**西安电子科技大学**

**微机系统综合实验 课程实验报告**

## **实验名称** 实验一 EMU8086使用及8086指令系统

网络与信息安全学院 2118021 班

成 绩

姓名 盖乐 学号 21009200991

同作者

实验日期 2023 年 4 月 12 日

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |

## **实验要求**

1. 熟悉并掌握EMU8086汇编语言编程调试环境;
2. 学习8086寻址方式（立即寻址、直接寻址、寄存器寻址、寄存器间接寻址、寄存器相对寻址、基址变址寻址、基址变址相对寻址、隐含寻址）。要求对以上寻址方式进行熟悉，观察不同寻址方式下指令运行结果（每种寻址方式2个实例，可参考PPT及教材相关章节）;
3. 学习8086指令系统，输入简单的指令（程序），观察各寄存器、内存相关单元以及处理器标志位的变化（数据传送类指令，算术运算类指令，逻辑运算类指令，标志处理和CPU控制类指令，移位和循环移位类指令，处理器控制类指令、程序控制类指令、输入/输出类指令等，要求每类指令至少2个用例。具体用例自行设计，可参考PPT及教材用例）;
4. 学习汇编语言程序设计的基本步骤和方法;
5. 学会使用EMU8086debug调试程序（单步运行、断点设置等）;
6. 学习汇编编程中DOS功能调用方法，掌握常用的DOS功能调用;
7. 编写一个简单的程序：将“ThisismyfirstASMprogram-姓名（汉语拼音各人的姓名）”放在DS=0700H，BX=0000H开始的存储器单元中，然后将该内容搬移到BX=0100H开始的单元中，最后将该字符串通过DOS功能调用显示在屏幕上。

## 实验目的

1. 学习EMU8086仿真开发环境的使用，理解和掌握汇编语言编程的基本步骤;
2. 熟悉并掌握8086/8088指令系统及内部结构;
3. 熟悉常用的DOS功能调用，掌握汇编语言编程的人机交互方法;
4. 熟悉变量、常量及伪指令的使用;
5. 熟悉内存单元的存储结构，字符串的处理以及简单的编程。

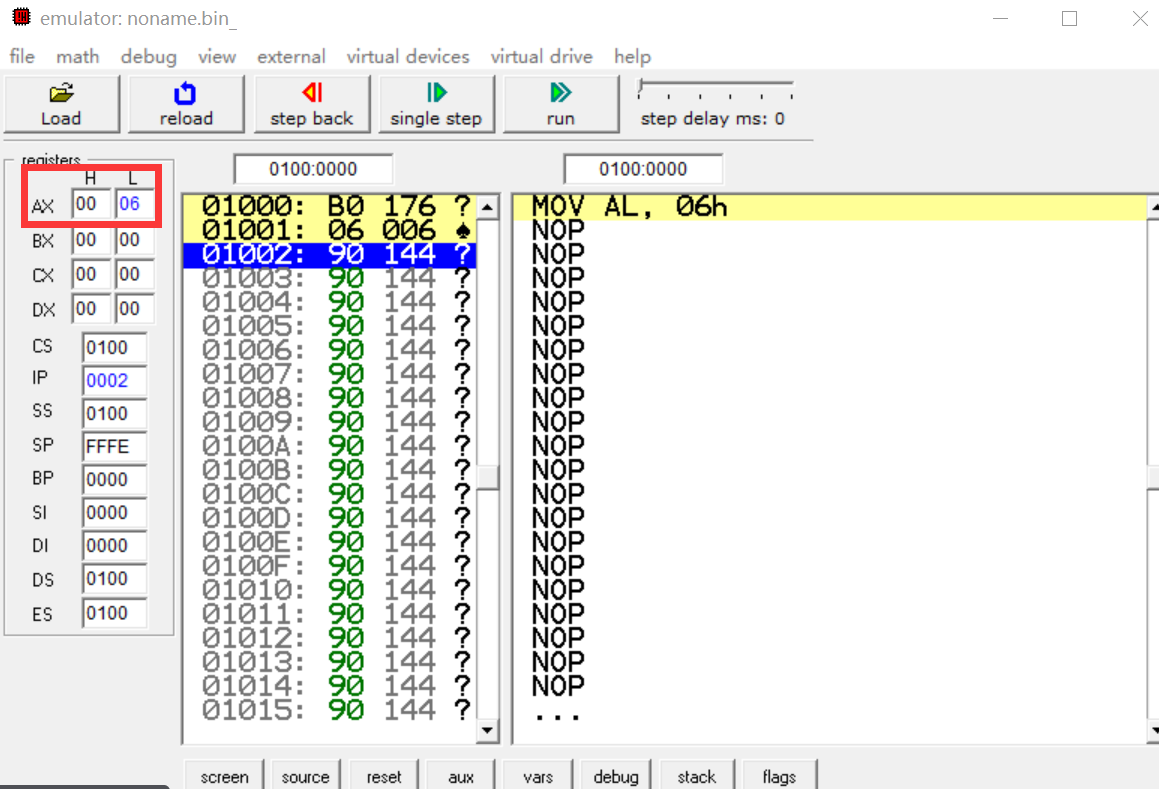
## 实验代码及实验结果

2.寻址方式

立即寻址：

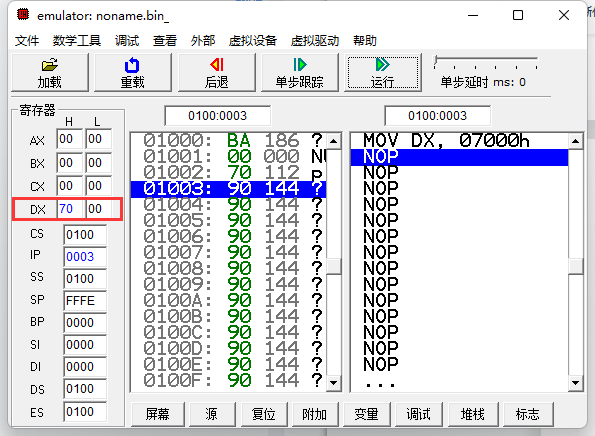
MOV AL,06H

指令含义：把立即数06H放入AL寄存器中



MOV DX,7000H

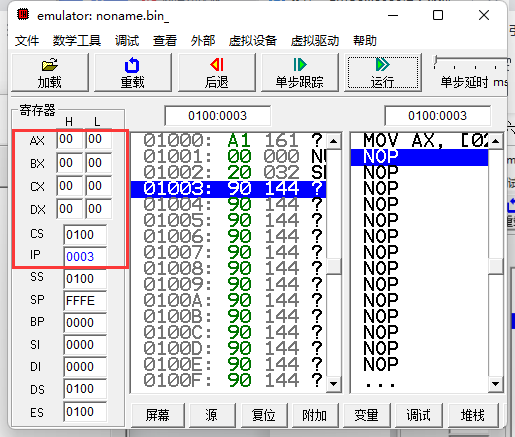
指令含义：把立即数7000H放入DX寄存器中



直接寻址：

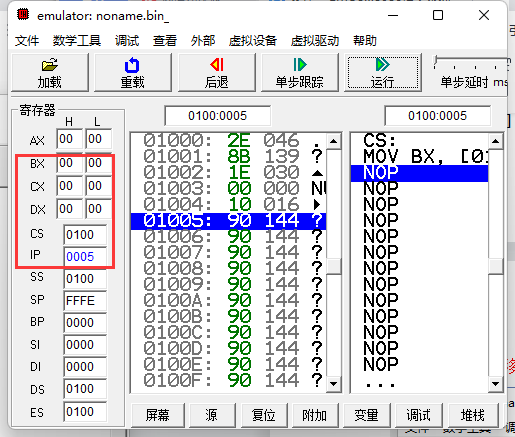
MOV AX,[2000H]

指令含义：将偏移地址为2000H的数据赋给AX



MOV BX,CS:[1000H]

指令含义：访问CS寄存器，将偏移地址为1000H的数据赋给BX



寄存器寻址：

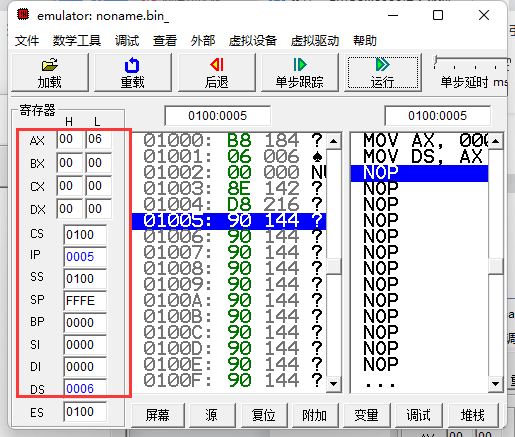
MOV AX,06H;

MOV DS,AX

指令含义：

把立即数06H放入AX寄存器

把AX寄存器中的数放入DS当中



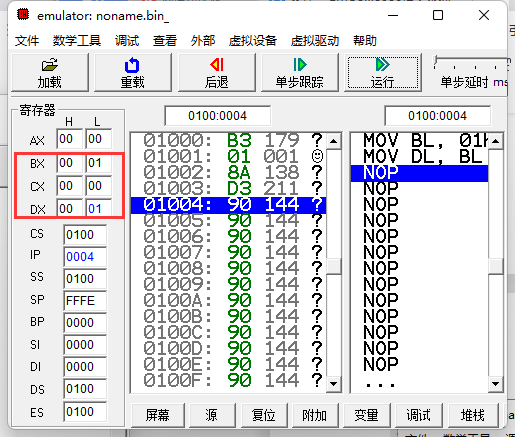
MOV BL,01H;

MOV DL,BL

指令含义：

把立即数01H放入BL寄存器中

把BL寄存器中的数赋给DL寄存器

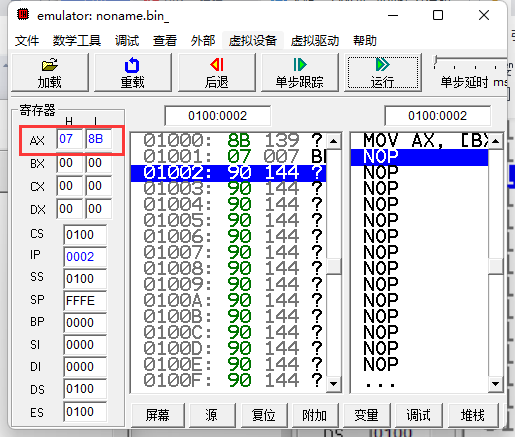


寄存器间接寻址：

MOV AX,[BX]

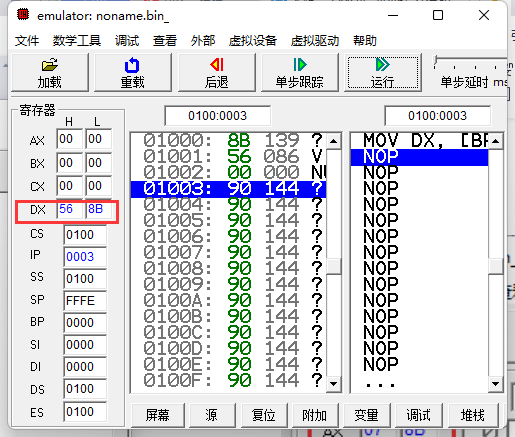
指令含义：

将BX寄存器中的值作为内存地址，然后将该地址处的16位数值读取到AX寄存器中。



MOV DX,[BP]

指令含义：

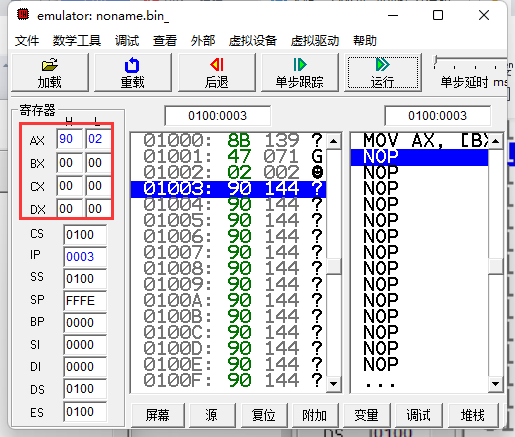
将存储在[BP]地址处的数据传送到DX寄存器中。

寄存器相对寻址：

MOV AX,[BX+2]

指令含义：

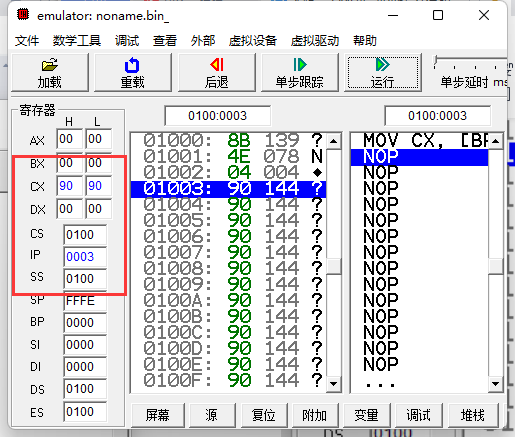
将内存地址中偏移量为BX+2的位置存储的数据传送到AX寄存器中。



MOV CX,[BP+4]

指令含义：

将内存地址中偏移量为BP+4的位置存储的数据传送到CX寄存器中。

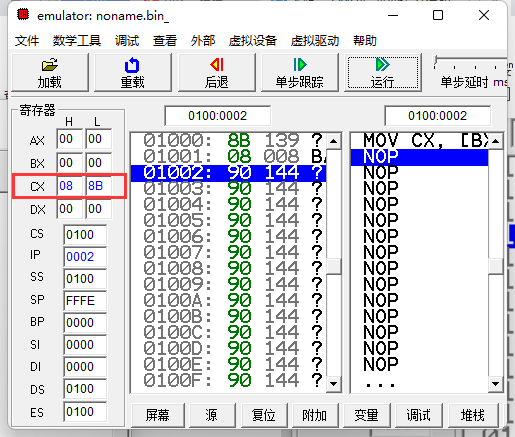


基址变址寻址：

MOV CX,[BX][SI]

指令含义：

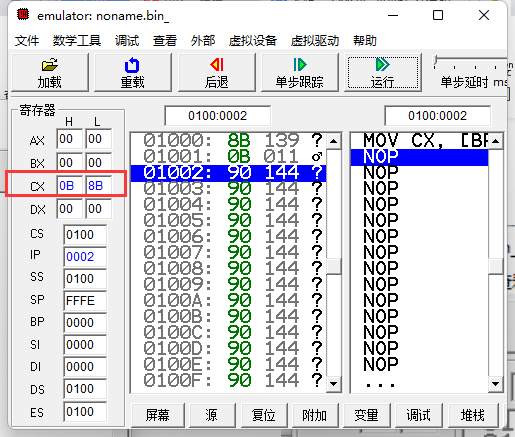
使用寄存器BX和SI中存储的数值作为偏移量，并将它们相加以得出内存地址。然后，该指令从该内存地址读取数据，并将其存储到CX寄存器中。



MOV CX,[BP][DI]

指令含义：

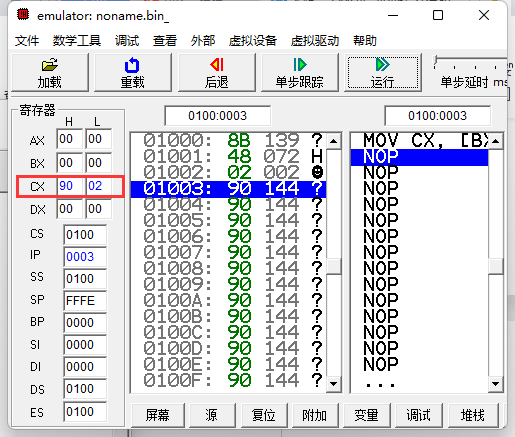
使用寄存器BP和DI中存储的数值作为偏移量，并将它们相加以得出内存地址。然后，该指令从该内存地址读取数据，并将其存储到CX寄存器中。



基址变址相对寻址：

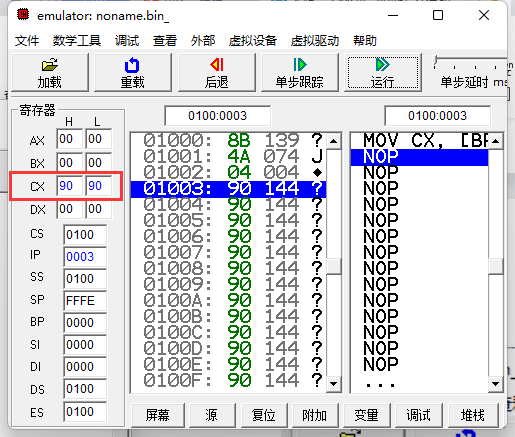
MOV CX,[BX][SI]+2

指令含义：

使用寄存器BX和SI中存储的数值作为偏移量，并将它们相加以得出内存地址,在地址计算完成后，将计算结果值加上2，作为最终的内存地址。然后，该指令从该内存地址读取数据，并将其存储到CX寄存器中。

MOV CX,[BP][SI]+4

指令含义：

使用寄存器BP和SI中存储的数值作为偏移量，并将它们相加以得出内存地址,在地址计算完成后，将计算结果值加上4，作为最终的内存地址。然后，该指令从该内存地址读取数据，并将其存储到CX寄存器中。

3.指令系统

数据传送指令：

LDS SI,[BX]

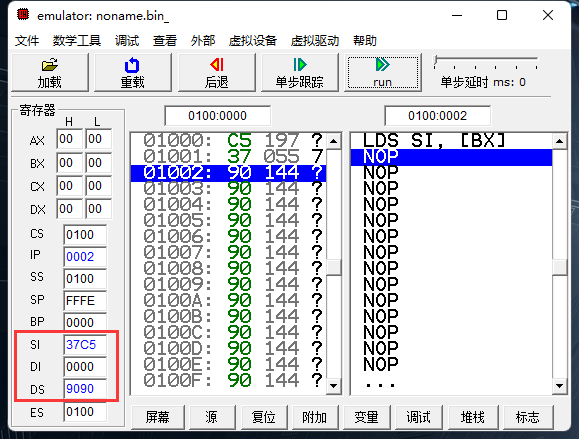
指令含义：

它用于将一个内存地址的16位偏移量和16位段地址加载到DS和SI寄存器中。

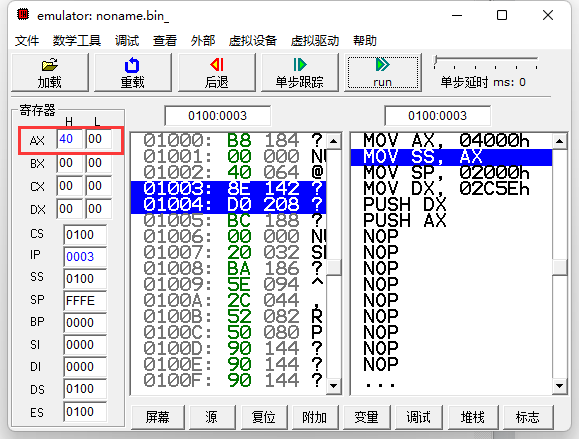
具体来说，这条指令的执行过程如下：

从内存地址[BX]处读取一个16位段地址，并将其存储到DS寄存器中。

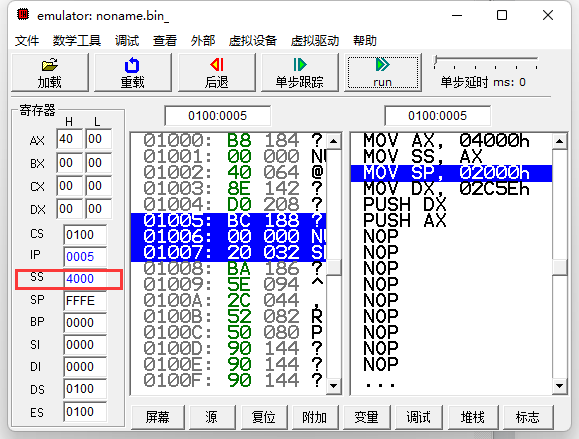
从内存地址[BX+2]处读取一个16位偏移量，并将其存储到SI寄存器中。



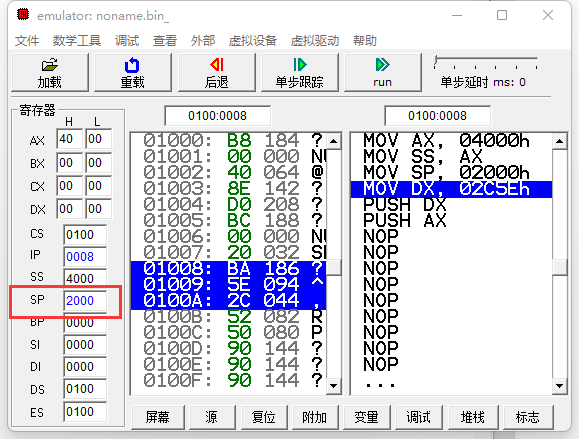
MOV AX,4000H把立即数4000H放入AX寄存器



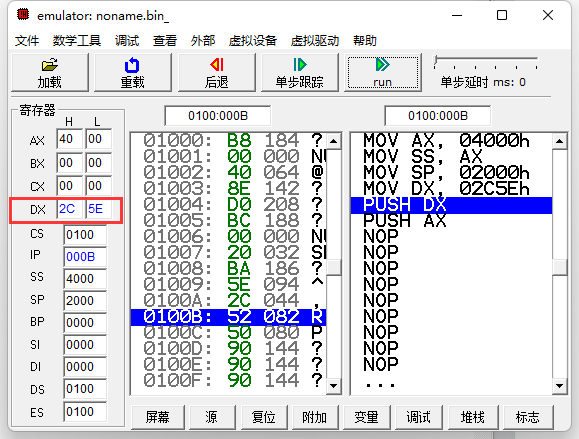
MOV SS,AX 把AX寄存器中的数放入SS



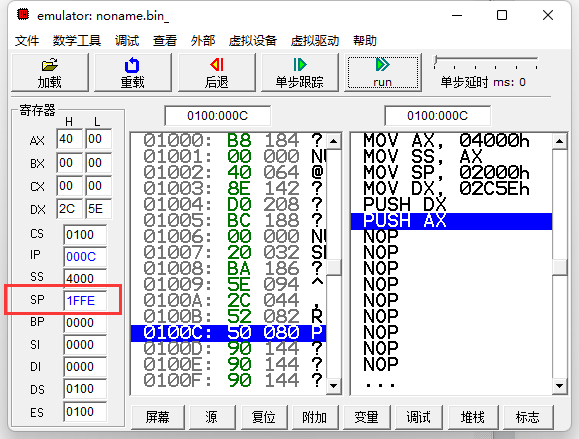
MOV SP,2000H 把立即数2000H放入SP寄存器



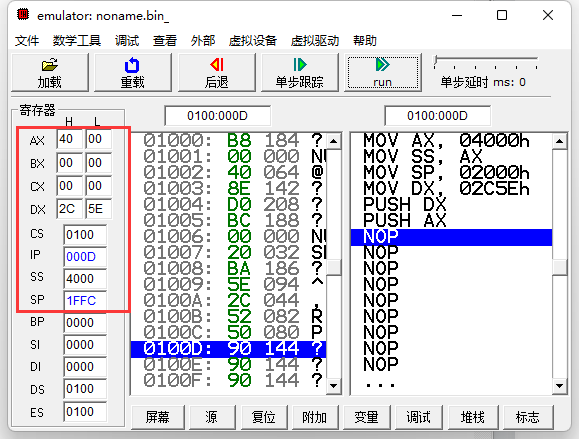
MOV DX,2C5EH 把立即数2C5EH放入DX寄存器



PUSH DX 把DX中的操作数压入栈内



PUSH AX 把AX中的操作数压入栈内

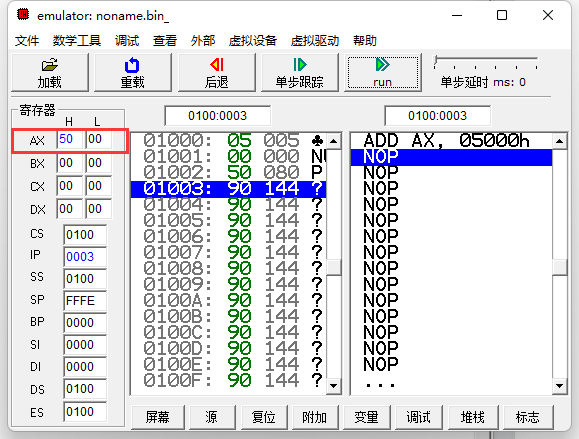


算术运算指令：

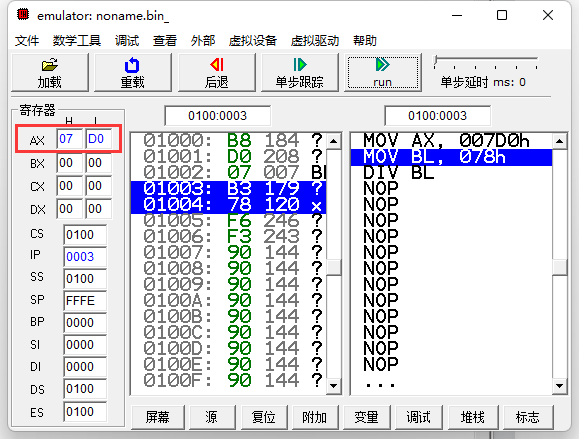
ADD AX,5000H

指令含义：

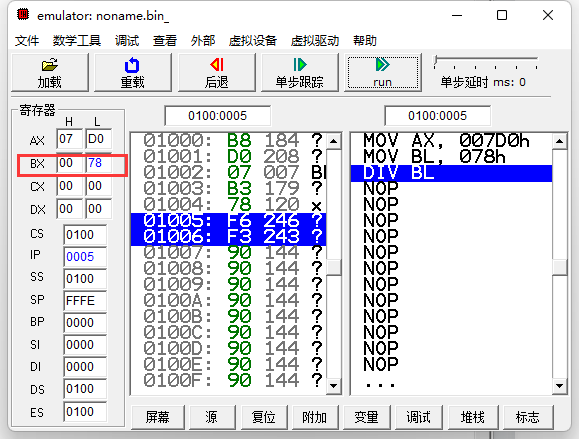
AX中内容为0000H，它用于将AX寄存器的值与立即数5000H相加，并将结果存储回AX寄存器中。



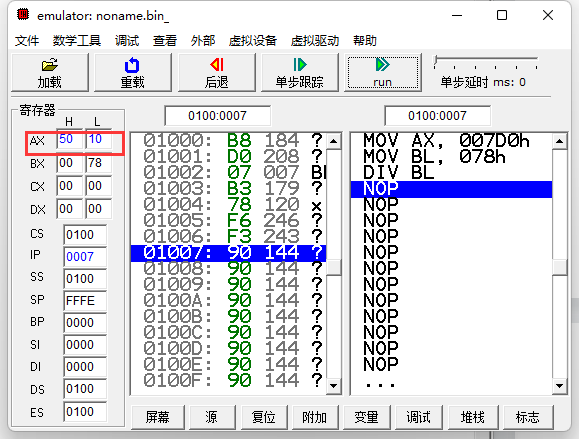
MOV AX,2000 把十进制数2000放入AX寄存器



MOV BL,120 把十进制数120放入BL寄存器

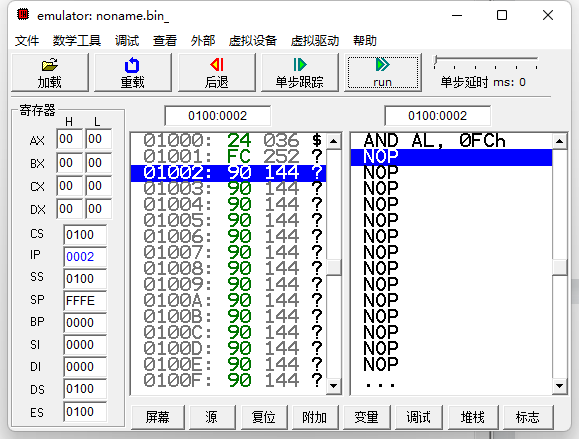


DIV BL 将AX中的操作数除BL中操作数，并把商保存在AH中，余数保存在AL中

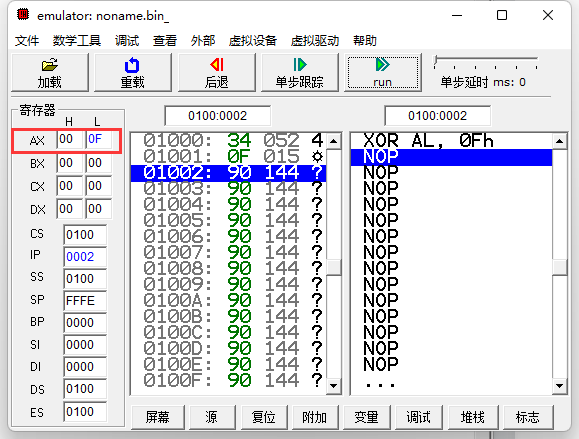


逻辑运算指令：

AND AL,0FCH 把立即数0FCH按位与AL与，结果放入AL



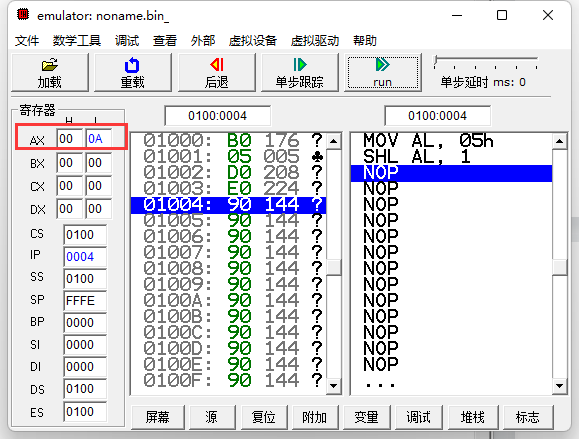
XOR AL,0FH 将立即数0FH与AL中操作数按位异或，放入AL中



移位和循环移位类指令:

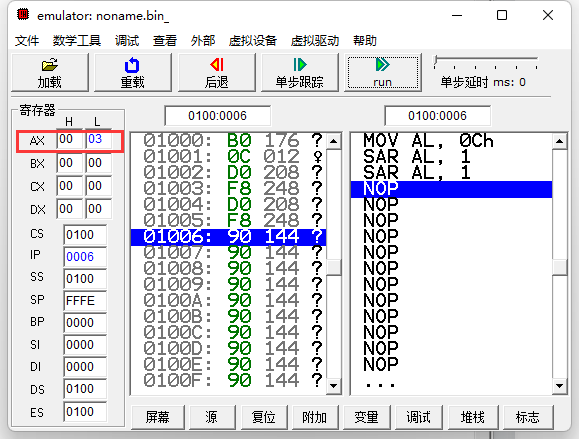
MOV AL,05H; 将立即数 05H 存储到寄存器 AL 中

SAL AL,1; 将寄存器 AL 中的值左移一位（相当于乘以 2），并将结果 保存回 AL 寄存器



MOV AL,12; 将立即数 12 存储到寄存器 AL 中

SAR AL,2; 将寄存器 AL 中的值右移两位（相当于除以 2 的平方，即除以 4），并将结果保存回 AL 寄存器



处理器控制指令：

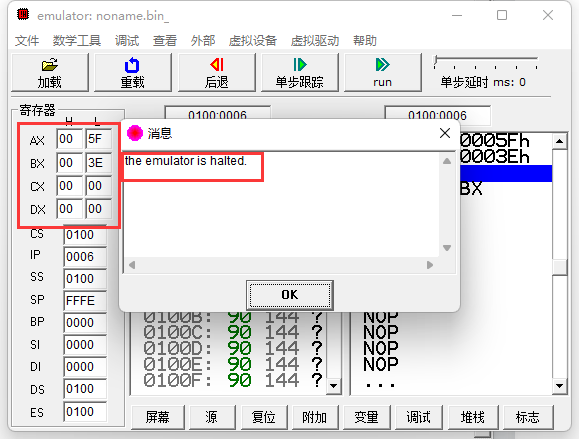
MOV AX,05FH; 把立即数05FH放入AX寄存器

MOV BX,03EH; 把立即数03EH放入BX寄存器

HLT; 中断

ADD AX,BX; 把BX中操作数加到AX中

HLT 指令是一个处理器控制指令，它会停止 CPU 的执行，并让 CPU 进入等待状态，直到外部中断被触发。因此，执行完 HLT 指令后，CPU 停止执行指令，不再继续执行 ADD AX, BX。



MOV AX,4000H; 将4000H放入AX

MOV SS,AX; 将地址位AX的操作数放入SS

MOV SP,2000H; 将2000H放入SP

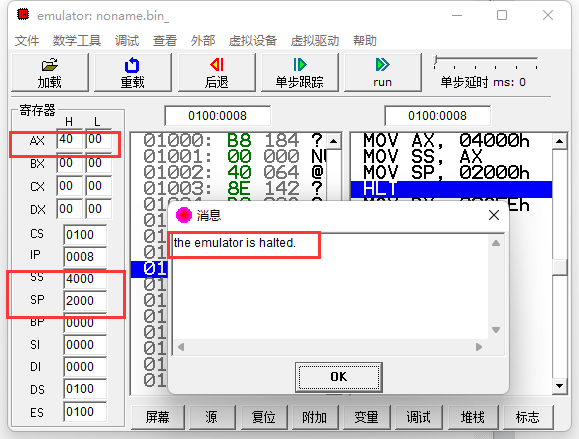
HLT; 中断

MOV DX,2C5EH; 将2C5EH放入DX

PUSH DX; 将DX压入栈

PUSH AX; 将AX压入栈

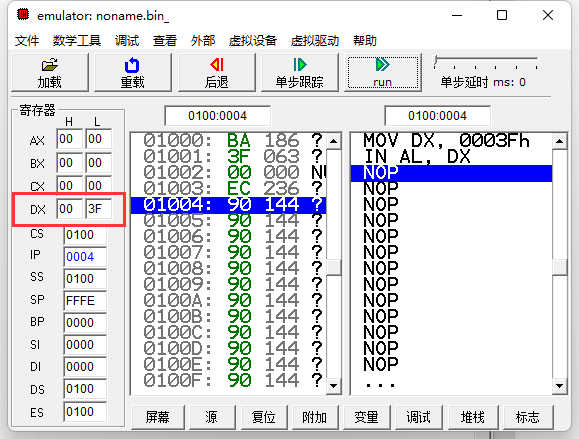
将一个值（4000H）存储到AX寄存器中，然后将AX寄存器的值复制到SS寄存器中。接着将2000H存储到SP寄存器中，这里的SP是栈指针寄存器，用于指示当前栈顶的位置。然后使用HLT指令停止处理器的执行，直到有外部中断信号触发后再继续执行。



输入/输出类指令：

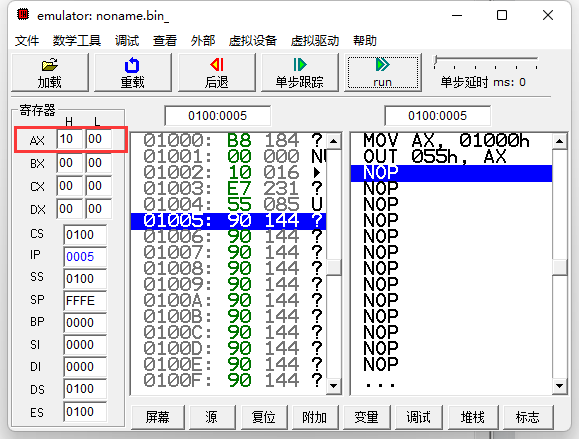
MOV DX,03FH; 将立即数 03FH 送入 DX 中

IN AL,DX; 由接口地址03H读入一个字节到 AL寄存器



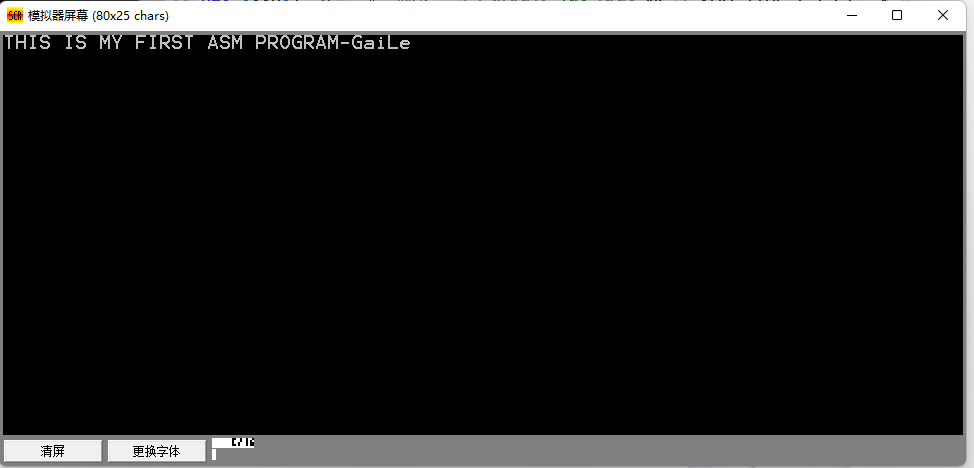
MOV AX,1000H； 将立即数 1000H 送入 AX 中

OUT 55H,AX; 将 AX 中内容由55H 接口地址输出



7.实验代码：

1. MOV AX, 0700H       把立即数0700H放入AX
2. MOV DS, AX          把地址为AX的操作数放到DS
3. MOV BX,0000H        把0000H放入BX
4. MOV [02H], 'T'      把字符T放入[02H]地址
5. MOV [04H], 'H'
6. MOV [06H], 'I'
7. MOV [08H], 'S'
8. MOV [0AH], ' '
9. MOV [0CH], 'I'
10. MOV [0EH], 'S'
11. MOV [10H], ' '
12. MOV [12H], 'M'
13. MOV [14H], 'Y'
14. MOV [16H], ' '
15. MOV [18H], 'F'
16. MOV [1AH], 'I'
17. MOV [1CH], 'R'
18. MOV [1EH], 'S'
19. MOV [20H], 'T'
20. MOV [22H], ' '
21. MOV [24H], 'A'
22. MOV [26H], 'S'
23. MOV [28H], 'M'
24. MOV [2AH], ' '
25. MOV [2CH], 'P'
26. MOV [2EH], 'R'
27. MOV [30H], 'O'
28. MOV [32H], 'G'
29. MOV [34H], 'R'
30. MOV [36H], 'A'
31. MOV [38H], 'M'
32. MOV [3AH], '-'
33. MOV [3CH], 'G'
34. MOV [3EH], 'a'
35. MOV [40H], 'i'
36. MOV [42H], 'L'
37. MOV [44H], 'e'
38. MOV CX,0028H        把0028H放入CX寄存器
39. MOV BX,0100H        把0100H放入BX寄存器
40. MOV SI,0002H        把0002H放入SI寄存器
41. COPY:
42. MOV AX,DS:[SI]
43. MOV DS:[BX+SI],AX
44. ADD SI,2
45. LOOP COPY           重复COPY操作
46. MOV BX,0102H
47. MOV CX,0028H
48. PRINTF:
49. MOV DL,[BX]
50. MOV AH,02H
51. INT 21H
52. ADD BX,2
53. LOOP PRINTF
54. MOV AH,1
55. INT 21H
57. MOV AH,4CH
58. INT 21H
59. RET



## 实验总结

1. 通过实践掌握了一些基础的汇编语言指令（数据传送指令 MOV、地址传送指

令 LDS、加法指令 ADD、堆栈操作指令 PUSH、除法指令 DIV、逻辑运算指

令等等）

2. 在实践的过程中更深入理解了 8086 的各种寻址方式（立即寻址、直接寻址、

寄存器寻址、寄存器间接寻址、寄存器相对寻址、基址变址寻址、基址变址

相对寻址、隐含寻址）

3. 学会了汇编语言程序设计的基本步骤和方法；

4. 学会了汇编编程中 DOS 功能调用方法，掌握了常用的 DOS 功能调用；

5. 学会了使用 EMU8086 debug 调试程序（单步运行、断点设置等）；