

汇编语言程序设计课程作业（十一）

姓名：袁昊男 学号：2018091618008

检测点 11.1

写出下面每条指令执行后，ZF、PF、SF 等标志位的值。

sub al,al ZF= 1 PF= 1 SF= 0

分析：结果为 0，1 的个数为偶数，非负。

mov al,1 ZF= 1 PF= 1 SF= 0

分析：mov 指令不影响标志位的值。

push ax ZF= 1 PF= 1 SF= 0

分析：push 指令不影响标志位的值。

pop bx ZF= 1 PF= 1 SF= 0

分析：pop 指令不影响标志位的值。

add al,bl ZF= 0 PF= 0 SF= 0

分析：结果为 00000010B，不为 0，1 的个数为奇数，非负。

add al,10 ZF= 0 PF= 1 SF= 0

分析：结果为 00001100B，不为 0，1 的个数为偶数，非负。

mul al ZF= 0 PF= 1 SF= 0

分析：结果为 10010000B，不为 0，1 的个数为偶数，非负。

检测点 11.2

写出下面每条指令执行后，ZF、PF、SF、CF、OF 等标志位的值。

	CF	OF	SF	ZF	PF
sub al,al	0	0	0	1	1

分析：结果为 0，无进位，无溢出，非负，1 的个数为偶数。

mov al,10H	0	0	0	1	1
------------	---	---	---	---	---

分析：mov 指令不影响标志位的值。

add al,90H	0	0	1	0	1
------------	---	---	---	---	---

分析：结果为 0A0H，无进位，无溢出，结果为负，1 的个数为偶数。

mov al,80H	0	0	1	0	1
------------	---	---	---	---	---

分析：mov 指令不影响标志位的值。

add al,80H	1	1	0	1	1
------------	---	---	---	---	---

分析：结果为(1)00H，有进位，有溢出，结果为非负，1 的个数为偶数。

mov al,0FCH	1	1	0	1	1
-------------	---	---	---	---	---

分析：mov 指令不影响标志位的值。

add al,05H	1	0	0	0	0
------------	---	---	---	---	---

分析：结果为(1)01H，有进位，无溢出，结果为非负，1 的个数为奇数。

mov al,7DH 1 0 0 0 0

分析：mov 指令不影响标志位的值。

add al,0BH 0 1 1 0 1

分析：结果为 88H，无进位，有溢出，结果为负，1 的个数为偶数。

检测点 11.3

(1) 补全下面的程序，统计 F000:0 处 32 个字节中，大小在[32,128]的数据的个数。

```
mov ax,0f000h
mov ds,ax

mov bx,0
mov dx,0
mov cx,32

s: mov al,[bx]
   cmp al,32
   jb s0
   cmp al,128
   ja s0
   inc dx
s0: inc bx
   loop s
```

分析：首先 cmp al,32 将 al 与 32 比较，因为要统计大于等于 32 的数据，因此如果数据小于 32，则跳过此数据，故使用 jb 指令（低于则转移）；再 cmp al,128 将 al 与 128 比较，因为要统计小于等于 128 的数据，因此如果数据大于 128，则跳过此数据，故使用 ja 指令（高于则转移）。

(2) 补全下面的程序，统计 F000:0 处 32 个字节中，大小在(32,128)的数据的个数。

```
mov ax,0f000h
mov ds,ax

mov bx,0
mov dx,0
mov cx,32

s: mov al,[bx]
   cmp al,32
   jna s0
   cmp al,128
   jnb s0
   inc dx
s0: inc bx
   loop s
```

分析：首先 cmp al,32 将 al 与 32 比较，因为要统计大于 32 的数据，因此如果数据小于等于 32，则跳过此数据，故使用 jna 指令（不高于则转移）；再 cmp al,128 将 al 与 128 比较，

因为要统计小于 128 的数据，因此如果数据大于等于 128，则跳过此数据，故使用 `jnb` 指令（不低于则转移）。

检测点 11.4

下面的程序执行后：(ax)=?

```
mov ax,0
push ax
popf
mov ax,0fff0h
add ax,0010h
pushf
pop ax
and al,11000101B
and al,00001000B
```

分析：`popf` 将栈中数据弹出至 PSW 寄存器，其中的值为 0000 0000 0000 0000B。`add` 指令后操作结果为 0000 0010h。有进位，CF=1；1 的个数为偶数，PF=1；结果为 0，ZF=1；结果非负，SF=1；无溢出，OF=0。因此 PSW 中的值在执行 `add` 指令后变为 0000 00__ 010__ 0101B（其中横线部分未知），后执行指令 `pushf` 和 `pop`，将 PSW 中的值传送到 ax 中，再进行 `and` 操作（刚好消除了未知位的影响），结果为 0000 0000 0100 0101B，即 45h。