

第3章

主教材课后习题答案

习题 1

1-1 有如下的数据段定义,写出内存单元中的数据存放形式。

```
DATA SEGMENT
    ORG 100H
    VAL DW 345BH
    DATA_BYTE DB 12, 3AH
    DATA_WORD DW 21, $ + 5, - 5
    DATA_DW DD 3 * 8, 04030201H
    MESSAGE dw 'AB'
    DATA1 DB 1, 2H
    EXPR DW 1, 2
    STR DB 'WELCOM!'
    S1 DW 'AB'
    S2 DD 'AB'
    OFFAB DW S1
DATA ENDS
```

解答:

具体的存放形式如表 3-1 所示。

表 3-1 内存单元中的内容及地址

存储单元	内存中的内容(十六进制)	偏移地址
VAL	5B	0100
	34	0101
DATA_BYTE	0C	0102
	3A	0103

续表

存 储 单 元	内存中的内容(十六进制)	偏 移 地 址
DATA_WORD	15	0104
	00	0105
	0B	0106
	01	0107
	FB	0108
	FF	0109
DATA_DW	18	010A
	00	010B
	00	010C
	00	010D
	01	010E
	02	010F
	03	0110
	04	0111
MESSAGE	42	0112
	41	0113
DATA1	01	0114
	02	0115
EXPR	01	0116
	00	0117
	02	0118
	00	0119
STR	57	011A
	45	011B
	4C	011C
	43	011D
	4F	011E
	4D	011F
	21	0120
S1	42	0121
	41	0122
S2	42	0123
	41	0124
	00	0125
	00	0126
OFFAB	21	0127
	01	0128
	00	0129
	00	012A

1-2 有符号定义语句如下,问 L 的值是多少?

```
BUF DB 1,2,3,'123'  
EBU DB 0  
L EQU EBU - BUF
```

解答: L 的值是存储单元 EBU 和 EBF 地址的差,也就是存储单元 EBF 的存储单元的大小,共 6 字节,所以 L 的值是 6。

1-3 有如下的数据定义,各条指令单独执行后,各寄存器的内容是什么?

```
a DB ?  
b dw 30 dup('b')  
c db 'ABCD'
```

- (1) mov ax, type a
- (2) mov ax, type b
- (3) mov cx, length b
- (4) mov dx, size b
- (5) mov cx, size c

解答:

- (1) mov ax, type a (ax) = 1
- (2) mov ax, type b (ax) = 2
- (3) mov cx, length b (cx) = 30
- (4) mov dx, size b (dx) = 60
- (5) mov cx, size c (cx) = 4

1-4 下列哪些指令需要添加 ptr 伪操作?

数据段定义如下:

```
adt db 10h,20h  
bdt dw 1234h
```

- (1) mov al, adt (4) mov cl, bdt
- (2) mov dl, [bx] (5) add al, adt + 1
- (3) sub [bx], 2

解答: 因为存储单元 adt 是以字节类型定义的,所以在存取该变量时,一次取出一个字节的数;存储单元 bdt 是以字类型定义的,所以在存取该变量时,一次取出一个字的数据。

(1) mov al, adt al 可以存放一个字节的数,adt 是以字节类型存储的,所以在该指令中,源操作数和目的操作数的类型一致,不需要添加 ptr。

(2) mov dl, [bx] 在该指令中,bx 存放的是操作数的地址,由于不知道该地址

是以字节的形式存放数据的还是以字的形式存放数据的,所以,该指令需要添加 ptr 以明确数据存取类型。应写为 `mov dl, byte ptr [bx]`。

(3) `sub [bx], 2` 该指令是从地址为 bx 的存储单元中取出数据,减去 2,然后再存入 bx 单元,立即数 2 可以自动转为 bx 单元中的数据类型,所以不需要添加 ptr。

(4) `mov cl, bdt` cl 是 cx 寄存器的低 8 位,是一个字节单元, bdt 是一个字单元,类型不一致,需要添加 ptr,使数据大小一致。所以指令应改为 `mov cx, bdt`。

(5) `add al, adt + 1` al 是 ax 寄存器的低 8 位,是一个字节单元, adt 是以字节类型的存储单元,数据类型一致,不需要添加 ptr。

习题 2

2-1 把一个 8 位的数据存入 AX 寄存器,如果只用 8 位的寄存器来表示,可以有几种表示方法? 如果放入 CX 寄存器,有几种表示方法?

解答: 在 AX 寄存器中,存储 8 位数据,可以存在 AL 或 AH 中; 在 CX 寄存器中,存储 8 位数据,可以存在 CL 或 CH 中; 所以,各有两种表示方法。

2-2 写出下列指令的物理地址,假设数据段 DS 的地址为 3000H,变量 VAL 的值为 0100, BX 寄存器的值为 1000H。

(1) `MOV AX, [2000]`

(2) `MOV AX, BX`

(3) `MOV AX, VAL`

解答: 在该题的 3 个指令中,默认的段地址都是存放在 DS 寄存器中,有效地址都在指令中直接计算。

(1) `MOV AX, [2000]` 物理地址 = 段地址 * 16 + 偏移地址 = 30000 + 2000 = 32000H。

(2) `MOV AX, BX` 物理地址 = 段地址 * 16 + 偏移地址 = 30000 + 1000 = 31000H。

(3) `MOV AX, VAL` 物理地址 = 段地址 * 16 + 偏移地址 = 30000 + 0100 = 30100H。

2-3 有如下的数据段定义,假设段地址为 13EB,偏移地址从 0 开始,写出存储单元的地址和内容。

```
a    DB    19, 2AH
b    DW    2 * 8 + 7
s1   DB    'abcd!'
s2   DB    '1234'
```

解答: 存储单元 a 以字节形式存放 2 个整数 19 和 2AH,先把 19 转换为十六进制 13H,存放在低地址中,再把 2AH 存放第一个字节中。

存储单元 b 以字类型定义,所以 $2 * 8 + 7 = 23$,转换为十六进制数是 17H,以字形式存储就是 0017H,所以,高字节存放 00H,低字节存放 17H;

存储单元 s1 以字节类型定义,每个字符占一个字节,在内存中存放该字符的 ASCII 值。字符 a 的 ASCII 值是 97,对应的十六进制数是 61H,其他以此类推。

存储单元 s2 以字节类型定义,每个字符占据一个字节,字符 '1' 的 ASCII 值是 31H,其他的以此类推。

具体内容和地址的对应关系如表 3-2 所示。

表 3-2 内存单元的内容和地址

内存单元内容	内存单元地址
13	13EB:0000
2A	13EB:0001
17	13EB:0002
00	13EB:0003
61 a	13EB:0004
62 b	13EB:0005
63 c	13EB:0006
64 d	13EB:0007
21 !	13EB:0008
31 1	13EB:0009
32 2	13EB:000A
33 3	13EB:000B
34 4	13EB:000C

习题 3

3-1 指出下列各种操作数的寻址方式。

- | | |
|--------------|--------------------------|
| (1) [BX] | (2) SI |
| (3) 435H | (4) [BP+DI+123] |
| (5) [23] | (6) data (data 是一个内存变量名) |
| (7) [DI+32] | (8) [BX+SI] |
| (9) [EAX+90] | (10) [BP+4] |

解答: (1) [BX] 寄存器间接寻址 (2) VAL[SI] 直接变址寻址

- | | |
|----------------------|------------------------------|
| (3) 435H 立即数寻址 | (4) [BP+DI+123] 相对基址变址寻址 |
| (5) [23] 直接寻址 | (6) data (data 是一个内存变量名)直接寻址 |
| (7) [DI+32] 寄存器相对寻址 | (8) [BX+SI] 基址变址寻址 |
| (9) [EAX+90] 寄存器相对寻址 | (10) [BP+4] 基址相对寻址 |

3-2 已知寄存器 BX、DI 和 BP 的值分别为 1234H、012F0H 和 42H,试分别计算下列各操作数的有效地址。

- | | |
|-------------|------------------|
| (1) [BX] | (2) [DI+123H] |
| (3) [BP+DI] | (4) [BX+DI+200H] |
| (5) [1234H] | (6) [BX*2+345H] |

解答：已知寄存器 BX、DI 和 BP 的值分别为 1234H、012F0H 和 42H。

- | | |
|------------------|-----------------------------|
| (1) [BX] | 有效地址 EA=1234H |
| (2) [DI+123H] | 有效地址 EA=12F0+123=1323H |
| (3) [BP+DI] | 有效地址 EA=42+12F0=1242H |
| (4) [BX+DI+200H] | 有效地址 EA=1234+12F0+200=2634H |
| (5) [1234H] | 有效地址 EA=1234H |
| (6) [BX*2+345H] | 有效地址 EA=1234*2+345=27ADH |

3-3 假定 DS=1123H, SS=1400H, BX=0200H, BP=1050H, DI=0400H, SI=0500H, LIST 的偏移量为 250H, 试确定下面各指令访问内存单元的地址。

- | | |
|-------------------------|---------------------------|
| (1) MOV AL, [1234H] | (2) MOV AX, [BX] |
| (3) MOV [DI], AL | (4) MOV [2000H], AL |
| (5) MOV AL, [BP+DI] | (6) MOV CX, [DI] |
| (7) MOV EDX, [BP] | (8) MOV LIST[SI], EDX |
| (9) MOV CL, LIST[BX+SI] | (10) MOV CH, [BX+SI] |
| (11) MOV EAX, [BP+200H] | (12) MOV AL, [BP+SI+200H] |
| (13) MOV AL, [SI-0100H] | (14) MOV BX, [BX+4] |

解答：

- | | |
|---|---|
| (1) MOV AL, [1234H]
PA = 11230 + 1234 = 12464H | (2) MOV AX, [BX]
PA = 11230 + 0200 = 11430H |
| (3) MOV [DI], AL
PA = 11230 + 0400 = 11630H | (4) MOV [2000H], AL
PA = 11230 + 2000 = 13230H |
| (5) MOV AL, [BP+DI]
PA = 11230 + 1050 + 0400 = 12680H | (6) MOV CX, [DI]
PA = 11230 + 0400 = 11630H |
| (7) MOV EDX, [BP]
PA = 11230 + 1050 = 12280H | (8) MOV LIST[SI], EDX
PA = 11230 + 250 + 0500 = 11980H |
| (9) MOV CL, LIST[BX+SI]
PA = 11230 + 250 + 0200 + 0500
= 11B80H | (10) MOV CH, [BX+SI]
PA = 11230 + 0200 + 0500 = 11930H |
| (11) MOV EAX, [BP+200H]
PA = 11230 + 1050 + 200 = 12480H | (12) MOV AL, [BP+SI+200H]
PA = 11230 + 1050 + 0500 + 200
= 12980H |
| (13) MOV AL, [SI-0100H]
PA = 11230 + 0500 + 0100 = 11830 | (14) MOV BX, [BX+4]
PA = 11230 + 0200 + 4 = 11434H |

3-4 按下列要求编写指令序列。

- (1) 清除 DH 中的最低 3 位而不改变其他位,结果存入 BH 中。
- (2) 把 DI 中的最高 5 位置 1 而不改变其他位。
- (3) 把 AX 中的 0~3 位置 1,7~9 位取反,13~15 位置 0。
- (4) 检查 BX 中的第 2、第 5 和第 9 位中是否有一位为 1。
- (5) 检查 CX 中的第 1、第 6 和第 11 位中是否同时为 1。
- (6) 检查 AX 中的第 0、第 2、第 9 和第 13 位中是否有一位为 0。
- (7) 检查 DX 中的第 1、第 4、第 11 和第 14 位中是否同时为 0。

解答: (1) AND DH,1000B MOV BH,DH

(2) OR DI,1111 1000 B

(3) OR AX,0001H XOR AX,0380H(011 1000 0000B) AND AX,1FFFH

(4) TEST BX, 0224H (01000100100B) JNZ HAVE1 (如果 ZF=0 则表示至少有一位为 1)

(5) AND CX,0842H(0100001000010B) CMP CX,0842H JZ ALLIS1(如果与运算后结果为 0842,说明第 1、第 6 和第 11 位中同时为 1)

(6) AND AX,2205H(0010001000000101B) CMP AX,2205H JNZ HAVE0(如果与运算后 AX 为 2205H,说明第 0、第 2、第 9 和第 13 位全部为 1; 如果不等于 2205H,则说明至少一位为 0)

(7) TEST DX,4812H(00100100000010010B) JZ HAVE0(测试运算即与以后,如果 ZF 为 1 说明测试结果为 0,也即所有这些测试位第 1、第 4、第 11 和第 14 位中同时为 0)

TEST 指令常用于测试操作数中某位是否为 1,而且不会影响目的操作数。如果测试某位的状态,对某位进行逻辑与 1 的运算,其他位逻辑与 0,然后判断标志位。运算结果为 0,ZF=1,表示被测试位为 0; 否则 ZF=0,表示被测试位为 1。

TEST 指令影响的标志位为 SF、ZF、PF,并且使 OF=CF=0。

3-5 选择适当的指令实现下列功能。

- (1) 右移 DI 3 位,并把 0 移入最高位。
- (2) 把 AL 左移一位,使 0 移入最低一位。
- (3) AL 循环左移 3 位。
- (4) EDX 带进位位循环右移 4 位。
- (5) DX 右移 6 位,且移位前后的正负性质不变。

解答:

(1) 右移 DI 3 位,并把 0 移入最高位 MOV CL,3 SHR DI,CL

(2) 把 AL 左移一位,使 0 移入最低一位 SAL AL,1

(3) AL 循环左移 3 位 MOV CL,3 ROL AL,CL

(4) EDX 带进位位循环右移 4 位 MOV CL,4 RCR EDX,CL

(5) DX 右移 6 位,且移位前后的正负性质不变 MOV CL,6 SAR DX,CL

3-6 假设(DS)=2000H,(BX)=0100H,(SI)=0002H,(20100)=12H,(20101)=34H,(20102)=56H,(20103)=78H,(21200)=2AH,(21201)=4CH,(21202)=B7H,(21203)=65H,试写出下列各指令执行完后 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]

解答:

(1) MOV AX,1200H (AX)=1200H

(2) MOV AX,BX (AX)=0100H

(3) MOV AX,[1200H] PA=20000+1200=21200 (AX)=4C2AH

(4) MOV AX,[BX] PA=20000+0100=20100(AX)=3412H

(5) MOV AX,1100[BX] PA=20000+1100+0100=21200H (AX)=4C2AH

(6) MOV AX,[BX][SI] PA=20000+0100+0002=20102 (AX)=7856H

(7) MOV AX,1100[BX][SI] PA=20000+1100+0100+0002=21202H(AX)
=65B7H

3-7 变量 datax 和 datay 的定义如下:

```
datax dw 0148h
```

```
      dw 2316h
```

```
datay dw 0237h
```

```
      dw 4052h
```

请按下列要求写出指令序列:

(1) datax 和 datay 两个字数据相加,和存放在 datay 单元中。

(2) datax 和 datay 两个双字数据相加,和存放在 datay 开始的字单元中。

(3) datax 和 datay 两个字数据相乘,积存放在 datay 开始的单元中。

(4) datax 和 datay 两个双字数据相加,积存放在 datay 开始的单元中。

(5) datax 和 datay 两个字数据相除。

(6) datax 双字和 datay 字数据相除。

解答:

(1) mov ax,datax

(2) mov eax, datax

mov dx,datax + 2

add datay, eax

add datay, ax

adc datay + 2 , dx

- | | |
|--|--|
| (3) <code>mov ax, datax</code>
<code>mov dx, datax + 2</code>
<code>imul ax, datay</code>
<code>imul datay + 2, dx</code>
<code>mov datay, ax</code> | (4) <code>mov eax, datax</code>
<code>imul datay, eax</code> |
| (5) <code>mov ax, datax</code>
<code>mov dx, datax + 2</code>
<code>div ax, datay</code>
<code>div dx, datay + 2</code> | (6) <code>mov eax, datax</code>
<code>mov edx, 0</code>
<code>div datay</code> |

3-8 假定 $(DX)=0B9H$, $(CL)=3$, $(CF)=1$, 下列各指令单独执行后 DX 的值是多少?

- | | |
|-----------------------------|-----------------------------|
| (1) <code>SHR DX, 1</code> | (2) <code>SAR DX, CL</code> |
| (3) <code>SHL DX, 1</code> | (4) <code>ROR DX, CL</code> |
| (5) <code>ROL DX, CL</code> | (6) <code>RCL DX, CL</code> |

解答:

- | | |
|--|--|
| (1) <code>SHR DX, 1</code> $(DX)=5CH$ | (2) <code>SAR DX, CL</code> $(DX)=17H$ |
| (3) <code>SHL DX, 1</code> $(DX)=0172H$ | (4) <code>ROR DX, CL</code> $(DX)=2017H$ |
| (5) <code>ROL DX, CL</code> $(DX)=05C8H$ | (6) <code>RCL DX, CL</code> $(DX)=05C8H$ |

3-9 假定 AX 和 BX 中的内容为带符号数, CX 和 DX 中的内容为无符号数, 请用比较指令和条件转移指令实现以下功能。

- (1) 若 DX 的内容超过 CX 的内容, 则转到 $J1$ 处执行。
- (2) 若 BX 的内容大于 AX 的内容, 则转到 $J2$ 处执行。
- (3) 若 CX 的内容等于 0, 则转到 $J3$ 处执行。
- (4) 若 BX 的内容小于 AX 的内容, 则转到 $J4$ 处执行。
- (5) 若 DX 的内容低于 CX 的内容, 则转到 $J5$ 处执行。

解答:

- | | | |
|---|-------------------------|--------------------|
| (1) 若 DX 的内容超过 CX 的内容, 则转到 $J1$ 处执行 | <code>cmp dx, cx</code> | <code>ja J1</code> |
| (2) 若 BX 的内容大于 AX 的内容, 则转到 $J2$ 处执行 | <code>cmp bx, ax</code> | <code>jg J2</code> |
| (3) 若 CX 的内容等于 0, 则转到 $J3$ 处执行 | <code>cmp cx, 0</code> | <code>je J3</code> |
| (4) 若 BX 的内容小于 AX 的内容, 则转到 $J4$ 处执行 | <code>cmp bx, ax</code> | <code>jl J4</code> |
| (5) 若 DX 的内容低于 CX 的内容, 则转到 $J5$ 处执行 | <code>cmp dx, cx</code> | <code>jb J5</code> |

习题 4

答案在 sourceasm 文件夹中, 源程序依次为 test41.asm、test42.asm、test43.asm、test44.asm、test45.asm、test46.asm、test47.asm。

4-1 从键盘输入一个小写字母,把它转换成大写字母并输出。

```
;test41.asm
dataarea segment
    mess1    db  'Please input letter:',13,10,'$ '
    mess2    db  'Please input again:',13,10,'$ '
dataarea ends
program segment
main    proc    far
        assume cs:program,ds:dataarea
start:
        push    ds
        sub     ax,ax
        push    ax
        mov     ax,dataarea
        mov     ds,ax
        lea     dx,mess1
        mov     ah,09
        int     21h
input:
        mov     ah,07
        int     21h
        cmp     al,61h
        jb      again
        cmp     al,7Ah
        ja      again
        sub     al,20h
        mov     dl,al
        mov     ah,02
        int     21h
        jmp     short exit
again:
        lea     dx,mess2
        mov     ah,09
        int     21h
        jmp     short input
exit:    ret
main    endp
program ends
        end     start
```

4-2 数据区中存有一个字符串,从键盘输入一个字母,找出它在字符串中的前导和后续字母,并输出这些字母。