

第二章 微处理器及其结构

2-7 什么是逻辑地址？什么是物理地址？在实地址方式下,如何求存储器的物理地址？设一个 16 字的数据区,它的起始地址为 70A0H:DDF6(段基址:偏移地址). 写出这个数据区的首字单元和末字单元的物理地址.

解:

1). 实模式下,逻辑地址由段基址和偏移地址组成.物理地址是真正的存储单元的地址.

2). 物理地址=段基址*16 + 偏移地址

3). 首字单元地址:

$$70A0H * 16 + DDF6H = 70A00H + DDF6H = 7E7F6H$$

末字单元地址:

$$7E7F6H + (16-1) * 2 = 7E7F6H + 1EH = 7E814H$$

注意:

相邻两个存储单元可构成一个字长为 16 位的字,在对准字时,用偶地址表示字的地址.

第三章 指令系统

3-6 分别指出下列指令中源操作数和目标操作数的寻址方式. 若是存储器寻址,用表达式表示 EA=?

- (1) AND AX, 00FFH
- (2) ADD BX, [00FFH]
- (3) MOV AX, [BX+10H]
- (4) ADD AX, [ESI*8]
- (5) SUB [BP][SI], AX
- (6) MOV AX, [BX+DI+20H]
- (7) CMP [SI], AX
- (8) OR AX, DX
- (9) MOV EAX, [ESI][EDI*2]
- (10) PUSH DS

解:

- (1) 目: 立即数寻址
- (2) 目: 直接寻址 EA=00FFH
- (3) 目: 相对寄存器间接寻址 EA=(BX)+10
- (4) 目: 比例间址 EA=ESI*8
- (5) 源: 基址加间址寻址 EA=(BP)+(SI)
- (6) 目: 相对基址加间址寻址 EA=(BX)+(DI)+20H
- (7) 源: 寄存器间址寻址 EA=(SI)
- (8) 目、源: 寄存器寻址
- (9) 目: 基址加比例间址寻址 EA=(ESI)+(EDI)*2
- (10) 源: 寄存器寻址

注意: ◆ 16 位寻址: BX 和 BP 作为基址寄存器. BX 以 DS 作为默认段寄存器, BP 以 SS 为默认段寄存器.

SI 和 DI 作为间址寄存器. 默认 DS 为段寄存器

- ◆ 32 位寻址: 8 个 32 位通用寄存器均可作为基址寄存器, 其中 ESP, EBP 以 SS 为默认段寄存器, 其余均以 DS 为默认段寄存器.

除 ESP 外的其它 7 个寄存器均可作间址寄存器, EBP 默认 SS 作段基址寄存器, 其它以 DS 作段基址寄存器

3-7 32 位微机工作在实地址模式下, 已知(DS) = 1000 和(SS) = 2000H, (SI) = 007FH, (BX) = 0040H, (BP) = 0016H, 变量 TABLE 的偏移地址为 0100H. 指出下列指令中源操作数的寻址方式, 求它的有效地址(EA)和物理地址(PA).

- (1) MOV AX, [1234H]
- (2) MOV AX, TABLE
- (3) MOV AX, [BX+100H]
- (4) MOV AX, TABLE[BP][SI]

解:

- (1) 直接寻址

EA=1234H PA=(DS)*16 + EA = 11234H

- (2) 直接寻址

EA=(TABLE)=0100H PA=(DS)*16+EA=10100H

- (3) 相对寄存器间接寻址

EA=(BX)+100H=0140H PA=(DS)*16+EA=10140H

- (4) 相对基址加间址寻址

EA=(BP)+(SI)+TABLE=0195H PA=(SS)*16+EA=20195H

注意: 当基址寄存器和间址寄存器默认的段寄存器不同时, 一般规定, 由基址寄存器来决定默认的段寄存器为段基址寄存器. 这里 BP 为基址寄存器, 所以默认 SS 为段基址寄存器.

3-8 指出下列指令的错误, 并加以改正.

- (1) MOV DS, 100
- (2) MOV 1020H, DX
- (3) SUB [1000H], [SI]
- (4) PUSH AL
- (5) IN AL, [80H]
- (6) MOV DS, ES
- (7) JMP BX
- (8) SHR DX, 4
- (9) OUT 380H, AX
- (10) ADD AL, BX
- (11) POP CS
- (12) MOV CL, 3300H

解:

- (1) 立即数不能直接传送到段寄存器中去

应改为: MOV AX, 100
MOV DS, AX

- (2) 立即数只能出现在源操作数位置
应改为: MOV DX, 1020H
- (3) 源操作数和目标操作数不能同时为寄存器寻址
应改为: MOV AX, [1000H]
SUB AX, [SI]
- (4) PUSH 指令不能操作 8 位数据
应改为: PUSH AX
- (5) [80H] 不是端口 IN AL, 80H
应改为: IN AL, 80H
- (6) 两个段寄存器之间不能直接传送
应改为: MOV AX, ES
MOV DS, AX
- (7) 对
- (8) 移位次数超过 1 的时候, 要把移位次数放入 CL 中
应改为: MOV CL, 4
SHR DX, CL
- (9) 端口地址大于 255 时, 要把地址放入 DX 中
应改为: MOV DX, 380H
OUT DX, AX
- (10) 源操作数和目标操作数不匹配
应改为: ADD AX, BX
- (11) POP 指令只能使用在存储器或通用寄存器
可改为: POP AX
- (12) 源操作数和目标操作数不匹配
应改为: MOV CX, 3300H

3-9 已知: (DS) = 091DH, (SS) = 1E4AH, (AX) = 1234H, (BX) = 0024H, (CX) = 5678H, (BP) = 0024H, (SI) = 0012H, (DI) = 0032H, [09226H] = 00F6H, [09228H] = 1E40H, [1E4F6H] = 091DH. 试求下列各指令单独执行后的结果.

- (1) MOV CL, 20H[BX][SI] ; (CL) = ?
- (2) MOV [BP][DI], CX ; [1E4F6H] = ?
- (3) LEA BX, 20H[BX][SI] : (BX) = ?
MOV AX, 2[BX] : (AX) = ?
- (4) LDS SI, [BX][DI]
MOV [SI], BX ; (SI) = ?
- (5) XCHG CX, 32H[BX] ; (AX) = ?
XCHG 20[BX][SI], AX ; [09226H] = ?

解:

- (1) (CL) = 00F6H
- (2) [1E4F6H] = 5678H
- (3) (BX) = 0056H
(AX) = 1E40H
- (4) (SI) = 0024H
- (5) (AX) = 5678H

[09226H] = 1234H

3-10 已知 (AL) = 0C4H, DATA 单元中内容为 5AH, 写出下列每条指令单独执行后的结果 (ODITSZAPC: 0---xxux0)

- (1) AND AL, DATA
- (2) OR AL, DATA
- (3) XOR AL, DATA
- (4) NOT DATA
- (5) AND AL, 0FH
- (6) OR AL, 1H
- (7) XOR AL, 0FFH
- (8) TEST AL, 80H

解:

- (1) (AL) = 40H CF=0,OF=0,SF=0,ZF=0,PF=0,AF 无定义
- (2) (AL) = DEH CF=0,OF=0,SF=1,ZF=0,PF=1,AF 无定义
- (3) (AL) = 9EH CF=0,OF=0,SF=1,ZF=0,PF=0,AF 无定义
- (4) (AL) = A5H 不影响任何标志位
- (5) (AL) = 04H CF=0,OF=0,SF=0,ZF=0,PF=0,AF 无定义
- (6) (AL) = C5H CF=0,OF=0,SF=1,ZF=0,PF=1,AF 无定义
- (7) (AL) = 3BH CF=0,OF=0,SF=0,ZF=0,PF=0,AF 无定义
- (8) (AL)不变=0C4H CF=0,OF=0,SF=1,ZF=0,PF=0,AF 无定义

3-12 (AL)=8EH,(BL)=72H, 执行以下指令后, 标志位 OF、SF、ZF、AF、PF 和 CF 的值是什么?

- (1) ADD AL,BL
- (2) AND BL,AL
- (3) CMP AL,BL
- (4) SHL AL,1

解:

- (1) OF=0,SF=0,ZF=1,AF=1,PF=1,CF=1
- (2) OF=0,SF=0,ZF=0,AF= (未定义),PF=0,CF=0
- (3) OF=1,SF=0,ZF=0,AF=0,PF=0,CF=0
- (4) OF=1,SF=0,ZF=0,AF= (未定义),PF=0,CF=1

3-15 试用 CMP 指令和无条件指令实现以下判断

- (1) AX 和 CX 中的内容均为无符号数
 - ① (AX) > (CX) 则转至 BIGGER 标号执行
 - ② (AX) < (CX) 则转至 LESS 标号执行
- (2) BX 和 DX 中的内容均为有符号数
 - ① (BX) > (DX) 则转至 BIGGER 标号执行
 - ② (BX) < (DX) 则转至 LESS 标号执行

解:

- (1) CMP AX,CX
JA BIGGER
JB LESS
- (2) CMP BX,DX
JG BIGGER
JL LESS

第四章 汇编语言程序设计

4-9 试用伪指令编写一数据段与下面程序等效。

```
MOV AX, 0913H
MOV DS, AX
MOV BX, 20H
MOV [BX], 'AB'
```

```
DATA SEGMENT AT 0931H
    ORG 20H
    DATA1 DW ?
(或 DATA1 DW 'AB')
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
        MOV DS,AX
        MOV DATA1,'AB'
CODE ENDS
```

4-10 下面各题有语法错误，分别用两种办法修改，使其正确。

- (1) M1 DW 5060H ... MOV BL, M1

改：

```
M1 DB 60H, 50H
```

```
...
```

```
MOV BL,M1
```

```
M1 DW 5060H
```

```
...
```

```
MOV BL, BYTE PTR M1
```

```
MOV BX, M1
```

```
M2 LABEL BYTE
```

```
M1 DW 5060H
```

```
...  
MOV BL,M2
```

(2) M2 EQU 10H ... MOV M2,AL

```
改: DATA SEGMENT  
    ORG 10H  
    M2 DB ?  
DATA ENDS  
CODE SEGMENT  
    ASSUME CS:CODE,DS:DATA  
START: MOV AX,DATA  
        MOV DS,AX  
        MOV M2,AL  
CODE ENDS  
END START
```

```
MOV SI,10H  
MOV [SI], AL
```

```
MOV DS:[0010H],AL
```

(3) M3 DW 'ABCD'

```
改: M3 DW 'AB','CD'  
    M3 DB 'ABCD'
```

(4) M4 DB 1234

```
改: M4 DB '1234'  
    M4 DW 1234
```

```
(5) DATA1 SEGMENT  
    DA1 DW 1234H  
DATA1 ENDS  
DATA2 SEGMENT  
    DA2 DW 5678H  
DATA2 ENDS  
CODE SEGMENT  
    ASSUME CS: CODE, DS: DATA1  
    ...  
    MOV BX, DA2  
    ...  
CODE ENDS
```

```
ASSUME CS: CODE, DS: DATA2
```

```
ASSUME CS:CODE, DS: DATA1
```

```
...
```

```
ASSUME DS: NOTHING
```

```
ASSUME DS: DATA2
```

```
ASSUME CD:CODE, DS: DATA1, ES: DATA2
```

```
MOV AX, DATA2
```

```
MOV ES, AX
```

```
...
```

```
MOV BX,DA2
```

4-12 按照下面要求写出程序的框架

- (1) 数据段的位置从 8000H 开始，数据段中定义一个有 100 个字节的数组。
- (2) 堆栈段名为 STACK1，留 100 个字的空间。
- (3) 代码段指定段寄存器，主程序从 10000H 开始，给有关段寄存器赋值。

```
DATA SEGMENT AT 8000H
```

```
    M1 DB 100 DUP (?)
```

```
DATA ENDS
```

```
STACK1 SEGMENT STACK
```

```
    DW 100 DUP (?)
```

```
STACK1 ENDS
```

```
CODE SEGMENT    AT 1000H
```

```
    ASSUME CS:CODE,DS:DATA, SS: STACK1
```

```
START:  MOV AX,DATA
```

```
        MOV DS,AX
```

```
CODE  ENDS
```

```
    END START
```

4-15 将 MBUF 为起始单元的 5 个数按相反次序传到 NBUF 单元开始的存储单元之中。

```
DATA  SEGMENT
```

```
    MBUF  DW  1,2,3,4,5
```

```
    NBUF   DW  5 DUP(?)
```

```

        M EQU 10H
DATA    ENDS
STACK1  SEGMENT PARA STACK
        DB  100 DUP(0)
STACK1  ENDS
CODE    SEGMENT
ASSUME  CS:CODE, DS:DATA, SS:STACK1
START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  SI, OFFSET MBUF
        MOV  DI, OFFSET NBUF
        MOV  CX, 5
        ADD  SI, 8
LOP:    MOV  AX, [SI]
        MOV  [DI], AX
        DEC  SI
        DEC  SI
        INC  DI
        INC  DI
        DEC  CX
        JNZ  LOP
        MOV  AH, 4CH
        INT  21H
CODE    ENDS
        END  START

```

4-16 试编写程序,完成下列数字表达式的功能

$$Y = \begin{cases} 1(X \geq 1) \\ 0(-1 < X < 1) \\ -1(X \leq -1) \end{cases}$$

解:

```

DATA    SEGMENT
        XX  DB  X
        Y   DB  ?
DATA    ENDS
STACK1  SEGMENT PARA STACK
        DB  100 DUP(0)
STACK1  ENDS
CODE    SEGMENT
ASSUME  CS:CODE, DS:DATA, SS:STACK1

```



```

START: MOV  AX, DATA
        MOV  DS, AX
        MOV  AL, XX
        CMP  AL, -1
        JG   BGR
        MOV  AL, -1
        JMP  JUST2
BGR:    CMP  AL, 1
        JGE  JUST1
        MOV  AL, 0
        JMP  JUST2
JUST1:  MOV  AL, 1
JUST2:  MOV  Y, AL
        MOV  AH, 4CH
        INT  21H
CODE    ENDS
        END  START

```

4-17 将内存中某数据块中的正数和负数分开,并分别将这些正数和负数送同一数据段的两个缓冲区中,并在屏幕上显示正数和负数的个数.

```

DATA    SEGMENT
D1      DB  -1, -2, 1, 2, 3, -4, 4,5,-5,0,0,9,-3
COUNT EQU  $-D1
D2      DB  100 DUP(?)
D3      DB  100 DUP(?)
PNUM    DW  1 DUP(?)
NNUM    DW  1 DUP(?)
CONST   DW  10000,1000,100,10,1
DECS    DB  5 DUP(0)
HEAD1   DB  'THE NUMBER OF POSITIVE NUMBER IS:$'
HEAD2   DB  'THE NUMBER OF NEGATIVE NUMBER IS:$'
DATA    ENDS
STACK1  SEGMENT PARA STACK
        DB  100 DUP(0)
STACK1  ENDS

```

```

CODE    SEGMENT
ASSUME  CS:CODE, DS:DATA,SS:STACK1
START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  BX, OFFSET D1
        MOV  SI, OFFSET D2
        MOV  DI, OFFSET D3

```

```

        MOV    CX, COUNT
LOP:    MOV    AL, [BX]
        CMP    AL, 0
        JGE    JUST1
        MOV    [DI], AL
        INC    DI
        INC    NNUM
        JMP    JUST2
JUST1:  JE     JUST2
MOV     [SI], AL
        INC    SI
        INC    PNUM
JUST2:  INC    BX
        DEC    CX
        JNZ    LOP

```

```

MOV AH,9H
MOV DX,OFFSET HEAD1
INT 21H
MOV AX,PNUM

```

```

MOV DL,0AH
MOV AH,02H
INT 21H
MOV DL,0DH
MOV AH,02H
INT 21H

```

```

MOV AH,9H
MOV DX,OFFSET HEAD2
INT 21H
MOV AX,NNUM
CALL DISPLAY

```

```

MOV DL,0AH
MOV AH,02H
INT 21H
MOV DL,0DH
MOV AH,02H
INT 21H

```

```

MOV  AH, 4CH
INT  21H

```

```

DISPLAY PROC NEAR
    MOV CX,5
    LEA SI,CONST
    LEA DI,DECS
CONV3:  MOV BL,0
LOP1:   SUB AX,[SI]
        JC NEXT
        INC BL
        JMP LOP1
NEXT:   ADD AX,[SI]
        OR BL,30H
        MOV [DI],BL
        INC SI
        INC SI
        INC DI
        LOOP CONV3
        MOV AH,02H
        MOV CX,5
        LEA DI,DECS
CONV4:  MOV DL,[DI]
        CMP DL,30H
        JZ NODIS
        JMP DISP
NODIS:  INC DI
        LOOP CONV4
DISP:   INT 21H
        CMP CX,0
        JZ QUIT
        INC DI
        MOV DL,[DI]
        LOOP DISP
QUIT:   RET
DISPLAY ENDP

CODE ENDS
        END START

```

4-19 内存 BLOCK 开始,存放着 256 个字节的带符号数.编写程序,从这些数中找出绝对值最大的数,将其放入 MAX 中.

解:

```
DATA SEGMENT
    BLOCK DB -1, -3, 2, 5, -4, -7, -8, 9, 10, -124
    COUNT EQU $-BLOCK
    MAX DB ?
    MAX1 DB ?
```

DATA ENDS

```
STACK1 SEGMENT PARA STACK
    DB 100 DUP(0)
```

STACK1 ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK1

```
START: MOV AX, DATA
        MOV DS, AX
        MOV SI, OFFSET BLOCK
        MOV CX, COUNT
        MOV BL, [SI]
        CMP BL, 0
        JL ABS1
        JMP CON1
```

ABS1: NEG BL

CON1: DEC CX

```
LOP: INC SI
      MOV AL, [SI]
      CMP AL, 0
      JL ABS2
      JMP CON2
```

ABS2: NEG AL

CON2: CMP BL, AL

JAE JUST1

XCHG BL, AL

MOV DI, SI

JUST1: DEC CX

JNZ LOP

MOV BH, [DI]

MOV MAX, BH

MOV MAX1, BL

MOV AH, 4CH

INT 21H

CODE ENDS

END START

第五章 存储器

5-7 若用 1024*1b 的 RAM 芯片组成 16K*8b 的存储器,, 需要多少芯片?在地址线中有多少位参与片内寻址?多少位用做芯片组选择信号?

解:

先进行位扩展,一组芯片需要 8 片

再进行字扩展,需要 16 组芯片.

所以共需要 $16*8=128$ 片

$1024=1K$,需要 10 位参与片内寻址

$16=2^4$,需要 4 位做选择信号.

5-8 试用 4K*8b 的 EPROM2732 和 8K*8b 的 SRAM6264,以及 74LS138 译码器,构成一个 8KB 的 ROM,32KB 的 RAM 存储系统,要求设计存储器扩展电路,并指出每片存储芯片的地址范围.

解:

		← 片选译码 →		← 片内译码 →	
芯片	型号	A15~A13	A12	A11~A0	地址范围
1#	2732	000	0	00...0	0000H~0FFFH
2#	2732	000	1	~ 11...1	1000H~1FFFH
3#	6264	001	00...0 ~ 11...1		2000H~3FFFH
4#	6264	010			4000H~5FFFH
5#	6264	011			6000H~7FFFH
6#	6264	100			8000H~9FFFH

5-9 用 EPROM2764 (8K*8b) 和 SRAM6264 (8k*8b) 各一片组成存储器,其地址范围为 FC000~FFFFFH, 试画出存储器与 CPU 的连接图和片选信号译码电路 (CPU 地址线 20 位, 数据线 8 位)。

1111 1100 0000 0000 0000

1111 1101 1111 1111 1111

1111 1110 0000 0000 0000

1111 1111 1111 1111 1111

选择 138 译码器 Y6 和 Y7

5-10 现有存储芯片:2K*1b 的 ROM 和 4K*1bde RAM,若用它们组成容量为 16KB 的存储器,前 4KB 为 ROM,后 12KB 为 RAM,问各种存储芯片分别用多少片?

解:

4KB=4K*8b 需要 $2*8=16$ 片

12KB=12K*8b 需要 $3*8=24$ 片

第六章 中断技术

6-18 某系统使用一片 8259A 管理中断, 中断请求由 IR2 引入, 采用电平触发、完全嵌套、普通 EOI 结束方式, 中断类型为 42H, 端口地址为 80H 和 81H,

试画出 8259A 与 CPU 的硬件连接图，并编写初始化程序。

解：

电路图参见教材 P178 图 6-19, \overline{CS} 由 $A_{19}-A_1=0000\ 0000\ 0000\ 1000\ 000$ 和 M/\overline{IO} 给出低电平。

```
MOV AL, 1BH(00011011B)
OUT 80H, AL          ; 设置 ICW1
MOV AL, 40H(01000000B)
OUT 81H, AL          ; 设置 ICW2
MOV AL, 01H(00000001B)
OUT 81H, AL          ; 设置 ICW4
MOV AL, FBH(11111011B)
OUT 81H, AL          ; 设置 OCW1
MOV AL, 20H(00100000B)
OUT 80H, AL          ; 设置 OCW2
```

6-19 某系统使用两片 8259A 管理中断，从片的 INT 连接到主片的 IR2 请求输入端。设主片工作于边沿触发、特殊完全嵌套、非自动结束和非缓冲方式，中断类型号为 70H，端口地址为 80H 和 81H；从片工作与边沿触发、完全嵌套、非自动结束和非缓冲方式，中断类型号为 40H，端口地址为 20H 和 21H。要求：

- (1) 画出主、从片级联图
- (2) 编写主、从片初始化程序

解：

电路图参见教材 P179 图 6-21，主片 \overline{CS} 由 $A_{19}-A_1=0000\ 0000\ 0000\ 1000\ 000$ 和 M/\overline{IO} 给出低电平，从片 \overline{CS} 由 $A_{19}-A_1=0000\ 0000\ 0000\ 0010\ 000$ 和 M/\overline{IO} 给出低电平。

主片 8259A

```
MOV AL, 11H(00010001B)
OUT 80H, AL ;定义 ICW1
MOV AL, 70H(01110000B)
OUT 81H, AL ;定义 ICW2
MOV AL, 04H(00000100B)
OUT 81H, AL ;定义 ICW3
MOV AL, 11H(00010001B)
OUT 81H, AL ;定义 ICW4
MOV AL, FBH(11111011B)
OUT 81H, AL ;定义 OCW1
```

(开放从片 IR2 的请求)

```
IN AL, 81H
AND AL, 11111011
```

OUT 81H, AL

MOV AL, 20H(00100000B)
OUT 80H, AL ; 定义 OCW2

从片 8259A
MOV AL, 11H(00010001B)
OUT 20H, AL ; 定义 ICW1
MOV AL, 40H(01000000B)
OUT 21H, AL ; 定义 ICW2
MOV AL, 02H(00000010B)
OUT 21H, AL ; 定义 ICW3
MOV AL, 01H(00000001B)
OUT 21H, AL ; 定义 ICW4

MOV AL, 20H(00100000B)
OUT 20H, AL ; 定义 OCW2

6-20 某系统由 8259A 的 IR2 引入外设中断请求（跳变信号有效），要求当 CPU 响应 IR2 请求时，输出显示字符串“****”，并中断 10 次退出，试编写主程序和中断服务程序。

解：

设 8259A 的 I/O 地址为 20H，21H，中断类型号 0AH，从 IR2 引入

```
DATA SEGMENT
    MESS DB '****',0AH,0DH,'$'
    INTA00 EQU 0020H
    INTA01 EQU 0021H
DATA ENDS
STACK SEGMENT STACK
    DB 100H DUP (?)
STACK ENDS
CODE SEGMENT
    ASSUME CS: CODE, DS: DATA, SS: STACK
MAIN: MOV AX, DATA
        MOV DS, AX
        MOV DX, INTA00 ; 8259A 初始化
        MOV AL, 13H ; 写 ICW1
        OUT DX, AL
        MOV DX, INTA01
        MOV AL, 08H ; 写 ICW2
        OUT DX, AL
```

```

MOV AL, 01H ; 写 ICW4
OUT DX, AL
PUSH DS
MOV AX, SEG INT-P ; 设置中断矢量
MOV DS, AX
MOV DX, OFFSET INT-P
MOV AL, 0AH
MOV AH, 25H
INT 21H
POP DS
MOV AL, 0FBH ; 写中断屏蔽字 OCW1
OUT DX, AL
MOV BX, 10
WAIT1: STI ; 开中断
JMP WAIT1 ; 等待中断
MOV AX, 4C00H ; 返回操作系统
INT 21H

INT-P: MOV AX, DATA ; 中断服务程序入口
MOV DS, AX
MOV DX, OFFSET MESS ; 输出指定字符串
MOV AH, 09H
INT 21H
MOV DX, INTA00 ; 写 OCW2, 送中断结束命令 EOI
MOV AL, 20H
OUT DX, AL
DEC BX ; 控制 10 次循环
JNZ NEXT
MOV DX, INTA01 ; 读屏蔽寄存器 IMR
IN AL, DX
OR AL, 04H ; 屏蔽 IR2 请求
OUT DX, AL
STI ; 开中断
MOV DX, INTA00
MOV AL, 20H ; 写中断结束方式 OCW2
OUT DX, AL
NEXT: IRET ; 中断返回
CODE: ENDS
END MAIN

```

第七章 I/O 接口技术

7-6 设 8255A 的 A 口工作于方式 1 输出, B 口工作于方式 0 输入, 试编写初始化程序 (设端口地址为 40H-43H)。

解:

```
MOV DX, 43H
MOV AL, 10100010B(0A2H)
OUT DX,AL
```

7-7 使用 8255A 作为开关和 LED 指示灯的接口。要求 8255A 的 A 口连接 8 个开关, B 口连接 8 个 LED 指示灯, 将 A 口的开关状态读入, 然后送至 B 口控制指示灯亮、灭。试画出接口电路设计图, 并编写程序实现之。

解:

电路图参见教材 P193 图 7-14。A 口接入 8 个开关, B 口用 LED 替代数码管, 共阴接法。

设 8255 四个端口的地址为 FFE0H~FFE3H。

```
DATA SEGMENT
    PORTA EQU 0FFE0H
    PORTB EQU 0FFE1H
    CONTR EQU 0FFE3H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:  MOV AX, DATA
        MOV DS, AX
        MOV AL, 90H      ;初始化 8255,PA 为输入, PB 为输出
        MOV DX, CONTR
        OUT DX, AL
L0:     MOV DX, PORTA     ;读 PA 口
        IN AL, DX
        NOT AL           ; 按下开关灯亮
        MOV DX, PORTB
        OUT DX,AL        ;送 PB 口显示
        MOV CX,1000      ;延时
L1:     DEC CX
        JNZ L1
        JMP L0
CODE ENDS
        END START
```

7-10 利用 8254 的通道 1,产生 500Hz 的方波信号.设输入时钟频率 CKL1=2.5MHz, 端口地址为 FFA0H~FFA3H,试编写初始化程序.

定时系数计算: $2.5\text{MHz}/500\text{Hz}=5000$

(特别注意: 1.和单片机计算方法不一样, 不要记混。

2.计数方式和定时方式计算计数器初值不一样

3.计数器是 16 位减 1 计数器

4. 设定 BCD 和二进制计数方式时，赋值的区别)

解:

```
MOV AL, 01110110B
MOV DX, 0FFA3H
OUT DX, AL
MOV AX, 5000          ;2.5MHz/500Hz=5000
MOV DX, 0FFA1H
OUT DX, AL
MOV AL, AH
OUT DX, AL
```

注意: 端口数大于 255, 应先装入 DX 中.

7-11 某系统使用 8254 的通道 0 作为计数器, 计满 1000, 向 CPU 发中断请求, 试编写初始化程序(端口地址自设)

解:

设端口地址为 80H~83H

```
MOV AL, 00110000B
OUT 83H, AL
MOV AX, 1000
OUT 80H, AL
MOV AL, AH
OUT 80H, AL
```

7-12 采用 8254 的通道 0 产生周期为 10ms 的方波信号, 设输入时钟的频率为 100kHz, 8254 的端口地址为 38H~3BH, 试编写初始化程序

解:

```
MOV AL, 00110110B
OUT 3BH, AL
MOV AX, 1000      ; 10ms*100kHz=1000
OUT 38H, AL
MOV AL, AH
OUT 38H, AL
```

注意: 计数值的高低字节的装入;

7-15 什么是波特率? 假设异步传输的一帧信息由 1 位起始位、7 位数据位、1 位校验和 1 位停止位构成, 传送的波特率为 9600, 则每秒钟能传输的字符个数是多少?

解: 波特率表示每秒钟传送二进制的位数。每秒钟能传输的字符个数为 $9600/10=960$ 。

7-16 一个异步串行发送器, 发送的字符格式为: 1 位起始位, 7 位数据位, 1 位奇偶校验位和 2 位停止位, 若每秒发送 100 个字符, 则其波特率为多少?

解: 波特率为 $100 \times 11 = 1100$ 。

7-19 设某系统使用一片 8250 进行串行通信, 要求波特率为 2400、8 位数据位、

2 位停止位、偶校验，对接收缓冲器满开中断，试编写初始化程序。

解：

设 8250 端口地址为 3F8H--3FEH，基准频率 1.8432MHZ，初始化程序如下：

```
MOV DX, 3FBH
MOV AL, 80H(10000000B)
OUT DX, AL ; 置线路控制寄存器 DLAB 为“1”
MOV DX, 3F8H
MOV AL, 30H ; 1843200 / (1200*16) = 48=30H
OUT DX, AL
INC DX
MOV AL, 0
OUT DX, AL ; 写除数 R 高位
MOV DX, 3FBH
MOV AL, 1FH(00011111B)
OUT DX, AL ; 写线路控制 R
MOV DX, 3FCH
MOV AL, 03H(00000011B)
OUT DX, AL ; 写 MODEM 控制 R，数据终端就绪，请求发送
MOV DX, 3F9H
MOV AL, 01H(00000001B)
OUT DX, AL ; 接收缓冲器满中断
```

7-21 设计一个应用系统，要求：8255A 的 A 口输入 8 个开关信息，并通过 8250 以串行的方式循环，将开关信息发送出去。已知：8255 的端口地址为 100H-103H。8250 输入的基准时钟频率为 1.8432MHz，传输波特率为 2400，数据长度为 8 位，2 位停止位，奇校验，屏蔽全部中断，端口地址为 108H-10EH，采用查询方式传送。要求：

- (1) 设计该系统的硬件接口电路（包括地址译码电路）。
- (2) 编写各芯片的初始化程序；
- (3) 编写完成上述功能的应用程序。

解：

电路图主要由 CPU 与 8255A 的连接、8255A 口输入电路、8250 与 CPU 的连接（应画出 $D_7 \sim D_0$ 、 \overline{IOR} 、 \overline{IOW} 、时钟等，参见教材 P223 图 7-45）、8255A 地址译码电路（ $A_1A_0=00-11$ ， \overline{CS} 由 $A_{19}-A_2=0000\ 0000\ 0001\ 0000\ 00$ 和 M/\overline{IO} 给出低电平）、8250 地址译码电路（ $A_2A_1A_0=000-111$ ， \overline{CS}_2 由 $A_{19}-A_3=0000\ 0000\ 0001\ 0000\ 1$ 和 M/\overline{IO} 给出低电平）等组成。

```
DATA SEGMENT
    A DB ?
DATA ENDS
```

```

STACK1 SEGMENT PARA STACK
    DW 100 DUP(?)
STACK1 ENDS
CODE SEGMENT
    ASSUME CS: CODE, DS: DATA, SS:STACK1
START:  MOV AX, DATA
        MOV DS, AX
        MOV DX, 103H
        MOV AL, 10010000B(90H)
        OUT DX, AL

        MOV  DX, 10BH
        MOV  AL, 80H(10000000B)
        OUT  DX, AL          ; 置线路控制寄存器 DLAB 为 “1”
        MOV  DX, 108H
        MOV  AL, 30H          ; 1843200 / (2400*16) = 48=30H
        OUT  DX, AL
        INC  DX
        MOV  AL, 0
        OUT  DX, AL          ; 写除数 R 高位
        MOV  DX, 10BH
        MOV  AL, 0FH(00001111B)
        OUT  DX, AL          ; 写线路控制 R
        MOV  DX, 10CH
        MOV  AL, 03H(00000011B)
        OUT  DX, AL          ; 写 MODEM 控制 R
        MOV  DX, 109H
        MOV  AL, 0
        OUT  DX, AL          ; 屏蔽全部中断

WAIT_FOR: MOV DX, 10DH          : 读线路状态寄存器
          IN  AL, DX

          TEST AL, 00100000B    ; 发送寄存器空否, 不空则返回等待
          JZ  WAIT_FOR
          MOV  DX, 100H          ; 读 A 口状态
          IN  AL, DX
          MOV  DX, 108H
          OUT  DX, AL          ; 发送
          JMP  WAIT_FOR
          MOVAH, 4CH
          INT  21H
CODE  ENDS
      END  START

```

7-25 采用 8237 的通道 1 控制外设与存储器之间的数据，设该片的 \overline{CS} 由地址线 A15-A4=031H 译码提供。试编写初始化程序，把外设中 1KB 的数据传送到内存 2000H 开始的存储区域，传送完毕停止通道工作。

解：

```
START:  MOV  DX, 031DH
        OUT  DX, AL    ; 软件复位，先/后触发器为 0
        MOV  DX, 0312H
        MOV  AL, 00H
        OUT  DX, AL    ; 2000H 写入基（当前）地址寄存器
        MOV  AL, 20H
        OUT  DX, AL
        MOV  AX, 1024   ; 传输的字节数 1024
        DEC  AX         ; 计数值调整为 1024-1
        MOV  DX, 0313H
        OUT  DX, AL    ; 计数值写入基（当前）字节计数器
        MOV  AL, AH
        OUT  DX, AL
        MOV  AL, 85H    ; 块传送，地址增 1，写传送，禁止自动预置
        MOV  DX, 031BH
        OUT  DX, AL    ; 写方式字
        MOV  AL, 00H    ; DACK1=0, DREQ1=0, 允许 8237 工作
        MOV  DX, 0318H
        OUT  DX, AL    ; 写命令字
        MOV  AL, 01H
        MOV  DX, 031AH
        OUT  DX, AL    ; 写屏蔽字，允许通道 1 请求
```