1. **实验内容**

**第一部分：debug的使用**

1、请自学并完成如下内容：

（1）DEBUG的使用方法(1)

（2）DEBUG的使用方法(2)

2、用DEBUG调试、运行可执行程序。

1）进入DEBUG后，首先用命令R查看当前DS和IP寄存器内容，记录此值。

2）学习R，D，A，U，T，P，G命令的使用。

3）用命令A汇编下列指令，并判断指令正误并说明原因。

（1） LEA DI，[AX]

（2） MOV DS，2010

（3） PUSH AL

（4） MOV [BX]，0

3． 用A命令汇编下列程序段，完成将DS：2000开始的5个字节内存内容相加，并把相加的结果存放在DS：2005内存单元中，将程序写到硬盘上，然后用DEBUG调试﹑运行该程序，查看运行结果，观察程序段能否完成指定功能，若不能，说明原因。

MOV CX，5

MOV AL，0

MOV BX，2000

NEXT： ADD AL，[BX]

INC BX

DEC CX

LOOP NEXT

MOV [BX]，AL

INT 3

**第二部分：编译汇编语言**

1、请自学并完成如下内容：

（1）自学汇编语言程序的上机过程

（2）学会编译自己的第一个汇编程序

（3）程序执行过程的跟踪

2、完成实训内容一；

3、将下面的程序编译连接，用DEBUG加载、跟踪，然后回答问题。

4、程序如下，编写code段中的代码，将a段和b段中的数据依次相加，将结果存到c段中。

5、编辑录入下列程序，然后编译连接并执行。

要求：正常运行并了解输出结果。

6、编辑录入下列程序，然后编译连接并执行。要求：正常运行并了解输出结果。

1. **实验目的**

1．  学习使用DEBUG程序的各种命令。

2．  掌握用DEBUG调试自编程序的方法，为以后实验打下基础。

3．  学习汇编语言程序的 编辑、编译与连接。

4．  掌握使用DEBUG调试汇编语言程序。

1. **实验原理**

Debug 使用和编译汇编程序

**四、实验过程**

（重点部分，必须包括：实验开发环境、实验步骤、实验源程序代码及其结构分析）

**第一部分：**

环境:windos，dosbox

步骤：

1、请自学并完成如下内容：

2、用DEBUG调试、运行可执行程序。

1）进入DEBUG后，首先用命令R查看当前DS和IP寄存器内容，记录此值。

2）学习R，D，A，U，T，P，G命令的使用。

3）用命令A汇编下列指令，并判断指令正误并说明原因。

（1） LEA DI，[AX]

（2） MOV DS，2010

（3） PUSH AL

（4） MOV [BX]，0

3． 用A命令汇编下列程序段，完成将DS：2000开始的5个字节内存内容相加，并把相加的结果存放在DS：2005内存单元中，将程序写到硬盘上，然后用DEBUG调试﹑运行该程序，查看运行结果，观察程序段能否完成指定功能，若不能，说明原因。

MOV CX，5

MOV AL，0

MOV BX，2000

NEXT： ADD AL，[BX]

INC BX

DEC CX

LOOP NEXT

MOV [BX]，AL

INT 3

**第二部分：**

**1、请自学并完成如下内容：**

**2、完成实训内容一；**

**3、将下面的程序编译连接，用DEBUG加载、跟踪，然后回答问题。**

**assume cs:code, ds:data, ss:stack**

**data segment**

**dw 0123h,0456h,0789h,0abch,0defh,0fedh,0cbah,0987h**

**data ends**

**stack segment**

**dw 0,0,0,0,0,0,0,0**

**stack ends**

**code segment**

**start: mov ax, stack**

**mov ss, ax**

**mov sp, 16**

**mov ax, data**

**mov ds, ax**

**push ds: [0]**

**push ds: [2]**

**pop ds: [2]**

**pop ds: [0]**

**mov ax, 4c00h**

**int 21h**

**code ends**

**end start**

**4、**

**程序如下，编写code段中的代码，将a段和b段中的数据依次相加，将结果存到c段中。**

**assume cs:code**

**a segment**

**db 1,2,3,4,5,6,7,8**

**a ends**

**b segment**

**db 1,2,3,4,5,6,7,8**

**b ends**

**c segment**

**db 0,0,0,0,0,0,0,0**

**c ends**

**code segment**

**start:**

**mov ax,a**

**mov ds,ax**

**mov cx,8**

**mov bx,0**

**s:**

**mov ax,[bx]**

**add bx,20h**

**mov [bx],ax**

**sub bx,10h**

**mov ax,[bx]**

**add bx,10h**

**add ax,[bx]**

**mov [bx],ax**

**sub bx,20h**

**inc bx**

**loop s**

**int 21h**

**code ends**

**end start**

**5、编辑录入下列程序，然后编译连接并执行。**

**assume cs:code;ds:data**

**data segment**

**out1 db 'Hello world$'**

**out2 db 'My name is MASM$'**

**data ends**

**code segment**

**start:**

**mov ax,data**

**mov ds,ax**

**lea dx,out1**

**mov ah,9**

**int 21h**

**mov dl,0ah**

**mov ah,2**

**int 21h**

**mov dl,0dh**

**mov ah,2**

**int 21h**

**lea dx,out2**

**mov ah,9**

**int 21h**

**mov ah,4ch**

**int 21h**

**code ends**

**end start**

**要求：正常运行并了解输出结果。**

**6、编辑录入下列程序，然后编译连接并执行。**

**CODE SEGMENT**

**ASSUME CS:CODE**

**START:**

**mov ah,1**

**int 21h**

**mov bl,al**

**sub bl,30h**

**mov dl,'+'**

**mov ah,2**

**int 21h**

**mov ah,1**

**int 21h**

**sub al,30h**

**add bl,al**

**mov dl,'='**

**mov ah,2**

**int 21h**

**mov al,bl**

**aam**

**or ax,3030h**

**xchg ah,al**

**mov dl,ah**

**mov ah,0eh**

**int 10h**

**mov al,dl**

**int 10h**

**mov ah,1**

**int 21h**

**MOV AH,4CH**

**INT 21H**

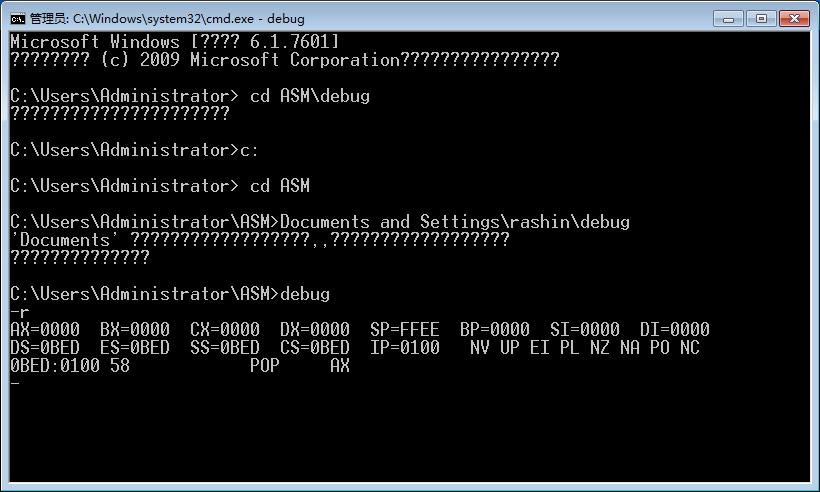
**CODE ENDS**

**END START**

**五、实验结果**

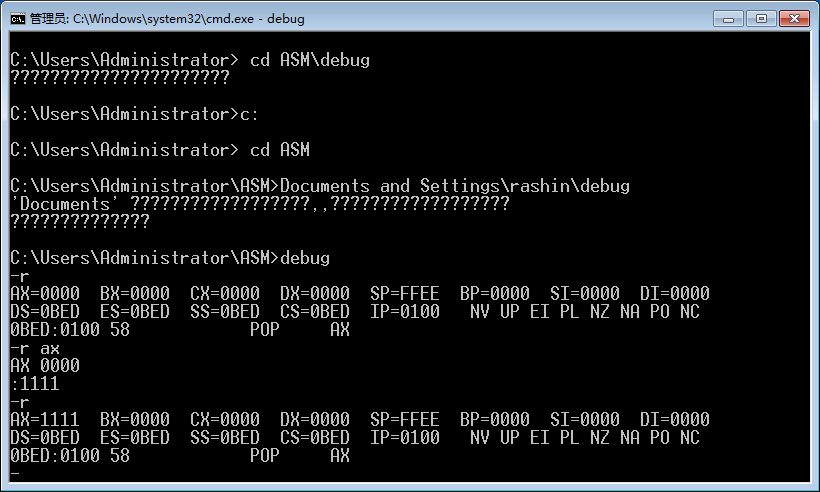
（把程序运行的结果通过截屏帖出来，并简单描述每个截图的意思）

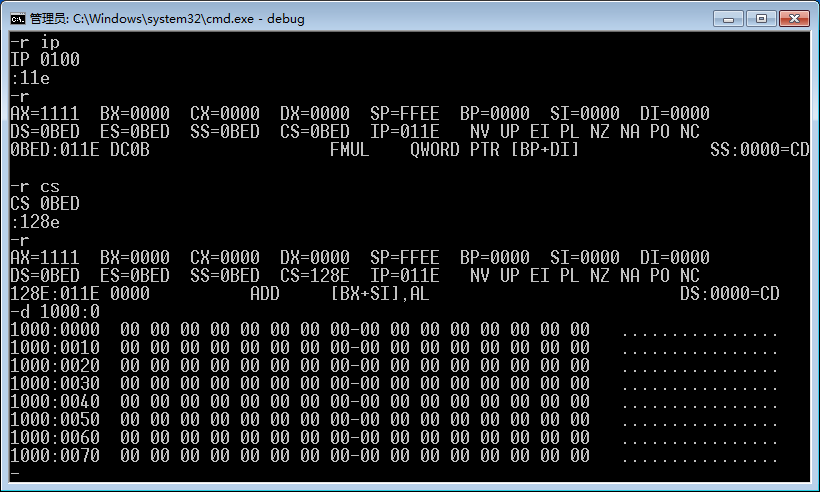
**第一部分**

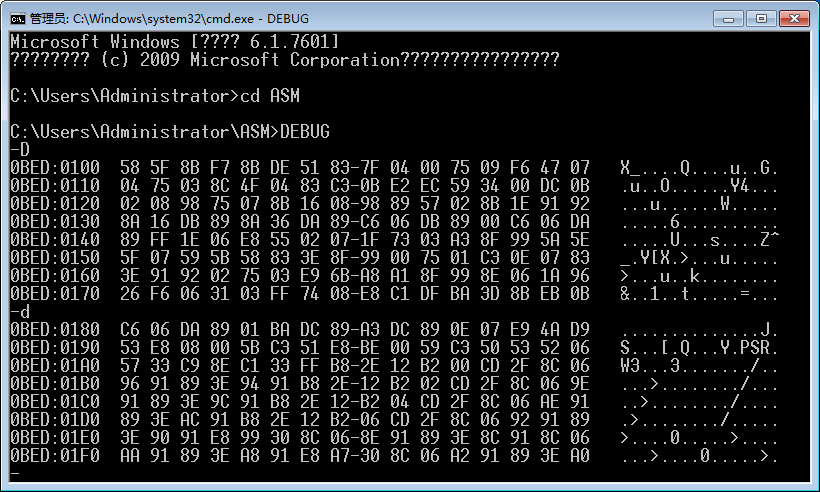


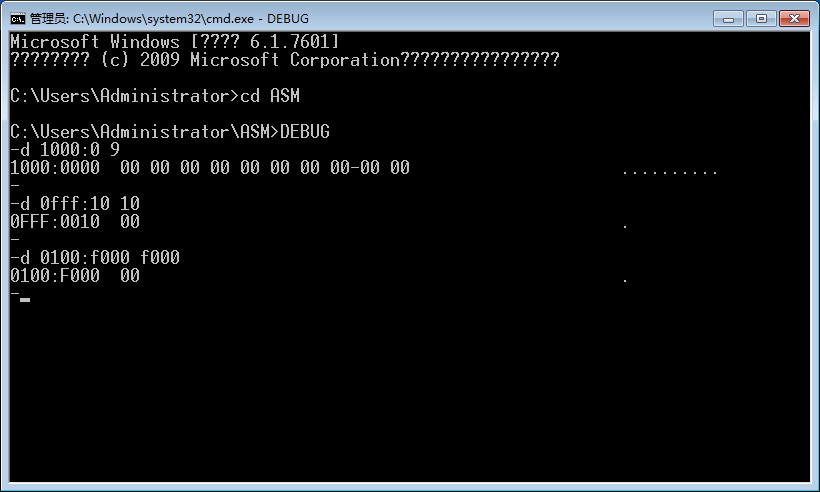
用r命令查看寄存器中内容

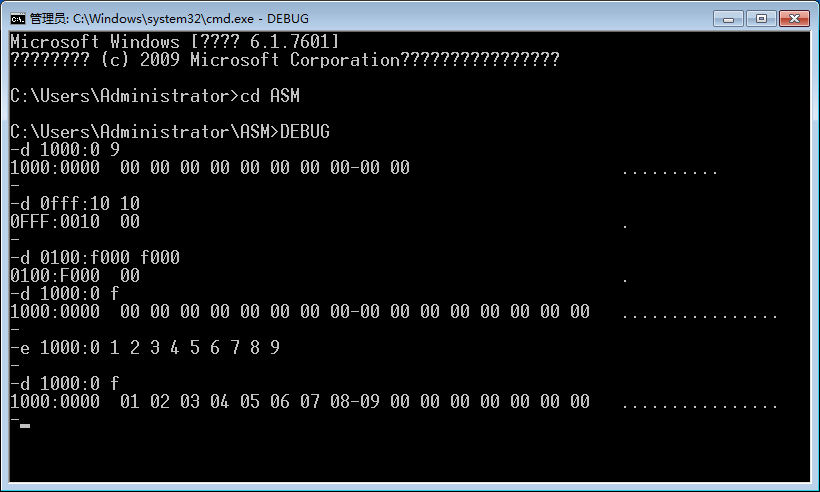
DS=0BED,IP=0100

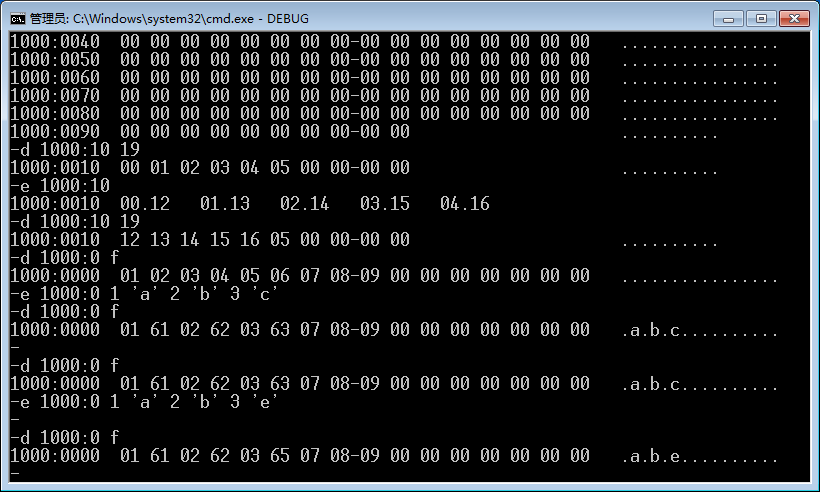
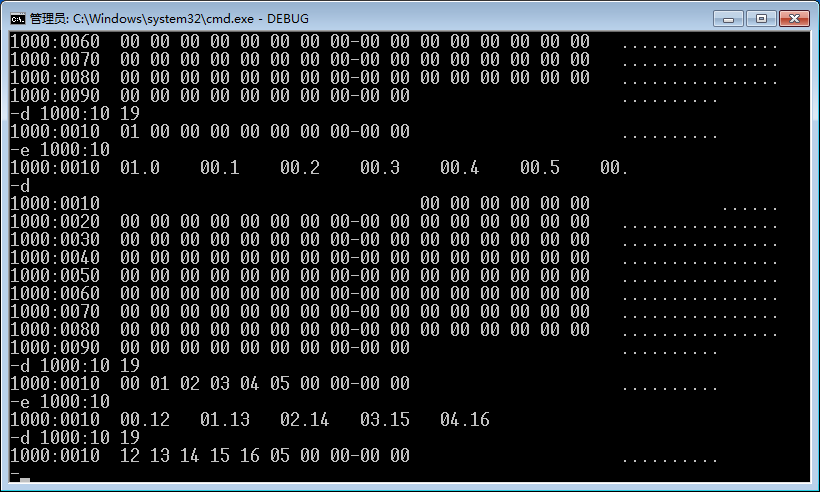


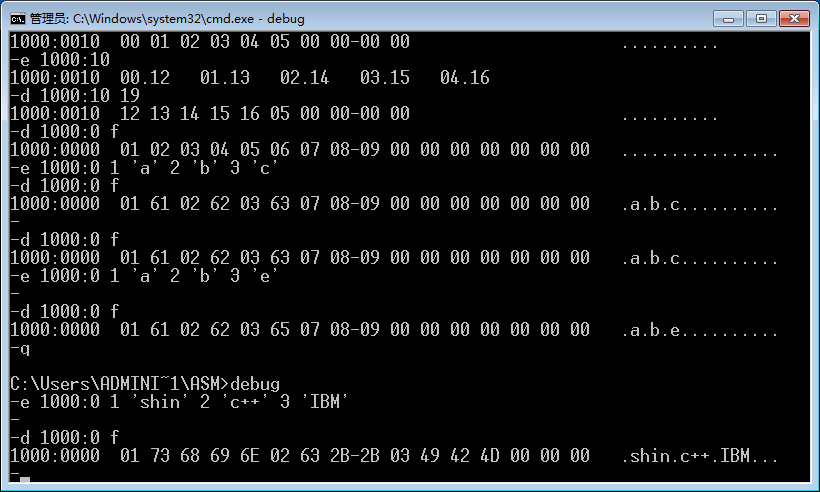


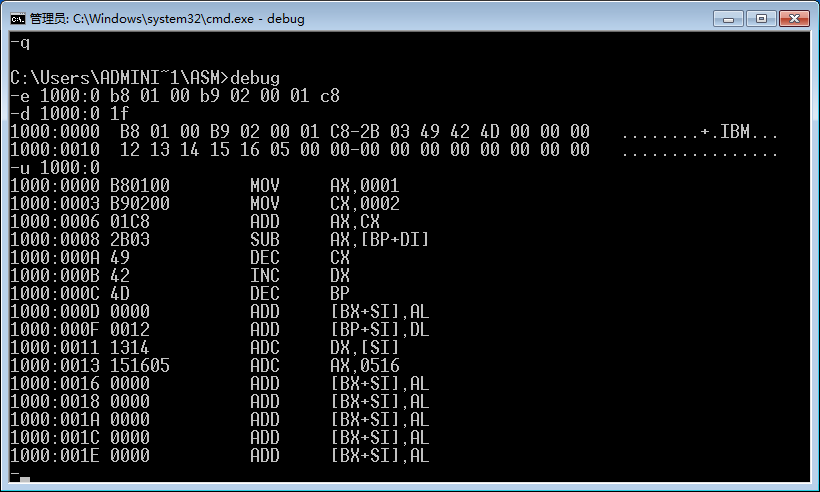


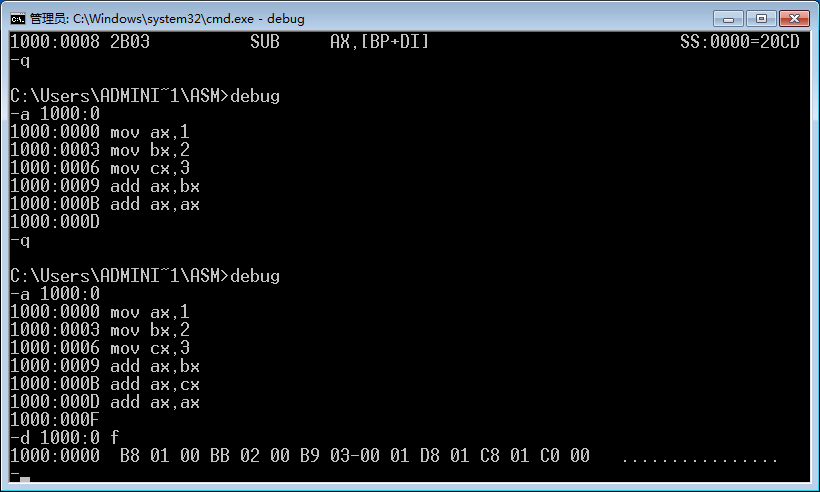




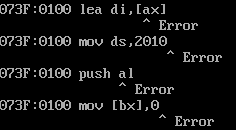








各种命令操作



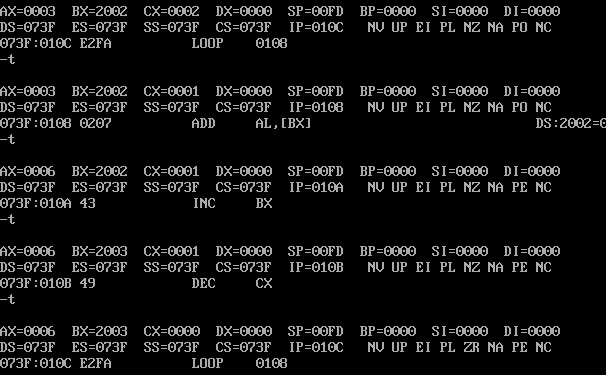
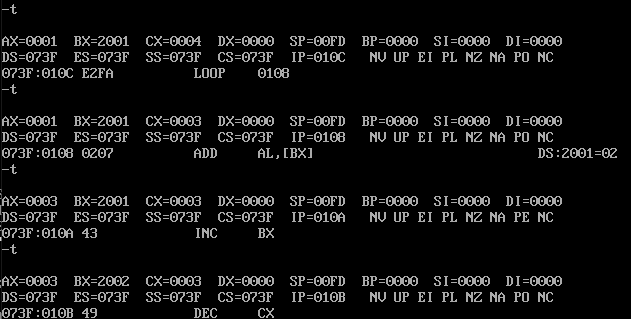
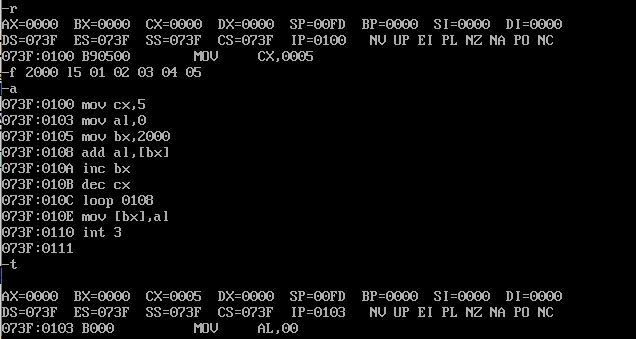
都错误，1.AX不能作为寄存器间接寻址

2.DS不能使用立即寻址法

3.AL不能使用PUSH

4.立即数不能赋给地址单元

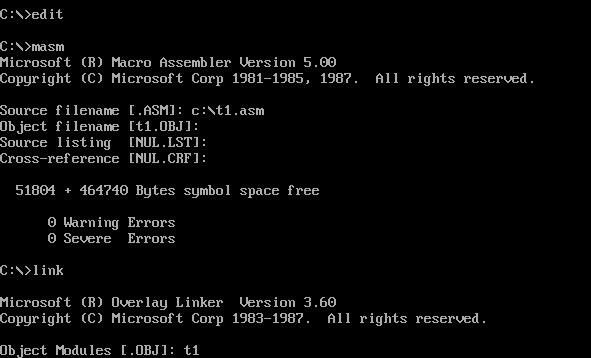
3.不能完成操作，loop和dex使cx减了两遍，不能将2000-20004中值全部相加。



第二部分

2.实训1

（1）程序运行体验实验

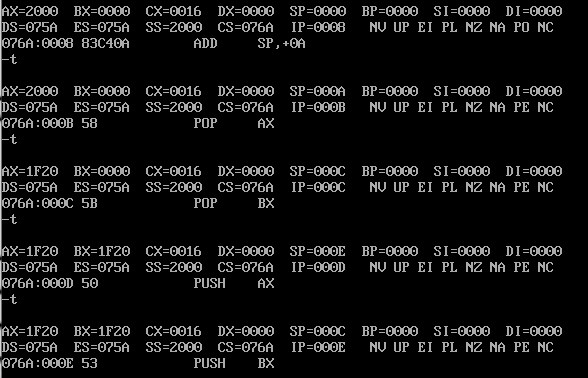




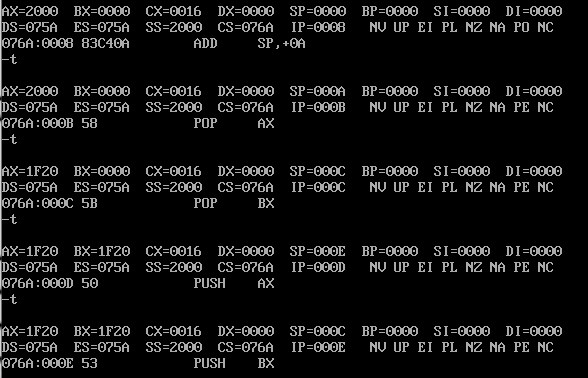
将2000存放在AX寄存器：



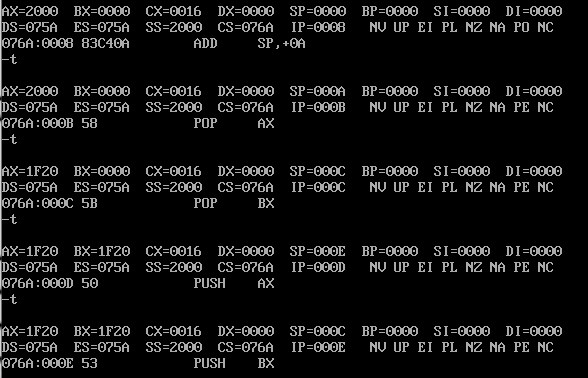
将AX中的操作数赋给SS堆栈段寄存器，并同时把0赋给SP：



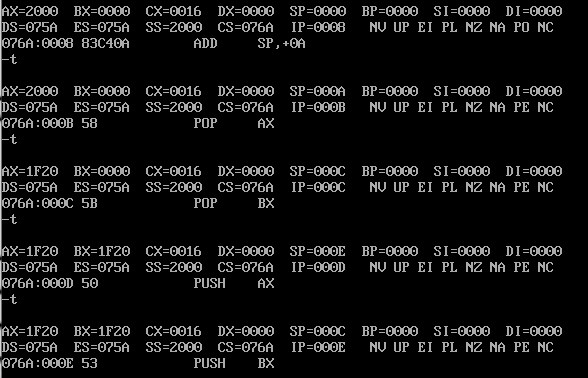
将0AH加上SP中操作数并赋给SP：



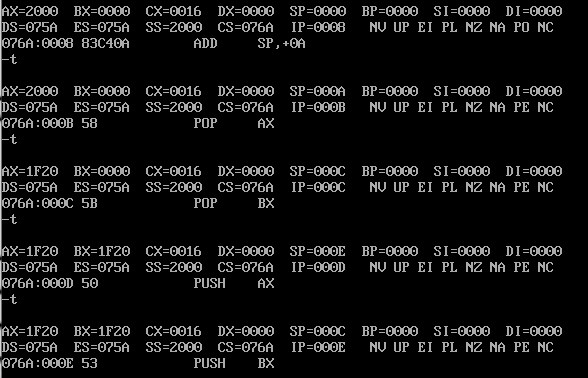
AX出栈，SP+2, AX中的值被堆栈地址覆盖,



BX出栈，SP+2, BX中的值被堆栈地址覆盖,



AX入栈，SP-2



BX入栈，SP-2



AX出栈，SP+2：



BX出栈，SP+2：



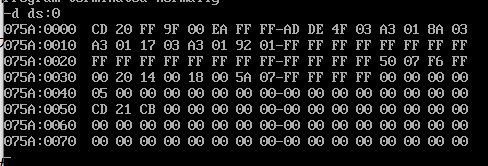
将4C00赋给AX



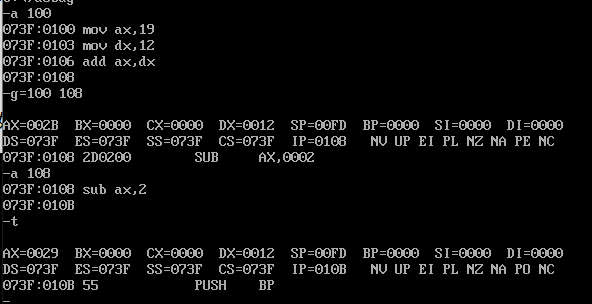
结束



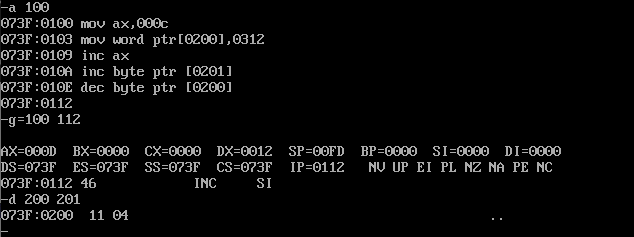
查看psp



**2、常用指令与寻址体验实验**

****

**和的结果为2B，存储在ax中，差的结果为29，存储在ax中。**

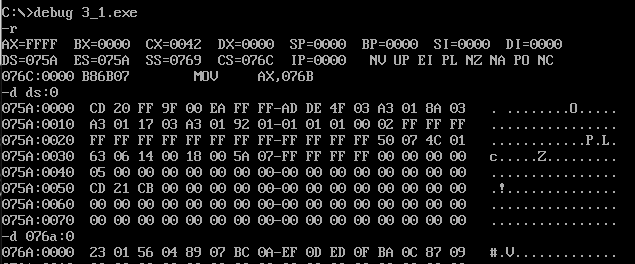
****

赋给AX 000C；将0312存入地址为[0200]中，其中低位12进入[0200],高位03进入[0201];

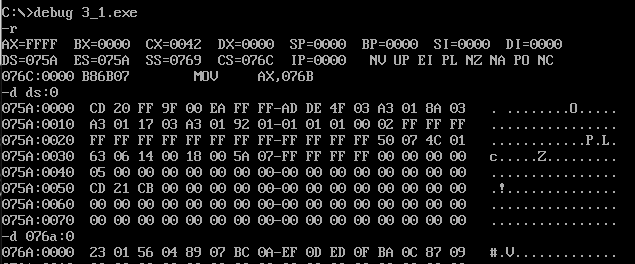
对AX中的值加一；对[0201]中值加一，对[0200]中值减一。

**3**

1. CPU执行程序，程序返回前，data段中的数据为多少？(回答并以屏幕截图证明)

Data值不变，****

1. CPU执行程序，程序返回前，cs、ss、ds的值分别为多少？(回答并以屏幕截图证明)

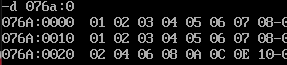
****

CS:076C; SS:0769; DS:075A

(3) 设程序加载后，code段的段地址为X，则data段的段地址为多少？stack的段地址为多少？

**data:X-2;stack:X-1**

**4**

****

**5**

****

**6.**

**输入6、5**

**六、实验心得体会**

（写出在做实验过程中碰到的问题、解决方法、存在的不足，实验过程中所获得的经验等等）

在做实验时，会遇到不清楚某个具体语句的用途，debug的时候遇到不清楚next怎么处理的困扰。在经过百度寻找以及实践操作debug后，获得了一些解决方法。在做实验时，我也存在很多不足，最开始的实验，但我花费了很长时间，对汇编的实验还没有十分上手，还需要多多训练。熟能生巧，多运用debug，能帮助自己理解更多，掌握更深。

**七、思考题**

**1.** **指令MOV [BX]，AX中，操作数[BX]的寻址方式是什么？在DEBUG下执行完该指令后，如何查看执行结果？**

寄存器间接寻址；输入r。

**2.** **可否随意在DEBUG提示符“-”后不带参数发出命令G？什么情况下使用命令G时，可不****用“=”给出执行首地址？**

不可以；当想要从CS:IP开始执行时，不需要用“=”给出执行首地址。

**八、参考文献**