

# 上机作业十

姓 名	学 号	日 期
刘源	201611210134	2018.12.13

## 实验目的

- 理解状态寄存器的读写方法。
- 理解ZF、CF、PF、SF、OF状态位的意义。
- 掌握联合DF位于movsb、movsw进行内存串搬移数据的搬移。

## 实验总结：

第一题的思考了许久不知道该如何计算CF、OF、PF的值，后来才发现也可以和ZF一样找到规律，但实现代码还是很艰难。第二题在做的时候，第一次使用的sub 20H的语句，但其中有几个地方的小写字母变成了"..", 但换成and 11011111B后就都输出正确了，猜想可能是进位或溢出的问题。

## 第 1 题 实现128位整数加法-回顾

讨论：

- 如果是无符号数，如何知道加法是否有进位？
  - 观察CF的值，若CF = 1，则加法产生进位。
- 如果是有符号数，如何确定最终结果的符号位和是否溢出？
  - 观察OF的值，若OF = 1，则溢出，且符号位由之前的符号决定。
- 如何确定结果是否等于0？
  - 若四次计算的ZF与完仍然为1，则最后结果为零。
- 如何在完成计算后，让状态寄存器的ZF, CF, OF, SF, PF表示对应的结果？
  - $ZF = ZF<1> \& ZF<2> \& ZF<3> \& ZF<4>$
  - $CF = CF<4>$
  - $OF = OF<4>$
  - $SF = SF<4>$
  - $PF = PF<1> \oplus PF<2> \oplus PF<3> \oplus PF<4>$

## 第 2 题

编写一个子程序letter，将以0（不是字符'0'）结尾的字符串中小写字母转换为大写字母。

### 源代码

```
assume cs:code,ds:data

data segment
    db 'Beginners All-purpose Symbolic Instruction Code.', 0
data ends

code segment
start:
    mov ax, data
    mov ds, ax
    mov si, 0
    call letterc

    mov ax, 4c00h
    int 21h
;子程序letterc
letterc:
    push ax
    push cx
s:
    mov cl, ds:[si]
    mov ch, 0

    jcxz ok      ;如果cx==0,则结束循环.判断是否为字符串末尾

    cmp cl, 61h
    jb check_out
    cmp cl, 91h
    ja check_out

    mov al, cl
    and al, 11011111b
    mov cl, al
    mov ds:[si], cl

check_out:
    inc si
    jmp short s

ok:
    pop cx
    pop ax
    ret
code ends
end start
```

### 第一次运行

```

D:\ASM>debug 11.exe
-T

AX=076C  BX=0000  CX=00AC  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075C  ES=075C  SS=076B  CS=0770  IP=0003  NV UP EI PL NZ NA PO NC
0770:0003 8ED8          MOV     DS,AX

```

#### 运行结果

```

-D 076C:0000
076C:0000  42 45 47 49 4E 4E 45 52-53 20 41 4C 4C 2D 50 55  BEGINNERS ALL-PU
076C:0010  52 50 4F 53 45 20 53 59-4D 42 4F 4C 49 43 20 49  RPOSE SYMBOLIC I
076C:0020  4E 53 54 52 55 43 54 49-4F 4E 20 43 4F 44 45 2E  NSTRUCTION CODE.
076C:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
076C:0040  B8 6C 07 8E D8 BE 00 00-E8 05 00 B8 00 4C CD 21  .l.....L.?
076C:0050  50 51 8A 0C B5 00 E3 15-80 F9 61 72 0D 80 F9 91  PQ.....ar....
076C:0060  77 08 8A C1 24 DF 8A C8-88 0C 46 EB E5 59 58 C3  w...$.F..YX.
076C:0070  FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86  .....+.P..
- ;

```