内存操作数**，但 VAR12 和 VAR34 都是内存操作数。

（2） SUB AL ，VAR12

VAR12 是字变量，而 AL 是字节型的，不匹配

（3） JMP LAB [DI]

JMP 跳转指令，后面直接跟指令标号，所以不应该有[DI]..

（4） JNZ VAR12

JNZ 条件跳转指令，后面直接跟指令标号，但是 VAR12 不是标号。

36. 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 0DH
    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
```

37. 设从 BUFFER 开始存放若干个以$为结束标志的带符号字节数据，试编写将其中的正数按由大到小的顺序存入 PLUS 开始的缓冲区中的完整源程序。（编程注意四个环节：段的定义；数据定义；提取带符号字节数据中的正数；由大到小排序。）

答：

```
DATA    SEGMENT
    BUFFER  DB
11H,12H,0FEH,0EAH,32H,0A3H,22H,14H,54H,0C3H,09H,45H,53H,37H,38H,'$'
    PLUS    DB 15 DUP(0)
    STR     DB 0AH,0DH,"PLUS==>$"
DATA    ENDS
CODE    SEGMENT
    ASSUME      CS:CODE,DS:DATA
MAIN    PROC FAR
    MOV AX,DATA
    MOV DS,AX
    CALL SUB1 ; Calculate / Positive Number to PLUS
    CALL SUB2 ; Sort PLUS
    CALL SUB3 ; output PLUS / for testing the program
    MOV AH,4CH
    INT 21H
```

```
MAIN    ENDP
SUB1    PROC NEAR
        PUSH AX
        PUSH BX
        LEA SI,PLUS
        LEA BX,BUFFER
        XOR AX,AX
        XOR DX,DX
AA1:    MOV AL,[BX]
        CMP AL,24H
        JZ AA3
        TEST AL,80H
        JNZ AA2
        MOV [SI],AL
        INC SI
        INC BX
        INC DX ; total of PLUS
        JMP AA1
AA2:    INC BX
        JMP AA1
AA3:    POP BX
        POP AX
        RET
SUB1    ENDP
SUB2    PROC NEAR
        PUSH DX
        DEC DX
L1:     MOV CX,DX
        LEA BX,PLUS
L2:     MOV AL,[BX]
        CMP AL,[BX+1]
        JNC L3
        XCHG AL,[BX+1]
        MOV [BX],AL
L3:     INC BX
        LOOP L2
        DEC DX
        JNZ L1
        POP DX
        RET
SUB2    ENDP
;;;;;;;;;;;;;;;;;;;;;;;;
SUB4    PROC NEAR
        PUSH AX
```

```
        PUSH CX
        PUSH DX
        MOV DL,[BX]    ; BX is the Parameter in this function
        MOV CL,4
        SHR DL,CL
        OR DL,30H
        CMP DL,3AH
        JC BB1
        ADD DL,7
BB1:    MOV AH,2
        INT 21H
        MOV DL,[BX]
        AND DL,0FH
        OR DL,30H
        CMP DL,3AH
        JC BB2
        ADD DL,7
BB2:    MOV AH,2
        INT 21H
        MOV DL,48H
        INT 21H
        MOV DL,20H ;Output Space
        INT 21H
        POP DX
        POP CX
        POP AX
        RET
SUB4    ENDP
SUB3    PROC NEAR
        PUSH DX
        MOV CX,DX
        LEA DX,STR
        MOV AH,9
        INT 21H
        LEA BX,PLUS
LL1:    CALL SUB4
        INC BX
        LOOP LL1
        POP DX
        RET
SUB3    ENDP
CODE    ENDS
        ENDMAIN
```

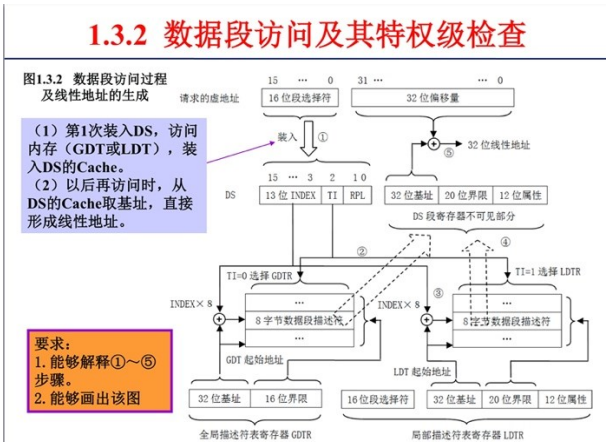
38. 已知在 ARRAY 数组中有 80 个无符号字节数据，编写一个完整的 8086 汇编语言程序，将 ARRAY 数组中的最大值放入 MAX 单元中。
39. 有 100 个无符号整数，编写完整汇编程序，求最小值及其位置。
40. 有 100 个无符号整数，编写完整汇编程序，求平均值及其位置。
41. 说明计算机中采用 Cache 和虚拟存储器的原因。
采用虚拟存储器：增加存储器的容量
采用 Cache：弥补主存速度的不足
42. 段描述符用于描述段的信息，它由 8 个字节组成。
43. Pentium 实地址模式的特点，8086 工作模式、Pentium 实地址模式、Pentium 虚拟 8086 模式之间的异同？
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。
44. IDTR、GDTR 和 LDTR 分别代表什么寄存器？其内容是什么信息？有什么作用？
IDTR：中断描述符表寄存器，保存门描述符，整个系统只有一个，包括中断门、陷阱门、任务门（通常没有调用门）
GDTR：全局描述符表寄存器，保存全局描述符表起始地址，
48 位 = 32 基址+16 界限，全局描述符表只有一个，各任务共享
LDTR：局部描述符表寄存器，保存局部描述符表的起始地址，保存某任务使用的段描述符，每个任务各有一个
45. 说明段描述符的组成及作用。
段描述符：8 个字节，用于描述段的基本信息，包括段的属性、段的大小、段在存储器中的位置以及控制和状态信息
46. 说明 CPL、RPL、DPL 的含义。

1.当前特权级 CPL：CPL 是当前正在执行的代码段所具有的访问特权级。

2.描述符特权级 DPL：DPL 是段被访问的特权级，保存在该段的段描述符的特权级 DPL 位。

3.请求特权级 RPL：RPL 是新装入段寄存器的段选择符的特权级，存放在段选择符的**最低两位**

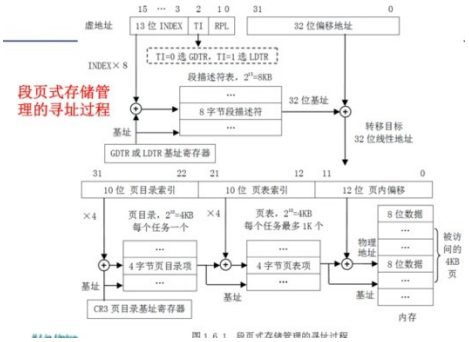
高特权级可以访问低（等于）特权级的数据;
低特权级可以调用高（等于）特权级的程序。
47. 任务状态段 TSS 的主要作用是什么？
保存现有任务的机器状态（指处理器的工作环境，比如各个寄存器的状态）及其任务间的关联信息。每个任务一个。
48. 简述 Pentium 通过 GDT 访问数据段的寻址 过程（也可画图说明）。



49. 简述 Pentium 通过 LDT 访问数据段的寻址过程（也可画图说明）。

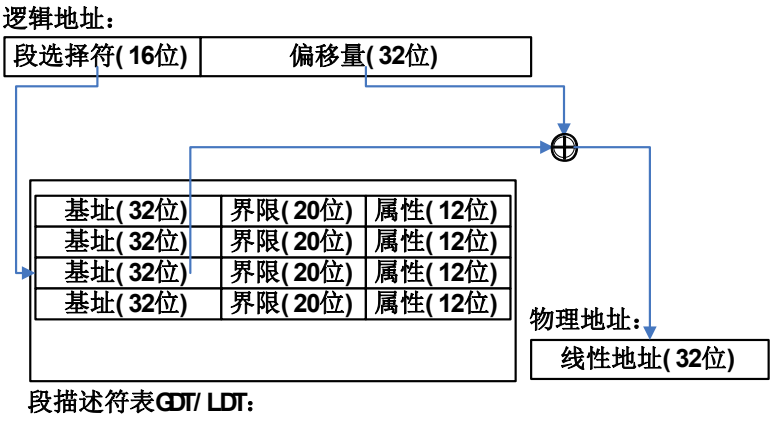
50. 说明段页式存储管理的寻址过程。

在段页式存储管理的寻址过程中，首先将虚地址通过**段式**存储管理部件转换为**线性地址**，然后将线性地址通过**页式**存储管理部件转换为**物理地址**。



51. Pentium 处理器采用分段存贮器管理时的数据访问逻辑地址转换为物理地址的过程。

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。



52. CPU 与外部设备之间为什么要使用接口？

微型计算机的外部设备多种多样，这些外设在工作原理、驱动方式、信息格式以及工作速度等方面彼此差别很大，它们不可能和 CPU 直接相连，必须借助于中间电路，即接口与 CPU 相连。

53. 什么是 I/O 接口？ 什么是 I/O 端口？

I/O 接口是位于 I/O 设备与 CPU（或系统总线）之间的电路；在微型计算机系统中，CPU 与外部设备之间的联系，需要有特定的硬件连接和相应的控制软件。完成这一任务的软、硬件的综合称为接口。对这种硬件、软件的设计，称为接口技术。

I/O 端口是指接口电路中那些完成信息传送，可由程序寻址并进行读写操作的寄存器，是 PC 访问外设资源（通常是寄存器）的一种特定方式。

54. 接口的基本功能是什么？

1.**地址译码或设备选择**：接口必须进行地址译码，从而产生设备选择信号，以使微处理器和指定的外部设备交换信息。

2.**数据缓冲和锁存**：在微计算机系统中，数据总线是系统各部分之间公用的双向总线，所有设备分时复用。所以，无论是存储器，还是外部设备，都不能长期占用数据总线，**只允许被选中的设备在读/写周期内可用其传送数据。**

3.**信息格式与电平的转换**：接口应该具有信号传送格式、信号类型、信号电平的转换能力。

4.**数据传送的协调**：CPU 工作是有一定的时序的，CPU 与外部设备交换数据时必须采用一定的传送方式进行控制。

55. 画出一个微型计算机 I/O 接口一般结构图，标明接口内部主要寄存器及外部主要信号线。

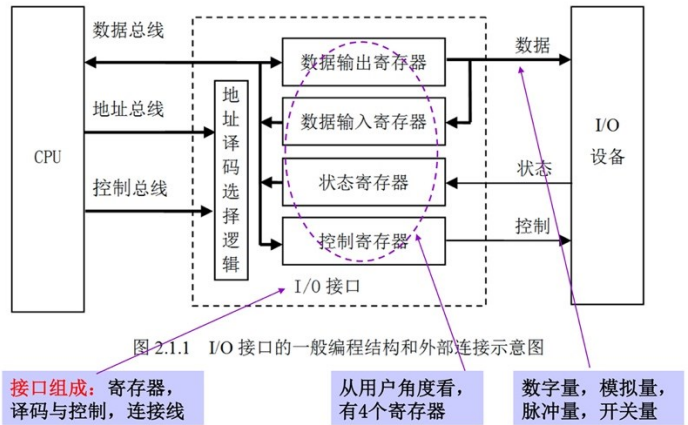


图 2.1.1 I/O 接口的一般编程结构和外部连接示意图

56. 计算机的 I/O 传送中，与程序查询传送和程序中断传送相比，DMA 传送的主要优点是什么？

在外设和内存之间直接传送数据，即直接存储器传输方式，进一步降低了对的 CPU 的打扰

57. I/O 接口有什么用途？

I/O 接口用来对 CPU 和外设的速度和工作方式进行匹配，协助完成 CPU 和外设之间数据传送和传送控制任务。其用途表现在以下几个方面：

（1）地址译码或设备选择

在微计算机系统中，可能有多个外部设备。当微处理器在不同时刻需要和不同的外部设备发生联系时，微处理器要用地址码来选择不同的外部设备。因此，接口必须进行地址译码，从而产生设备选择信号，以使微处理器和指定的外部设备交换信息。

（2）数据缓冲和锁存

在微计算机系统中，数据总线是系统各部分之间公用的双向总线，所有设备分时复用。所以，无论是存储器，还是外部设备，都不能长期占用数据总线，只允许被选中的设备在读/写周期内可用其传送数据。未选中的设备必须对总线呈高阻抗状态，与总线“脱离”，不影响其他设备使用总线。

（3）信息格式与电平的转换

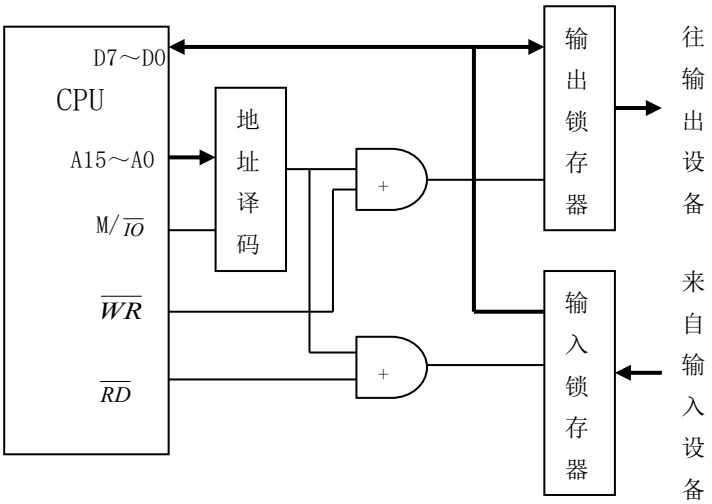
在微计算机系统中，信息是并行二进制代码。CPU 和内存的信息交换就采用并行处理。而有些外部设备，比如 CRT 显示器，其信息是串行数据，这就要求接口能把 CPU 输出的并行数据转换成串行数据，而把外部设备送来的串行数据转换成并行数据。此外，有些外部设备的信号电平与 TTL 电平不能兼容，所以还要有信号电平的转换。所以说，接口应该具有信号传送格式、信号类型、信号电平的转换能力。

（4）数据传送的协调

CPU 工作是有一定的时序的，CPU 与外部设备交换数据时必须采用一定的传送方式进行控制。比如采用查询方式传送数据时，就要先询问外部设备是否已具备了与 CPU 交换数据的条件。具体地说，输入设备要给出“数据是否准备就绪”的状态信号。输出设备要给出“忙”或“闲置”的状态信号，由 CPU 决定是否可以进行数据交换。

58. 设置无条件输入端口的前提条件是什么?画图说明一个典型的无条件输入端口的组成?

条件：输入设备可随时提供数据。



59. CPU 与外设之间的数据传送控制方式。

（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。

60. 说明 DMAC8237 的主要功能与特点。

基本功能：实现 DMA 控制器的功能

1．在一片 8237A 内**有 4 个独立的 DMA 通道**。 2．每个通道的 DMA 请求**可分别编程允许或禁止**。 3．每个通道的 DMA 请求优先级有两种：**固定优先级和循环优先级**。固定优先级的顺序是通道 0 最高，通道 3 最低。 4．可在外设与存储器，存储器与存储器之间传送数据。

61. 说明 8237A 四种基本传送方式的特点。

单字节传送方式：一次只传送一个字节，然后释放总线。每次传送后字节计数器减一，由 0 变为 FFFFH 后终止

数据块传送方式：响应一次 DMA 请求，将完成设定的字节数的全部传送

请求传送方式：又称查询方式，类似数据块传送，但每传送一个字节，检测 DREQ 状态，若无效，则终止，否则继续传送

级连方式：这种方式允许连接一个以上的芯片来扩展 DMA 通道数

62. 说明 8237A 三种传送类型。

1.DMA 读：把存储器中的数据读到 I/O 设备

2.DMA 写：把 I/O 设备中的数据写到存储器

3.DMA 校验：是一种伪传输，实际上是校验 8237 芯片内部的读写功能

63. 说明 8237A 中正常时序、压缩时序、扩展时序的含义。

正常时序：传送**一个字节数据包含 4 个时钟脉冲周期**，即 S1-S4 状态。产生的读写脉冲信号与这 4 个状态有确定的对应关系。若是数据块传送中不改变高 8 位地址，则省去 S1，只占用 S2、S3、S4 三个时钟周期。
压缩时序：把读命令的宽度压缩到等于写命令的宽度，省掉了 S3，即由 S4 完成读和写的操作。

在正常时序操作下，可选择**扩展写方式，即写命令提前到读命令**，从 S3 状态开始（一般情况下，读为 S3、S4 状态，写为 S4 一个状态）。

64. DMA 控制器 8237A 的信号线 IOW 和 IOR 是单向的还是双向的？为什么？

65. 说明 8237A 四种基本传送方式的特点。

①单字节传送方式：8237A 每次获得总线控制权后仅传送一个字节，传送之后修改通道　内当前地址寄存器和当前字节计数器内容，然后就释放系统总线。

②数据块传送方式：8237A 一旦获得总线控制权将连续传送完一个数据块，然后释放系统总线归还给 CPU 使用。

③请求传送方式：与数据块传送方式类似，但要求在数据传送期间必须保持 DREQ 信号有效。每传送一个字节的数据后 8237A 都要采样 DREQ，若 DREQ 无效，则停止 DMA 传送。

④级联传送方式：从片的 HRQ 和 HLDA 引脚分别连接到主片的 DREQ 和 DACK 上，从片的优先级和所连主片通道的优先级相对应。这种情况下，主片的连接通道只起两个作用，一是优先级连接作用，即将从片的 4 个通道纳入到主片的优先级管理机制；二是向 CPU 输出 HRQ 和传递 HLDA。

66. Pentium 微处理机内部有哪几类中断？简要说明各类的特点是什么

四类中断源：

可屏蔽中断 INTR：高电平有效，IF=1 时，且 CPU 在当前指令执行结束后，

会响应 INTR 引脚的中断请求

非屏蔽中断 NMI：用于重要事件处理，如电源掉电、存储器读写错误、总线

奇偶位出错等，不受 IF 的影响，上升沿触发，中断类型码固定为 2

软件中断（执行 INT0、INT3、INT n、BOUND 指令引起的中断）：

人为设置。中断类型码包含在指令中，所以**不执行中断响应总线周期**。除 BOUND 指令外，中断返回地址指向软件中断指令的**下一条指令**

异常：既不是外部硬件产生的也不是用软件指令产生的，而是在 CPU 执行一条指令的过程中出现错误或检测到的异常情况自动产生的

67. 什么叫中断向量？试说明 Pentium 微处理机可屏蔽硬件中断是怎样获得中断向量，从而进入中断程序的。

中断向量：中断服务程序的入口地址。

中断类型码*4 得到偏移地址，加上中断向量表的基地址，得到中断服务程序的入口地址

68. 中断描述符表的作用是什么？其内保存的是什么信息？

保存门描述符的起始地址，整个系统只有一个，包括中断门、陷阱门、任务门（通常没有调用门）

69. 说明 8259A 的中断优先权管理方式的特点。

每片管理 8 级中断优先权，通过级联，最多可管理 64 级优先权的中断源

70. 说明 8259A 的中断结束方式的特点。

分为自动中断结束方式 AEOI（8259A 自动地在最后一个 INTA 脉冲的后沿将 ISR 的相应位清零）和命令中断结束方式 EOI（在中断服务程序返回前，向 8259A 发中断结束命令，使 ISR 相应位清零）

71. 说明 8259A 中断控制器中的 IRR、ISR 和 IMR 三个寄存器的功能。

中断请求寄存器 IRR：用于寄存所有 IR 输入线输入的中断请求信号，即保存正在请求服务的中断级。接收来自某一引脚的中断请求后，IRR 寄存器中对应位置 1，也就是对这一中断请求中断做了锁存

中断服务寄存器 ISR：保存当前被 CPU 服务的中断级，也就是记录正在被处理的中断请求

中断屏蔽寄存器 IMR：存放 CPU 送来的中断屏蔽信号，某位=1，对应的中断请求被屏蔽

72. 8086 发出 INTA 的条件是什么？

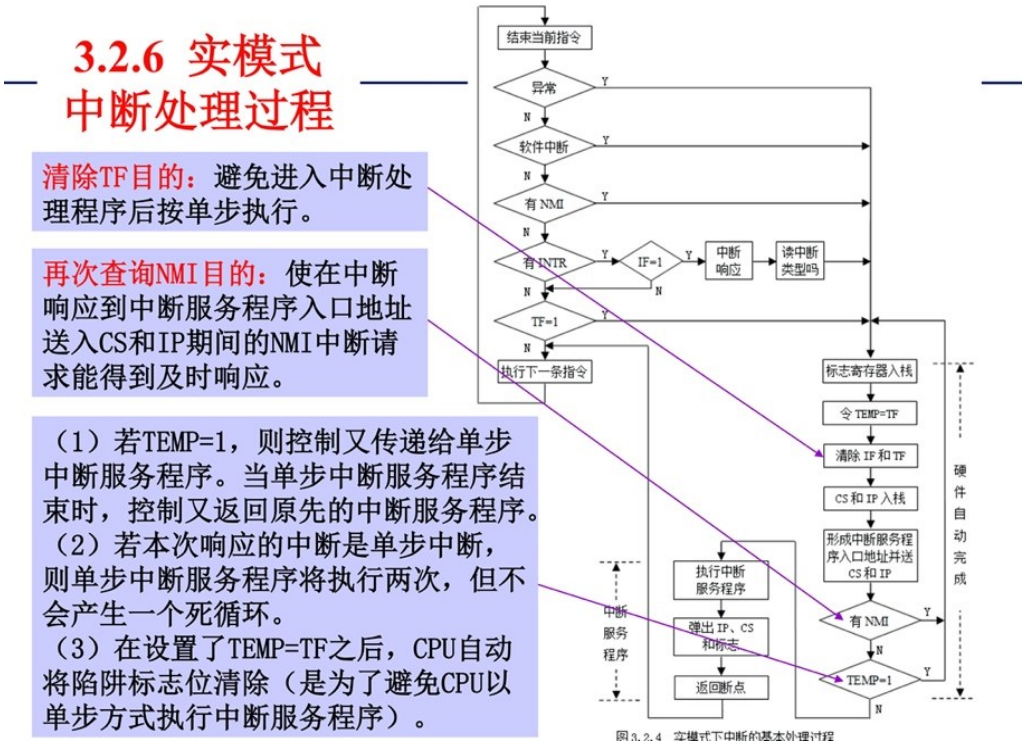
控制逻辑向 CPU 发送中断请求，CPU 响应时向 8259A 送两个 INTA 负脉冲

73. 说明中断控制器 8259A“特殊全嵌套方式”的含义和功能。

适用于多片级联，与普通的全嵌套工作方式的不同点在于：

1.当某从片的一个中断请求被 CPU 响应后，该从片的中断仍未被禁止，即**该从片中的高级中断仍可提出申请**（全**嵌套方式中这样的中断是被屏蔽的**，因为这种中断对从片而言后者是高级中断，可以嵌套，但对主片而言，由于它们来自于同一个从片，故中断优先级相同，而在全嵌套方式中，同级和低级中断是被禁止的）。2. 在某个中断源退出中断服务程序之前，CPU 要用软件检查它是否是这个从片中的唯一中断。

74. 简述 Pentium CPU INTR 的中断响应过程。



75. 试述 8259A 的初始化编程过程。

依次确认 ICW1-ICW4 命令字的内容，按顺序进行初始化

```
        CLI
        MOV     AL，13H         ；ICW1
        OUT     8CH，AL
        MOV     AL，77H         ；ICW2
        OUT     8DH，AL
        MOV     AL，01H         ；ICW4
        OUT     8DH，AL
        MOV     AL，7FH         ；OCW1
        OUT     8DH，AL
        STI
```

76. 中断响应有哪些条件？为什么 CPU 响应中断后立即关中断？

条件：1.一条指令执行结束后 2.有中断请求 3.开中断

防止本次中断服务结束前同级的其他中断源产生另一次中断进行干扰

77. 中断响应周期主要完成哪些工作？这些工作是由硬件完成还是软件完成？

第一个总线周期给出应答，表示中断被响应，第二个总线周期去查找对应的中断类型码（指 CPU 响应中断）

由硬件完成：关中断→保护现场（CPU 正在处理的现场，如标志）→ 保护断点→形成中断服 务程序入口地址→转入中断服务程序。

78. 什么是外部中断？什么是内部中断？内部中断有什么特点？

外部中断 INTR，NMI（由外部事件引起）、内部中断：INT n（由软件执行引发）

内部中断的特点：

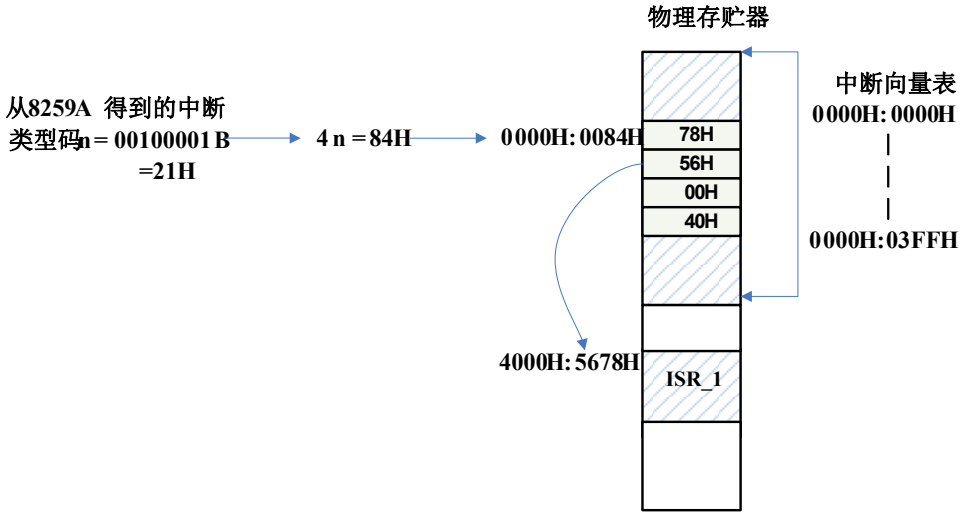
1. 内部中断由一条指令 INT n 产生中断类型码或者由指令规定，或者是预定的。
2. 不执行 INTA 总线周期，CPU 也不发出响应信号 INTA。
3. 除单步中断外，内部中断无法用软件禁止。
4. 除单步中断外，内部中断比外部中断具有更低的优先级别。

79. 试简要说明中断服务程序的一般组成结构。

保护现场、中断处理、恢复现场、开中断、返回断点

80. 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）。画图说明从中断源，到中断向量表，再到中断服务程序的对应关系。

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。



81. 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。

82. 8259A 的中断结束是指什么？简要说明 8259A 的 3 种中断结束方式的特点。

使 ISRx 位清 0 的动作就是 8259A 的中断结束处理。

包括：自动中断结束、一般中断结束命令和特殊中断结束命令。

自动中断结束：8259A 在接收到第二个 $\overline{INTA}$ 响应信号的后沿自动清除 ISRx。

一般中断结束命令用在**全嵌套**情况下。当 CPU 用输出指令往 8259A 发出一般中断结束命令时，8259A 就会把当前中断服务寄存器中的最高的非零 ISR 位复位。

特殊中断结束命令用在**非全嵌套**方式下，特殊中断结束命令中带有用于指定 ISR 中要复位的三位（即 L2～L0）编码信息，所以特殊中断结束命令**可以作为任何优先级管理方式的中断结束命令**。

83. 从方式名称、启动方法、计数值有效期限角度，说明 8253 的方式 1 与方式 2 的工作特点。

方式 1 是可重**触发单稳态**方式，由硬件启动，**计数值多次有效**，计数过程中输出保持为低，知道计数到 0，OUT 向高电平跳变。

方式 2 是**脉冲发生器**，有硬件和软件两种启动方式，计数值重复有效，OUT 可以产生一个连续的波形。开始计数后当减到 1 时输出将变低一个时钟的宽度，然后恢复为高电平又重新计数。

84. 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

85. 用 8253A 产生 5ms 的周期信号，做为 8259A 中断请求输入，CLK=1MHz，分析 8253A 计数器需采用哪种工作方式，计数初值是多少。

方式 3 计数初值 1000

86. 说明 8253 的方式 2 与方式 3 的工作特点。

方式 2 为脉冲频率发生器方式，软件或硬件启动，计数值重复有效，计数值为 1 时 OUT 输出一个 CLK 周期的负脉冲，之后回到正脉冲，正脉冲持续时间为计数初值*CLK 周期

方式 3 为方波发生器方式，软件或硬件启动，计数值重复有效，计数值奇偶有差别：偶数时，每个 CLK 周期减 2，到零后 OUT 端输出反转，重新装入计数值，继续计数；奇数时，第一个 CLK 先减一，变成偶数，之后每周期减 2，到零时反转，装入最近的不大于计数值的偶数，继续重新计数，

87. 说明 8253 的方式 1 与方式 5 的工作特点。

方式 1 为计数结束中断方式，软件启动，计数值一次有效，方式控制字输入时 OUT 变高，等待计数值输入，输入后变低，计数值为 0 时升高

方式 5 为硬件触发选通方式，硬件启动，计数值多次有效，写入控制字后 OUT 变高，之后写入计数值，等 GATE 端输入触发脉冲时，进行计数

88. 简述 8253 的主要特点。

3 个独立的 16 位计数器，能够进行 **3 个 16 位的独立计数**。

每一个计数器具有**六种工作方式**。

能进行**二进制/十进制计数（减法计数）**。所谓十进制计数，是指 BCD 码计数，每个计数器可表示 4 位十进制数的 BCD 码，每来一个计数脉冲，**按照十进制数减 1 规律进行计数**。

计数频率为 **0～2MHz**。

可作**计数器或定时器**。

89. 8253 有几个通道？各采用几种工作方式？简述这些工作方式的特点。

3 个通道，每个通道 6 种工作方式

90. 8253 有几种读操作方式？简述之。

直接读：直接读计数器

锁存读：从锁存器中读出（读 16 位时一般用锁存读）

91. 在 8253 计数器的六种工作方式中，方式 2 和方式 3 各输出何种波形，它们有何特点。

方式 2 输出脉冲波，持续周期数和取决于计数值，负脉冲一个周期

方式 3 输出方波，偶数情况下正负脉冲持续时间一致，奇数情况下负脉冲少一个周期

92. 某系统中有一片 8253，其计数器 0 至控制口地址依次为 40H-43H，请按如下要求编程：

（1）通道 0：方式 3，CLK0＝2MHz，要求在 OUT0 输出 1KHz 方波。

（2）通道 1：方式 2，CLK1＝1MHz，要求 OUT1 输出 1KHz 脉冲波。

（3）通道 2：方式 4，CLK2＝OUT1，计数值为 1000，计数到 0 时输出一个控制脉冲。

通道 0 计数值为 2000，方式 3；通道 1 计数值为 1000，方式 2；通道 0 计数值 1000

93. 某系统采用一片 8253 产生周期为 2ms、个数为 10 的脉冲序列，已知有一个时钟源，频率为 2MHz。要求：

（1）画出硬件接口电路图，并确定 8253 的端口地址。

（2）编写相应程序。

通道 0 计数值 2000，方式 3；通道 1 计数值 10，方式 1

94. 画出 8255 工作于方式 1 输入的波形，并说明其工作过程。

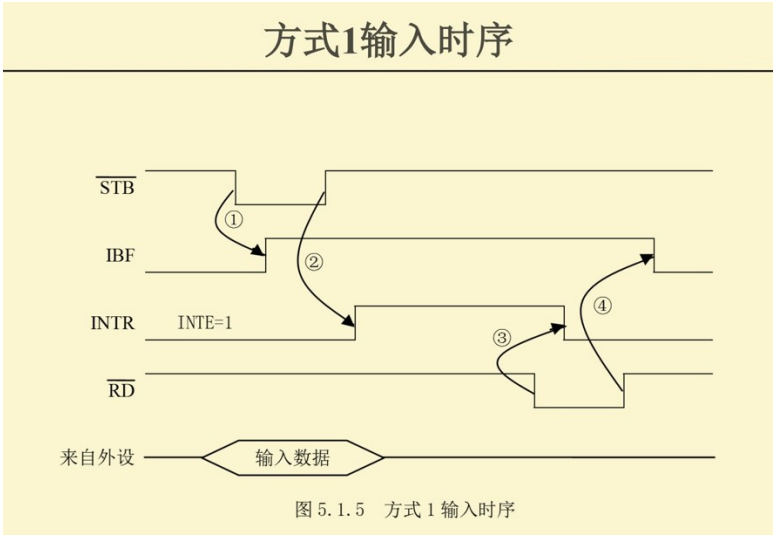


图 5.1.5 方式 1 输入时序

外设向 8255 输入数据：外设将一个 8 位数送 A/B 端口，通过一个选通信号输入数据，输入同时 IBF 变高，表示有一个新数据，如果中断允许，INTR 变高。CPU 了解到数据就绪后，可以读数据，选通数据端口，#RD 下降，读出后 INTR 无效，同时 IBF（输入缓冲器满）信号无效

95. 画出 8255 工作于方式 1 输出的波形，并说明其工作过程。

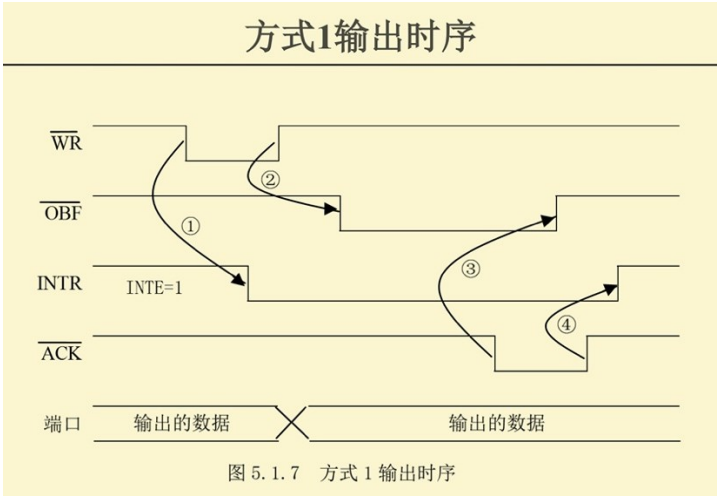


图 5.1.7 方式 1 输出时序

CPU 将一个新数据写入 A/B 口的数据输出寄存器时，#WR 有效，之后#OBF 有效，如果中断允许，INTR 变低；数据就绪

外部设备取出数据时，反馈一个应答信号#ACK，之后#OBF（输出缓冲器满）无效。中断请求变为有效，如果是中断方式，此时中断通知 CPU 数据已经取走，如果是查询方式，则通过查询 INTR 有效，通知 CPU 数据已经取走。

96. 并行接口芯片 8255A 的 A 口-控制口的端口地址依次为 60H-63H。编一段程序使从 PC5 输出一个负脉冲。另外，若脉冲宽度不够，应如何解决。

不会

97. 串行通信中为什么要用 Modem？Modem 在接收和发送中的作用是什么？

使用 Modem（调制解调器）进行调制解调

调制：用数字信号控制载波的一个参数的变化，就可以实现数字信号变换为频带信号，这种变换就是调制

解调：已调信号经过信道传输到接收端，在接收端通过反变换，将已调信号恢复成数字信号，这一变换过程称为解调

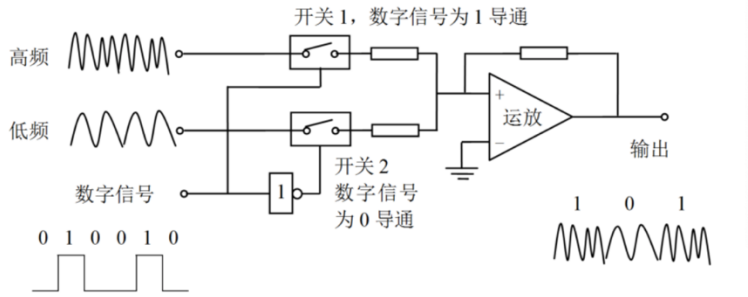
98. 简述 FSK 的调制原理。

FSK：频移键控法

原理：两个不同频率的模拟信号分别由电子开关控制，在运算放大器的输入端相加，而电子开关需由传输的数字信号来控制.

信号为 1 时，控制开关 1 导通，送出一串频率较高的模拟信号

信号为 0 时，控制开关 2 导通，送出一串频率较低的模拟信号



99. 当串行通信的波特率是 2400 波特时，数据位时间周期是多少？

比特率：单位时间（每秒）内通信系统所传送的信息量，计作 Rb，单位为比特/秒（即每秒传输的二进制位数）

如果每一位传送速度一样，1/Rb 就是每位传送的时间

码元传输速率（传码率）：单位时间（每秒）内通信系统所传送的码元数目，记做 RB，单位为波特

每个码元占有的时间 TB 叫做码长 RB=1/TB

RBM 表示 M 进制的码元

Rb=RB.Log_2(M)

100. 简述异步通信和同步通信的主要区别。

101. 简述 8251A 初始化的一般步骤。

102. 串行异步通信字符格式中的停止位和空闲位有什么不同？

103. 在串行异步通信中，为什么接收时时钟频率一般是波特率的 16 倍频？

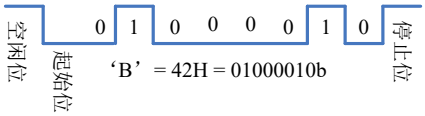
104. 8251A 可检测到几种接收数据错误，详细说明。

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。

105. 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



106. A/D 和 D/A 转换器在微机应用中起什么作用？

107. 为什么 DAC 0832 特别适用于多个模拟量同时输出的场合？其工作过程如何？

108. 简述采样定理。

109. 用 CD4051 设计一个 32 路模拟开关，画出电路连接图。

110. 采样/保持器有什么作用？

111. 简述 A/D 转换的基本过程。

112. ADC0809 的输入电压=5V、参考电压=5V，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

113. A/D 转换接口中采样保持电路的作用是什么？省略采样保持电路的前提条件是什么？

A/D 转换接口中采样保持电路的作用是：能把一个时间边续的信号变换为时间离散的信号，并将采样信号保持一段时间。

114. 当外接模拟信号的变化速度相对于 A/D 转换速度来说足够慢，在转换期间可视为直流信号的情况下，可以省略采样保持电路。

115. PC/XT 控制 ADC0809 构成一个压力参数采集系统，要求以查询方式采集 400 个压力值，存入 ADBUF 开始的存储单元，试设计硬件接口电路，并编写采集程序。

116. 采用 8255A、ADC0809、DAC0832、8253 和 8088CPU 构成一个数据采集控制系统，要求画出接口电路图，确定各芯片端口地址，说明系统工作原理。

117. DAC0832 的参考电压=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？

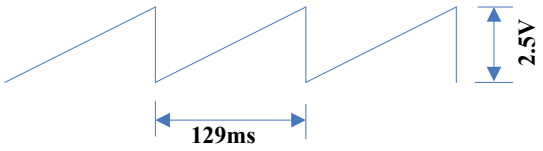
3/256 = V/128， V = 128*3/256 = 1.5V

118. DAC 单极性输出，参考电压=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
```

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



119. DAC0832 采用单极输出，参考电压=5V，输出电压最大值+5V。若输出 2V 的电压，需要写入 DAC0832 的数字量是多少？若输出 250Hz 方波，写入 DAC0832 的两个数字量之间的软件延时时间是多少？

120. 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。

121. LED 显示器的工作原理是什么？何谓共阳极？何谓共阴极？

122. LED 显示器接口有哪几种方式？

123. 设计一个通过 8255A 芯片控制的一个两位 LED 的动态显示接口电路。

124. 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。

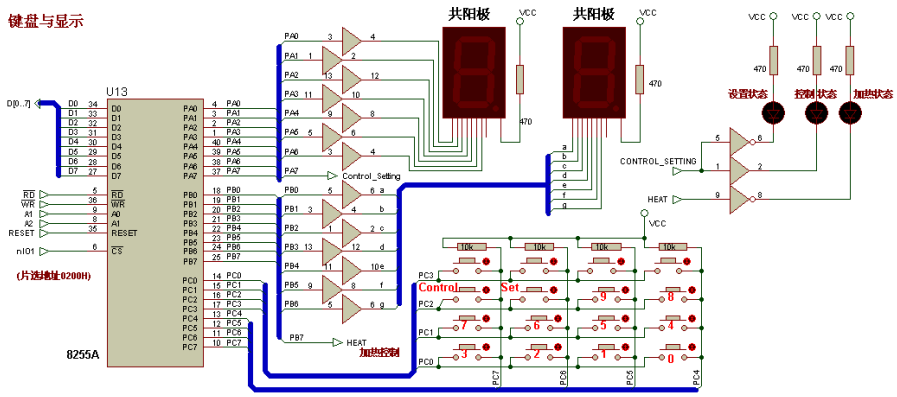
系统有两位七段数码管显示温度值（0~99℃）。在设置状态时，系统显示设定温度；在控制状态时，系统显示当前检测温度。系统通过 4x4 键盘输入设定温度值和启动控制，键盘有 0~9 键、Setting 键和 Control 键共 12 个键可用。

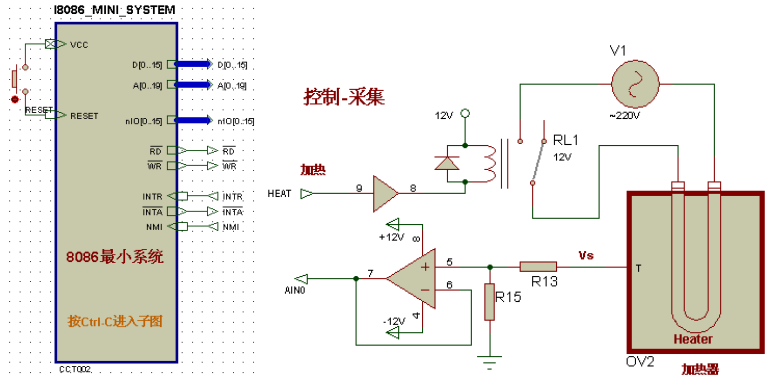
（1）画出系统的硬件连接原理图，并标明分配给各元器件的端口地址。

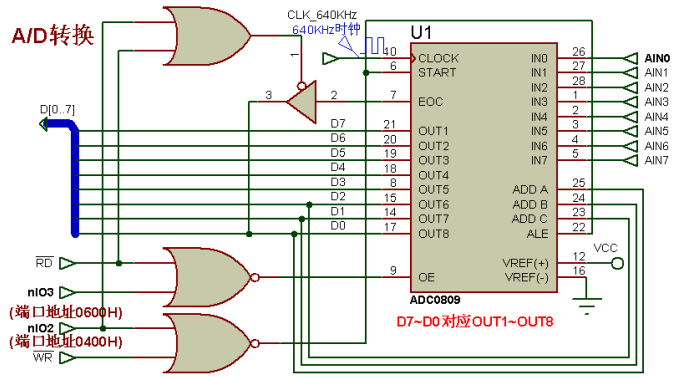
（2）写出“0”对应的七段数码管译码值；编写 8255、8253 初始化程序；

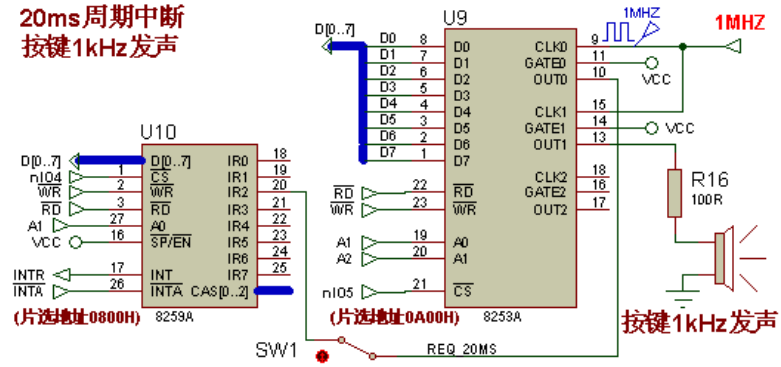
（3）编写 AD 转换子程序（adc）、显示子程序（display）、按键识别子程序（key）和主程序（main）。

（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
      incal
     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
```