实验报告

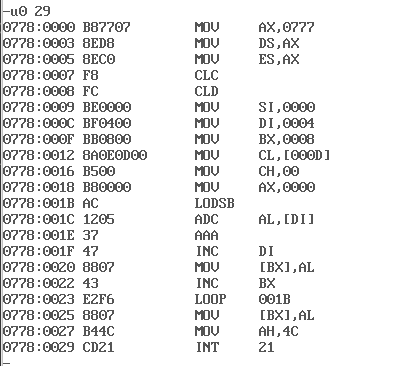
## 两个非压缩的BCD数的加法

非压缩BCD数，一个数字就占一个字节，加数被放在SBCD1为首地址的顺序单元中，加数放在以SBCD2为首地址的顺序单元中 考虑到是４位BCD数相加，那么应该使用5个字节去存储结果．

STACK SEGMENT PARA STACK 'STACK'  
 DB 200 DUP(?)  
STACK ENDS  
DATA SEGMENT  
SBCD1 DB 5, 6, 9, 2;2965  
SBCD2 DB 2, 3, 7, 8;8732  
SSUM DB 5 DUP(0);５个字节去存储结果 算式为2965+8732=11697  
CONT DB 4  
DATA ENDS  
CODE SEGMENT  
 ASSUME CS:CODE, DS:DATA, SS:STACK, ES:DATA  
SBCDAD: MOV AX, DATA  
 MOV DS, AX  
 MOV ES, AX  
 CLC  
 CLD;DF=0 与LODSB操作有关，使LODSB后SI自动加1  
 MOV SI, OFFSET SBCD1  
 MOV DI, OFFSET SBCD2  
 MOV BX, OFFSET SSUM  
 MOV CL, CONT  
 MOV CH, 0  
 MOV AX, 0;AX清零  
SBCDAD1:LODSB;把[SI]的数据送入AL，并使得SI++（DF=0）  
 ADC AL, [DI]  
 AAA;对未压缩的BCD数之和做调整  
 INC DI  
 MOV BYTE PTR[BX], AL  
 INC BX  
 LOOP SBCDAD1  
 MOV BYTE PTR[BX], AL;把万位送给以SSUM为首地址的第五个顺序单元  
 MOV AH, 4CH  
 INT 21H  
CODE ENDS  
 END SBCDAD

### DEBUG

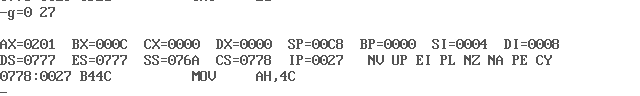
* 首先查看代码段: \_u 0 29



代码段反编译

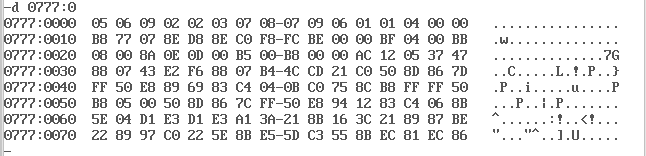
可以看出数据段的**段地址为0777**, 结果SSUM的首个单元的偏移地址为**0008**

* 运行: \_g=0 27 在MOV AH,4C处打断点



运行

* 结果查看: \_d 0777:0

 只需查看0777:0000即可 SSUM的首个单元的偏移地址为**0008**,那么从第9个单元到第13个单元的值连起来为11697(高位在前) **结果正确**