上海理工大学光电信息与计算机工程学院

**《计算机组成》实验报告**

****

**专　　业 智能科学与技术**

**姓 名　 高浩琦**

**学　 号 2035060413**

**年　　级 2020级**

**指导教师 邵 清**

**成 绩：**

**教师签字：**

目录

[1. 实验一：算术运算指令 3](#_Toc6563749)

[1.1 实验目的 3](#_Toc6563750)

[1.2 实验内容 3](#_Toc6563751)

[1.3 主要源代码 3](#_Toc6563752)

[1.4 实验结果与分析 3](#_Toc6563753)

[2. 实验二：数组操作 4](#_Toc6563754)

[2.1实验目的 4](#_Toc6563755)

[2.2实验内容 4](#_Toc6563756)

[2.3主要源代码 4](#_Toc6563757)

[2.4实验结果与分析 4](#_Toc6563758)

[3. 实验三：逻辑运算和移位指令 5](#_Toc6563759)

[3.1实验目的 5](#_Toc6563760)

[3.2实验内容 5](#_Toc6563761)

[3.3主要源代码 5](#_Toc6563762)

[3.4 实验结果与分析 5](#_Toc6563763)

[4. 实验四：字符串的输入输出 6](#_Toc6563764)

[4.1实验目的 6](#_Toc6563765)

[4.2实验内容 6](#_Toc6563766)

[4.3主要源代码 6](#_Toc6563767)

[4.4 实验结果与分析 6](#_Toc6563768)

[5. 总结 7](#_Toc6563769)

# 实验一：算术运算指令

## 实验目的

1. 熟悉Intel8086指令系统的汇编语言编程工具MASM实验环境，掌握编译、调试、运行程序的全过程。
2. 学习算术运算指令的用法

## 实验内容

1. 将十六进制数123AH存于存储器中（变量名自定义）

计算表达式result = -x + (y-z), 其中result、x、y、z均为双字节整数操作数和运算结果均存于寄存器中

1. 定义两个变量，并交换其内容

计算表达式result = -x + (y-z), 其中result、x、y、z均为单字节整数操作数和运算结果均存于变量中

1. 利用加减运算指令测试零标志位、进位标志位、符号标志位和溢出标志位的变化

## 主要源代码

(一)

DATAS SEGMENT

D1 DW 0

x DW 2222H

y DW 6666H

z DW 3333H

re DB 0

DATAS ENDS

;123AH

mov D1,123AH

;result = -x + y - z

mov ax,y

sub ax,z

sub ax,x

(二)

DATAS SEGMENT

D1 DW 0234H

D2 Dw 0123H

x DB 01H

y DB 04H

z DB 02H

re DB 0

DATAS ENDS

;交换

mov ax,D1

mov bx,D2

mov D1,bx

mov D2,ax

;result = -x + y – z

mov ah,00H

mov al,y

sub al,z

sub al,x

mov re,al

(三)

DATAS SEGMENT

x DB 0

y DB 0

;此处输入数据段代码

DATAS ENDS

;进位ov 溢出标志

mov x,88H

mov y,0AAH

mov al,x

add al,y

;零标志位

mov x,88H

mov y,88H

mov al,x

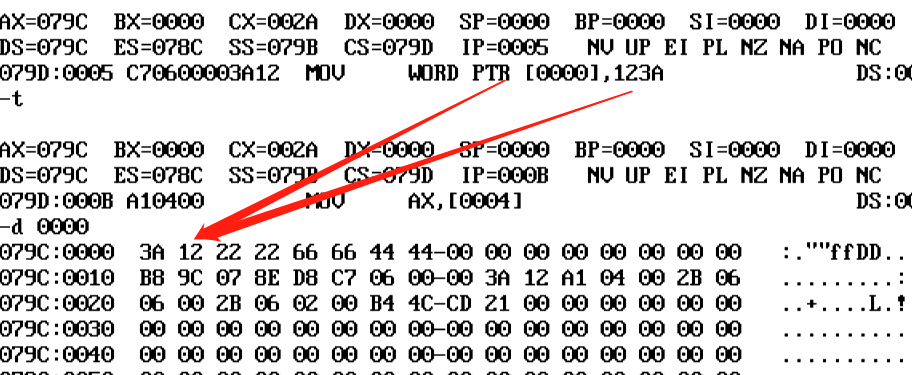
sub al,y

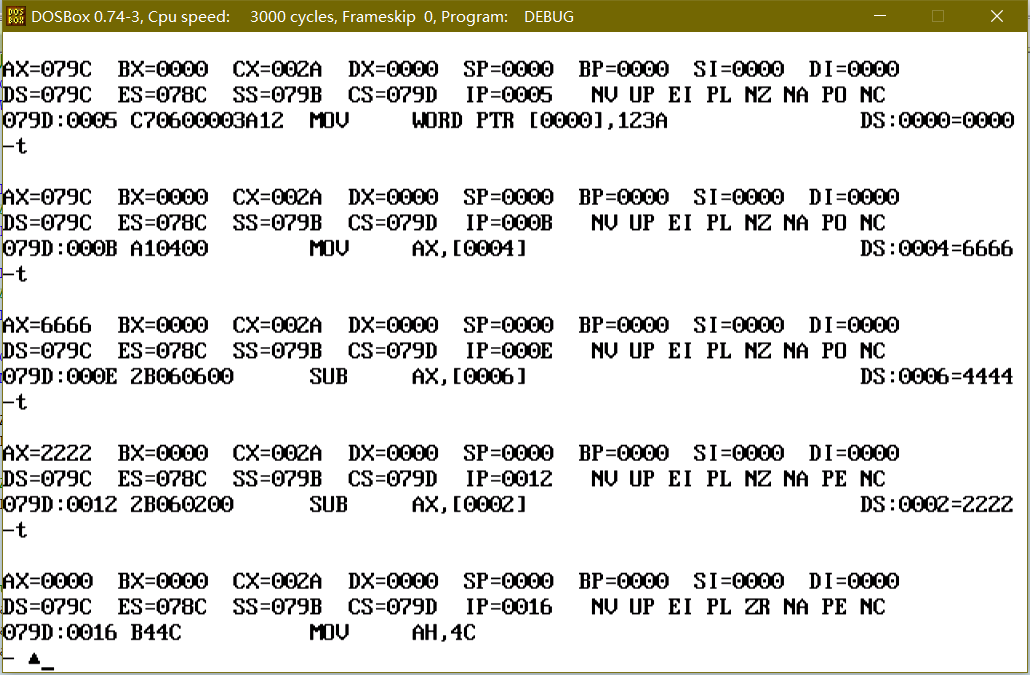
;符号标志

sub al,01H

## 实验结果与分析

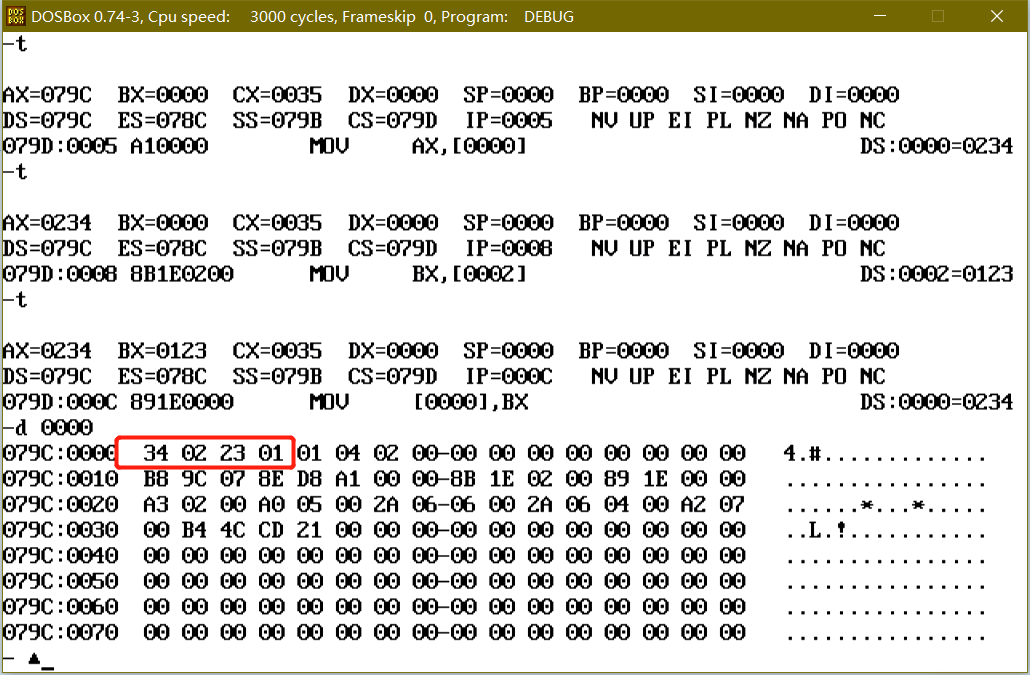
(一)

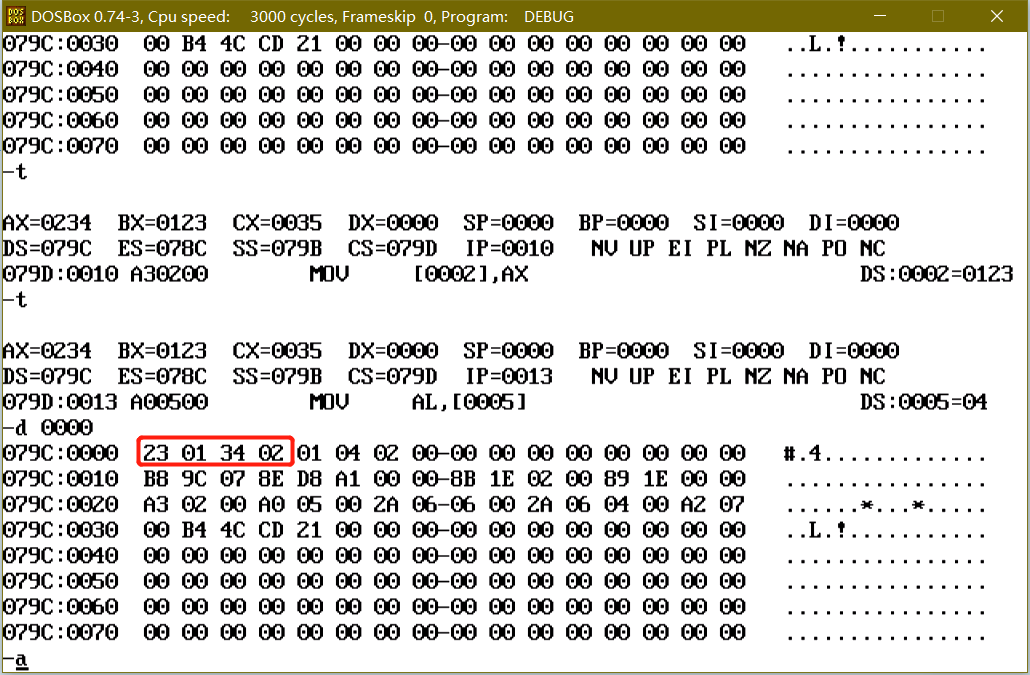




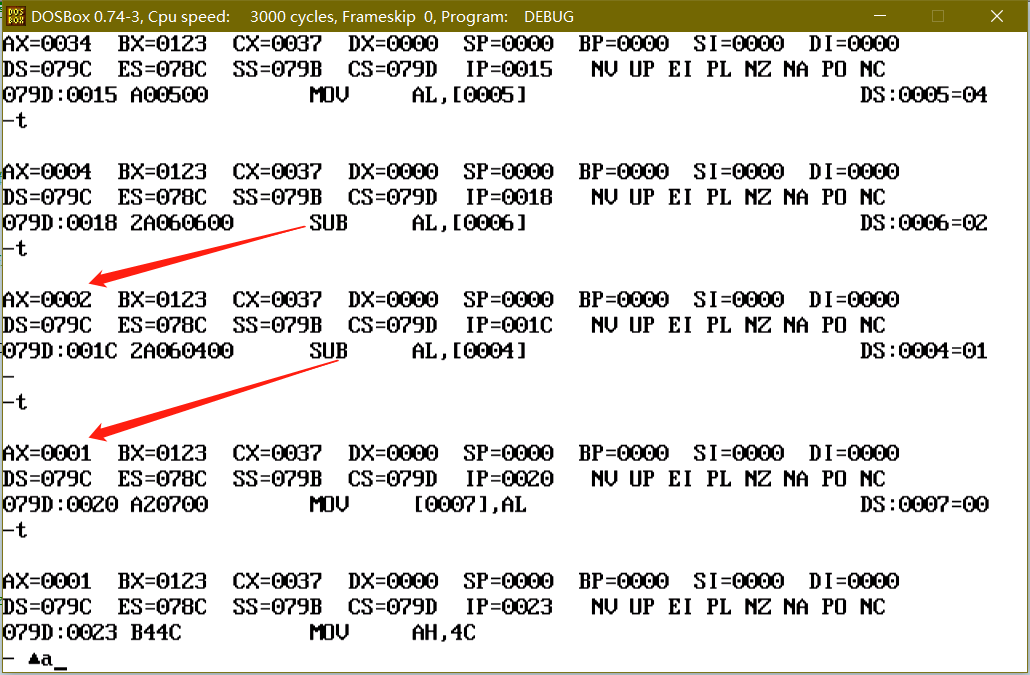
(二)

**交换**

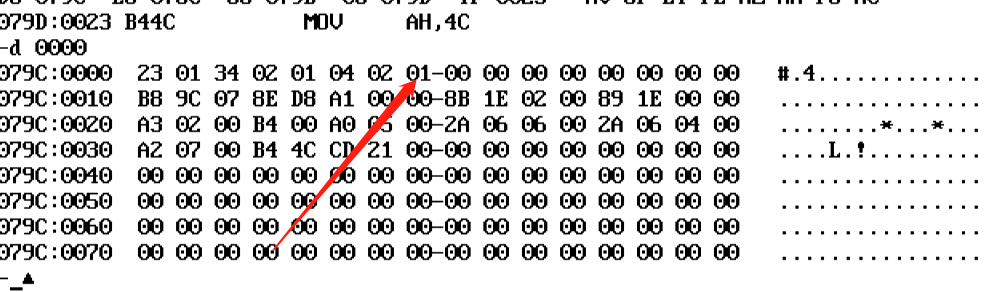




**计算 -x+y-z**

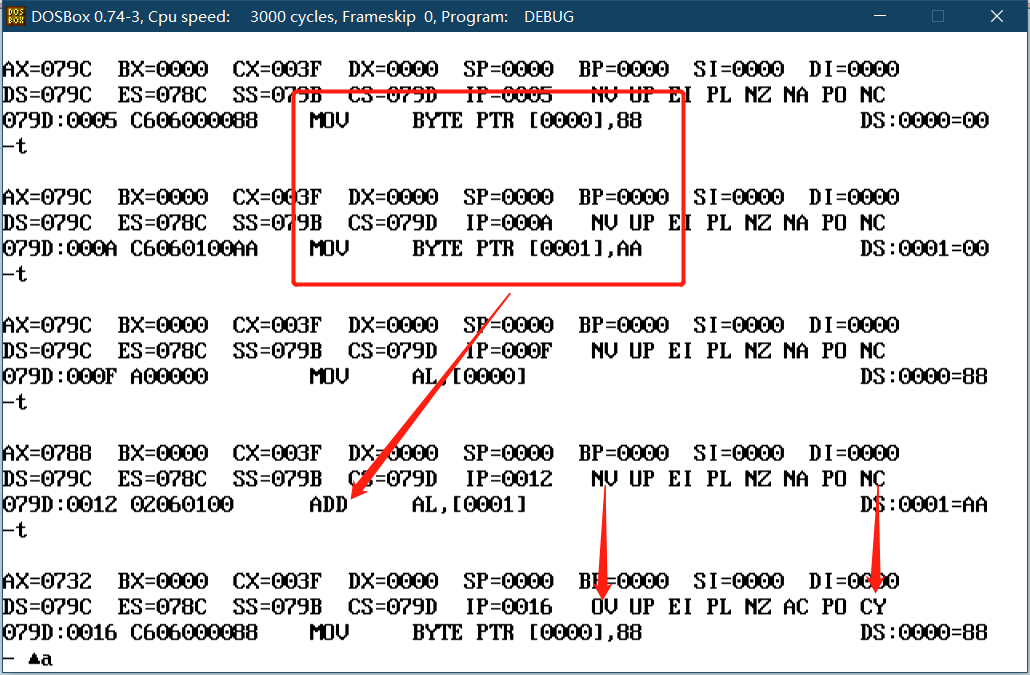


**存入re**

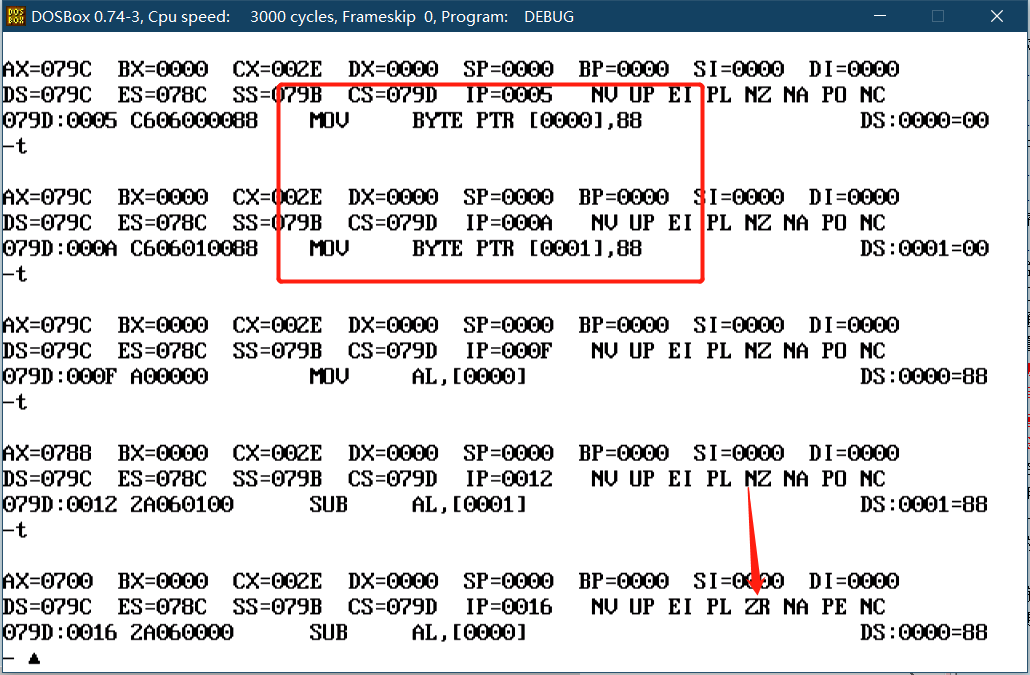


(三)

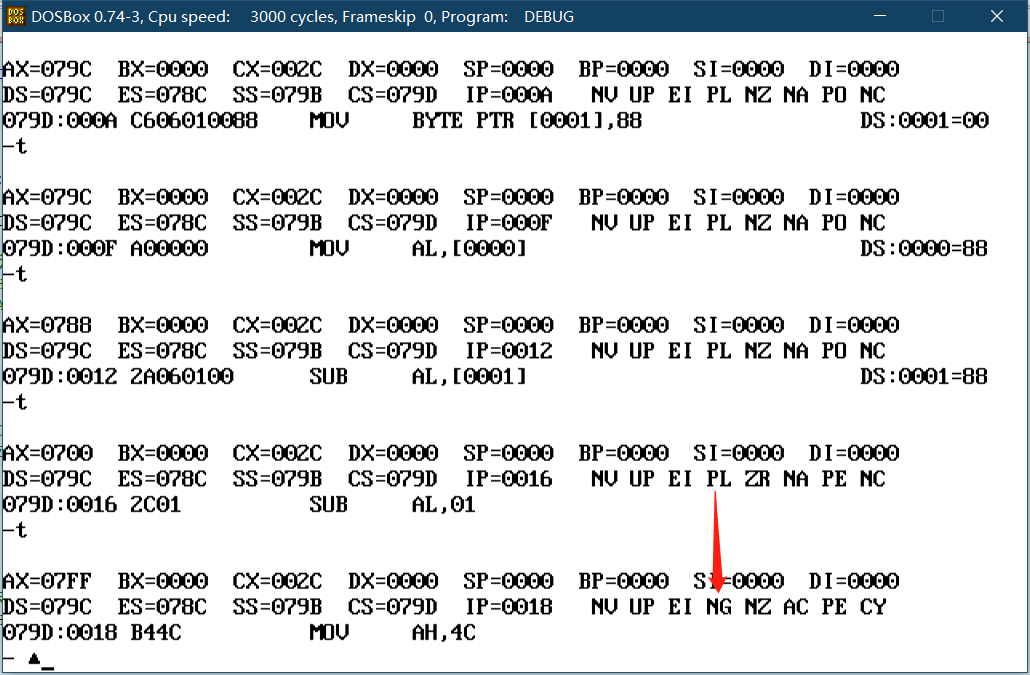
**进位标志位 溢出标志位**



**零标志位**



**符号标志位**



# 实验二：数组操作

## 2.1实验目的

学习使用数组和转移指令的用法

## 2.2实验内容

编程实现以下功能：

1. 为源数组赋初值（要求每个数组元素的值不相同）
2. 将该数组搬到主存的另一段连续的存储单元中，目标数组的起始地址是AD
3. 求数组中N个元素的最小值，结果存于MIN中
4. 将AS中的元素向左循环移动一个位置，例如{10H,20H,30H,40H}左循环移动后变成{20H,30H,40H,10H}要求–采用循环结构

**注:在实现循环移位操作时不能将数组元素复制到其它数组中**

## 2.3主要源代码

(一) 源数组赋初值

DATAS SEGMENT

ARRAY1 DB 40H,20H,60H,80H

AD DB 4 DUP(?)

MIN DB 00H

;此处输入数据段代码

DATAS ENDS

(二) 转移至AD地址

;赋给数组地址

MOV SI,OFFSET ARRAY1

MOV DI,OFFSET AD

MOV CX,3

;转移AD

L1: MOV AX,[SI]

MOV [DI],AX

INC DI

INC SI

LOOP L1

(三)最小值

;求min

MOV SI,OFFSET ARRAY1

MOV CX,3

MOV AH,00H

MOV BL,[SI]

MOV MIN,BL

L2: INC SI

MOV AL,[SI]

CMP AL,MIN

JC L3

JNC LOOP1

L3: MOV MIN,AL

JMP LOOP1

LOOP1: LOOP L2

(四)左循环

;左循环转移

MOV SI,OFFSET ARRAY1

MOV CX,3

MOV SI,2

MOV AH,[SI+1]

L4: MOV AL,[SI]

MOV [SI+1],AL

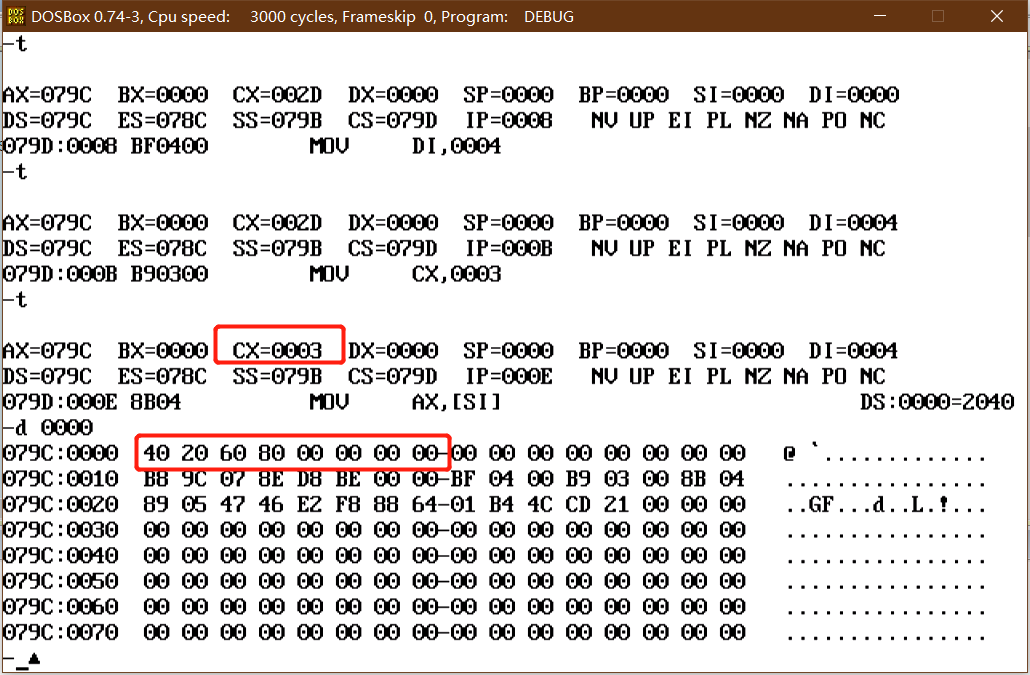
DEC SI

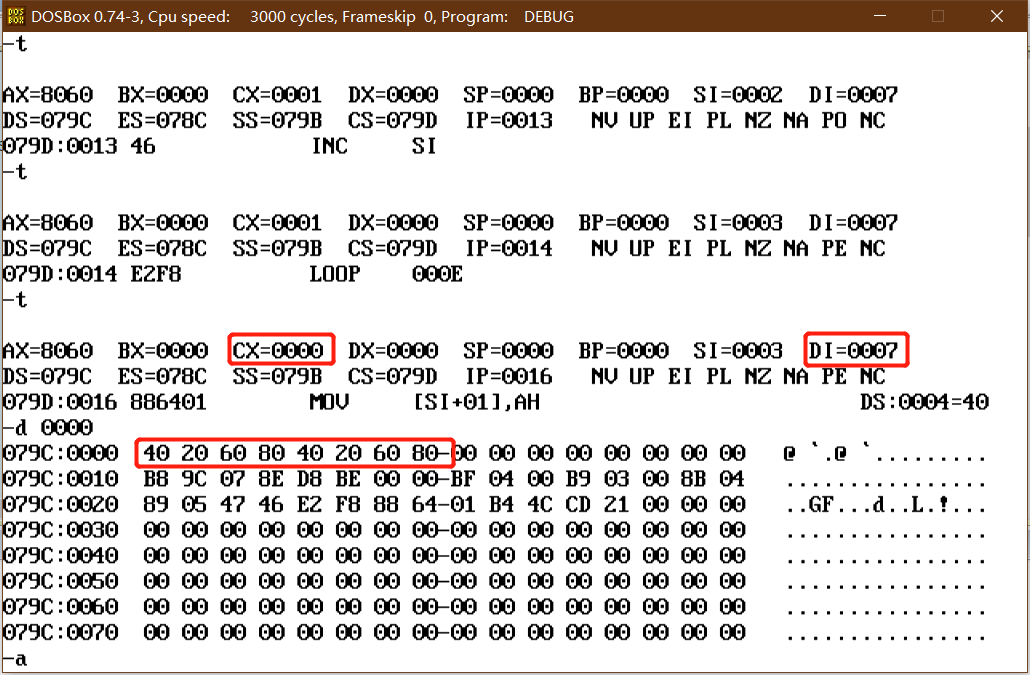
LOOP L4

MOV [SI+1],AH

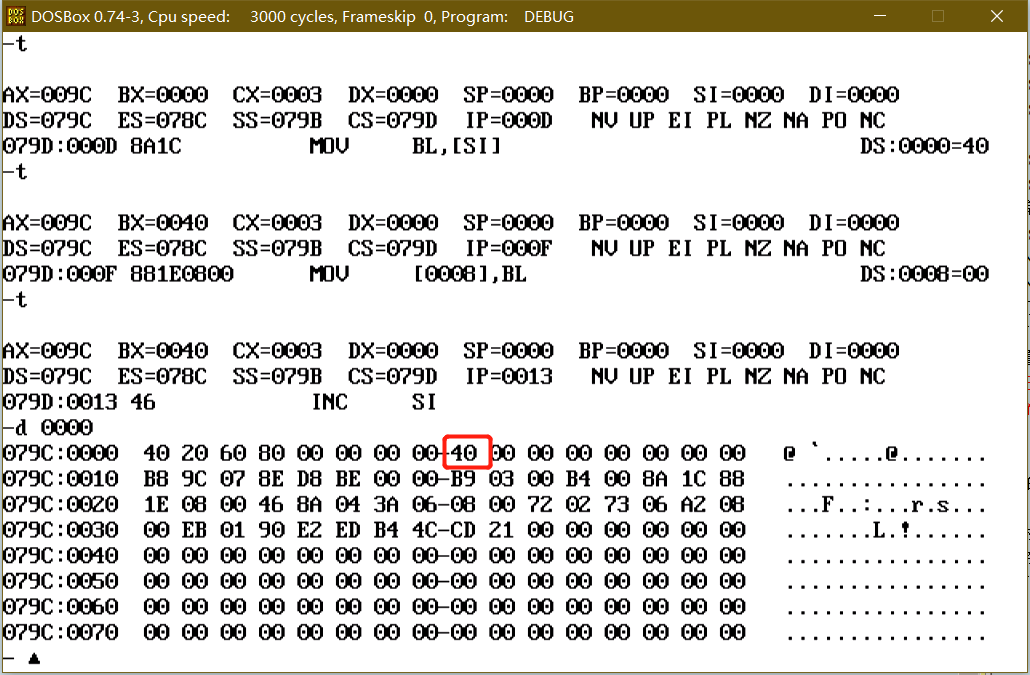
## 2.4实验结果与分析

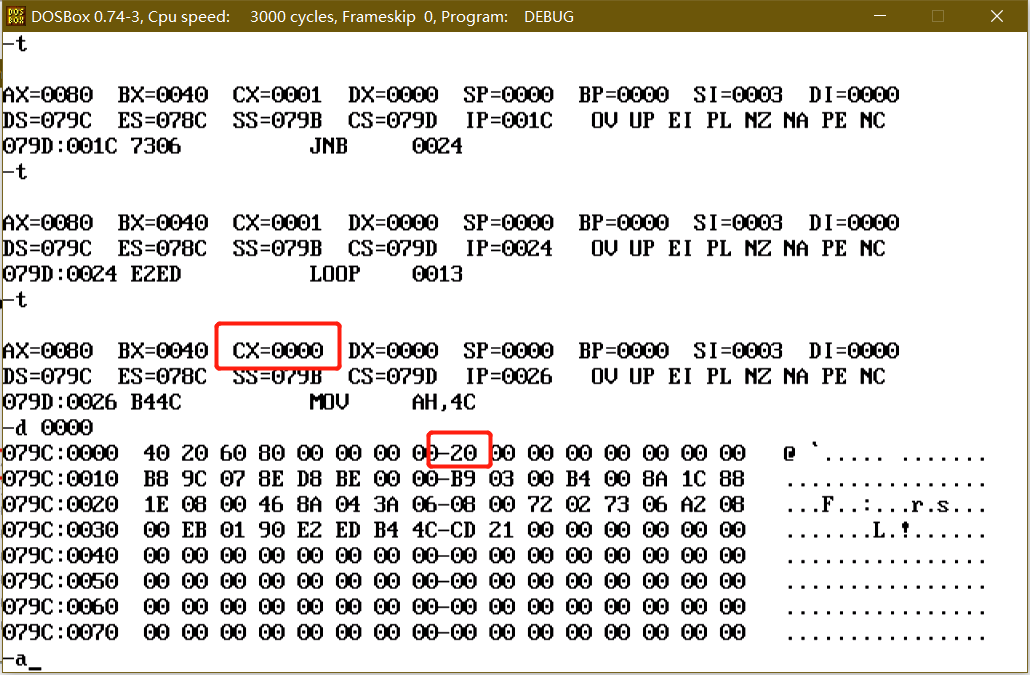
1. 转移至AD数组



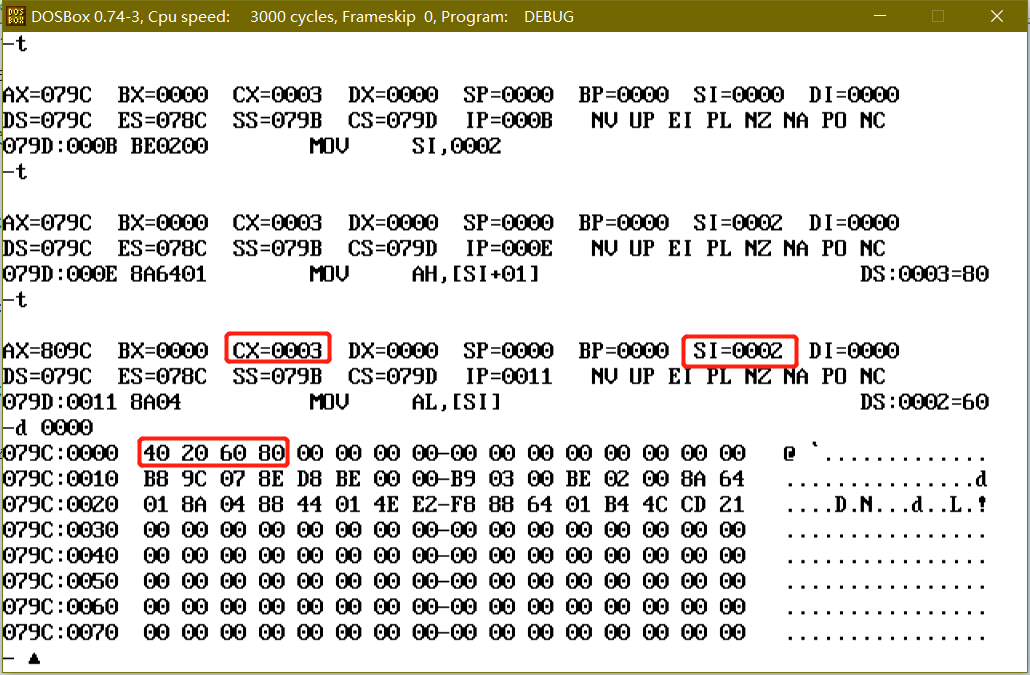


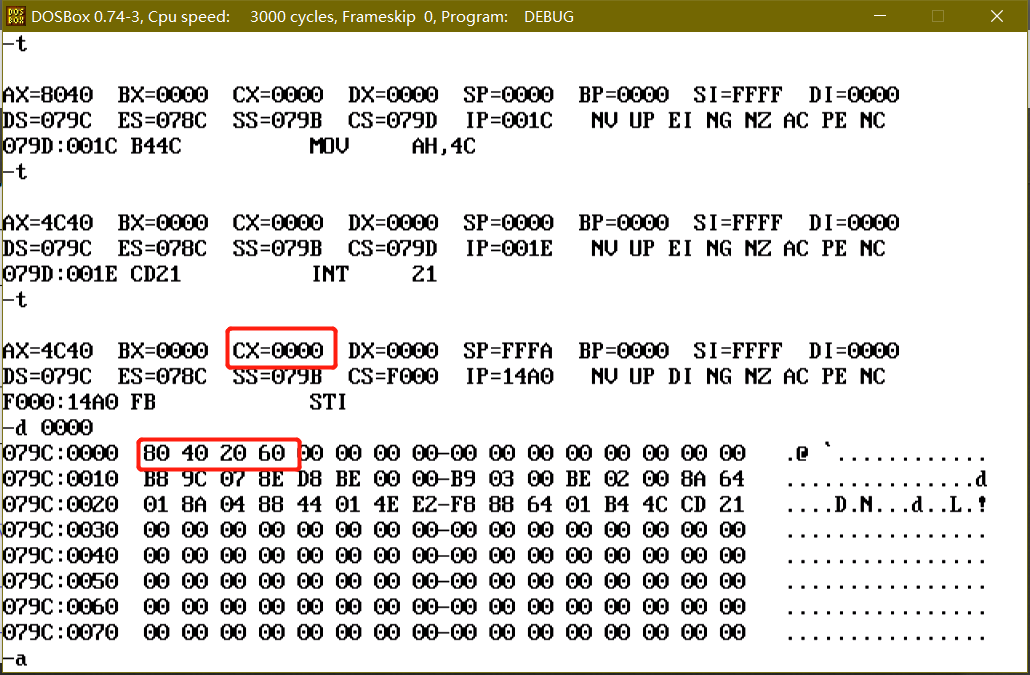
2. 求MIN





3. 左循环数组





# 实验三：逻辑运算和移位指令

## 3.1实验目的

学习逻辑运算指令和移位指令的用法

## 3.2实验内容

1. 实现两个无符号整数的乘法，使用SHL和ADD指令（不使用乘法指令），结果存于product中

**说明：操作数不能是2的整数次幂**

2. 对一个长度为n的字符串进行加密和解密操作。每个单字符采用不同的密钥。密钥存于数组Key中。

例：Source=‘20210428’ key= {5，0，6，2，3，1，4，6}

方法一：采用XOR指令。 单字符密钥范围(0～255)

程序运行结果依次显示：

**所产生的密文**

**解密后的明文**

方法二:循环移位加密–单字符密钥范围(-7~ 7) 。

其中负数表示循环左移,正数表示循环右移，0不变，数字表示移动的位数。

## 3.3主要源代码

(一)

;初始化

MOV AL,AN

MOV AH,00H

MOV BL,BN

MOV BH,00H

MOV CX,08H

;al右移动 bl左移

L1: SHR AL,1

JC L2

SHL BL,1

JMP L3

L2: ADD BH,BL

SHL BL,1

JMP L3

;如果al=1 跳出循环

L3: CMP AL,1

JE L4

LOOP L1

L4: ADD BH,BL

MOV PRODUCT,BH

(二)

DATAS SEGMENT

plainText DB '20226745',13,10,'$'

key DB '1','2','3','4','5','6','7','8'

;此处输入数据段代码

DATAS ENDS

MOV SI,OFFSET key

MOV DI,OFFSET plainText

MOV CX,08H

L1: MOV BL,[SI]

XOR [DI],BL

INC SI

INC DI

LOOP L1

LEA DX,plainText

MOV AH,09H

INT 21H

;再来一遍

MOV SI,OFFSET key

MOV DI,OFFSET plainText

MOV CX,08H

L2: MOV BL,[SI]

XOR [DI],BL

INC SI

INC DI

LOOP L2

LEA DX,plainText

MOV AH,09H

INT 21H

(三)

DATAS SEGMENT

plainText1 DB '20226745',13,10,'$'

key DB -1,2,3,-4,5,-6,7,-8

;此处输入数据段代码

DATAS ENDS

MOV SI,OFFSET key

MOV DI,OFFSET plainText1

MOV CX,08H

L1: MOV DH,[DI]

MOV DL,[SI]

CMP DL,80H

JC L2

;判断正负

;取反

NEG DL

JMP L3

L2: JMP L4

;负数左移

L3: ROL DH,1

DEC DL

CMP DL,00H

JE L5

JMP L3

;正数右移

L4: ROR DH,1

DEC DL

CMP DL,00H

JE L5

JMP L4

L5: MOV [DI],DH

INC DI

INC SI

LOOP L1

;输出

LEA DX,plainText1

MOV AH,09H

INT 21H

;再来一遍

MOV SI,OFFSET key

MOV DI,OFFSET plainText1

MOV CX,08H

L21: MOV DH,[DI]

MOV DL,[SI]

CMP DL,80H

JC L22

;判断正负

;取反

NEG DL

JMP L23

L22:JMP L24

;负数右移

L23: ROR DH,1

DEC DL

CMP DL,00H

JE L25

JMP L23

;正数左移

L24:ROL DH,1

DEC DL

CMP DL,00H

JE L25

JMP L24

L25: MOV [DI],DH

INC DI

INC SI

LOOP L21

LEA DX,plainText1

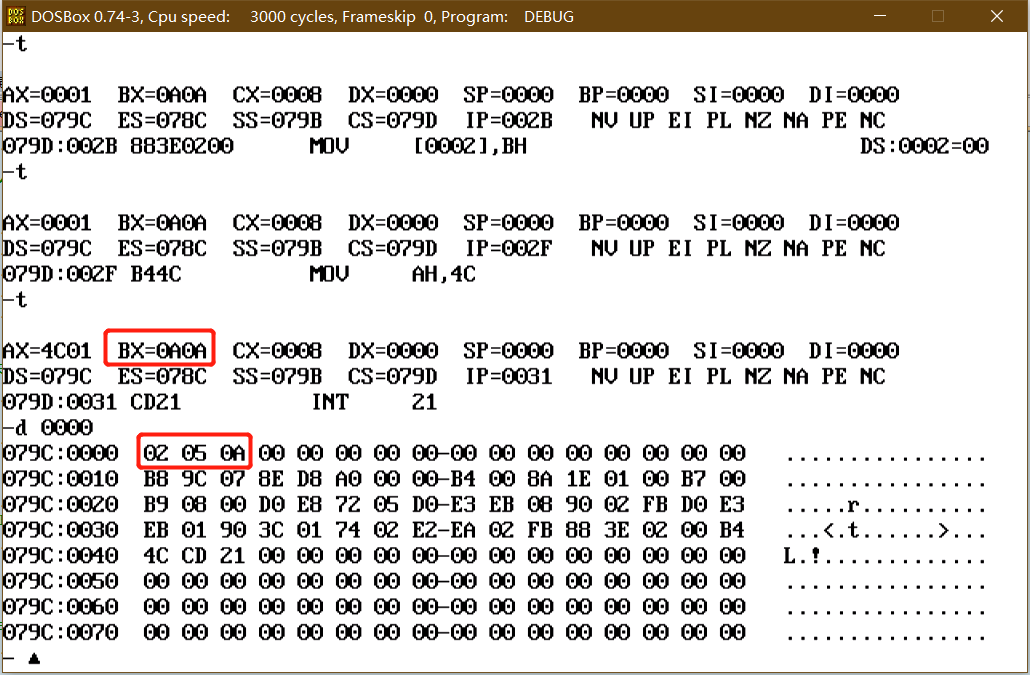
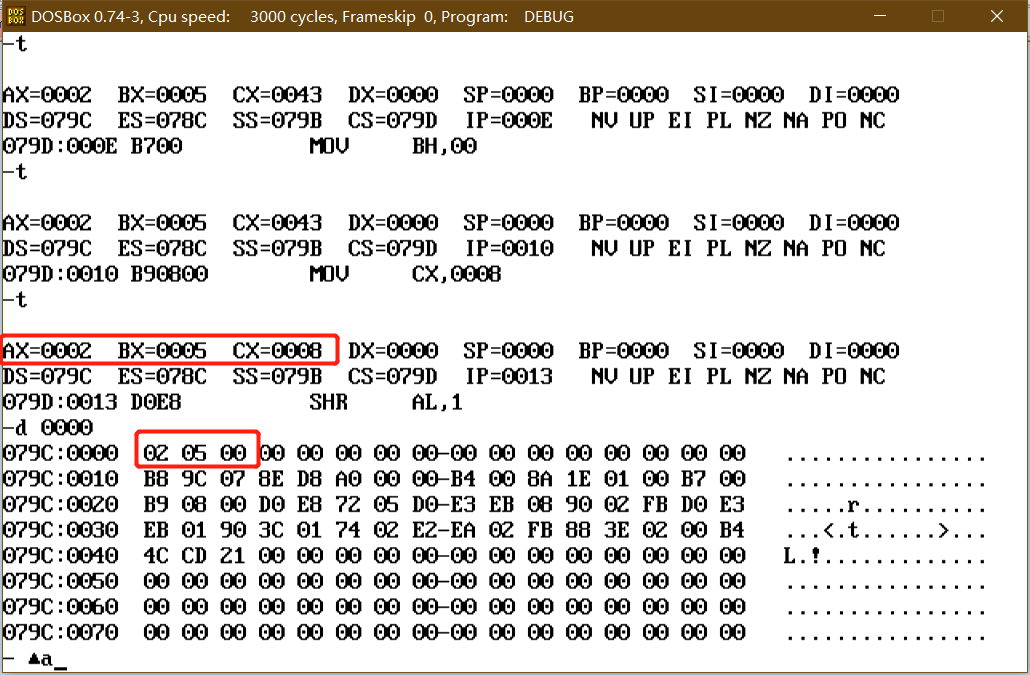
MOV AH,09H

INT 21H

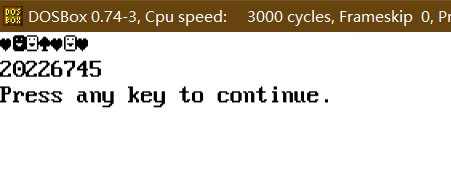
## 3.4 实验结果与分析

**(一)**

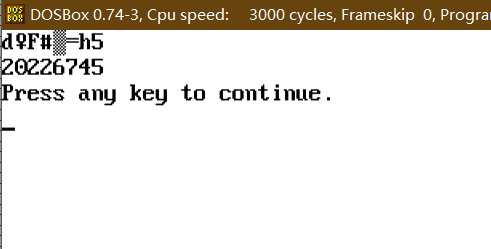
**2\*5 = A**



**(二)**



**(三)**



# 实验四：字符串的输入输出

## 4.1实验目的

掌握简单的字符串输入输出方法

## 4.2实验内容

输入指定长度的字符串（最后一位是数字）

要求:**转换成小（大）写字母，数字乘以2，并反转顺序**

用不同颜色显示该字符串(颜色任选)

## 4.3主要源代码

DATAS SEGMENT

string1 DB 4 DUP(?),13,10,'$'

;此处输入数据段代码

DATAS ENDS

DATASES SEGMENT

string2 DB '4000D'

DATASES ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS,ES:DATASES

MOV CX,04H

MOV BP,00H

;注: 不能打回车 要连续输入

L1: MOV AH,01H

INT 21H

MOV string1[BP],AL

INC BP

LOOP L1

MOV BP,00H

MOV CX,04H

L2: MOV BL,string1[BP]

CMP BL,'9'

JC L3 ;小于9 L3 数字

CMP BL,'Z'

JC L4 ;小于Z L4 大写字母

CMP BL,'z'

JC L5 ;小于z L5 小写字母

L3: ADD BL,BL

JMP L6

L4: ADD BL,20H

JMP L6

L5: SUB BL,20H

JMP L6

L6: MOV string1[BP],BL

INC BP

LOOP L2

MOV DL,0AH ;输出换行

MOV AH,02H

INT 21H

;LEA DX,string1

;MOV AH,09H

;INT 21H

MOV AX,0B800H

MOV ES,AX

MOV DI,1660

MOV SI,0

MOV BP,00H

MOV CX,04H

MOV AH,0AH

L7: MOV AL,string1[BP]

MOV string2[DI],AL

MOV string2[DI+1],AH

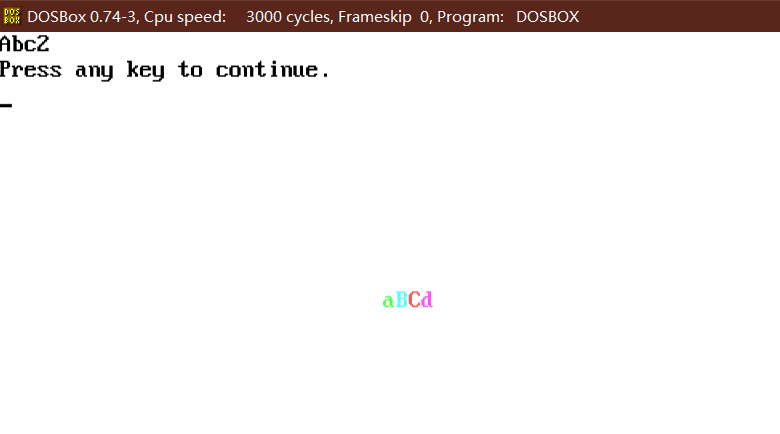
ADD DI,02H

INC BP

INC AH

LOOP L7

## 实验结果与分析



# 总结

本次实验终于结束了,学习汇编的最大功能是让我们了解了计算机体系结构，这是学习C、Java这类高级语言难以达到的。

编写汇编时要对体系结构非常清楚，熟知其编程模型：如寄存器有哪些、指令集是什么、中断怎么处理及内存怎么处理等。

但是对于Windows平台上的x86汇编而言由于有操作系统的存在，不像嵌入式的汇编能了解那么多。但是也很有学习价值，可以对x86的构架有个大体的了解。

汇编语言的大部分语句直接对应机器指令，执行速度快，效率高，代码体积小，在某些存储器容量有限，可以认识到需要快速和实时响应的场合比较有用。