中山大學

微机实验

题　　目：	作业3
上课时间：	第16-17周
授课教师：	何涛
姓　　名：	周德峰
学　　号：	21312210
日　　期：	2023-6-10

题一

设数据段中有 10 个字节的无符号数(每个数 8 位)。要求统计这些 数中高电平(即 1)与低电平(即 0)的个数。将数据段中所有数高电平的 个数和与低电平的个数和分别存入 3000H 与 3100H 中。

要求：使用子程序完成 (样例: 51H, 3AH, 95H, 8DH, 90H, 0A7H, 0C1H, 77H, 24H, 0B1H)

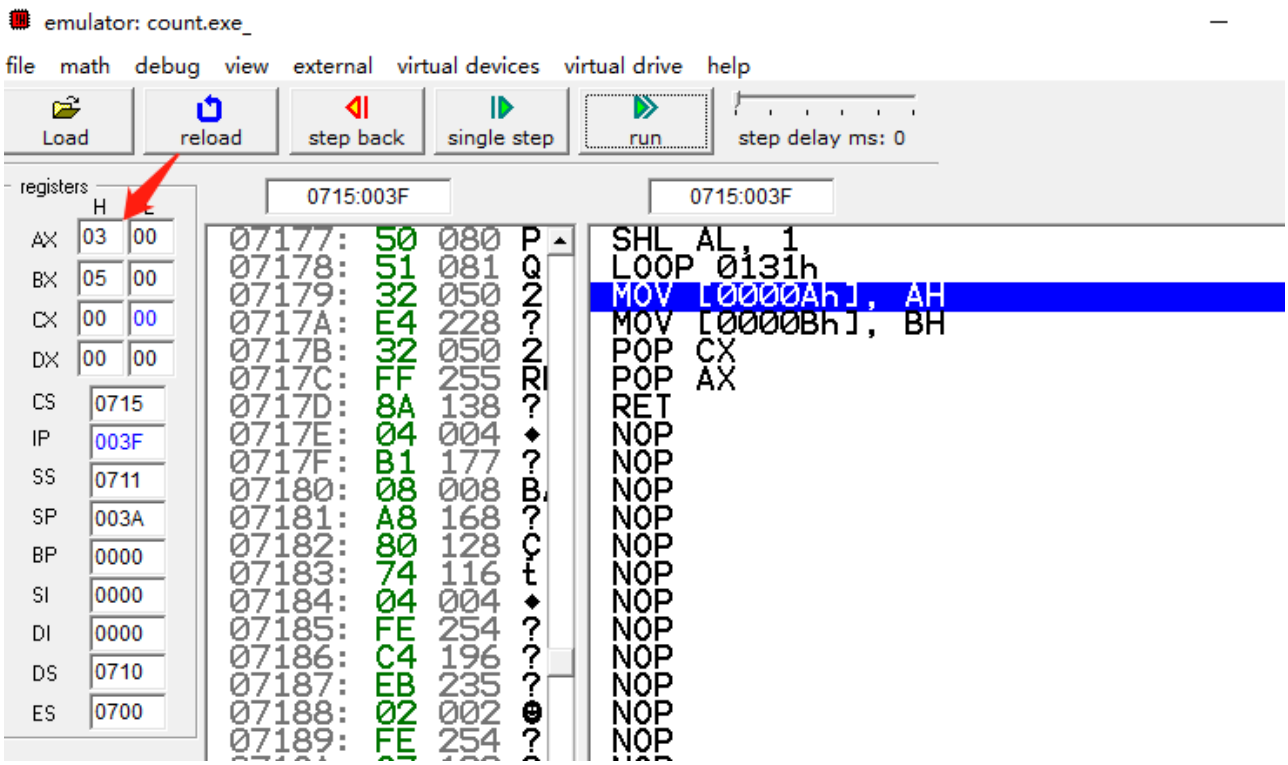
1 实验思路

在主程序中循环10次，每次将一个字节放入子程序中，计算其对应的高电平数与低电平数，然后将其累加到AH和AL中，AH存放高电平，AL存放低电平

2 实验结果

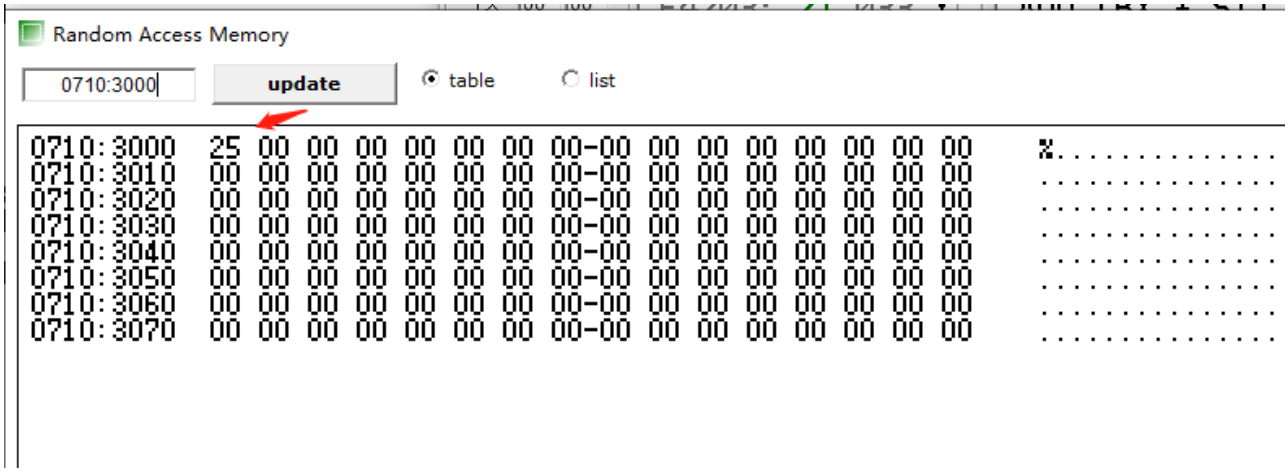
验证第一个数

当第一次循环时，将51H（0101 0001B）送入子程序中

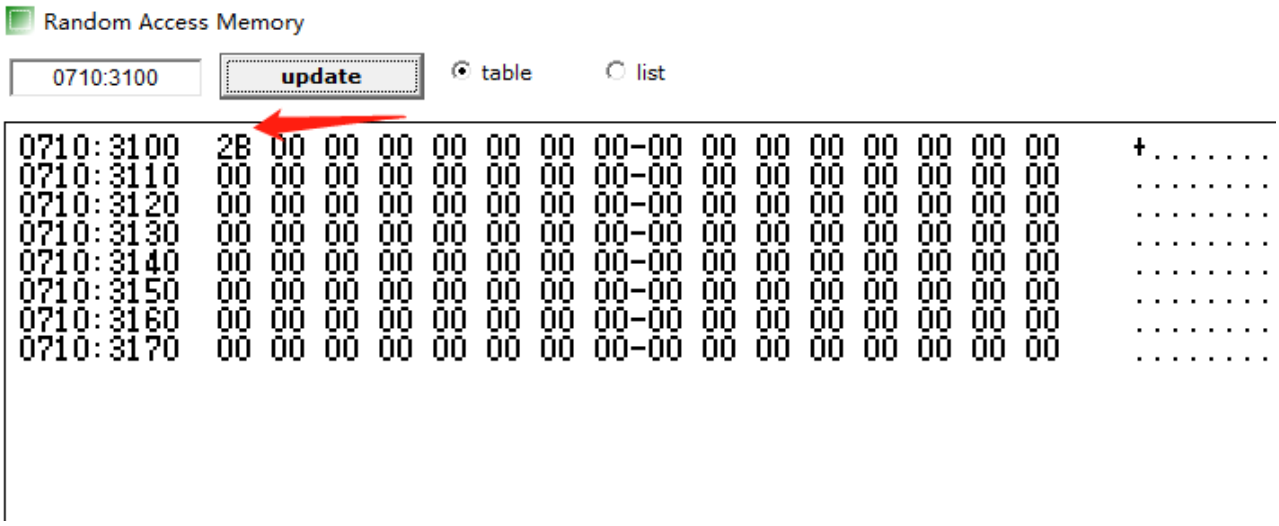


当CX=0时，对应的AH=3，AL=5，符号结果

对应高电平个数为2*16+5=37个



对应低电平个数为2*16+11=43个



总和刚好为80个，进一步验证程序的正确性

count.asm源码：

```
DATA SEGMENT
    X DB 51H,3AH,95H,8DH,90H,0A7H,0C1H,77H,24H,0B1H
    COUNT_HIGH DB 0
    COUNT_LOW DB 0
DATA ENDS

STACK SEGMENT PARA  STACK
    DW 20H DUP(0)

STACK ENDS


CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START: MOV AX, DATA
       MOV DS, AX
```

```
        XOR AH,AH    ;CLEAR
        XOR AL,AL
        MOV CL,10          ; LOOP 10 TIMES
        LEA SI,X
COUNT:
        CALL COUNT_BITS    ;调用子程序
        ADD AH,[COUNT_HIGH]

        ADD AL,[COUNT_LOW]
        INC SI
        LOOP COUNT
        MOV [3000H],AH
        MOV [3100H],AL
        MOV AH,4CH
        INT 21H
COUNT_BITS PROC
        PUSH AX
        PUSH CX
        XOR AH, AH      ; 将AH寄存器清零，用于存储高电平计数
        XOR BH, BH      ; 将BH寄存器清零，用于存储低电平计数
        MOV AL, [SI]    ; 读取数据段中的字节
        MOV CL,8
SHL_LOOP:
    TEST AL, 80H    ; 检查字节的最高位是否为1
    JZ LOW_BIT      ; 如果最高位为0，跳转到LOW_BIT标签

HIGH_BIT:
    INC AH          ; 如果最高位为1，增加高电平计数
    JMP NEXT            ; 跳转到NEXT标签

LOW_BIT:
    INC BH          ; 如果最高位为0，增加低电平计数

NEXT:
    SHL AL, 1
    LOOP SHL_LOOP

BIT_END:
    ;INC SI          ; 增加SI寄存器的值，指向下一个字节
    ;LOOP COUNT_LOOP  ; 循环COUNT_LOOP标签，直到CX寄存器为0
    MOV [COUNT_HIGH], AH    ; 将高电平计数存储到COUNT_HIGH变量
    MOV [COUNT_LOW], BH     ; 将低电平计数存储到COUNT_LOW变量
    POP CX
    POP AX
    RET

```

```
63  COUNT_BITS ENDP
64
65  CODE ENDS
66  END START
```

题二

设数据段中初始存放了 10 个字节的无符号数，要求使用冒泡排序法 将这些数从小到大排序。排序完成后放在以 3000H 为初始单元的地 址中。

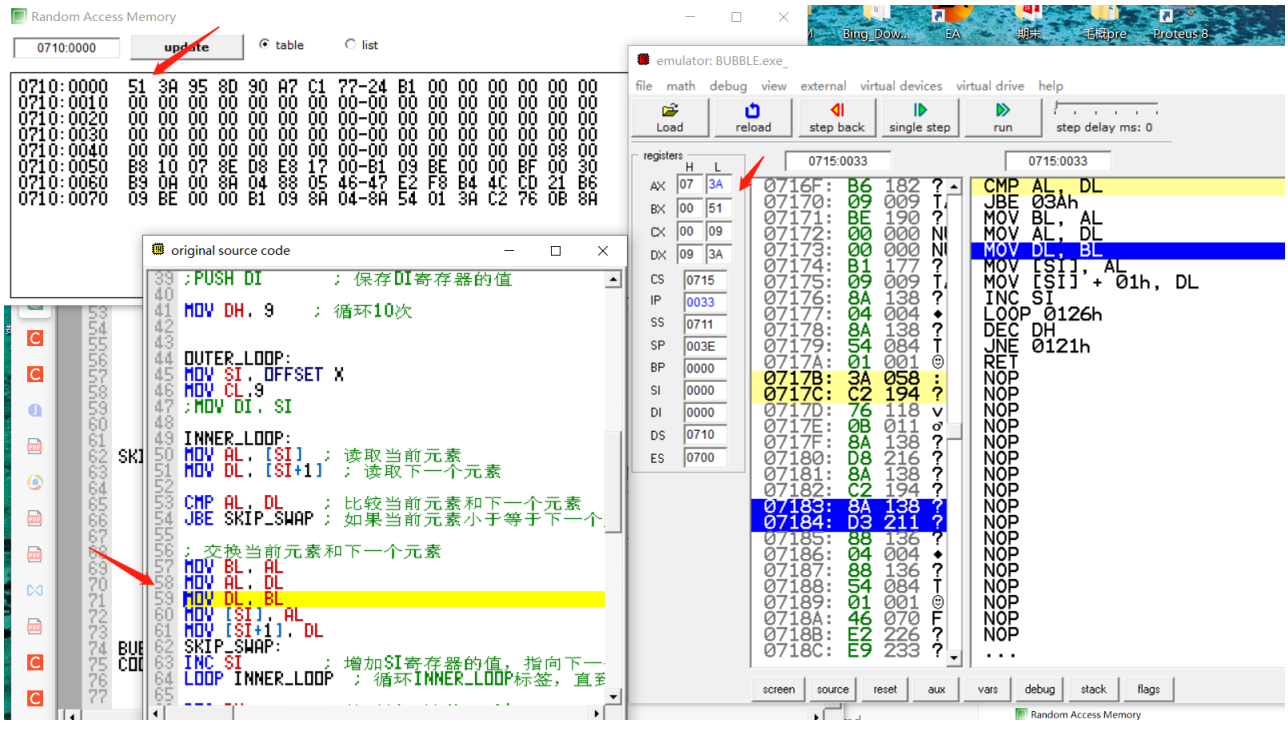
(样例: 51H, 3AH, 95H, 8DH, 90H, 0A7H, 0C1H, 77H, 24H, 0B1H)

1　实验思路

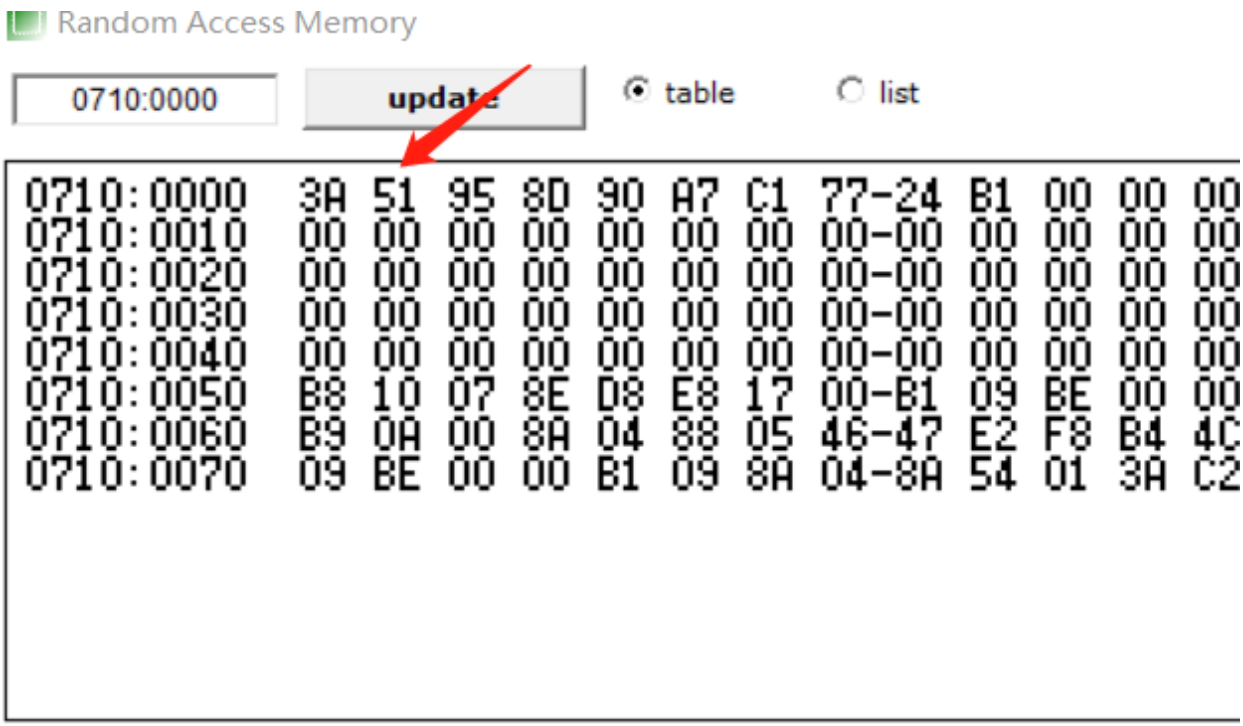
将数据段放在DS：0000处，在子程序段中进行初步冒泡排序，然后再将所有数据迁移到对应地址中

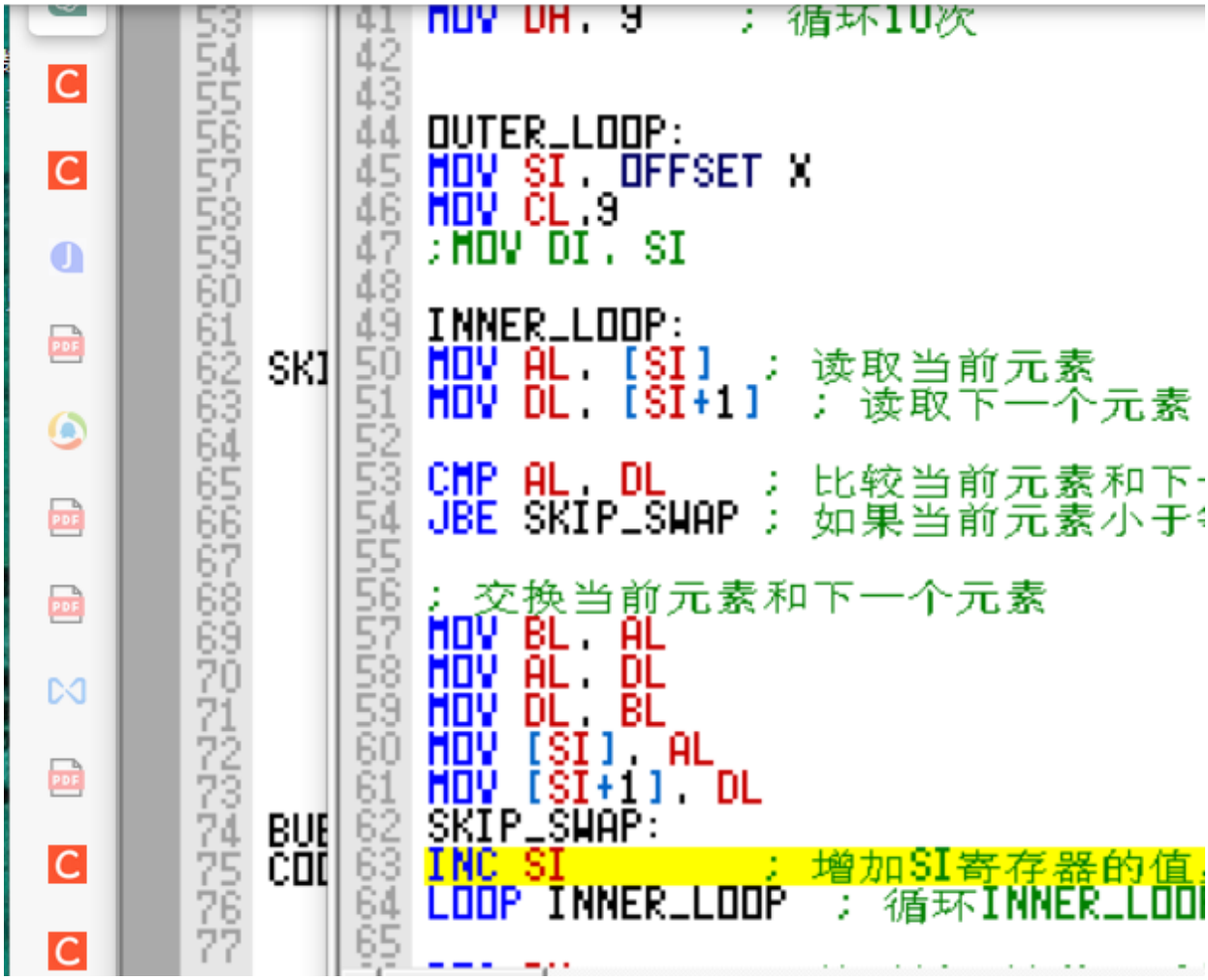
2　实验结果

判断第一个数51H和3AH时，3AH小于51H，因此进行交换，将BL作为数据中转寄存器

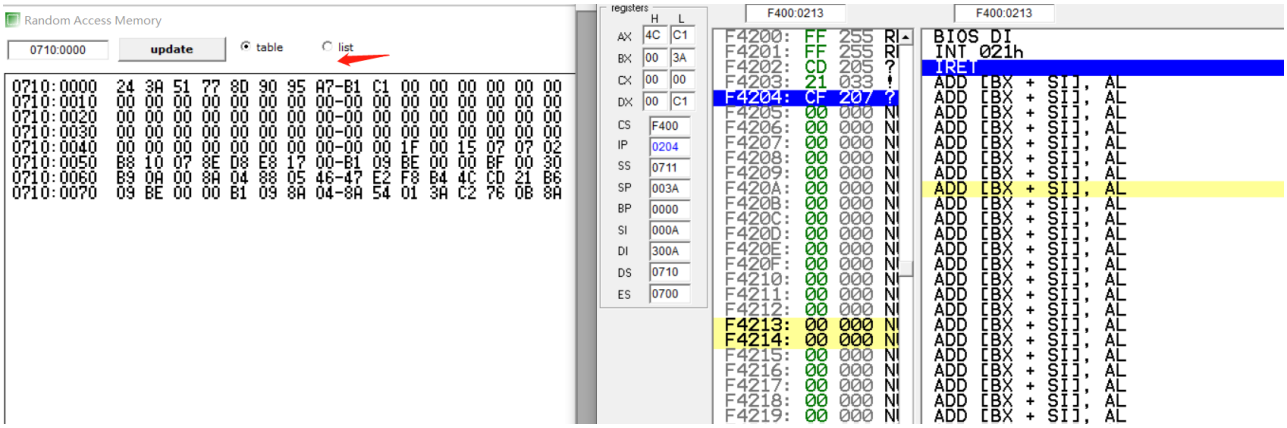


可以看到，交换后，3AH排到了51H前

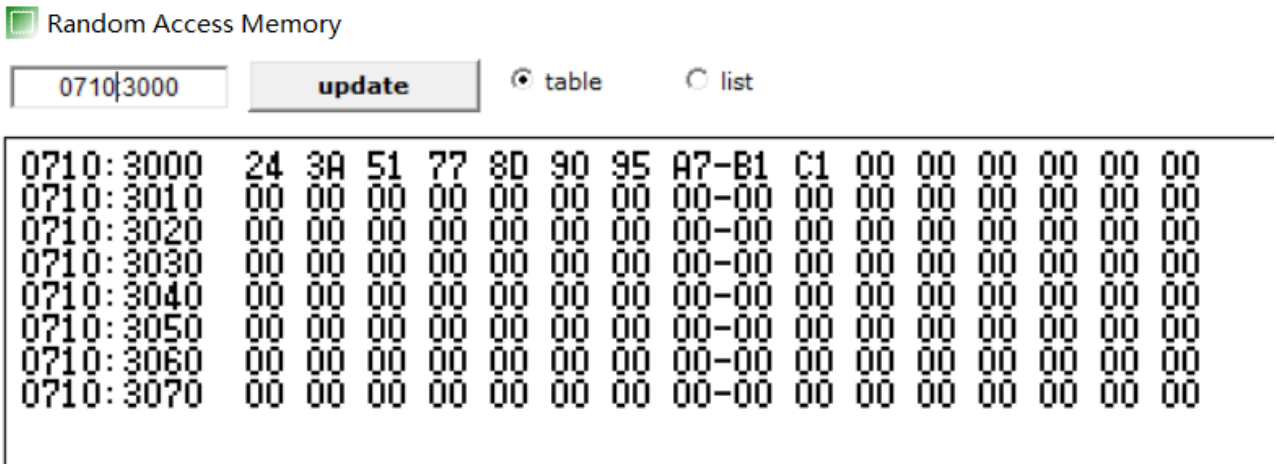




可以看到，当冒泡排序结束后，对应初地址的结果已经按照从小到大的顺序排好



对应偏移地址也已经成功排好序



Note：本文比较采用无符号比较，因为对于16进制而言，无符号更容易判断程序是否正确，若要进行有符号数的冒泡排序，只需将CODE中第54行的比较结果跳转更换成`JLE`

BUBBLE.asm源码：

```
1  DATA SEGMENT
2      X DB 51H, 3AH, 95H, 8DH, 90H, 0A7H, 0C1H, 77H, 24H, 0B1H
3  DATA ENDS
4
5  STACK SEGMENT PARA  STACK
6      DW 20H DUP(0)
7  STACK ENDS
8
9
10 CODE SEGMENT
11     ASSUME DS:DATA, CS:CODE
12 START: MOV AX,DATA
13        MOV DS,AX
14        ;LEA SI,X              ; 将SI寄存器指向数据段的起始地址
15
16        CALL BUBBLE_SORT      ; 调用子程序BUBBLE_SORT进行冒泡排序
17        MOV CL,9
```

```
        MOV SI, OFFSET X
        MOV DI, 3000H
        MOV CX, 10    ; 循环10次

COPY_LOOP:
        MOV AL, [SI]  ; 读取当前元素
        MOV [DI], AL  ; 将当前元素存储到目标地址

        INC SI        ; 增加SI寄存器的值，指向下一个元素
        INC DI        ; 增加DI寄存器的值，指向下一个目标地址
        LOOP COPY_LOOP  ; 循环COPY_LOOP标签，直到CX寄存器为0

        MOV AH, 4CH   ; 程序结束
        INT 21H


; 子程序: BUBBLE_SORT
; 使用冒泡排序法将数据段中的字节从小到大排序
BUBBLE_SORT PROC
    ;PUSH CX       ; 保存CX寄存器的值
    ;PUSH SI       ; 保存SI寄存器的值
    ;PUSH DI       ; 保存DI寄存器的值

    MOV DH, 9    ; 循环10次


OUTER_LOOP:
    MOV SI, OFFSET X
    MOV CL,9
    ;MOV DI, SI

INNER_LOOP:
    MOV AL, [SI]  ; 读取当前元素
    MOV DL, [SI+1]  ; 读取下一个元素

    CMP AL, DL    ; 比较当前元素和下一个元素
    JBE SKIP_SWAP ; 如果当前元素小于等于下一个元素，跳过交换

    ; 交换当前元素和下一个元素
    MOV BL, AL
    MOV AL, DL
    MOV DL, BL
    MOV [SI], AL
    MOV [SI+1], DL
SKIP_SWAP:
    INC SI        ; 增加SI寄存器的值，指向下一个元素
```

```
    LOOP INNER_LOOP  ; 循环INNER_LOOP标签，直到CX寄存器为0

    DEC DH         ; 外层循环计数器减1
    JNZ OUTER_LOOP  ; 如果外层循环计数器不为0，继续外层循环

    ;POP DI         ; 恢复DI寄存器的值
    ;POP SI         ; 恢复SI寄存器的值
    ;POP CX         ; 恢复CX寄存器的值
    RET

BUBBLE_SORT ENDP
CODE    ENDS
     END START

```

题三

设数据段中初始存放了分数在 1~100 的 10 个成绩，将这些成绩放入 初始地址为 3000H 的单元当中，3000H+I 表示第 I 位同学的成绩。 编写程序，将排出的学生成绩名次放在初始地址为 3100H 的单元中， 3100H+I 表示第 I 位同学的名次。 本题编号 I 的范围是 0~9

(样例: 56H, 4DH, 5DH, 52H, 64H, 47H, 51H, 5BH, 4FH, 61H)

1 实验思路

创建Rank_segment来存放排名

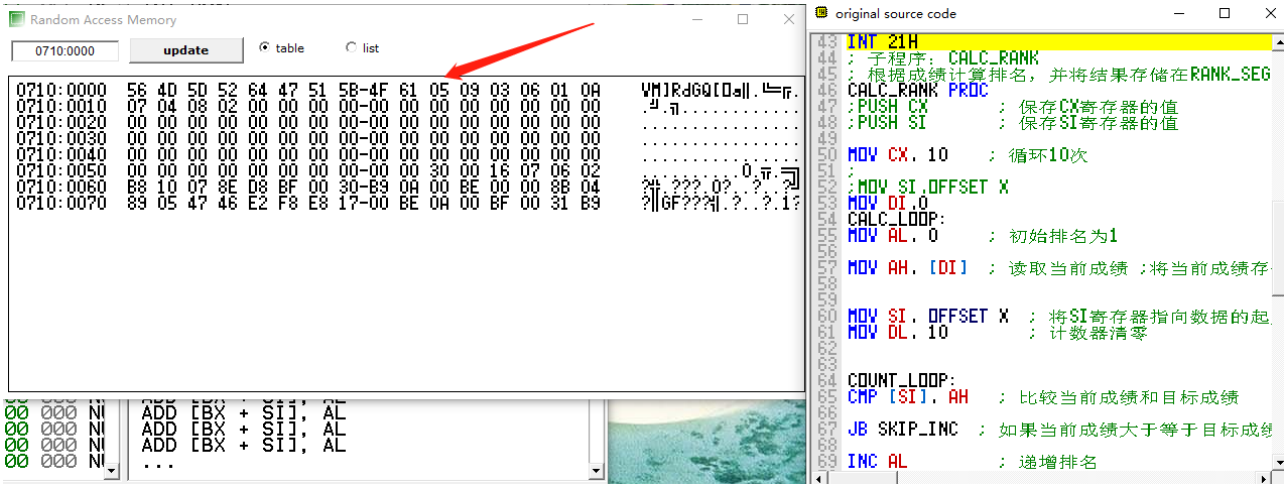
如何计算排名：AL存放排名，AH存放当前成绩，计算每位数据的排名时，将其初始化为1，然后从头开始遍历，如果有成绩大于当前成绩，将当前成绩排名+1

如此循环十次，可以得到每个数据的排名

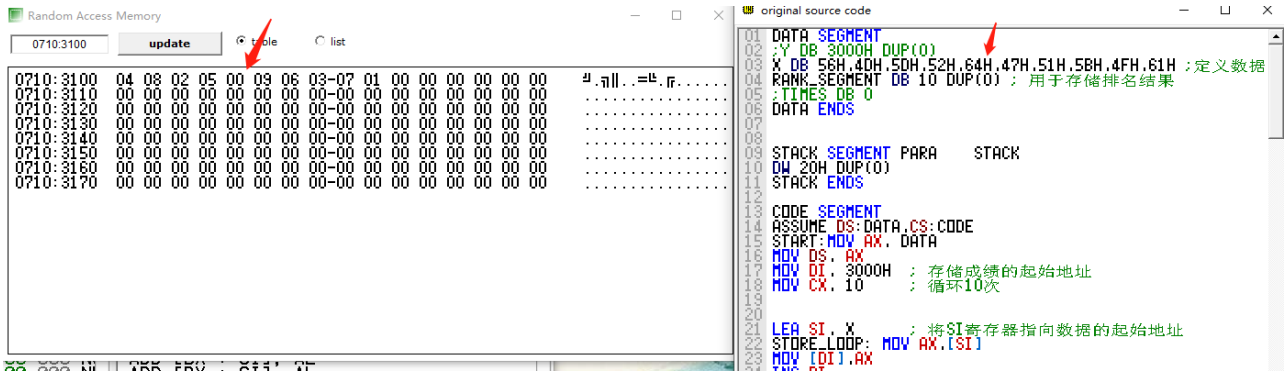
2 实验结果

可以看到当程序运行结束后，每个数据对应的排名，且范围为1-10

因此在将排名迁移到3100H时，要将其减1



所以在3100H的排名范围为0-9



源码score_proc.asm

```
DATA SEGMENT
    ;Y DB 3000H DUP(0)
    X DB 56H,4DH,5DH,52H,64H,47H,51H,5BH,4FH,61H ;定义数据
    RANK_SEGMENT DB 10 DUP(0) ; 用于存储排名结果
    ;TIMES DB 0
DATA ENDS


STACK SEGMENT PARA  STACK
    DW 20H DUP(0)
STACK ENDS

CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START:MOV AX, DATA
      MOV DS, AX
      MOV DI, 3000H  ; 存储成绩的起始地址
      MOV CX, 10     ; 循环10次


    LEA SI, X      ; 将SI寄存器指向数据的起始地址
```

```
STORE_LOOP: MOV AX,[SI]
            MOV [DI],AX
            INC DI
            INC SI
            LOOP STORE_LOOP

    CALL CALC_RANK  ; 调用子程序CALC_RANK进行排名计算

    MOV SI, OFFSET RANK_SEGMENT
    MOV DI, 3100H  ; 存储排名的起始地址
    MOV CX, 10     ; 循环10次

COPY_LOOP:
    MOV AL, [SI]  ; 读取当前排名
    MOV [DI], AL  ; 将当前排名存储到目标地址
    DEC [DI],1
    INC SI        ; 增加SI寄存器的值，指向下一个排名
    INC DI        ; 增加DI寄存器的值，指向下一个目标地址
    LOOP COPY_LOOP  ; 循环COPY_LOOP标签，直到CX寄存器为0

    MOV AH, 4CH   ; 程序结束
    INT 21H
; 子程序: CALC_RANK
; 根据成绩计算排名，并将结果存储在RANK_SEGMENT中
CALC_RANK PROC
    ;PUSH CX       ; 保存CX寄存器的值
    ;PUSH SI       ; 保存SI寄存器的值

    MOV CX, 10    ; 循环10次
    ;
    ;MOV SI,OFFSET X
    MOV DI,0
CALC_LOOP:
    MOV AL, 0     ; 初始排名为1

    MOV AH, [DI]  ; 读取当前成绩 ;将当前成绩存储在AH寄存器中


    MOV SI, OFFSET X  ; 将SI寄存器指向数据的起始地址
    MOV DL, 10        ; 计数器清零


COUNT_LOOP:
    CMP [SI], AH   ; 比较当前成绩和目标成绩

    JB SKIP_INC  ; 如果当前成绩大于等于目标成绩，跳过递增排名
```

```
68
69      INC AL          ; 递增排名
70
71  SKIP_INC:
72      INC SI          ; 增加SI寄存器的值，指向下一个成绩
73
74      DEC DL          ;递减DL，循环次数为0
75      JNZ COUNT_LOOP
76      MOV [RANK_SEGMENT + DI], AL   ; 将排名存储在RANK_SEGMENT中
77      INC DI
78      LOOP CALC_LOOP  ; 循环CALC_LOOP标签，直到CX寄存器为0
79      RET
80
81  CALC_RANK ENDP
82
83  CODE ENDS
84   END START
```

3 心得

NOTE:

- 判断符号要是无符号数，
- 在子程序中，如果没有PUSH，不能有POP，push与pop是成对的