

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

姓名 盖乐 学号 21009200991

同作者 _____

实验日期 2023 年 4 月 12 日

成 绩

指导教师评语：

指导教师：

_____年____月____日

一、 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。

二、 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。

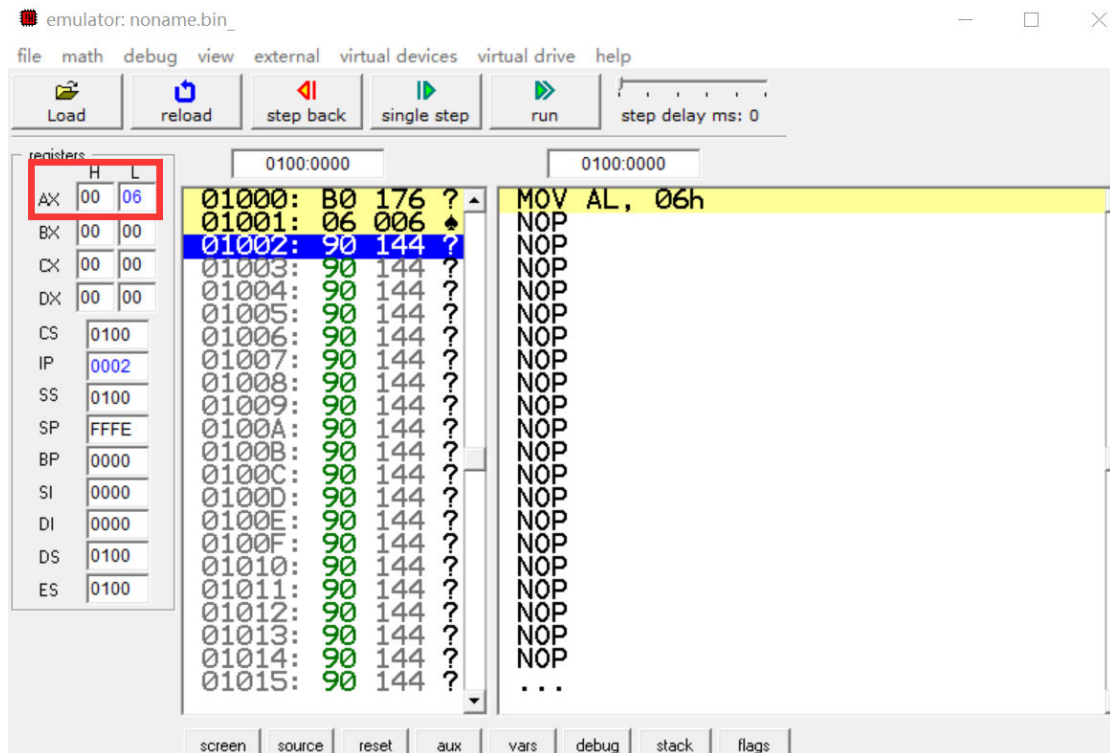
三、 实验代码及实验结果

2.寻址方式

立即寻址：

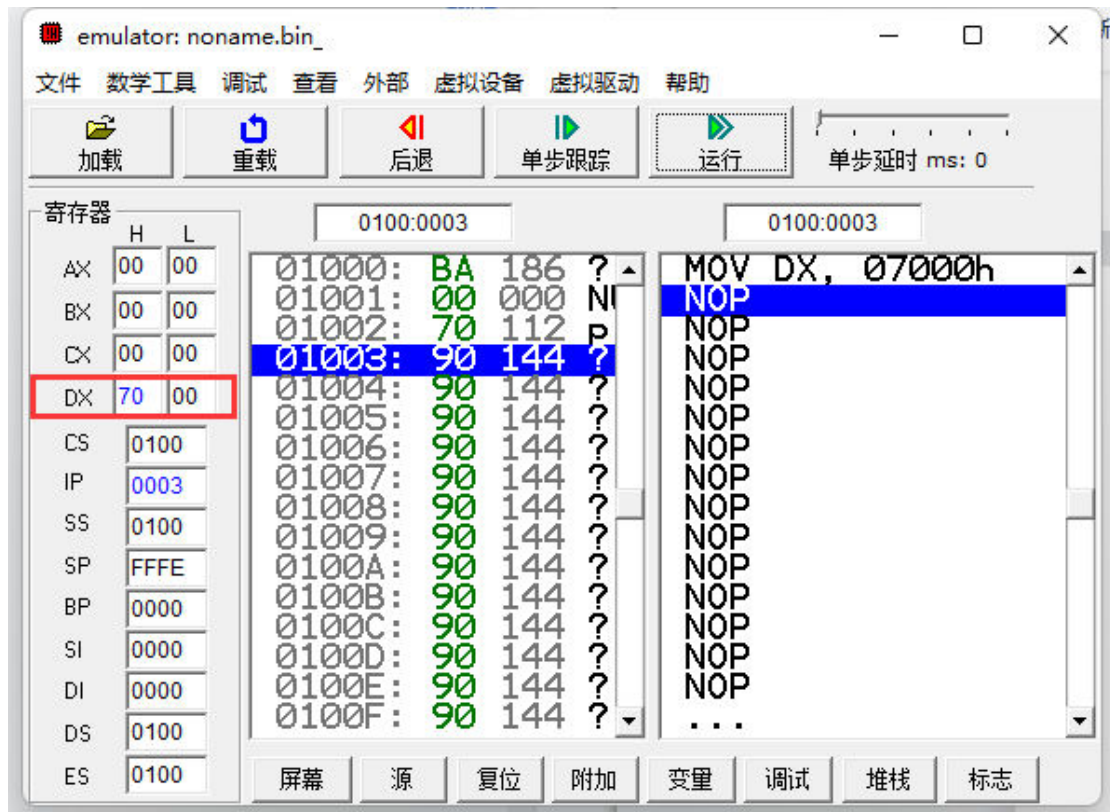
MOV AL,06H

指令含义：把立即数 06H 放入 AL 寄存器中



MOV DX,7000H

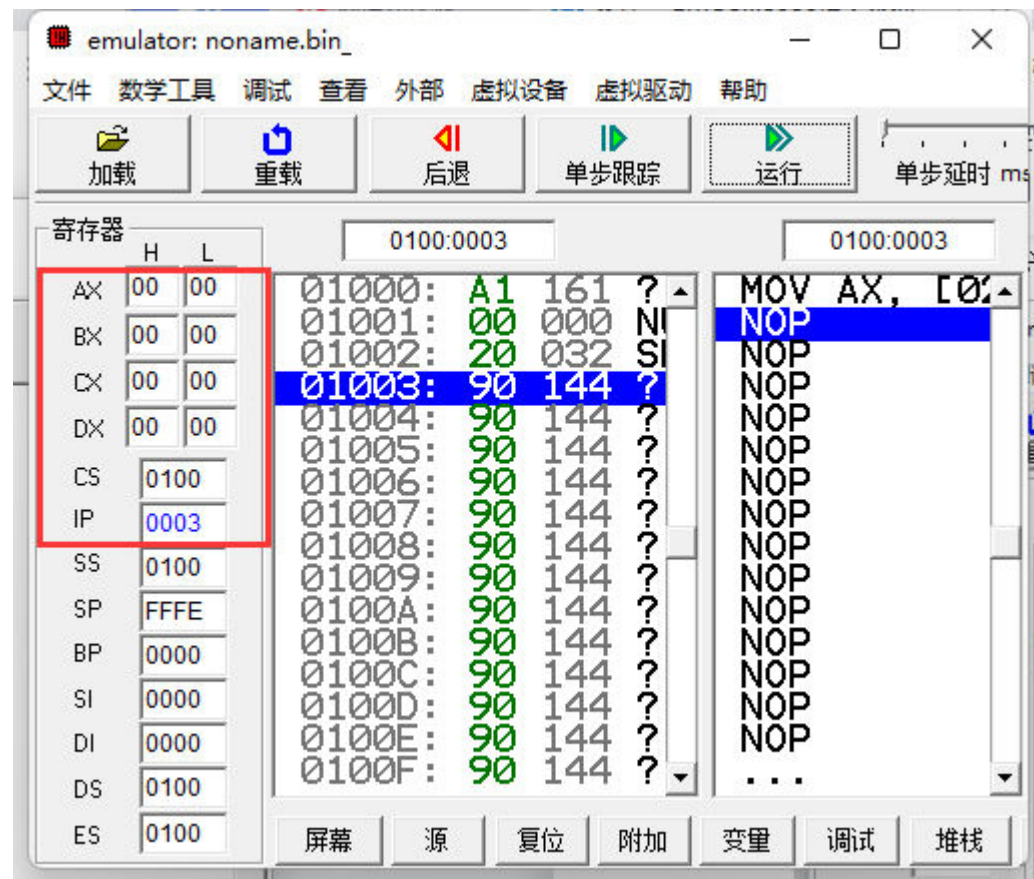
指令含义：把立即数 7000H 放入 DX 寄存器中



直接寻址：

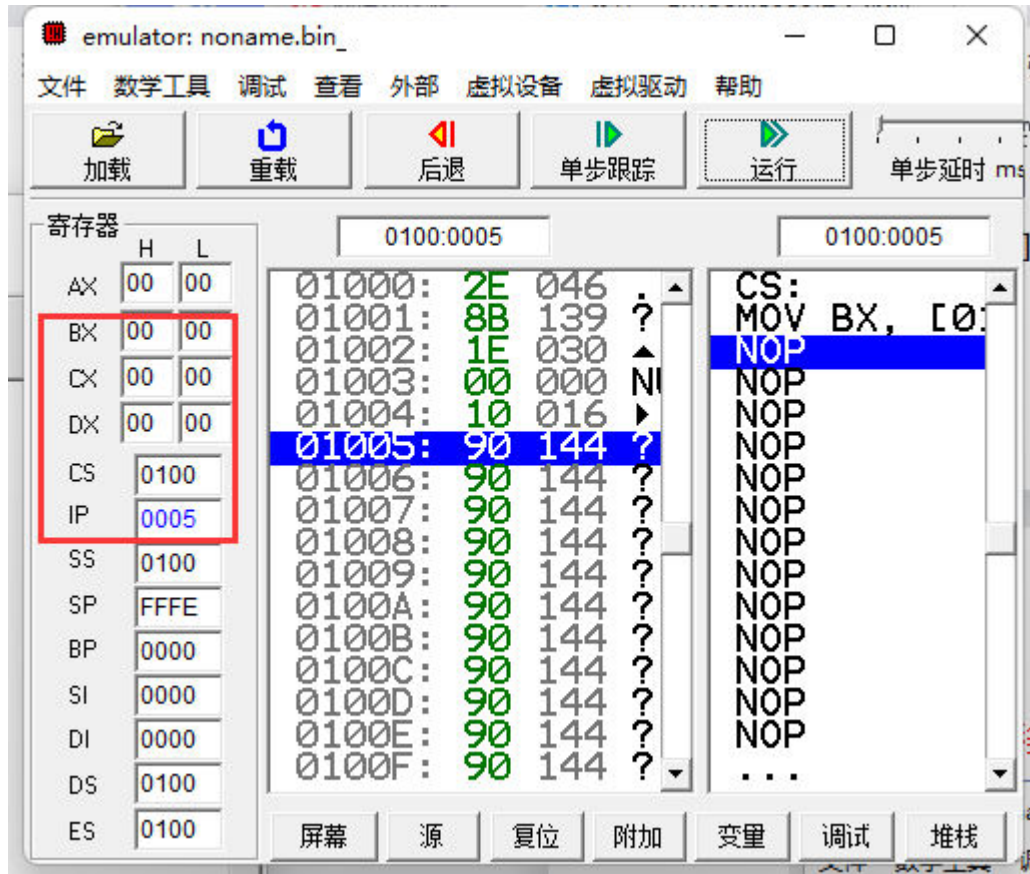
MOV AX,[2000H]

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:

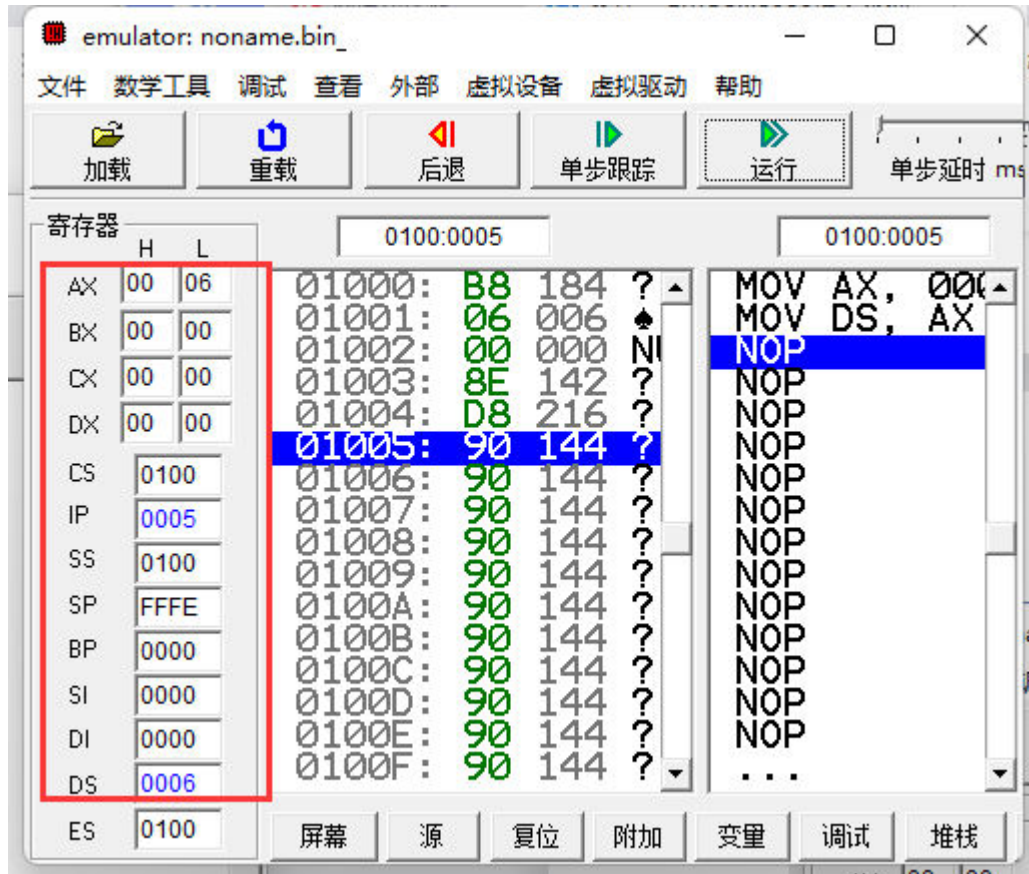
MOV AX,06H;

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



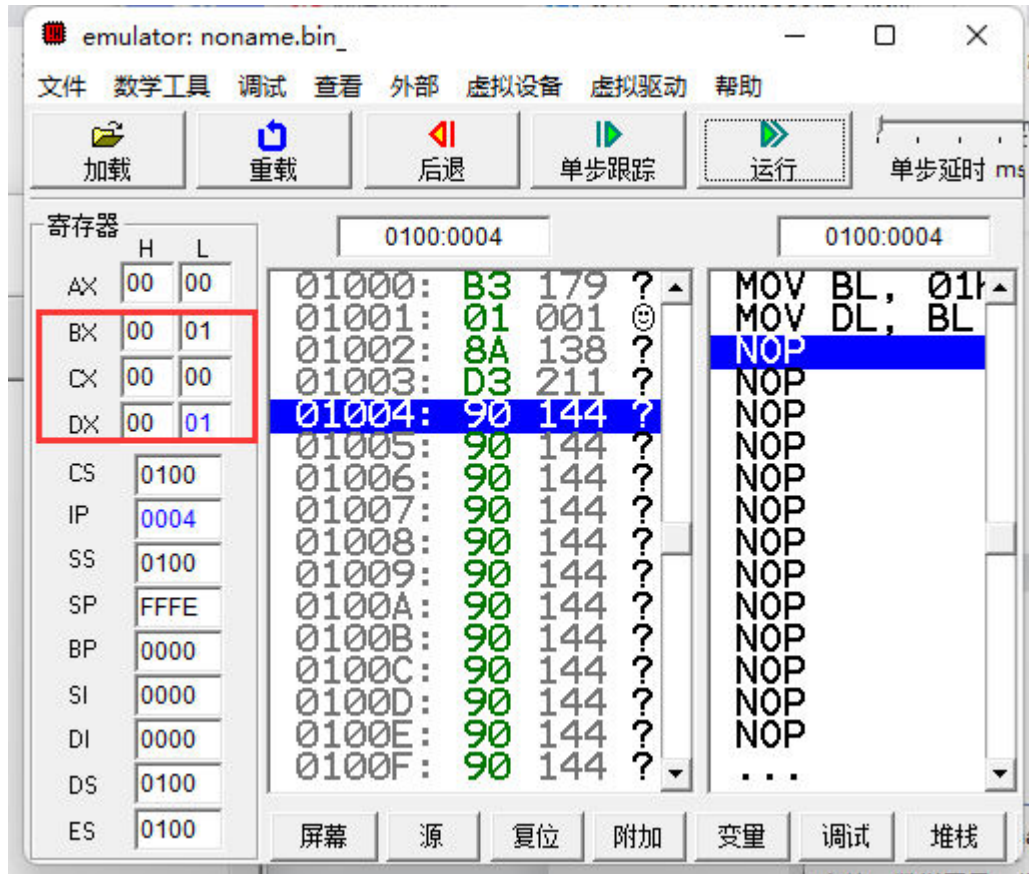
MOV BL,01H;

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

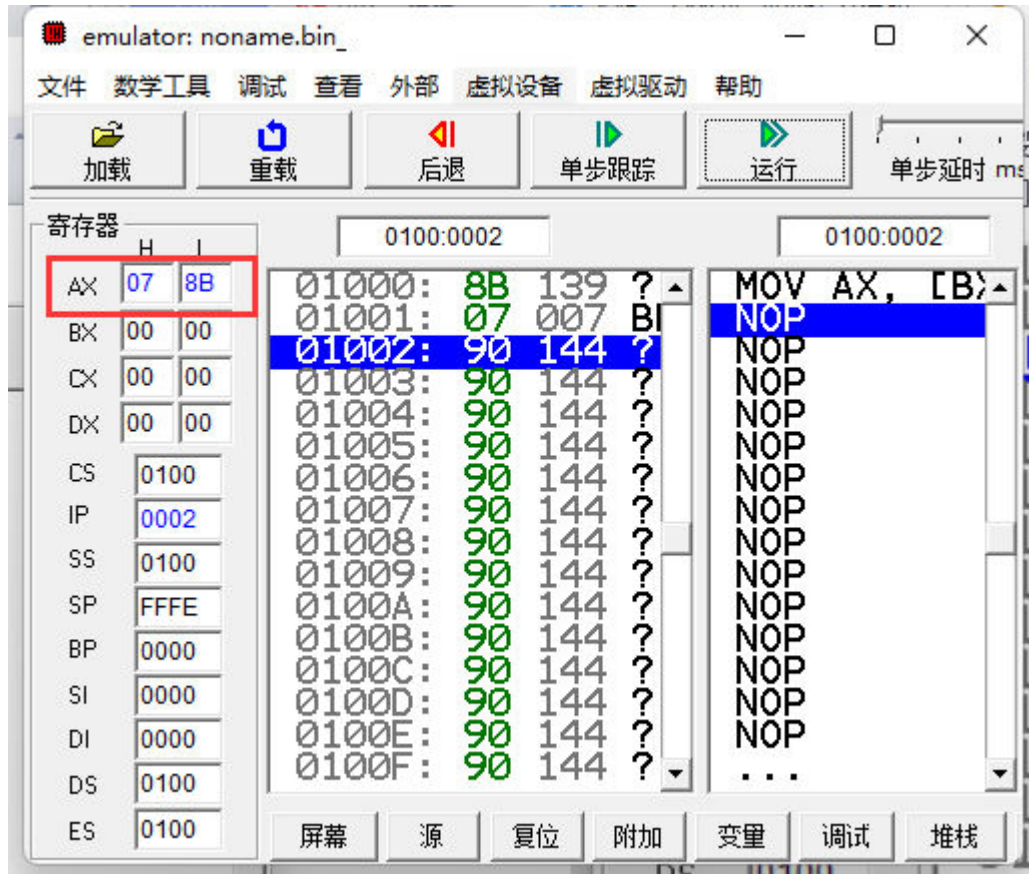


寄存器间接寻址：

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。

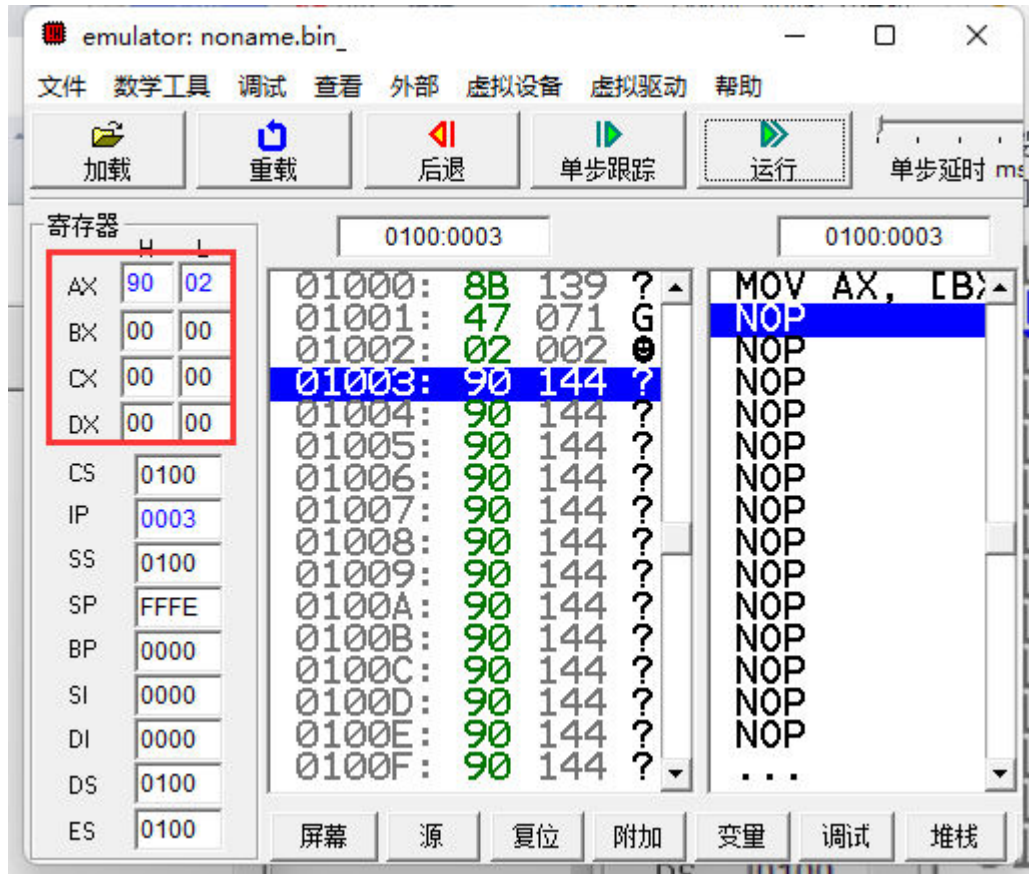


寄存器相对寻址:

MOV AX,[BX+2]

指令含义:

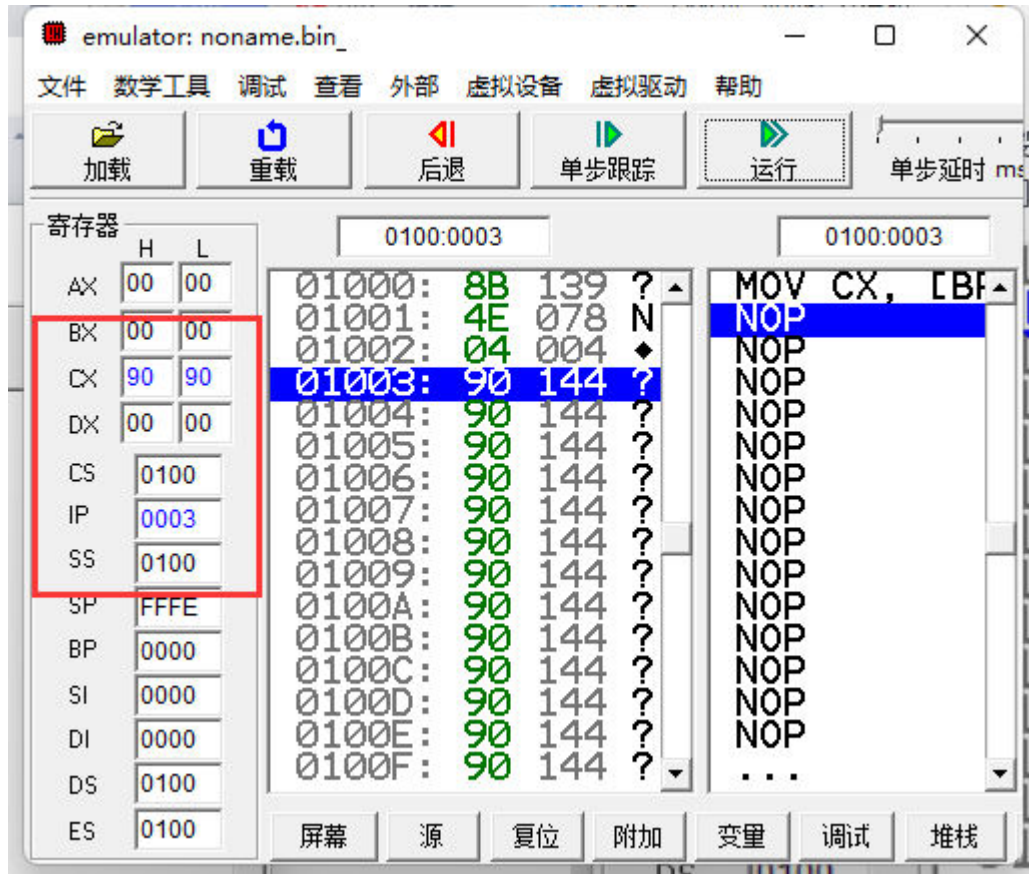
将内存地址中偏移量为 BX+2 的位置存储的数据传送到 AX 寄存器中。



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。

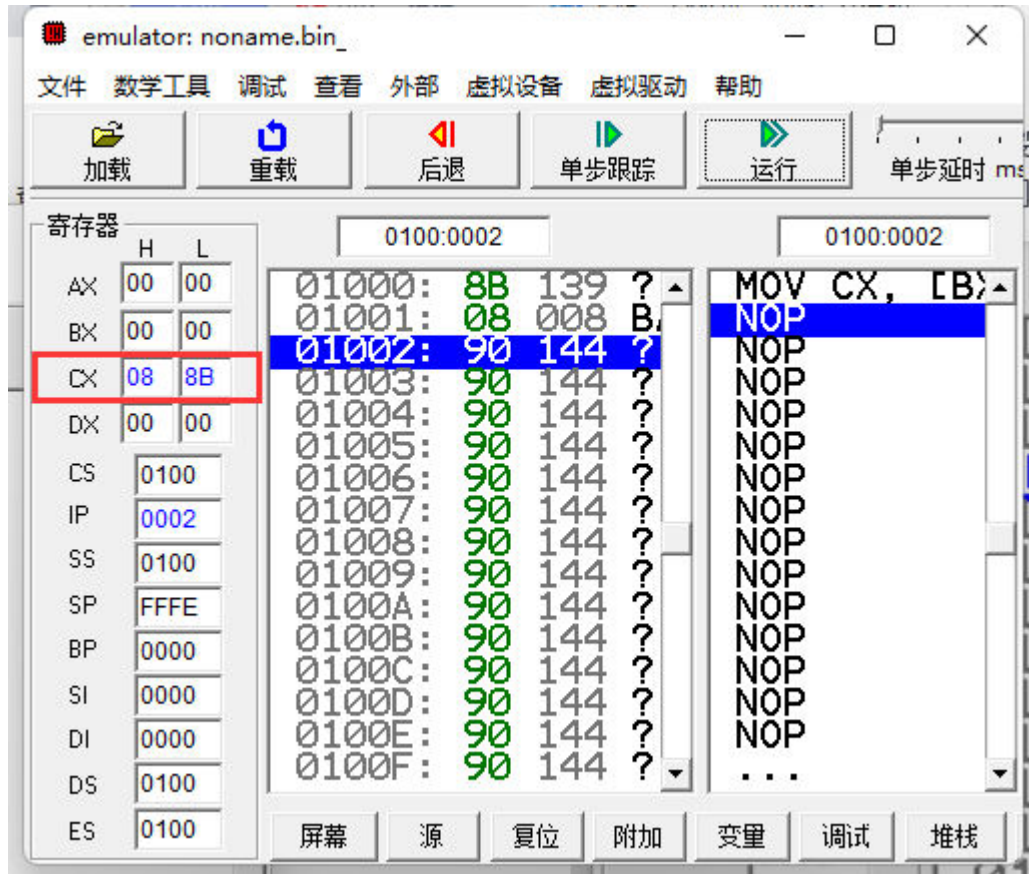


基址变址寻址：

MOV CX,[BX][SI]

指令含义：

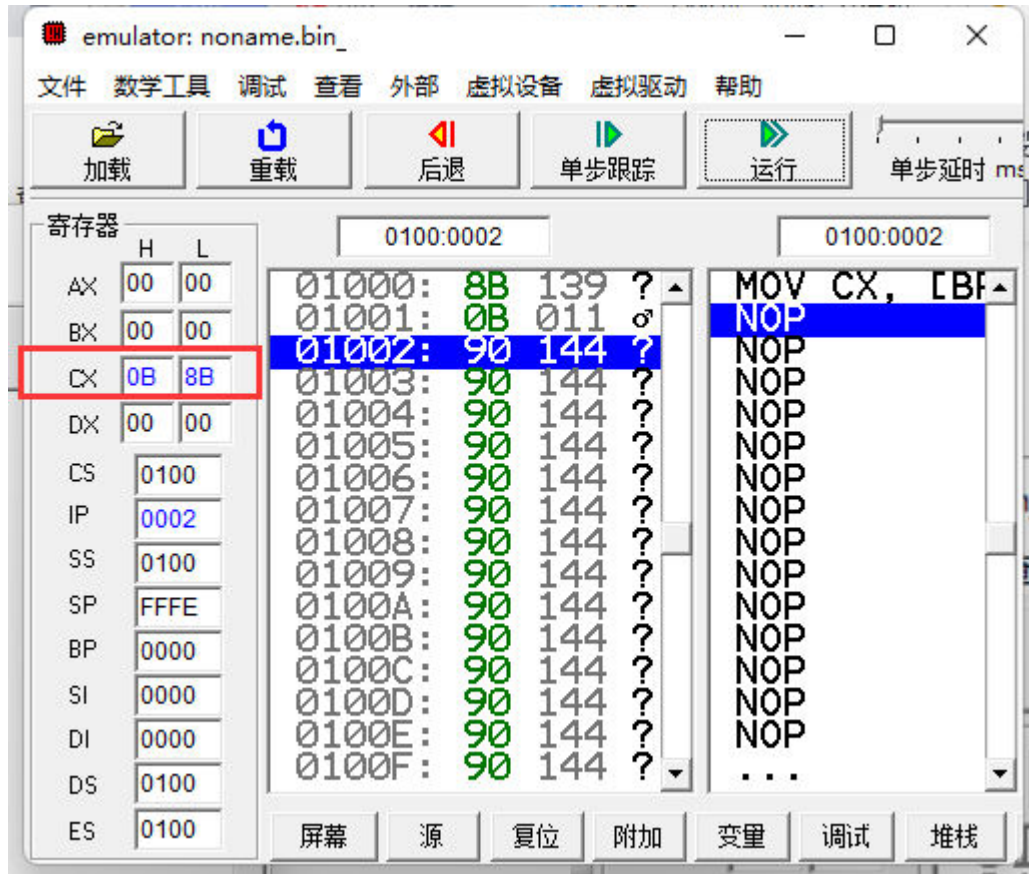
使用寄存器 BX 和 SI 中存储的数值作为偏移量，并将它们相加以得出内存地址。然后，该指令从该内存地址读取数据，并将其存储到 CX 寄存器中。



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：

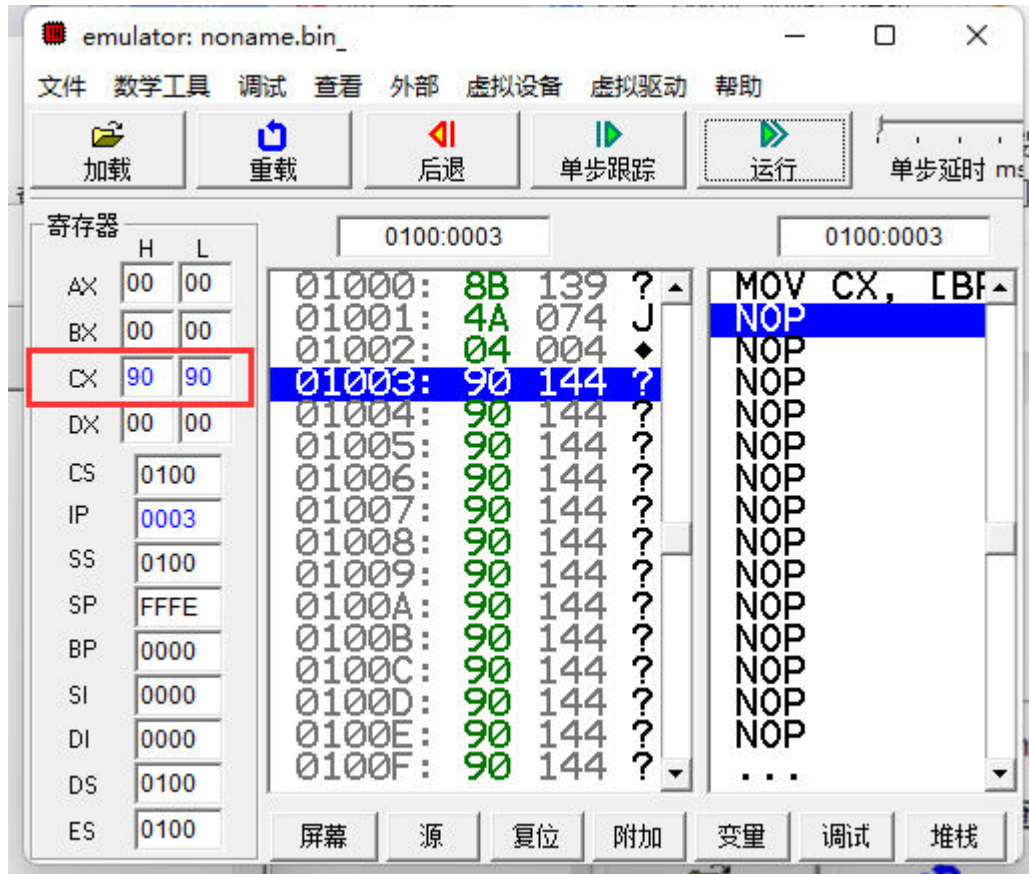
使用寄存器 BX 和 SI 中存储的数值作为偏移量，并将它们相加以得出内存地址，在地址计算完成后，将计算结果值加上 2，作为最终的内存地址。然后，该指令从该内存地址读取数据，并将其存储到 CX 寄存器中。



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]

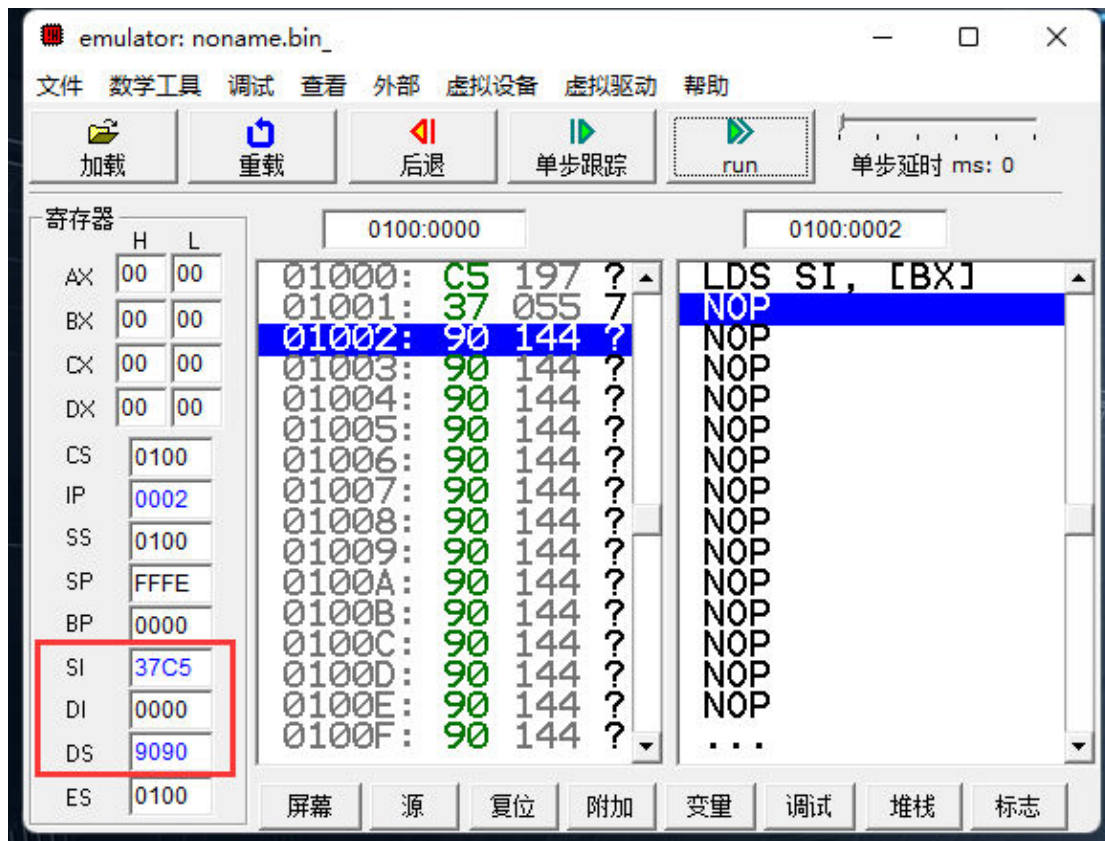
指令含义:

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。

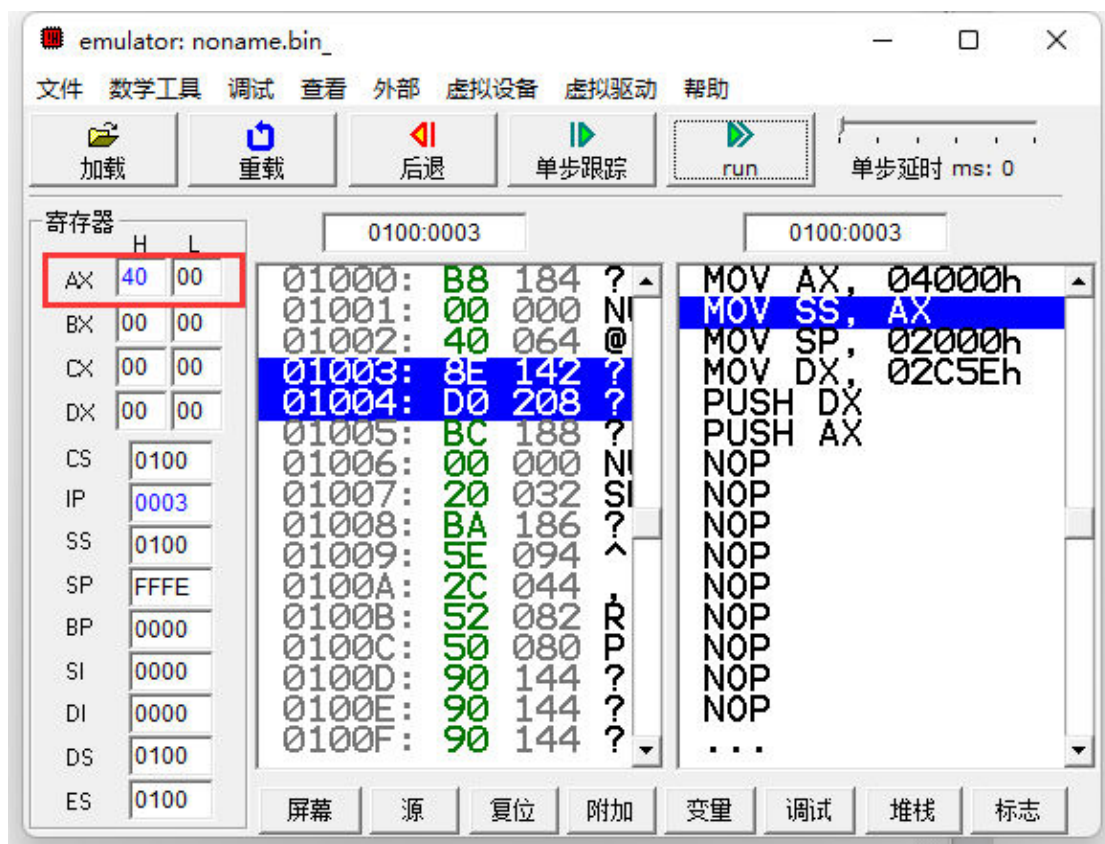
具体来说，这条指令的执行过程如下：

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。

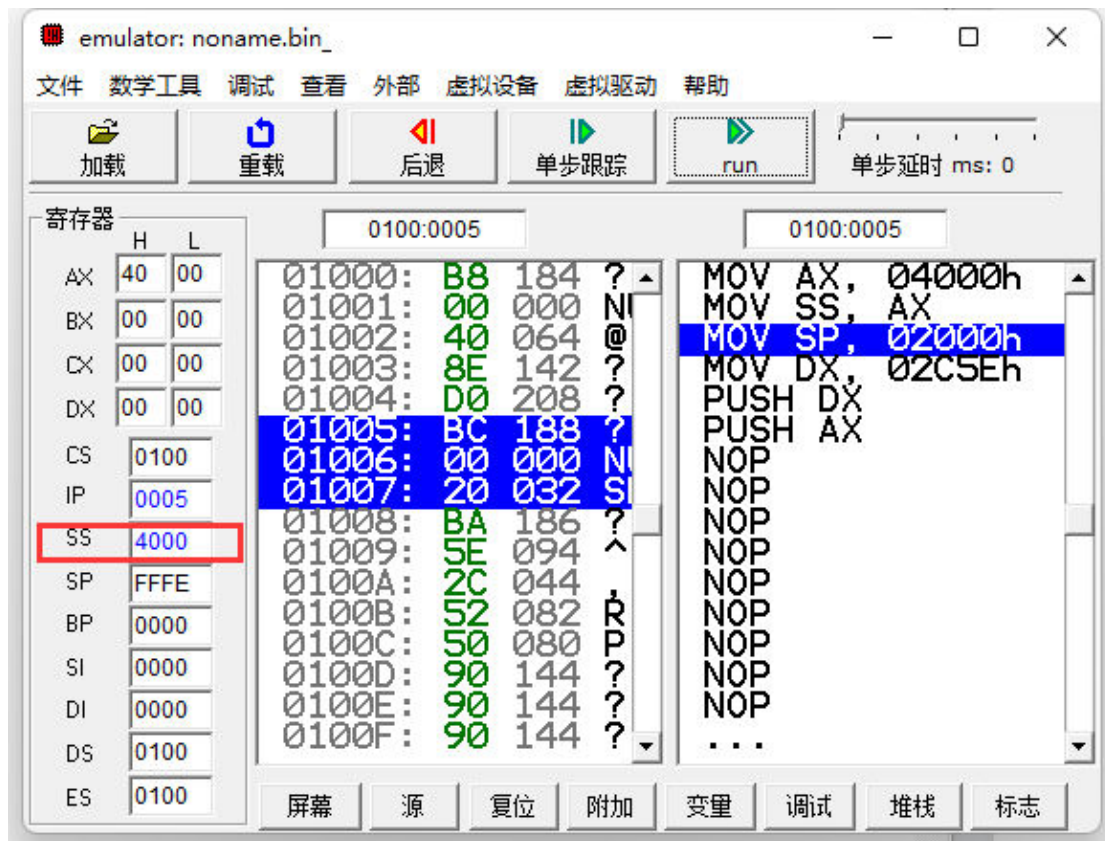
从内存地址[BX+2]处读取一个 16 位偏移量，并将其存储到 SI 寄存器中。



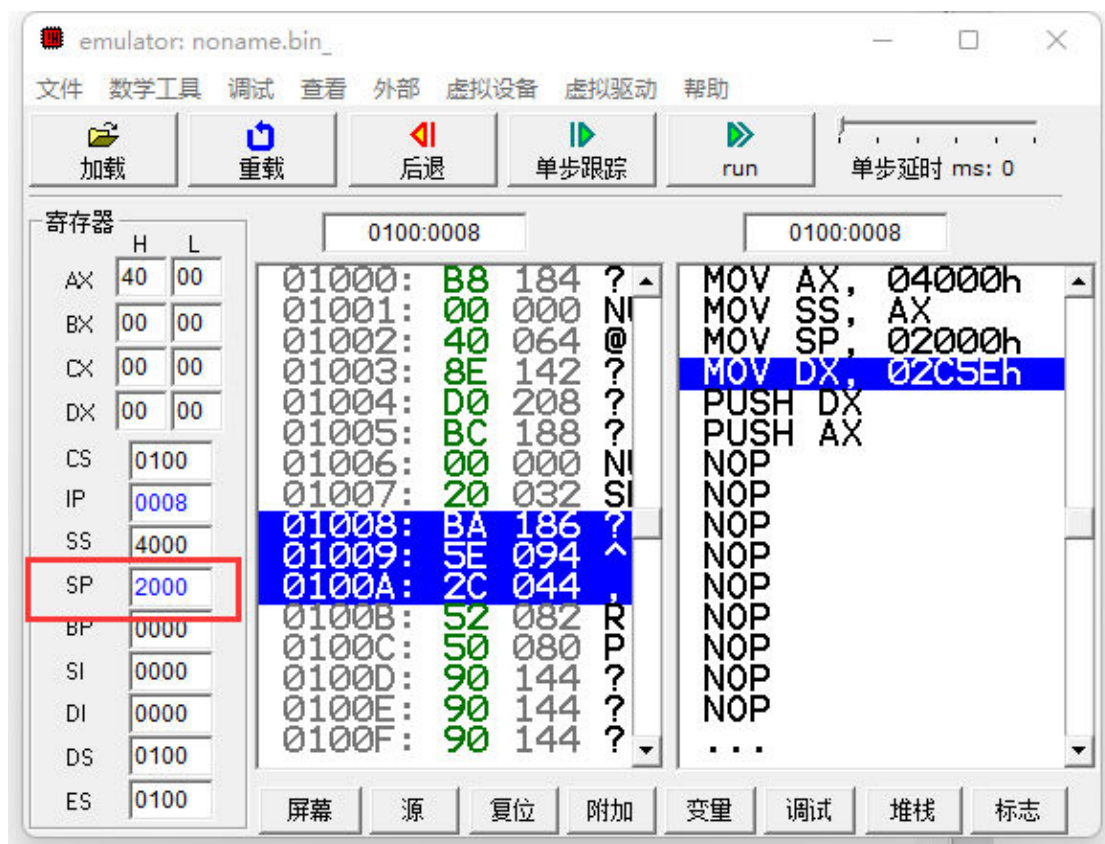
MOV AX,4000H 把立即数 4000H 放入 AX 寄存器



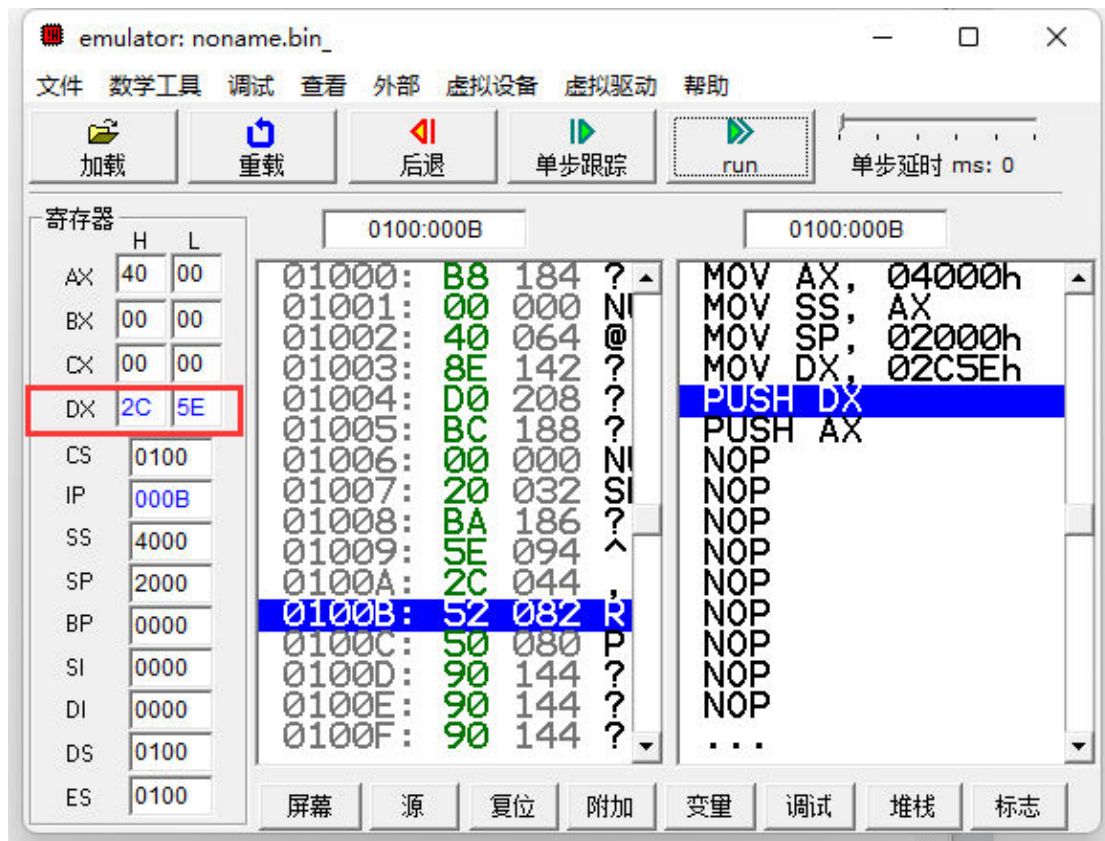
MOV SS,AX 把 AX 寄存器中的数放入 SS



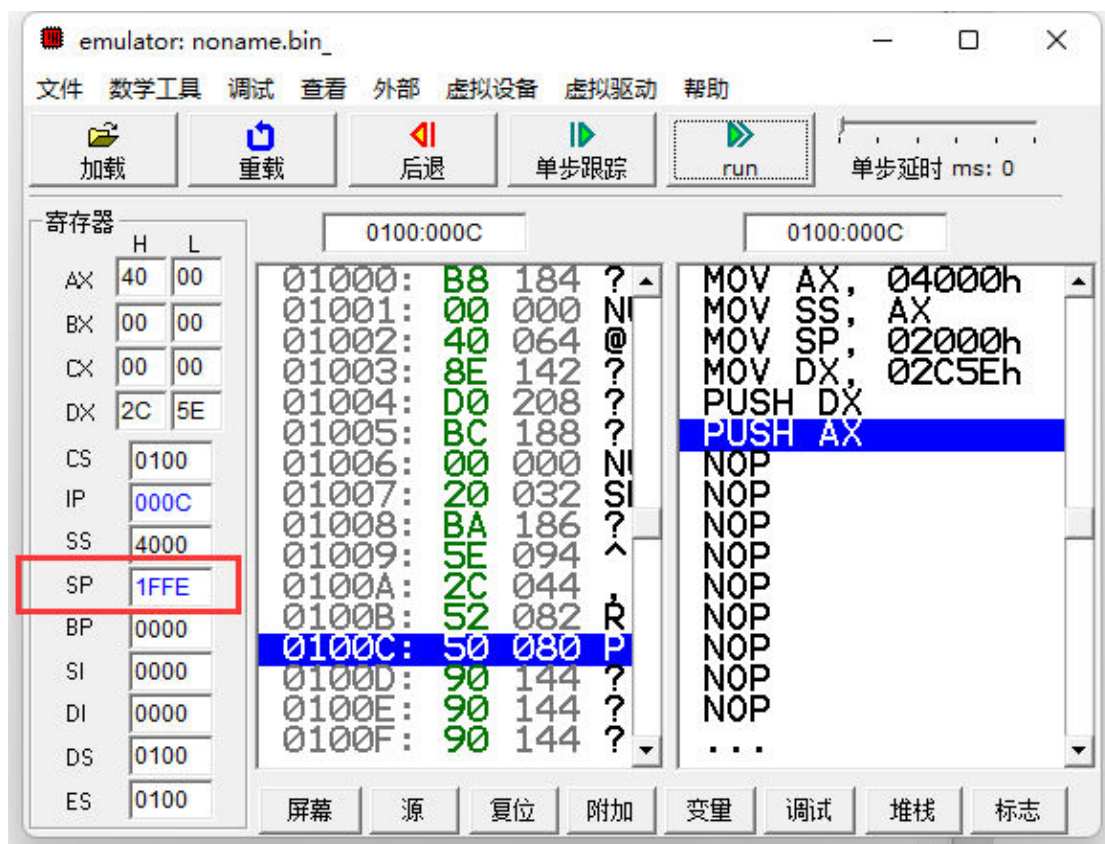
MOV SP,2000H 把立即数 2000H 放入 SP 寄存器



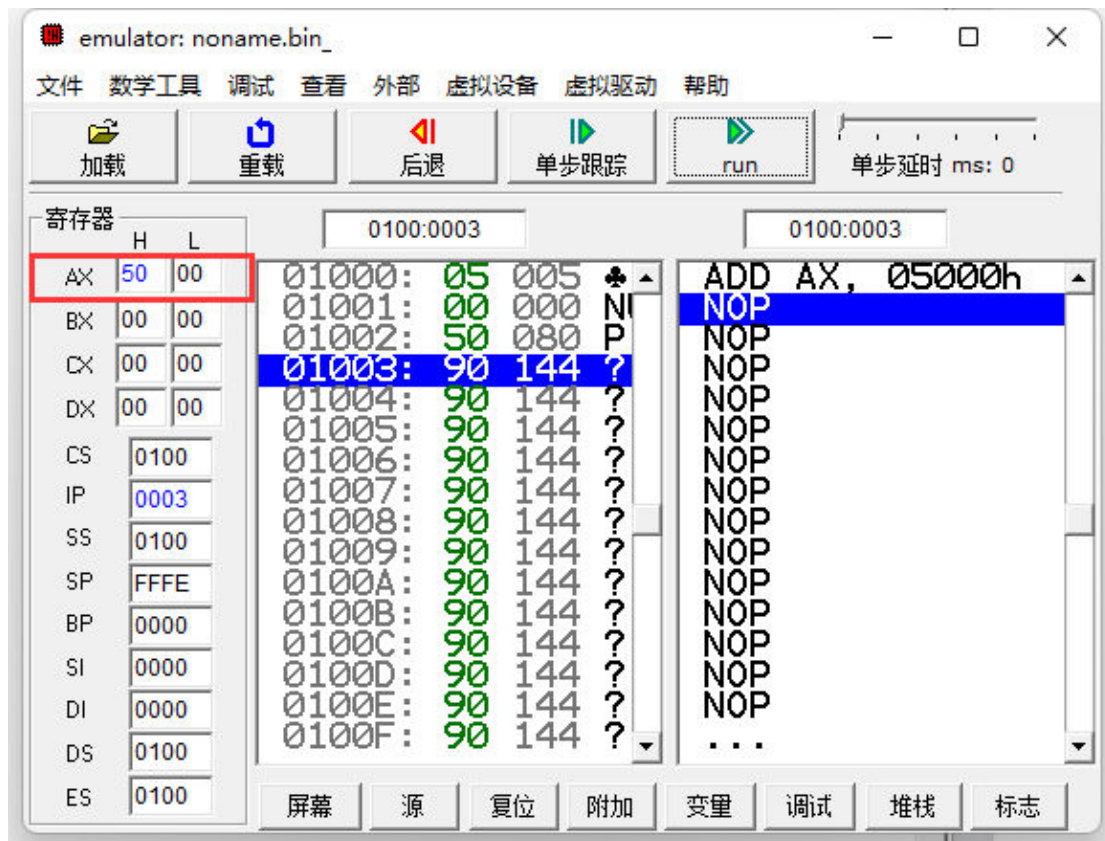
MOV DX,2C5EH 把立即数 2C5EH 放入 DX 寄存器



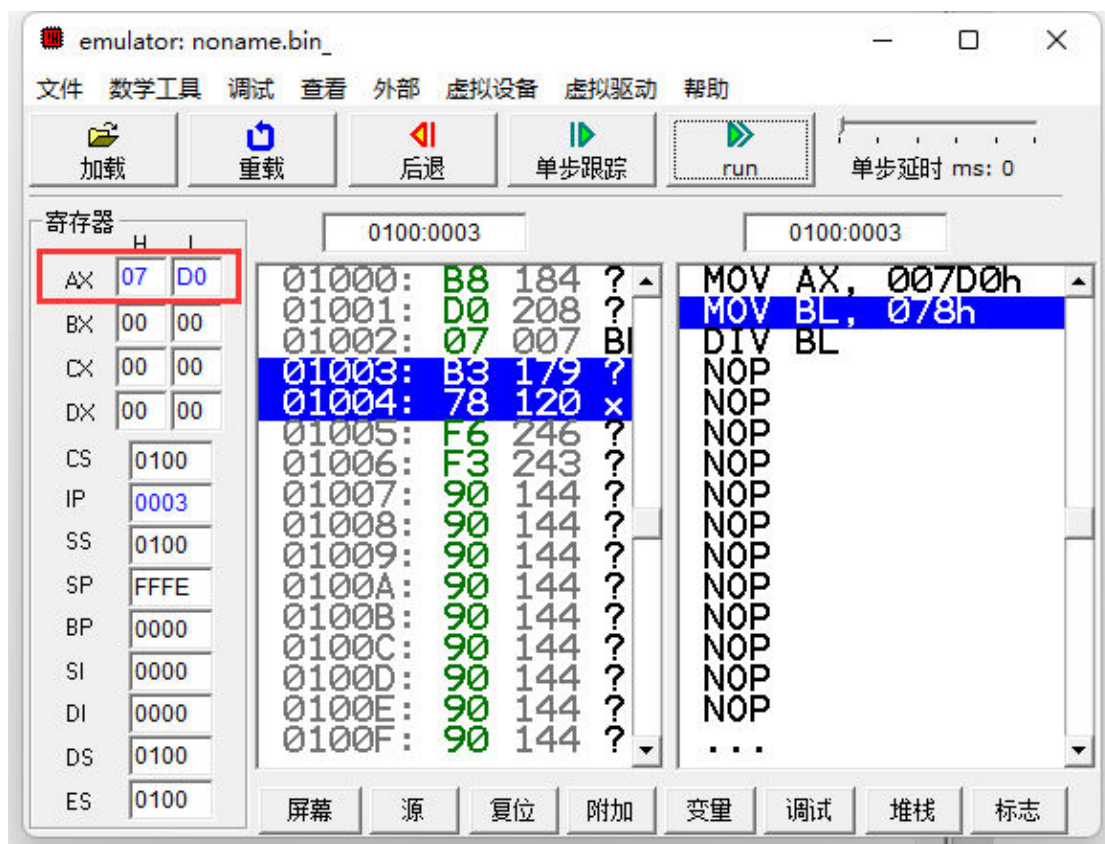
PUSH DX 把 DX 中的操作数压入栈内



PUSH AX 把 AX 中的操作数压入栈内



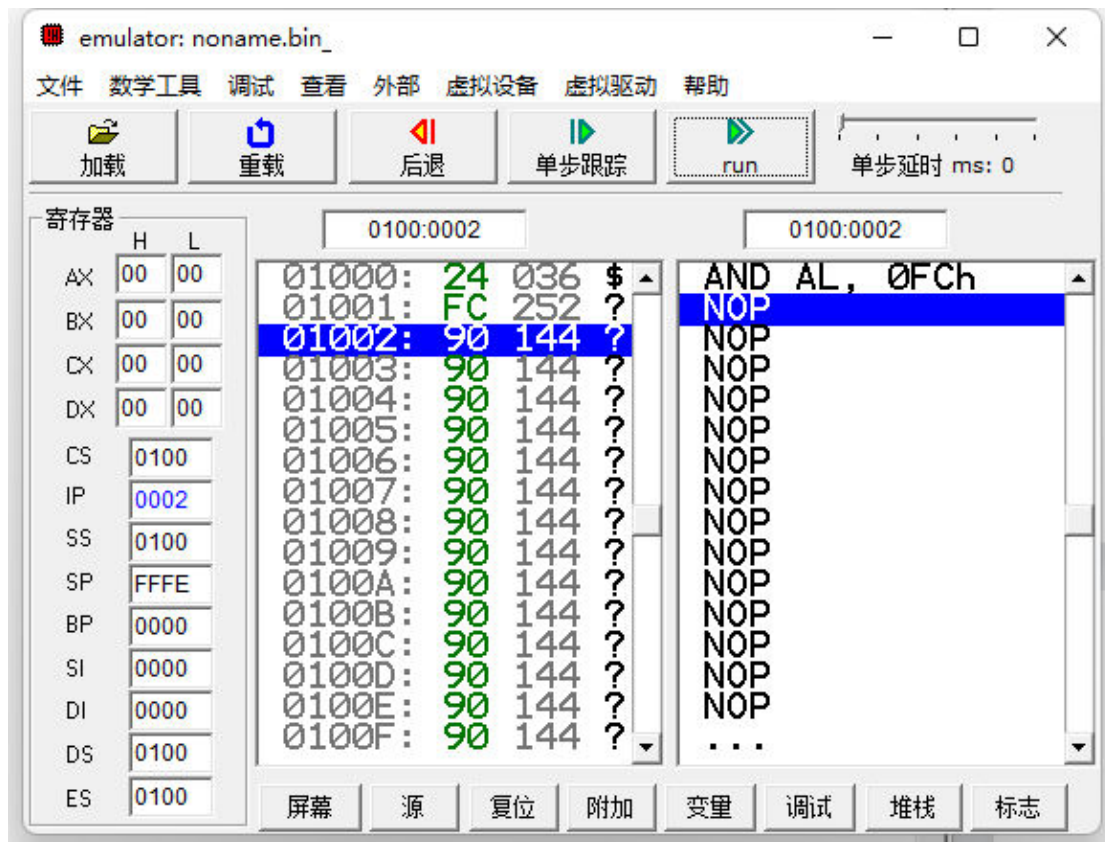
MOV AX,2000 把十进制数 2000 放入 AX 寄存器



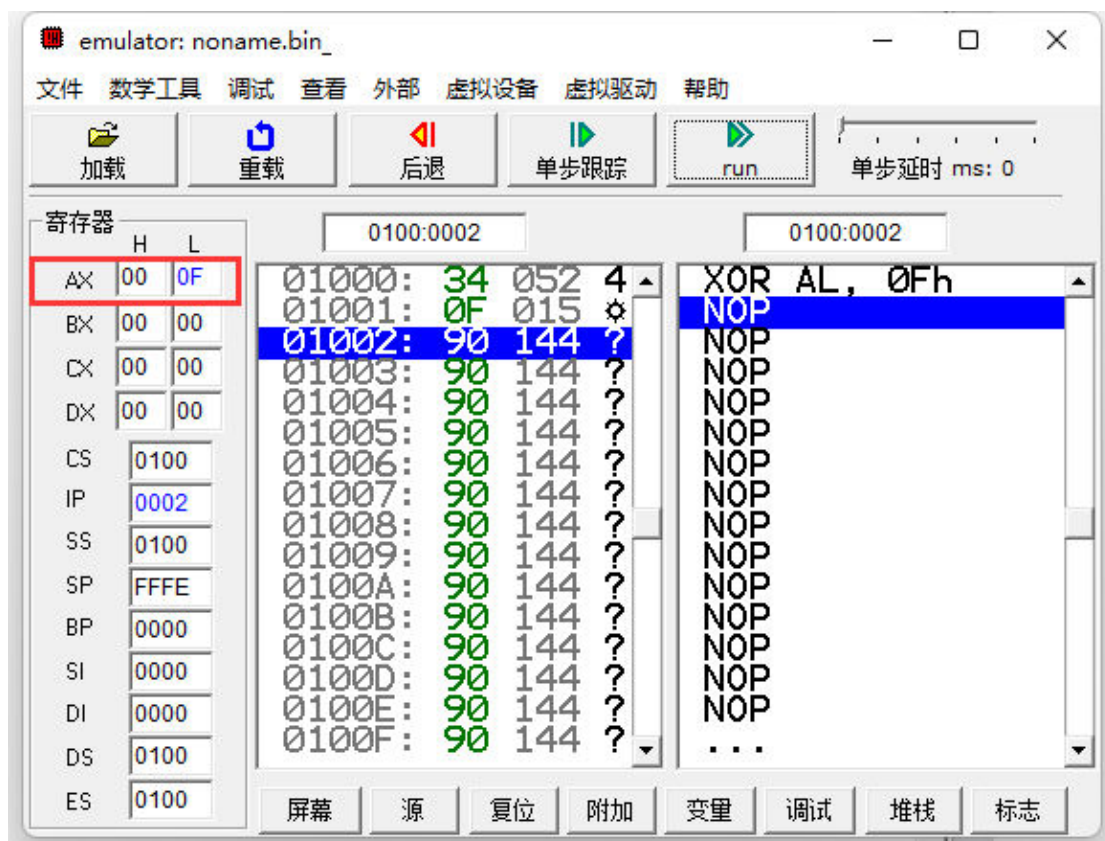
MOV BL,120 把十进制数 120 放入 BL 寄存器

逻辑运算指令：

AND AL,0FCH 把立即数 0FCH 按位与 AL 与，结果放入 AL



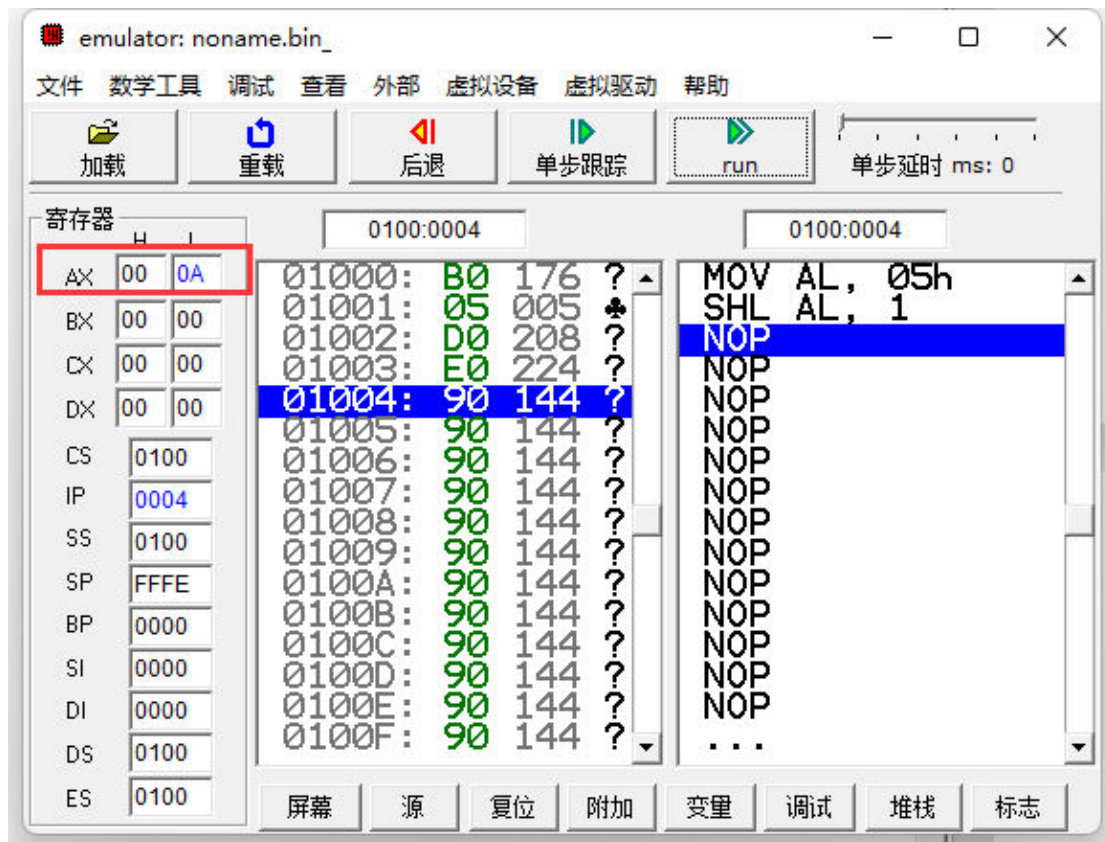
XOR AL,0FH 将立即数 0FH 与 AL 中操作数按位异或，放入 AL 中



移位和循环移位类指令:

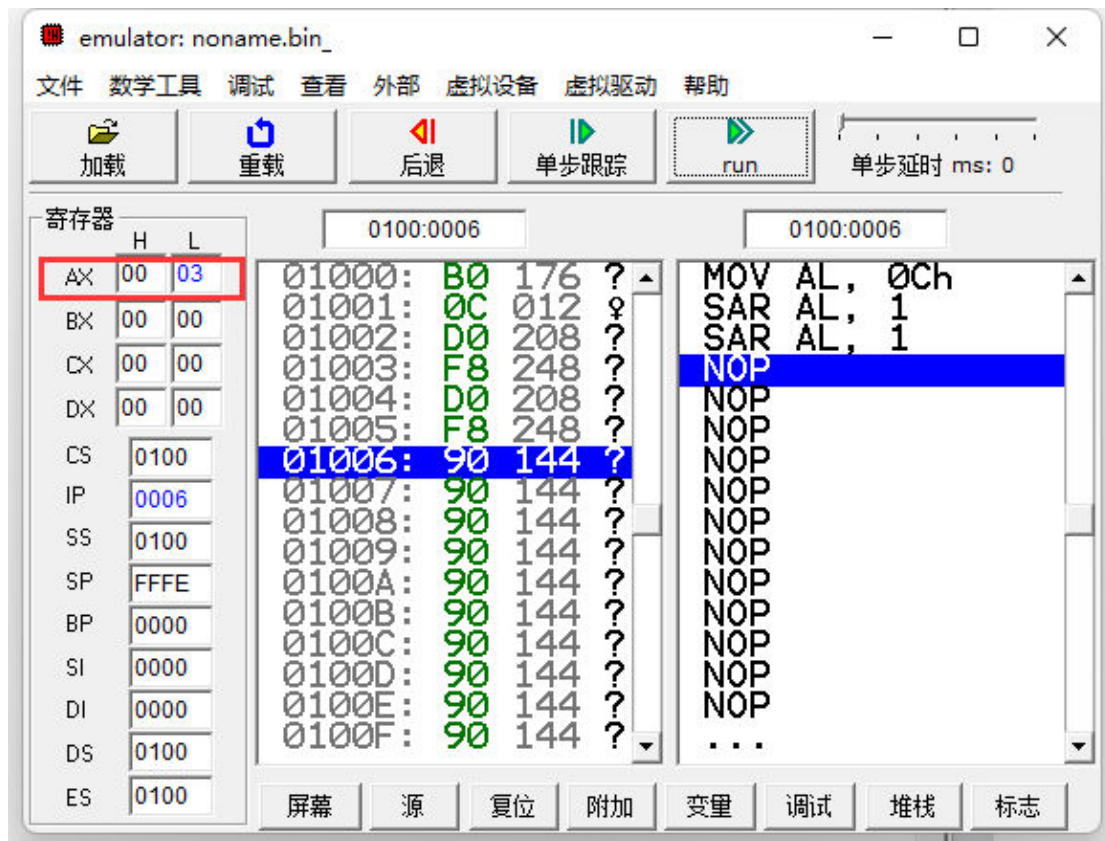
MOV AL,05H; 将立即数 05H 存储到寄存器 AL 中

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

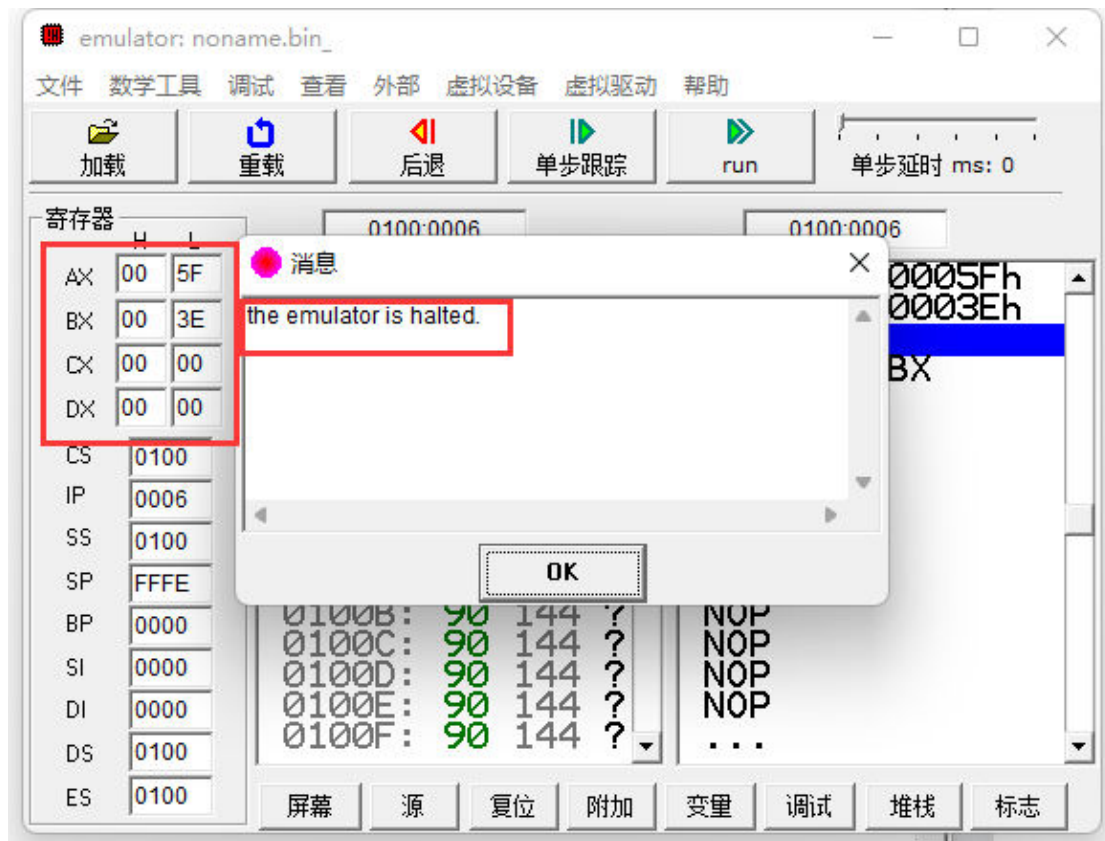
SAR AL,2; 将寄存器 AL 中的值右移两位（相当于除以 2 的平方，即除以 4），并将结果保存回 AL 寄存器



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

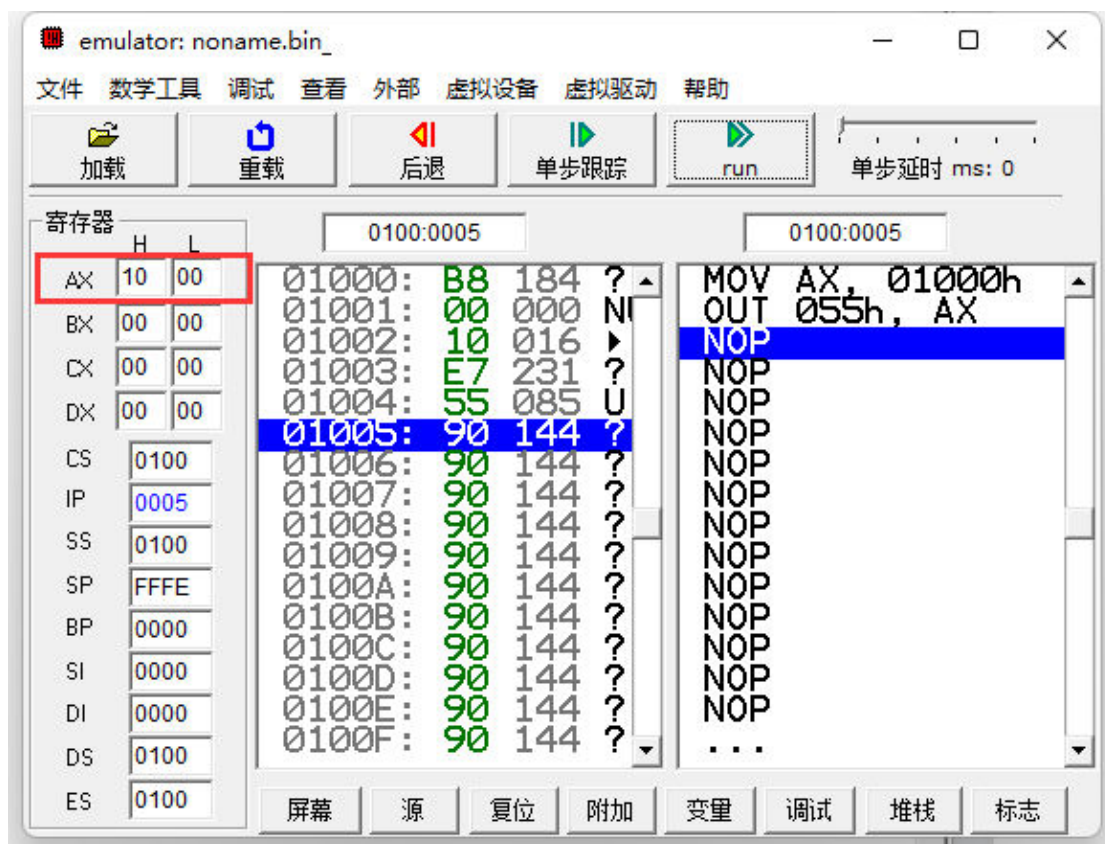
HLT 指令是一个处理器控制指令，它会停止 CPU 的执行，并让 CPU 进入等待状态，直到外部中断被触发。因此，执行完 HLT 指令后，CPU 停止执行指令，不再继续执行 ADD AX, BX。



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。

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
56.
57. MOV AH,4CH
58. INT 21H
59. RET
```



四、 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）；