

指令系统作业一答案

本节其余的习题使用下面的变量定义:

```
.data
var1 SBYTE -4,-2,3,1
var2 WORD 1000h,2000h,3000h,4000h
var3 SWORD -16,-42
var4 DWORD 1,2,3,4,5
```

7. 对于下面的每条语句,指明其是否有效:

- a. `mov ax,var1`
- b. `mov ax,var2`
- c. `mov eax,var3`
- d. `mov var2,var3`
- e. `movzx ax,var2`
- f. `movzx var2,al`
- g. `mov ds,ax`
- h. `mov ds,1000h`

8. 下列每条指令按顺序执行后,目的操作数的十六进制数值是什么?

```
mov    al,var1                ; a.
mov    ah,[var1+3]            ; b.
```

9. 下列每条指令按顺序执行后,目的操作数的值是什么?

```
mov    ax,var2                ; a.
mov    ax,[var2+4]            ; b.
mov    ax,var3                ; c.
mov    ax,[var3-2]            ; d.
```

10. 下列每条指令按顺序执行后,目的操作数的值是什么?

```
mov    edx,var4               ; a.
movzx  edx,var2               ; b.
mov    edx,[var4+4]           ; c.
movsx  edx,var1               ; d.
```

7

- (a) 无效 两个操作数必须是同样大小
- (b) 有效
- (c) 无效 理由同 a
- (d) 无效 两个操作数不能同时为内存操作数
- (e) 无效 目的操作数需要比原操作数大
- (f) 无效 目的操作数必须是寄存器
- (g) 有效
- (h) 无效 不能将立即数直接移动到段寄存器

8

(a) FCh

1 Byte = 8 bit. -4 (SByte) = 1000 0100。存储用补码，所以实际上为1111 1100 = FCh

(b) 01h

请按规范书写，尽量不要直接写1

9

(a) 1000h

(b) 3000h

x86汇编数组里的+1不代表索引+1，而是偏移一个字节，所以对于WORD数组，+4表示index+2

(c) FFF0h

原理同 8(a)

(d) 4000h

-2在WORD数组表示index-1。做实验的过程中可以看到你生命的data seg里的数据都是紧邻的，所以var3-2实际上已经指向了var2

10

(a) 00000001h

(b) 00001000h

(c) 00000002h

(d) FFFFFFFCh

下面几个问题使用以下数据：

```
.data
val1 BYTE 10h
val2 WORD 8000h
val3 DWORD 0FFFFh
val4 WORD 7FFFh
```

1. 写一条 val2 加 1 的指令。
2. 写一条从 EAX 中减掉 val3 的指令。
3. 写一条从 val2 中减去 val4 的指令。
4. 如果使用 ADD 指令对 val2 加 1，那么进位标志和符号标志的值分别是什么？
5. 如果使用 ADD 指令对 val4 加 1，那么溢出标志和符号标志的值分别是什么？
6. 在每条指令执行后，在提示处写下进位标志、符号标志、零标志和溢出标志的值：

```
mov ax,7FF0h
add al,10h           ; a. CF =      SF =      ZF =      OF =
add ah,1             ; b. CF =      SF =      ZF =      OF =
add ax,2             ; c. CF =      SF =      ZF =      OF =
```

7. 用汇编语言实现下面的表达式：AX = (-val2 + BX) - val4。
8. (是/否)：正整数和负整数相加的时候是否有可能设置溢出标志位？
9. (是/否)：负整数和负整数相加产生正整数时是否会设置溢出标志位？
10. (是/否)：NEG 指令是否可能设置溢出标志位？
11. (是/否)：符号标志和零标志是否可能同时被设置？

1. `inc val2`

2. `sub eax, val3`

3.

```
1 | mov ax,val4
2 | sub val2,ax
```

4. CF=0, SF=1

符号位为1，表示负数SF=1，无进位CF=0。

5. OF=1, SF=1

相加后符号位为1，表示负数。相加后相当于是正数+正数=负数，溢出

6.

1. (a) CF=1, SF=0, ZF=1, OF=0

有进位CF=1。符号位为0，SF=0。结果全0，ZF=0。负数+正数=0，无溢出

2. (b) CF=0, SF=1, ZF=0, OF=1

无进位，符号位为1，不为0，正数+正数=负数，溢出

3. (c) CF=0, SF=1, ZF=0, OF=0

(a)中al=00，(b)中ah=80，所以计算是8000h+2。无进位，是负数，非0，负数+正数=负数，无溢出。

7.

```
1 mov ax, bx
2 sub ax, val2
3 sub ax, val4
4 ;(请不要更改 bx)
```

8. 否

溢出产生条件

两个正数相加，如果结果为负数，就产生了溢出。

两个负数相加，如果结果为正数，就产生了溢出。

9. 是

10. 是

```
1 mov al, -128
2 neg al
```

neg相当于乘-1。众所周知只有-128-127。

11. 否