## 第八周作业

## 3.34

- (1) AX, BX相加结果为9557H, 没有溢出, 转向L1
- (2) AX,BX相加结果为0F830H,没有溢出,转向L1
- (3) AX,BX相加结果为0A355H,两个正数相加得负数,溢出,没有进位,故转到L2
- (4) AX,BX相加结果为6FF3H,两个负数相加得正数,溢出,有进位,相加后,AX,BX相减结果为0B023H,有借位,正数减负数得负数,有溢出,故转向L5
- (5) AX,BX相加结果为05A1FH,两个负数相加得正数,溢出,有进位,相加后,AX,BX相减结果为94B7H,有借位,正数减负数得负数,有溢出,故转向L5

## 3.36

首先了解ADC为带进位加法,首先用不带进位的加法给AX赋值2\*p,然后用带进位的加法给DX赋值2\*p,比较带进位加法的结果2\*p和q两个数的大小关系,如果2\*p小于q,则AX赋值为2,如果2\*p小于等于q,则AX赋值为1,

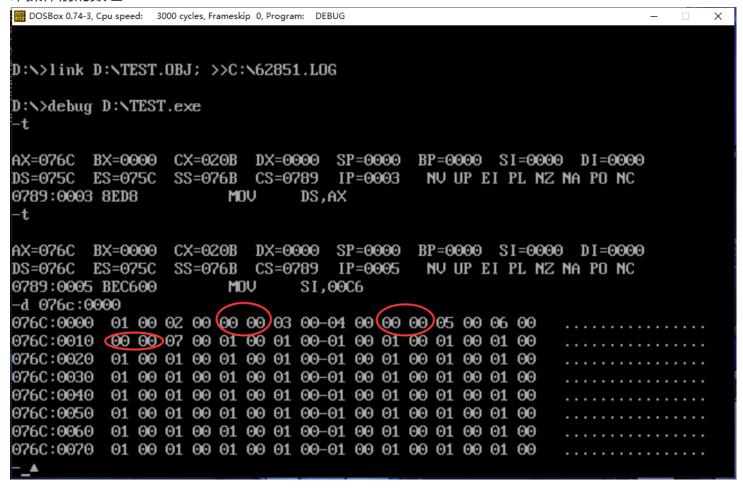
```
data segment
    MEM
              dw 1,2,0,3,4,0,5,6,0,7
              dw 90 dup(1)
    changerow db 0dh,0ah,'$'
    cnt
              dw 0
data ends
stack segment
          dw 128 dup(0)
stack ends
code segment
          assume cs:code,ds:data,ss:stack
    start:
                 ax,data
         mov
                 ds,ax
         mov
         mov
                 si,100*2-2
         mov
                 bx,-2
         mov
                 cx,100
    loop1:
                 bx,2
          add
                MEM[bx],0
          cmp
                 deal
          je
          loop
                 loop1
          jmp
                 exit
    deal:
         mov
                 di,bx
    ;开始向前移动
    loop2:
    ;看是否到了最后一个元素
                 di,si
          cmp
                 add_0
          je
                 ax,MEM[di+2]
         mov
                MEM[di],ax
         mov
                 di,2
          add
          jmp
                 loop2
    add_0:
         mov
                 word ptr MEM[si],0
                 loop1
          loop
```

exit:

mov ah,4ch int 21h

code ends end start

## 未操作前的数组



操作后的数组

