# 

**计算机科学与技术学院**

**2016-2017学年第2学期**

**《汇编语言程序设计》**

**实验3：顺序程序设计**

**专业： 信息安全**

**学号： 150341221**

**姓名： 路伟饶**

**教师： 王家亮**

**完成日期：2017年4月10日**

1. 实验要求
2. 复习数据传送和减法等运算类指令
3. 请运行以上源程序，并在 DEBUG 状态下观察程序运行结果。
4. 实验目的
5. 熟悉运算类指令对标志位的状态影响以及标志位状态的表示方法;
6. 熟悉逻辑类指令的用法;
7. 掌握最基本的程序设计方法。
8. 程序说明
9. 求内存单元中已定义的两个数据(855CH 与 AB43H)之差值。下面已给出一种方法，请用定义字的方法改写实现。

已有的程序：

DATA SEGMENT

A DB 5CH,85H

B DB 43H,0ABH

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA

MOV DS,AX

MOV SI,0

MOV AL,A[SI]

SUB AL,B[SI]

MOV A[SI],AL

INC SI

MOV AL,A[SI]

SBB AL,B[SI]

MOV A[SI],AL

MOV AH,4CH

INT 21H

CODE ENDS

END START

用定义字改写后的程序：

DATA SEGMENT

A DW 855CH

B DW 0AB43H

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA

MOV DS,AX

MOV SI,0

MOV AX,A

MOV BX,B

SUB AX,BX

MOV A,AX

MOV AH,4CH

INT 21H

CODE ENDS

END START

1. 试编写一程序，实现将一存放在 DX、AX 中的 32 位操作数循环右移 4 位。

本题主要考虑的是如何将DX的最低位传递至AX的最高位，以及如何将AX的最低位传递到DX的最高位。首先SAR DX,1就完成了将DX的最低位传递至CF标志位的任务，然后用RCR AX,1完成了CF标志位传递给AX的最高位，并且AX的最低位也进入了CF标志位，因为在SAR操作中，原来DX最高位被“复制”了一次，也就是说DX现在的最高位和次高位一定相同。如果直接使用ROR操作，高位将出现错误，并且实际上DX多右移了一次，故要先进行RCL DX,1操作，将CF标志位（也就是之前AX的最低位）先暂存至DX的最低位，然后再进行ROR DX,1操作，这样原来的AX最低位才能顺利的抵达DX的最高位，并且保证次高位的正确。使用ROR的原因是因为CF此时存储的是一个无用的值（原来DX的最高位），所以不应该使用RCR。再配合CX和LOOP执行四次相同指令，即可循环右移4位。

DATA SEGMENT

A DW 4321H

D DW 8765H

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV BX,DATA

MOV DS,BX

MOV AX,A

MOV DX,D

MOV CX,4

s: SAR DX,1

RCR AX,1

RCL DX,1

ROR DX,1

LOOP s

MOV AH,4CH

INT 21H

CODE ENDS

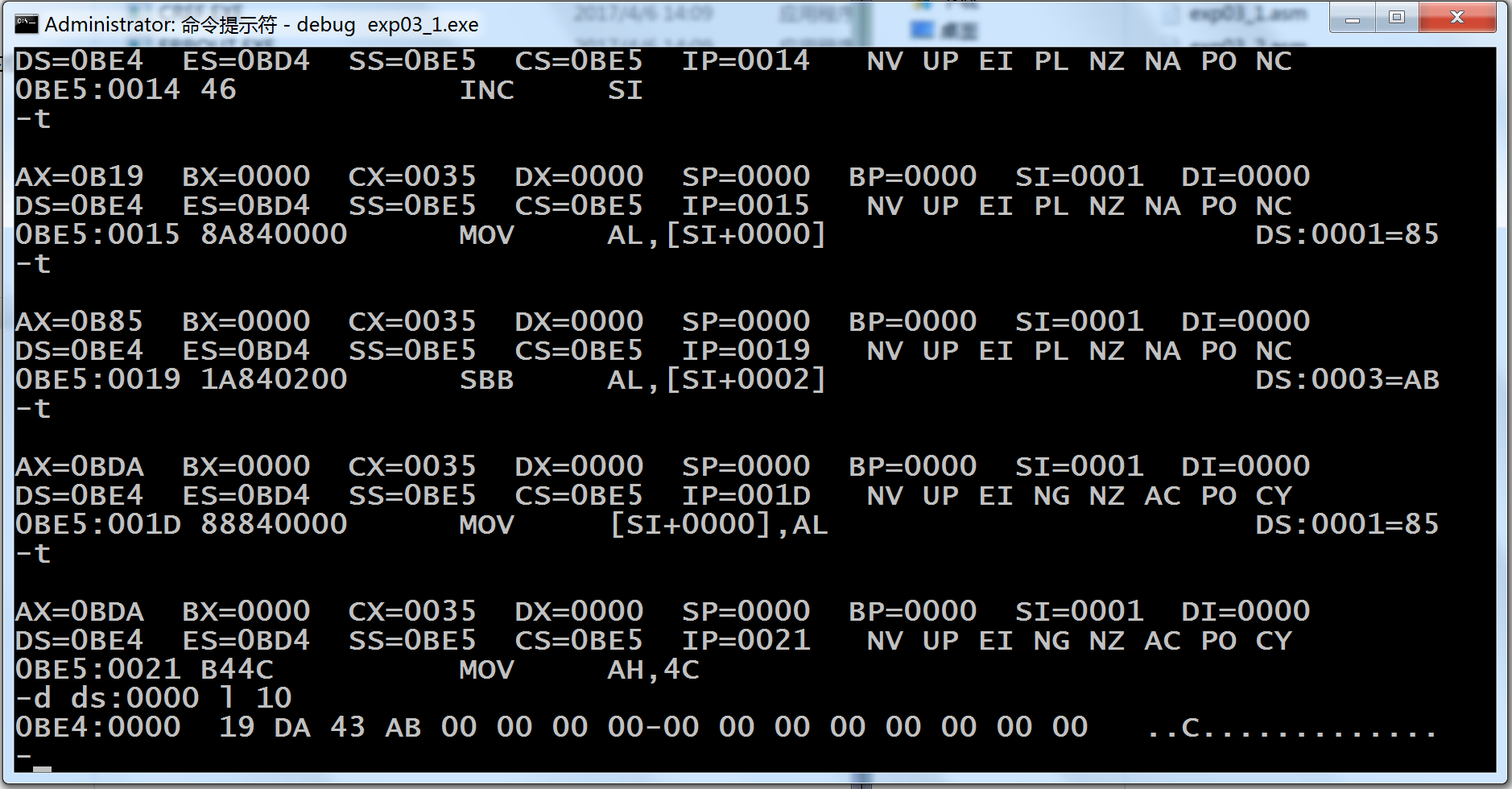
END START

1. 上机调试说明

本次上机使用的是Windows 7 32位操作系统，发现32位的7也可以直接运行masm，所以果断抛弃XP。在调试过程中没有遇到明显的问题。

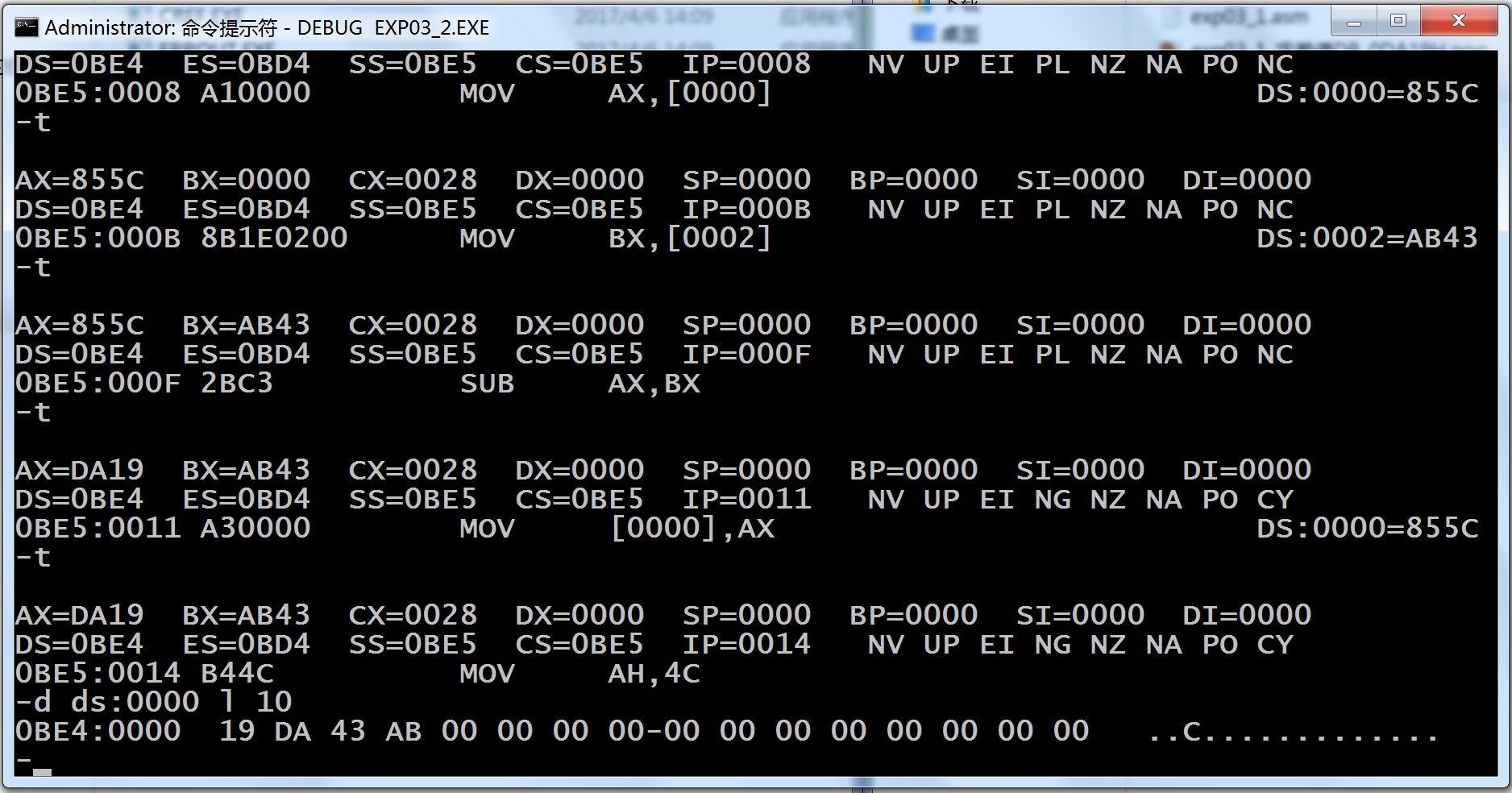
1. 写出源程序清单及执行结果
2. 求内存单元中已定义的两个数据(855CH 与 AB43H)之差值。下面已给出一种方法，请用定义字的方法改写实现。

已有的程序



已有程序实际上是只利用了8位寄存器进行操作，使用了带借位的减法SBB。

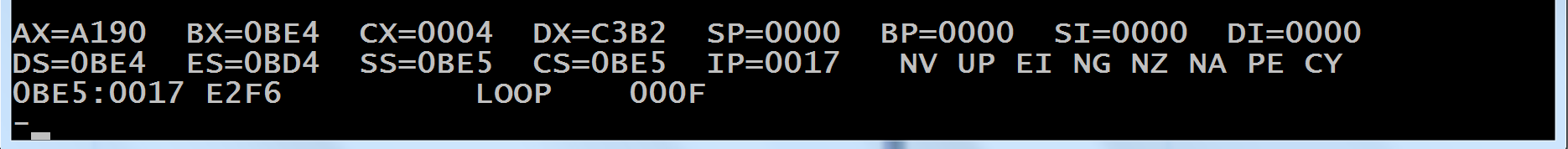
用字实现的程序



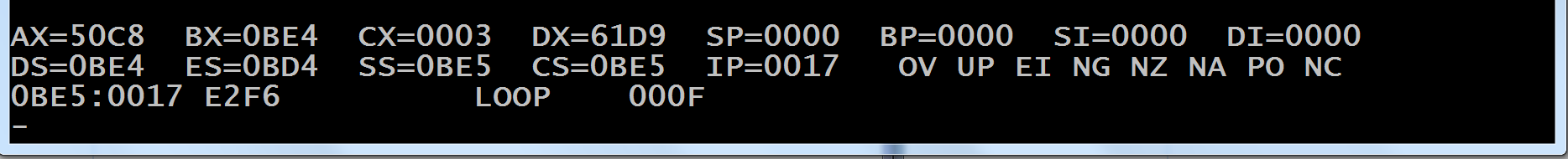
用字实现十分直观，直接进行SUB减法即可。

1. 试编写一程序，实现将一存放在 DX、AX 中的 32 位操作数循环右移 4 位。

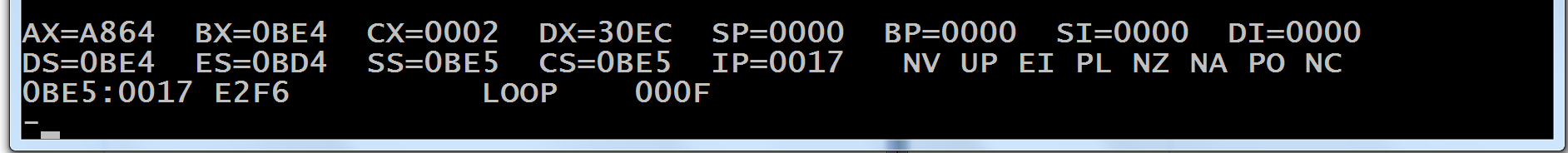
初始：DX=8765H，AX=4321H



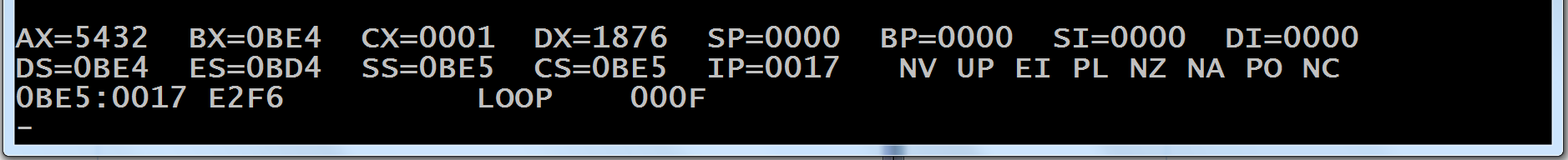
循环右移1位后：DX=0C382H，AX=0A190H



循环右移2位后：DX=61D9H，AX=50C8H



循环右移3位后：DX=30ECH，AX=0A864H



循环右移1位后，最终结果为：DX=1876H，AX=5432H

1. 心得体会

第三次上机主要是练习SUB指令和移位指令，重点理解了逻辑移位、算数移位、循环移位（不进位），循环移位（带CF进位）的使用。听其他同学说第二个32位操作数循环右移4位的题目，在PPT上的一种解法是先临时保存值，再直接移四位。考虑到此法用到了辅助空间，我认为直接移动的方式有助于加深对移位的理解，并且锻炼自己的思维，于是我就摒弃了传统的做法。不过现在的方法不得不配合了LOOP使用，这也违背了本实验名为“顺序程序设计”的含义，而成为了“循环程序设计”。但是话说回来，有时间的话学学新知识又未尝不可呢？