

8086CPU 与寻址方式

1. 有两个 16 位字 1EE5H 和 2A3CH 分别存放在微机存储器 000B0H 和 000B3H 单元中，请用图示表示出它们在存储器里的存储情况。

答：

E5H	000B0H
1EH	000B1H
3CH	000B3H
2AH	000B4H

2. 微机存储器里存放的信息如右图示，试读出 30022H 和 30024H 字节单元的内容，以及 30021H 和 30022H 字单元中的内容。

答：

30022H 字节单元中的内容为 ABH;
 30024H 字节单元中的内容为 EFH;
 30021H 字单元中的内容为 AB34H;
 30022H 字单元中的内容为 CDABH。

12H	30020H
34H	30021H
ABH	30022H
CDH	30023H
EFH	30024H

3. 段地址和偏移地址为 3017:000A 的存储单元的物理地址是什么？如果段地址和偏移地址是 3015:002A 和 3010:007A 呢？

答：

物理地址为段地址左移 4 位再加上偏移地址值；
 所以段地址和偏移地址为 3017:000A 的存储单元的物理地址为 3017AH;
 段地址和偏移地址是 3015:002A 和 3010:007A 的存储单元的物理地址也都为 3017AH。

4. 如果在一个程序开始执行之前 (CS) = 0A7F0H (如果十六进制数的最高位为字母，则应在其前加 1 个 0)，(IP) = 2B40H，试问该程序的第 1 个字的物理地址是多少？

答：

第一个字的物理地址 = 0A7F00H + 2B40H = 0AAA40H

5. 1MB 最多能分成多少逻辑段？请将下列逻辑地址转换成物理地址？

(1) 0FFFFH: 0 (2) 40H: 17H (3) 2000H: 2400H

答：

每 16B 就可为一个逻辑段，因此 1MB 最多能分成 $1\text{MB}/16\text{B} = 2^{16} = 65536$ 个逻辑段

(1) 物理地址 = 0FFFF0H + 0H = 0FFFF0H;

- (2) 物理地址 = $400H + 17H = 00417H$;
(3) 物理地址 = $20000H + 2400H = 22400H$.

6. 现有 $(DS)=2000H$, $(BX)=0100H$, $(SI)=0002H$, $(20100)=12H$, $(20101)=34H$, $(20102)=56H$, $(20103)=78H$, $(21200)=2AH$, $(21201)=4CH$, $(21202)=0B7H$, $(21203)=65H$, 试说明下面各条指令执行后 AX 寄存器的内容。

- (1) MOV AX, 1200H
- (2) MOV AX, BX
- (3) MOV AX, [1200H]
- (4) MOV AX, [BX]
- (5) MOV AX, 1100[BX]
- (6) MOV AX, [BX][SI]
- (7) MOV AX, 1100[BX][SI]

答:

- (1) 立即寻址, $(AX) = 1200H$;
- (2) 寄存器寻址, $(AX) = 0100H$;
- (3) 直接寻址, $PA=20000H+1200H=21200H$, $(AX) = 4C2AH$;
- (4) 寄存器间接寻址, $PA=20000H+0100H=20100H$, $(AX) = 3412H$;
- (5) 寄存器相对寻址, $PA=20000H+1100H+0100H=21200H$, $(AX) = 4C2AH$;
- (6) 基址变址寻址, $PA=20000H+0100H+0002H=20102H$, $(AX) = 7856H$;
- (7) 相对基址变址寻址, $PA=20000H+1100H+0100H+0002H=21202H$, $(AX) = 65B7H$.

7. 假设 $(DS)=2000H$, $(ES)=2100H$, $(SS)=1500H$, $(SI)=00A0H$, $(BX)=0100H$, $(BP)=0010H$, 数据段中的变量名 VAL 的偏移地址值为 0050H, 试指出下列源操作数的寻址方式是什么? 其物理地址值是什么?

- (1) MOV AX, 0BAH
- (2) MOV AX, [100H]
- (3) MOV AX, [BX]
- (4) MOV AX, [BP]
- (5) MOV AX, [BX+10]
- (6) MOV AX, [BX][SI]
- (7) MOV AX, BX
- (8) MOV AX, VAL
- (9) MOV AX, ES: [BX]
- (10) MOV AX, [SI]
- (11) MOV AX, VAL[BX]
- (12) MOV AX, VAL[BX][SI]

答:

- (1) 立即寻址, 无物理地址;
- (2) 直接寻址, $PA=16 \times (DS)+100H=20000H+100H=20100H$;
- (3) 寄存器间接寻址, $PA=16 \times (DS)+(BX)=20000H+0100H=20100H$;
- (4) 寄存器间接寻址, $PA=16 \times (SS)+(BP)=15000H+0010H=15010H$;
- (5) 寄存器相对寻址,
 $PA=16 \times (DS)+(BX)+10H=20000H+0100H+10H=20110H$;
- (6) 基址变址寻址,
 $PA=16 \times (DS)+(BX)+SI=20000H+0100H+00A0H=201A0H$;
- (7) 寄存器寻址, 无物理地址;
- (8) 直接寻址, $PA=16 \times (DS)+VAL=20000H+0050H=20050H$;

- (9) 寄存器间接寻址, $PA=16 \times (ES) + (BX) = 21000H + 0100H = 21100H$;
- (10) 寄存器间接寻址,
 $PA=16 \times (DS) + (SI) = 20000H + 00A0H + 200A0H$;
- (11) 寄存器相对寻址,
 $PA=16 \times (DS) + VAL + (BX) = 20000H + 0050H + 0100H = 20150H$;
- (12) 相对基址变址寻址,
 $PA=16 \times (DS) + VAL + (BX) + (SI) = 20000H + 0050H + 0100H + 00A0H = 201F0H$.

8086 的伪指令

8. 画图说明下列语句所分配的存储空间及初始化的数据

(1) AA DB 'BYTE', 12, -12H,

3 DUP (0, ?, 2 DUP (1, 2), ?)

(2) BB DW 2 DUP (0, 1, 2), ?, -5, 'BY', 'TE', 256H

答:

(1)

AA-->	42H
	59H
	54H
	45H
	0CH
	EEH
	00H
	—
	01H
	02H
	01H
	02H
	—
	00H
	—
	01H
	02H
	01H
	02H
	—
	00H
	—
	01H
	02H
	01H
	02H
	—

(2)

BB->	00H
	00H
	01H
	00H
	02H
	00H
	00H
	00H
	00H
	01H
	00H
	02H
	00H
	—
	—
	FBH
	FFH
	59H
	42H
	45H
	54H
	56H
	02H

9. 假设程序中的数据定义如下:

PARTNO DW ?

PNAME DB 16 DUP (?)

COUNT DD ?

PLENTH EQU \$-PARTNO

问 PLENTH 的值是多少? 它表示什么意义?

答:

PLENTH 的值为 $2+16+4 = 22$ ，它表示当前地址的大小- PARTNO 的地址，即 PLENTH 与 PARTNO 之间有 22 个字节空间。

10. 请设置一个数据段 DATASG，段中变量及数据如下：

- (1) FLD1 为字符串变量：‘computer’
- (2) FLD2 为十进制字节变量：32
- (3) FLD3 为十六进制字节变量：20H
- (4) FLD4 为二进制字节变量：01011001
- (5) FLD5 为 10 个 0 的字节变量
- (6) FLD6 为数字的 ASCII 字符字节变量 32654
- (7) FLD7 为包括 5 个十进制数的字变量：5，6，7，8，9
- (8) FLD8 为 100 个字变量

答：

DATASG SEGMENT

```
FLD1  DB  'computer'
FLD2  DB  32
FLD3  DB  20H
FLD4  DB  01011001B
FLD5  DB  10 DUP(0)
FLD6  DB  '32654'
FLD7  DW  5, 6, 7, 8, 9
FLD8  DW  100 DUP(?)
```

DATASG ENDS

8086 的指令系统

11. 指出下列错误的指令及错误的原因

- (1) MOV AH, BX
- (2) MOV [BX], [SI]
- (3) MOV AX, [DI][SI]
- (4) MOV MYDAT[BX][SI], ES:AX
- (5) MOV BYTE PTR[BX], 1000
- (6) MOV BX, OFFSET MYDAT[SI]
- (7) MOV CS, AX

答：

上述指令全部错误，原因如下：

- (1) 源操作数 BX 为 16 位，目的操作数 AH 为 8 位，两者长度不相同
- (2) 汇编语言中不允许在两个内存单元之间传数据
- (3) 不能连用两个变址，基址变址寻址应为一个基址一个变址
- (4) 汇编语言中不允许在两个内存单元之间传数据
- (5) 目的操作数和源操作数必须有一个在寄存器中
- (6) OFFSET 不能与复杂操作数相连
- (7) 汇编语言中目的操作数不能为 CS

12. 指出下列哪些指令是非法的，设 OP1 和 OP2 均为字变量

- (1) CMP 15, BX
- (2) CMP OP1, 25

(3) CMP OP1, OP2

(4) CMP AX, OP1

答:

(1) 非法, 目的操作数不允许为立即数

(2) 非法, 源操作数和目的操作数类型不匹配

(3) 非法, 目的操作数和源操作数不能同时为存储器寻址

(4) 合法

13. 假设下列指令中的所有标示符均为类型属性为字的变量, 请指出下列指令中哪些是非法的? 它们的错误是什么?

(1) MOV BP, AL

(2) MOV WORD_OP[BX+4*3][DI], SP

(3) MOV WORD_OP1, WORD_OP2

(4) MOV AX, WORD_OP1[DX]

(5) MOV SAVE_WORD, DS

(6) MOV [BX][SI], 2

答:

(1) 非法, 操作数类型不匹配

(2) 合法

(3) 非法, 目的操作数和源操作数不能同时为存储器寻址

(4) 非法, DX 不能用于寄存器间接寻址

(5) 合法

(6) 非法, 目的操作数[BX][SI]类型不确定

14. 假设程序中的数据定义如下:

LNAME DB 30 DUP(?)

ADDRESS DB 30 DUP(?)

CITY DB 15 DUP(?)

CODE_LIST DB 1, 7, 8, 3, 2

(1) 用一条 MOV 指令将 LNAME 的偏移地址放入 AX。

(2) 用一条指令将 CODE_LIST 的头两个字节的内容放到 SI。

(3) 写一条伪操作使 CODE_LENGTH 的值等于 CODE_LIST 域的实际长度

答:

(1) MOV AX, OFFSET LNAME

(2) MOV SI, WORD PTR CODE_LIST

(3) CODE_LENGTH EQU \$-CODE_LIST

15. 试说明下述指令中哪些需要加上 PTR 伪操作:

BVAL DB 10H, 20H

WVAL DW 1000H

(1) MOV AL, BVAL

(2) MOV DL, [BX]

(3) SUB [BX], 2

(4) MOV CL, WVAL

(5) ADD AL, BVAL+1

答:

(1) 不需要加上 PTR 伪操作

(2) 不需要加上 PTR 伪操作

- (3) 需要加上 PTR 伪操作, 可以变为 SUB BYTE PTR [BX], 2
- (4) 需要加上 PTR 伪操作, 可以变为 MOV CL, BYTE PTR WVAL
- (5) 不需要加上 PTR 伪操作

16. 若 TABLE 为数据段 0032H 单元的符号名, 其中存放的内容为 1234H, 试问下列两条指令有什么区别? 执行完指令后, AX 寄存器中的内容是什么?

```
MOV AX, TABLE
LEA AX, TABLE
```

答:

MOV AX, TABLE 是将 TABLE 单元中的内容取出来送到 AX 中, 所以 (AX)=1234H;
而 LEA AX, TABLE 是将 TABLE 的偏移地址取出并放到 AX 中, 所以 (AX)=0032H.

17. 已知程序段如下:

```
MOV AX, 1234H
MOV CL, 4
ROL AX, CL
DEC AX
MOV CX, 4
MUL CX
```

试问:

- (1) 每条指令执行后, AX 的内容是什么?
- (2) 每条指令执行后, CF、SF 和 ZF 的值是什么?
- (3) 程序执行完后, AX 和 CX 的内容是什么?

答:

(1) (2) 如下:

```
MOV AX, 1234H ; (AX)=1234H, CF=0、SF=0、ZF=0
MOV CL, 4      ; (AX)=1234H, CF=0、SF=0、ZF=0
ROL AX, CL     ; (AX)=2341H, CF=1、SF=0、ZF=0
DEC AX         ; (AX)=2340H, CF=1、SF=0、ZF=0
MOV CX, 4      ; (AX)=2340H, CF=1、SF=0、ZF=0
MUL CX         ; (AX)=8D00H, CF=0、SF=1、ZF=0
```

(3) 程序执行完 (AX)=8D00H, (CX)=0004H

分支程序设计

1. 内存有一个字节变量 VAL 中存放着小写字符 'a', 请将该字符转换为大写字符并在屏幕中显示出来。

代码如下:

```
DSEG SEGMENT
    VAL DB 'a'
DSEG ENDS

CSEG SEGMENT
    ASSUME CS:CSEG, DS:DSEG
```

```

START:
    MOV AX, DSEG
    MOV DS, AX

    MOV DL, VAL
    SUB DL, 20H ; ASCII码-20H转换成大写字母'A'
    MOV AH, 2H
    INT 21H ; 在屏幕打印出来

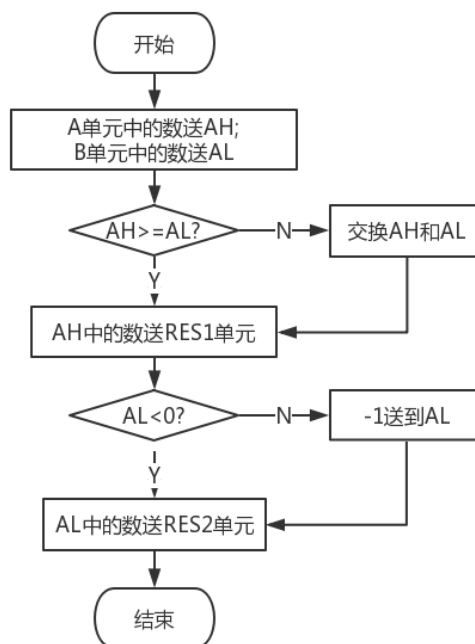
    MOV AH, 4CH ; 结束程序
    INT 21H

CSEG ENDS
    END START

```

2. 设存储单元 A 和 B 各有一带符号字节数，比较大小和正负，要求将较大数送 RES1 单元；如果有负数，将一个负数送 RES2，否则 RES2 送-1。

我的思路主要是先把 A 和 B 单元中的数分别存在 AH 和 AL 中，然后比较大小，若 AH 小于 AL，则交换，即使 AH 中始终放较大的数，AL 中始终放较小的数，然后 AH 送 RES1，这样之后判断有没有负数，只需要比较 AL 和 0 即可，若 $AL < 0$ ，则有负数，将 AL 送到 RES2 即可。程序设计流程图如下：



代码如下：

```

DSEG SEGMENT
    A DB 2H
    B DB 0FAH
    RES1 DB ?
    RES2 DB ?
DSEG ENDS

CSEG SEGMENT
    ASSUME CS:CSEG, DS:DSEG

```

```

START: MOV AX,DSEG
      MOV DS,AX

      MOV AH,A
      MOV AL,B
      CMP AH,AL
      JNL NEXT1 ;A>=B跳转
      XCHG AH,AL ;A<B则交换AH、AL, 使AH中放较大数, AL中放较小数

NEXT1: MOV RES1,AH ;较大数送RES1
      CMP AL,0
      JL NEXT2 ;有负数跳转
      MOV AL,0FFH ;没有负数将-1送到AL

NEXT2: MOV RES2,AL ;负数送到RES2
      MOV AH,4CH
      INT 21H

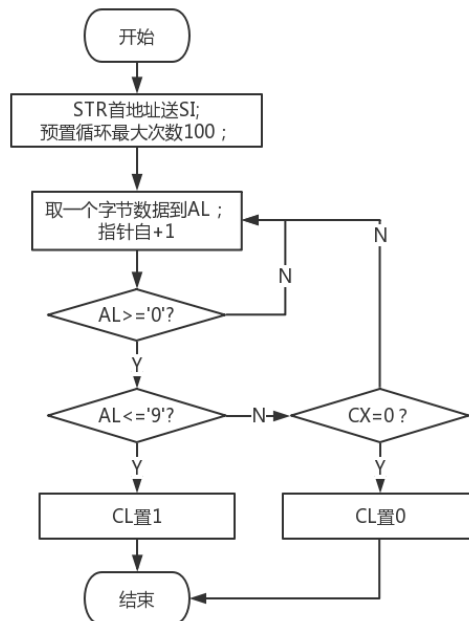
CSEG ENDS
      END START

```

循环程序设计

1. 在 STR 到 STR+99 单元中存放着一个字符串, 试编写程序测试该字符串中是否有数字, 若有将 CL 置 1, 否则 CL 置 0。

程序设计的流程图如下:



代码如下:

```

DSEG SEGMENT
    STR DB 'sda1dsa'
DSEG ENDS

CSEG SEGMENT

```



```

ASSUME CS:CSEG,DS:DSEG

START: MOV AX,DSEG
      MOV DS,AX
      LEA SI,STR
      MOV CX,100

LOP:   MOV AL,[SI] ;取一个字节
      INC SI ;指针+1
      CMP AL,30H ;判断是否>=0
      JNB NEXT1 ;>=0, 跳转到NEXT1
      LOOP LOP ;跳转回LOP

NEXT1: CMP AL,39H ;判断是否<=9
      JBE NEXT2 ;<=9跳转
      LOOP LOP

      MOV CL,0 ;循环结束还没有数字则CL置0
      JMP EXIT

NEXT2: MOV CL,1 ;找到数字CL置1

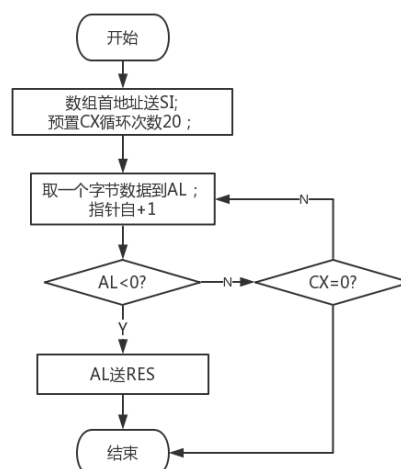
EXIT:  MOV AH,4CH
      INT 21H

CSEG ENDS
      END START

```

2. 在字节数组中找出第一个负数，并将该负数存入 RES 单元中；假设该数组中包含 20 个带符号数，且至少有 1 个负数。

程序设计流程图如下：



代码如下：

```

DSEG SEGMENT
    BUF DB 2,-2,3,10,43,22,-12,-7,8,4
        DB 55,34,12,-32,67,9,20,-4,18,20
    RES DB ?

```

```

DSEG ENDS

CSEG SEGMENT
    ASSUME CS:CSEG, DS:DSEG

START: MOV AX, DSEG
        MOV DS, AX
        MOV SI, OFFSET BUF
        MOV CX, 20    ;循环初值

LOP:    MOV AL, [SI]    ;取值
        INC SI    ;指针+1
        CMP AL, 0    ;判断是否<0
        JL NEXT    ;<0, 跳转到NEXT
        LOOP LOP    ;跳转回LOP
        JMP EXIT

NEXT:    MOV RES, AL    ;送到RES

EXIT:    MOV AH, 4CH
        INT 21H

CSEG ENDS
        END START

```

子程序设计

3. 试编写一个汇编程序，能对键盘输入的小写字母用大写字母显示出来（要求采用子程序格式，即采用子程序完成将小写字母转化成大写字母）。

答：

我先在主程序中读入一个小写字母，然后调用子程序，子程序中通过将输入的小写字母的 ASCII 码-20H 将小写字母转化为大写字母，最后再在主程序中显示转换后的大写字母。

代码如下：

```

DSEG SEGMENT
DSEG ENDS
CSEG SEGMENT
    ASSUME CS:CSEG, DS:DSEG
MAIN PROC FAR    ;主程序
START: MOV AX, DSEG
        MOV DS, AX

        MOV AH, 1
        INT 21H    ;读入字符
        CALL UPPER    ;调用子程序

```

```
PRINT: MOV DL, AL ;显示转换后的字符
        MOV AH, 2
        INT 21H

EXIT:   MOV AH, 4CH ;退出
        INT 21H

MAIN    ENDP

UPPER PROC NEAR ;小写字母转化成大写字母子程序
        SUB AL, 20H; ASCII码-20H转化成大写字母
        RET
UPPER ENDP

CSEG ENDS
        END START
```

4. 有 2 个数组：

ary1 db 12, -35, 0, 126, -90, -5, 68, 120, 1, -19

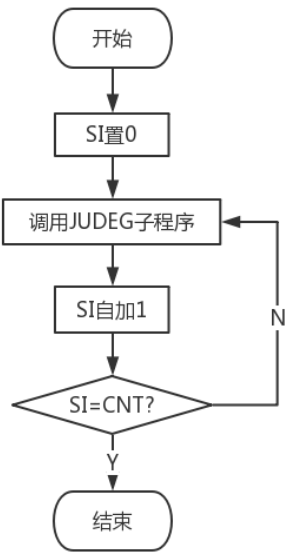
ary2 db 24, 25, 0, -38, -89, 99, 68, 100, 2, -20

比较两个数组的对应位，将大的数放在 ary1 数组中，小的数放在 ary2 中（要求采用子程序格式）。

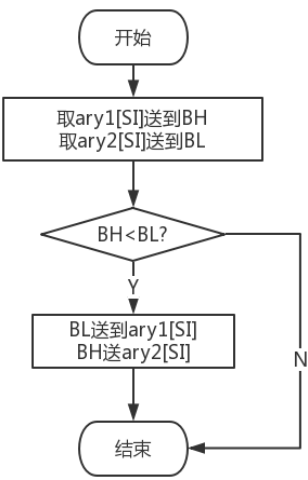
答：

我在主程序中设置了一个循环，循环中每次调用子程序来比较当前 ary1 和 ary2 中对应位数的大小，如果 ary1<ary2，就交换两个数，即把大的数放在 ary1 数组中，小的数放在 ary2 中。

程序设计流程图如下：



主程序流程图



子程序流程图

代码如下：

```
DSEG SEGMENT
    ary1 DB 12,-35,0,126,-90,-5,68,120,1,-19
    CNT EQU $-ary1
    ary2 DB 24,25,0,-38,-89,99,68,100,2,-20
DSEG ENDS

CSEG SEGMENT
    ASSUME CS:CSEG,DS:DSEG
MAIN PROC FAR ;主程序
START: MOV AX,DSEG
        MOV DS,AX
        MOV SI,0 ;设置循环与数组指针初值

LOP:    CALL JUDGE
        INC SI
        CMP SI,CNT
        JNZ LOP
```

```

EXIT:  MOV AH,4CH
        INT 21H
MAIN   ENDP

JUDGE  PROC NEAR ;子程序
        MOV BH,ary1[SI]
        MOV BL,ary2[SI]
        CMP BH,BL
        JL NEXT ;BH<BL,跳转NEXT1
        RET

NEXT:   MOV ary1[SI],BL ;大的放ary1
        MOV ary2[SI],BH ;小的放ary2
        RET
JUDGE  ENDP

CSEG  ENDS
        END START

```

定时计数技术

假设 82C54A 的端口地址为 304H-307H，试按下列要求，分别编写 3 个计数通道的初始化及计数初值装入程序段(指令序列)。

计数器 0：二进制计数，工作在 0 方式，计数初值为 1234H

计数器 1：BCD 码计数，工作在 2 方式，计数初值为 100H

计数器 2：二进制计数，工作在 4 方式，计数初值在 55H。

答：

● 计数器 0：

方式命令字为 00110000，代码如下：

```

MOV DX,307H ;命令口
MOV AL,00110000B ;方式字
OUT DX,AL
MOV DX,304H ;计数器0
MOV AL,34H ;低8位计数初值
OUT DX,AL
MOV AL,12H ;高8位计数初值
OUT DX,AL

```

● 计数器 1：

方式命令字为 01110101，100H 的十进制为 256，代码如下：

```

MOV DX,307H ;命令口
MOV AL,01110101B ;方式字
OUT DX,AL
MOV DX,305H ;计数器1
MOV AL,56H ;低8位计数初值
OUT DX,AL
MOV AL,2H ;高8位计数初值
OUT DX,AL

```

● 计数器 2:

方式命令字为 10011000，代码如下：

```
MOV DX, 307H ;命令口
MOV AL, 10011000B ;方式字 只读/写低字节
OUT DX, AL
MOV DX, 306H ;计数器2
MOV AL, 55H ;低8位计数初值
OUT DX, AL
```

4. 19 要求产生 25kHz 的方波，则应向方波发生器写入的计数初值是多少？方波发生器的 GATE=1, CLK=1. 19318MHz。

答：

计数初值=1.19318MHz / 25kHz = 47 = 2FH

4. 23 采用计数通道 1，设计一个分频器。输入的时钟信号 CLK1=1000Hz，要求 OUT1 输出的高电平和低电平均是 20ms 的方波。GATE1=1，端口地址为 304H-307H。试编写出初始化程序段和计数初值装入程序段。

答：

方波的频率为 $1/(20\text{ms}+20\text{ms}) = 25\text{Hz}$

所以计数初值为 $1000\text{Hz} / 25\text{Hz} = 40 = 28\text{H}$

方式命令字为 01010110

代码如下：

```
MOV DX, 307H ;命令口
MOV AL, 01010110B ;方式字 只读/写低字节
OUT DX, AL
MOV DX, 305H ;计数器2
MOV AL, 28H ;低8位计数初值
OUT DX, AL
```

并行接口

如果要求将 82C55A 的 A 端口、B 端口和 C 端口设置为 0 方式，且 A 端口和 B 端口用于输入而 C 端口用于输入，那么应向命令寄存器写入什么方式的命令字？

答：

按照下图定义：

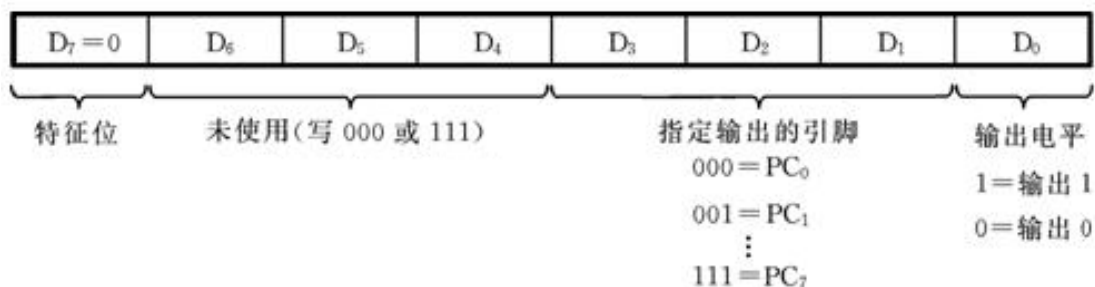
D ₇ =1	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
特征位	A 组方式		A 端口	PC ₄ ~PC ₇	B 组方式	B 端口	PC ₀ ~PC ₃
	00=0 方式		0=出	0=出	0=0 方式	0=出	0=出
	01=1 方式		1=入	1=入	1=1 方式	1=入	1=入
	10=2 方式						
	11=不用						

A、B、C 端口都为 0 方式，则 D₆D₅=00，D₂=0，且 A、B、C 三个端口都是输入，则 D₄=D₃=D₁=D₀=1

所以完整的方式命令字为：10011011

7. 16 如果把 03H 代码写入 82C55A 的命令寄存器，那么这个“按位置 1/清 0”命令将对 C 端口的哪一位进行操作？该位被置 1 还是清 0？

答：
按照下图定义：



命令字 03H 即 00000011B，所以对应 D₃D₂D₁=001，且 D₀=1，即将对 C 端口的 PC₁ 引脚置 1。

7.17 试分别编写产生从 C 端口的 PC₇ 引脚输出一个正脉冲和从 PC₃ 引脚输出一个负脉冲的程序段？

答：

- 从 C 端口的 PC₇ 引脚输出一个正脉冲

```

MOV  DX, 303H      ; 8255A 命令口
MOV  AL, 00001110B ; 置 PC7=0
OUT  DX, AL
NOP                      ; 维持低电平
NOP
MOV  AL, 00001111B ; 置 PC7=1
OUT  DX, AL
NOP                      ; 维持高电平
NOP
MOV  AL, 00001110B ; 置 PC7=0
OUT  DX, AL

```

- 从 C 端口的 PC₃ 引脚输出一个负脉冲

```

MOV  DX, 303H      ; 8255A 命令口
MOV  AL, 00000111B ; 置 PC3=1
OUT  DX, AL
NOP                      ; 维持高电平
NOP
MOV  AL, 00000110B ; 置 PC3=0
OUT  DX, AL
NOP                      ; 维持低电平
NOP
MOV  AL, 00000111B ; 置 PC3=1
OUT  DX, AL

```

7.20 为了允许 82C55A 的 1 方式下 A 组输出中断请求，应向命令寄存器写入何值？

答：

根据书上对 1 方式状态字格式的定义，如下图：

2. 状态字的格式

状态字的格式如图 7.16 所示。状态字有 8 位,分 A 和 B 两组,A 组的状态位占高 5 位,B 组的状态位占低 3 位,并且输入时与输出时的状态字不相同。

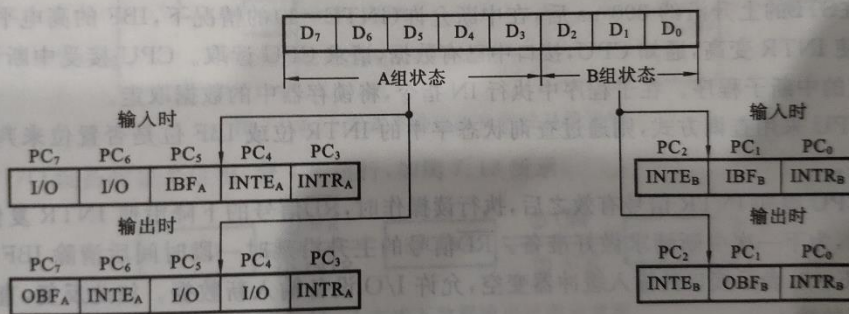


图 7.16 1 方式下状态字的格式

可以发现当允许 A 组输出中断请求时,影响的是 PC6 的 INTEA 这一状态位,置 1 则允许中断请求,置 0 则禁止中断请求。而本题要求允许中断请求,则命令字为 00001101,代码如下:

```
MOV DX,303H
MOV AL,10100000B ;1方式, A口输出
OUT DX,AL
MOV AL,00001101B ;置输出中断允许INTEA为1, 即置PC6=1
OUT DX,AL
```

7.21 为了允许 82C55A 的 1 方式下 A 组输入中断请求,应向命令寄存器写入何值?

答:

和上一题图一样,可以发现当允许 A 组输入中断请求时,影响的是 PC4 的 INTEA 这一状态位,置 1 则允许中断请求,置 0 则禁止中断请求。而本题要求允许中断请求,则命令字为 00001001,代码如下:

```
MOV DX,303H
MOV AL,10110000B ;1方式, A口输入
OUT DX,AL
MOV AL,00001001B ;置输入中断允许INTEA为1, 即置PC4=1
OUT DX,AL
```

中断技术

5.12 微机系统配置了两片(主片和从片)82C59A 中断控制器芯片,可以处理 15 级可屏蔽中断,试说明它们的中断号及中断优先级?

答:

中断号 08H~0FH 对应 IRQ0~IRQ7

中断号 70H~77H 对应 IRQ8~IRQ15

主片优先级从 IRQ0-IRQ7 依次降低;从片优先级从 IRQ8-IRQ15 依次降低

5.14 中断向量修改的目的是什么? 修改中断向量的方法与步骤?

答:

中断向量修改的目的:

中断向量修改是解决系统中中断资源共享的一种手段,也是用户利用系统中中断资源来开发可屏蔽中断服务程序的常用方法,具有实际意义。当用户要用自行开发的中断服务程序

去代替系统原有的中断服务程序时，就必须修改原有的中断向量，使其改为用户的中断服务程序的中断向量。若产生中断，并被响应，就可转到用户的服务程序来执行。

修改中断向量的方法：

MS-DOS 程序中，中断向量修改的方法是利用 DOS 功能调用 INT 21H 的 35H 号功能和 25H 号功能。

修改中断向量的步骤：

- 1) 调用 35H 号功能，从向量表中读取某一中断号的原中断向量，并保存在字变量中；
- 2) 调用 25H 号功能，将新中断向量写入中断向量表中原中断向量的位置，取代原中断向量；
- 3) 新中断服务程序完毕后，再用 25H 号功能将保存在字变量中的原中断向量写回去，恢复原中断向量。

5.17 如何利用微机系统的主片 82C59A 设计一个中断应用程序？

答：

首先明确用哪个引脚来接收中断请求，从而确定中断号，然后修改中断向量，开放中断屏蔽字，并打开中断，并在中断服务子程序中编写相应用户需要实现的功能。

AD 转换接口

9.5 A/D 转换器与 CPU 的接口电路设计时，需要给 A/D 转换器提供哪些基本信号线？

答：

- 模拟信号输入线，有单通道与多通道之分；
- 数字量输出线，线的数目决定了分辨率；
- 转换启动线（输入），每次启动只能转换一次数据；
- 转换结束线（输出），表示 ADC 作一次转换结束的状态。

9.18 如何设计一个采用查询方式的 A/D 转换器接口？

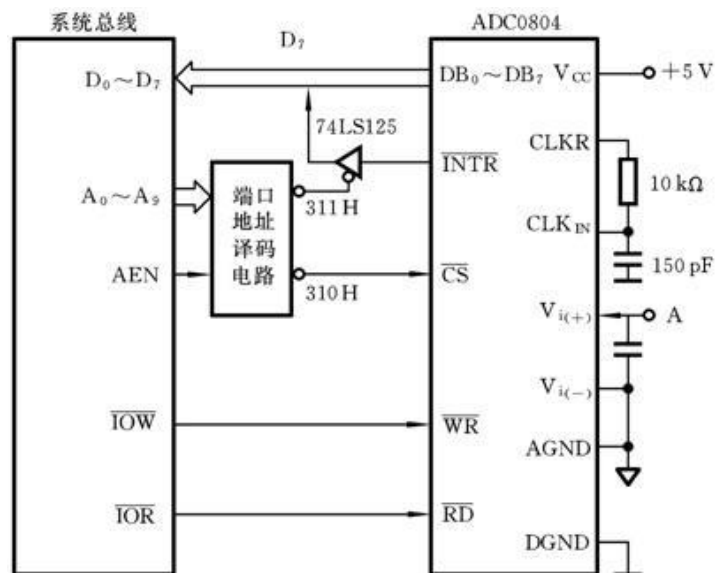
答：

1) 分析

我这里以 ADC0804 为例进行分析，ADC0804 是单个模拟量输入；ADC0804 的分辨率为 8 位，并具有三态输出锁存器，所以可以与系统数据总线直接相连；ADC0804 的启动方式为脉冲启动，当它的输入引脚 CS 和 WR 同时有效时，开始转换；数据传输方式为查询方式，所以需要将转换结束状态信号作为查询的对象。

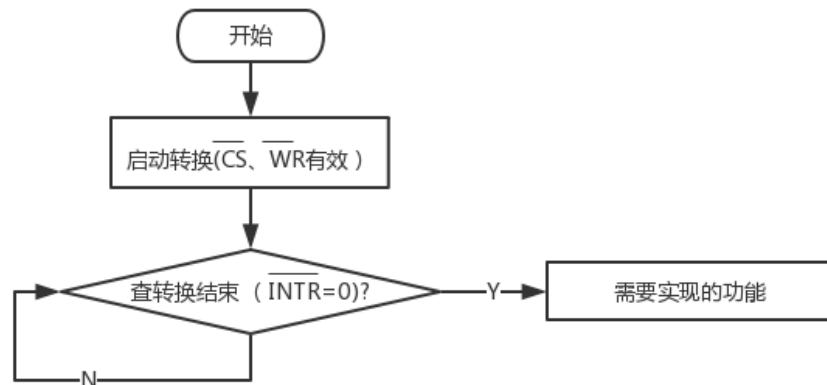
2) 硬件设计

根据上面分析，本接口电路只需要提供转换启动信号和提供读取转换结束状态信号的通路。而数据线不作处理，直接连接。为此，要设计端口地址译码电路，产生 CS，并由 CS 和 WR 共同组成启动信。，同时还要设置一个三态门，将转换结束信号 INTR 引到数据线的某一位上，以便 CPU 读取状态。接口电路如下图所示：



3) 软件设计

程序流程图如下：



主要程序如下：

```

DATA SEGMENT
    START_P EQU 310H    ; 转换启动端口
    STATE_P EQU 311H    ; 状态端口
    DATA_P EQU 310H    ; 数据端口
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
BEGIN:  MOV AX,DATA
        MOV DS,AX

; 启动转换
START:  MOV DX,START_P
        MOV AL,00H      ; (可以是其它值)
        OUT DX,AL       ; 使CS和WR同时有效

; 查转换结束
WAIT1:  MOV DX,STATE_P
        IN AL,DX
        AND AL,80H      ; 查D7=0 (INTR=0?)
        JNZ WAIT1       ; 未结束, 等待

; 需要实现的功能
  
```

.....

9.19 如何设计一个采用中断方式的 A/D 转换器接口？

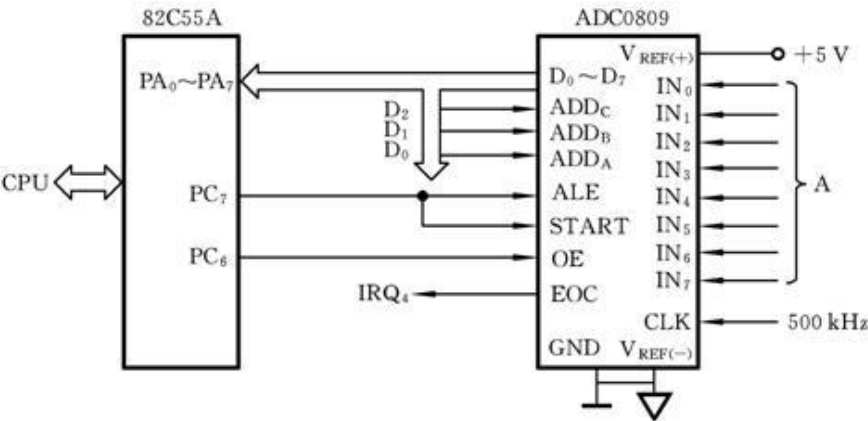
答：

1) 分析

我这里以 ADC0809 为例进行分析，我们主要需要考虑 ADC0809 的外部特性、接口电路结构形式、中断处理这三个方面。

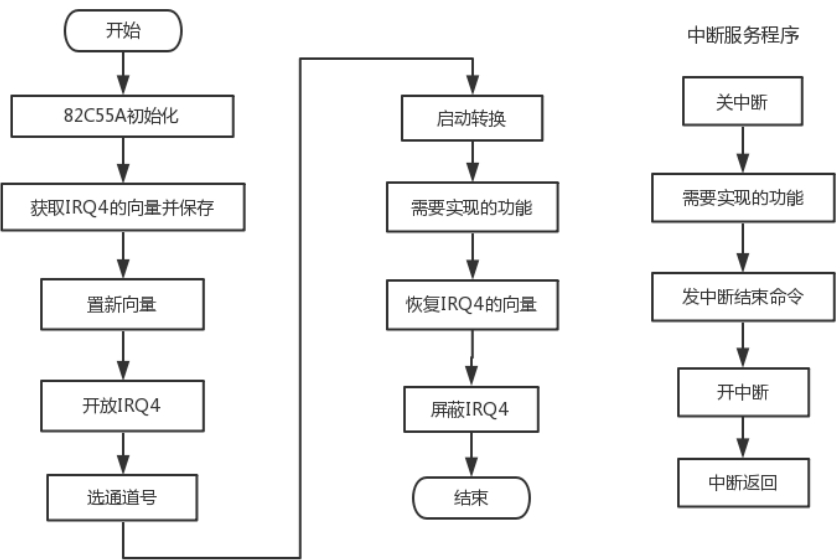
2) 硬件设计

根据上面分析，本接口电路要提供 ADC0809 模拟量通道号选择信号、启动转换信号、读允许信号。这些信号都可由 82C55A 接口芯片实现。而 EOC 的中断请求直接连到系统总线的 IRQ4 上。接口电路如下图所示：



3) 软件设计

程序流程图如下：



主要程序如下：

```
MOV DX, 303H ; 82C55初始化, A口输出 (为锁地址)
MOV AL, 80H
OUT DX, AL
MOV AL, 0EH ; 置PC7=0, 使START和ALE无效
OUT DX, AL
MOV AL, 0CH ; 置PC6=0, 使OE无效
OUT DX, AL
CLI
```

```

MOV AX, 350CH ; 获取IRQ4的中断向量并保存
INT 21H
MOV OLD_OFF, BX
MOV BX, ES
MOV OLD_SEG, BX
MOV AX, 250CH ; 置新中断向量
MOV DX, SEG A_D
MOV DS, DX
MOV DX, OFFSET A_D
INT 21H
MOV AX, DATA ; 恢复数据段
MOV DS, AX

IN AL, 21H ; 开放IRQ4
AND AL, 0EFH
OUT 21H, AL
STI

BEGIN: MOV DX, 303H ; 82C55初始化, A口输出 (为锁地址)
MOV AL, 80H
OUT DX, AL
MOV DX, 300H ; 通道地址送到A口
MOV AL, 07H
OUT DX, AL
MOV DX, 303H ; 启动转换
MOV AL, 0FH ; 产生ALE锁存信号
OUT DX, AL
NOP
MOV AL, 0EH ; 产生START启动脉冲信号
OUT DX, AL

; 需要实现的功能

CLI ; 已完, 关中断
MOV AX, 250CH ; 恢复IRQ4的原中断向量
MOV DX, OLD_SEG
MOV DS, DX
MOV DX, OLD_OFF
INT 21H
MOV AX, DATA ; 恢复数据段
MOV DS, AX
IN AL, 21H
OR AL, 10H ; 屏蔽IRQ4
OUT 21H, AL
MOV AX, 4C00H ; 返回DOS
INT 21H
A_D PROC FAR ; 中断服务程序
PUSH AX ; 保护现场

```

```

PUSH DX
MOV DX, 303H ; 82C55初始化, A口输入
MOV AL, 90H
OUT DX, AL
MOV DX, 303H ; 打开三态锁存器
MOV AL, 0DH
OUT DX, AL ; 置PC6=1高
MOV DX, 300H

; 需要实现的功能

MOV DX, 303H ; 关闭三态锁存器
MOV AL, 0CH
OUT DX, AL ; 置PC6=0低
MOV AL, 20H ; 发中断结束命令
OUT 20H, AL
POP DX ; 恢复现场
POP AX
STI ; 开中断
IRET ; 中断返回

A D ENDP

```

9.21 利用 DAC 作为函数波形发生器，可以产生任何一种波形。如何设计一个产生三角波与锯齿波的 D/A 转换器接口？

答：

1) 分析

首先应该分析使用的 DAC 的连接特性和工作方式，然后根据连接特性和工作方式进行接口设计。我以 DAC0832 为例进行设计：

● 连接特性：

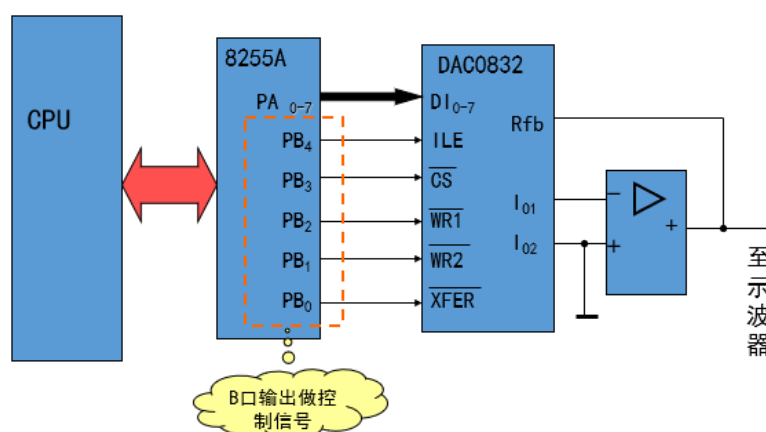
DAC0832 是单通道模拟量输出，不需通道选择；

● 硬件设计：

DAC0832 有两级缓冲锁存器，它有 3 种工作方式：双缓冲方式、单缓冲方式和直通方式。

2) 硬件设计：

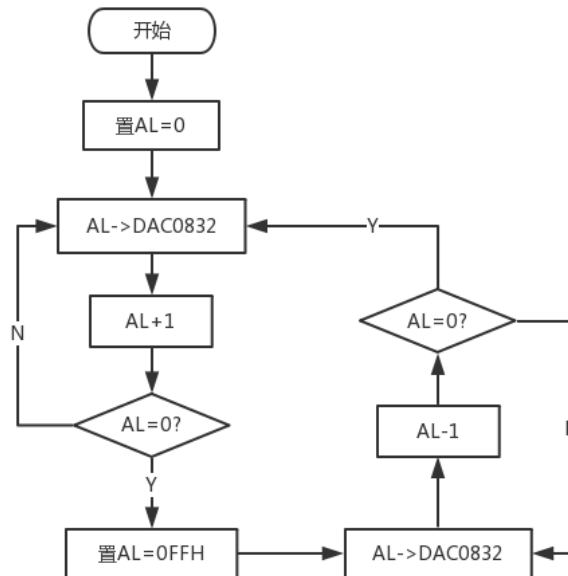
采用 8255 做 CPU 与 DAC 的接口，并把 82C55A 的 A 端口作为数据输出，而 B 端口的 PB0~PB4 根线作为控制信号来控制 DAC0832 的工作方式及转换操作，硬件设计图如下：



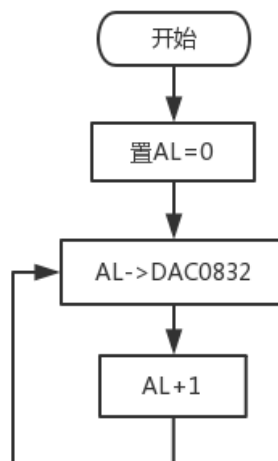
3) 软件设计

因为要求产生三角波和锯齿波，所以 D/A 转换程序主体应该是一个循环结构。

产生三角波程序流程图如下：



产生锯齿波程序流程图如下：



产生三角波的程序如下：

```
; 8255初始化
MOV DX, 303H    ; 8255A的命令口
MOV AL, 10000000B ; 8255A的方式字
OUT DX, AL

; 指定B口控制DAC的转换
MOV DX, 301H    ; 8255A的B口地址
MOV AL, 00010000B ; 置DAC0832为直通工作方式
OUT DX, AL

; 生成三角波的循环
MOV DX, 300H    ; 8255A的A口地址
MOV AL, 0H      ; 输出数据从0开始
L1: OUT DX, AL
    INC AL      ; 输出数据加1
    JNZ L1      ; AL是否加满? 未加满继续
    MOV AL, 0FFH ; 已满, AL置全1
L2: OUT DX, AL
```

```
DEC AL      ;输出数据减1
JNZ L2      ;AL是否减到0? 不为0继续
JMP L1      ;为 0, AL 加 1
```

产生锯齿波的程序如下：

```
; 8255初始化
MOV DX, 303H      ; 8255A的命令口
MOV AL, 10000000B ; 8255A的方式字
OUT DX, AL

; 指定B口控制DAC的转换
MOV DX, 301H      ; 8255A的B口地址
MOV AL, 00010000B ; 置DAC0832为直通工作方式
OUT DX, AL

; 生成锯齿波的循环
MOV DX, 300H      ; 8255A的A口地址
MOV AL, 0H        ; 输出数据从0开始
LOP: OUT DX, AL
      INC AL
      JMP LOP
```