**西安电子科技大学**

**微机原理综合实验 课程实验报告**

**实验名称 EMU8086使用及8086指令系统**

xxxxxxxxxxxxx 学院 xxxxxxx 班

成 绩

姓名 xxxxxx 学号 xxxxxxxxxxx

同作者 无

实验日期 2021 年 5 月 28 日

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验所用仪器（或实验环境）  三、实验基本原理及步骤（或方案设计及理论计算）  四、实验数据记录（或仿真及软件设计）  五、实验结果分析及回答问题（或测试环境及测试结果） |

1. **实验目的**
2. 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；
3. 熟悉并掌握8086/8088指令系统及内部结构；
4. 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；
5. 熟悉变量、常量及伪指令的使用；
6. 熟悉内存单元的存储结构，字符串的处理以及简单的编程。
7. **实验内容**
8. 熟悉并掌握EMU8086 汇编语言编程调试环境；
9. 8086寻址方式实验（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）；
10. 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）；
11. 学习汇编语言程序设计的基本步骤和方法；
12. 学会使用EMU8086 debug调试程序（单步运行、断点设置等）；
13. 学习汇编编程中DOS功能调用方法，掌握常用的DOS功能调用；
14. 编写一个简单的程序：将“This is my first ASM program-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。
15. **程序清单**

2．8086寻址

（1）立即数寻址: 操作数是由指令直接提供，用于对寄存器赋初值

MOV AL, 06H

MOV BX, 3306H

（2）寄存器寻址: 操作数在CPU内部的寄存器中，指令给出寄存器名

MOV AL, 06H

MOV BX, 3306H

MOV CL, AL

MOV DX, BX

（3）直接寻址: []表示存储器地址，里面存放偏移量

MOV [34H],56H

MOV AL,[34H]

MOV [1000H],1024H

MOV BX,[1000H]

（4）寄存器间接寻址：操作数在存储器中，偏移地址由指针寄存器提供

MOV AX,0000H;

MOV DS,AX;//DS段地址设定

MOV AX,1000H;

MOV SS,AX;//SS段地址设定

MOV BX,0100H;//基址寄存器设定

MOV BP,0100H;//基址指针寄存器设定

MOV [BX],1111H;//[BX]寄存器间接寻址

MOV [BP],2222H;//[BP]寄存器间接寻址

MOV CX,SS:[BP];

MOV DX,DS:[BX];

（5）寄存器相对寻址：操作数在存储器中，并且一般指定BX、BP、SI、DI的内容进行间接寻址，但是操作数的有效地址EA（即偏移量）还要加上指令中指定的8位或16位位移量（Displacement）

MOV BX,0100H;

MOV [012AH],1234H;

MOV AL,2AH[BX];//BX+2AH寄存器相对寻址

MOV AH,2AH[BX+1H];//BX+2BH寄存器相对寻址

（6）基址变址寻址：操作数一定在存储器中，操作数的有效地址EA是由指令指定的一个基址寄存器（如BX、BP）内容加上一个变址寄存器（如SI、DI）的内容

MOV AX,0000H;

MOV DS,AX;

MOV BX,1000H;

MOV DI,0001H;

MOV [BX][DI],1111H;

（7）基址变址相对寻址：在基址变址寻址基础上加上一定地址偏移，同样可以使用常量或者变量作为偏移，方式同寄存器相对寻址

MOV AX,0000H

MOV DS,AX;

MOV BX,1000H;

MOV DI,0001H;

MOV [BX][DI+0010H],1111H;//BX+DI+0010H基址变址相对寻址

2．8086指令

（1）数据传送类指令：

通用传送指令：MOV DEST, SRC；DEST可以是寄存器，存储器，累加器(acc)，段寄存器;SRC 可以是立即数，存储器，寄存器，累加器，段寄存器

MOV AX, 2000H

MOV DS, AX

堆栈操作指令：

PUSH SRC ：入栈指令，SRC可以为通用寄存器 段寄存器 存储器单元(地址连续的两个储存单元)

POP DEST ：出栈指令，DEST可以为通用寄存器 段寄存器（CS寄存器除外） 存储器单元(地址连续的两个储存单元)

（2）算术运算类指令：

不带进位加法指令ADD(addition)：

指令格式：ADD DEST, SRC；执行操作：DEST<—DEST+SRC

MOV AX,1000H

ADD AX,0010H

不带借位的减法指令SUB（subtraction）：

指令格式：SUB DEST, SRC；执行操作：DEST<—DEST-SRC

MOV BX,1000H

SUB BX,0010H

（3）逻辑运算类指令

MOV AX,1000H

MOV AX,1000H

AND AX,0110H

OR BX,0110H

（4）标志处理和CPU控制类指令

CLC: CF置0

STC: CF置1

（5）移位和循环移位类指令

SAL：算数左移

RCL：循环左移

7.编写程序并通过DOS功能调用显示

; 设置段寄存器

mov AX, 0700h

mov BX, 0000h

mov DS, AX

;存储字符串

mov [00h], 'T'

mov [02h], 'h'

mov [04h], 'i'

mov [06h], 's'

mov [08h], ' '

mov [0ah], 'i'

mov [0ch], 's'

mov [0eh], ' '

mov [10h], 'm'

mov [12h], 'y'

mov [14h], ' '

mov [16h], 'f'

mov [18h], 'i'

mov [1ah], 'r'

mov [1ch], 's'

mov [1eh], 't'

mov [20h], ' '

mov [22h], 'A'

mov [24h], 'S'

mov [26h], 'M'

mov [28h], ' '

mov [2ah], 'p'

mov [2ch], 'r'

mov [2eh], 'o'

mov [30h], 'g'

mov [32h], 'r'

mov [34h], 'a'

mov [36h], 'm'

mov [38h], '-'

mov [3ah], 'W'

mov [3ch], 'a'

mov [3eh], 'n'

mov [40h], 'g'

mov [42h], 'S'

mov [44h], 'h'

mov [46h], 'e'

mov [48h], 'n'

mov [4ah], 'a'

mov [4ch], 'o'

;一共39个字符

mov CX, 0027h

;搬移到BX=0100h开始的存储单元中

mov BX, 0000h

move:

mov AX, [BX]

mov [BX+0100h], AX

add BX, 2

loop move

;打印字符串

mov BX, 0100h

mov CX, 0027h

print:

mov DL, [BX] ;dl为要显示的字符

mov AH, 02h ;显示输出

int 21h

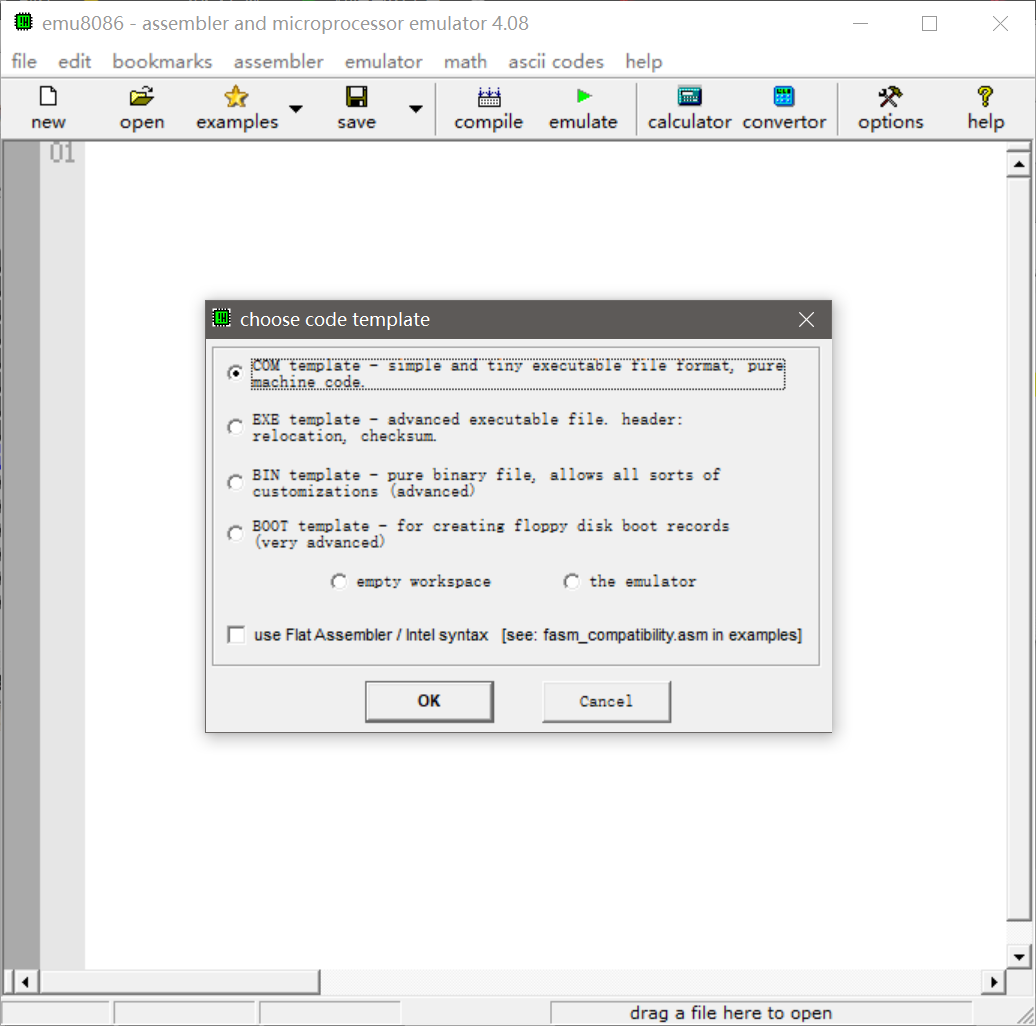
add BX, 2

loop print

1. **运行结果及截图**

1.EMU8086 汇编语言编程调试环境

双击emu8086.exe文件，可以选择新建来编写汇编语言代码，或者选择程序示例查看自带的例程代码。

点击新建可以选择4个模板：

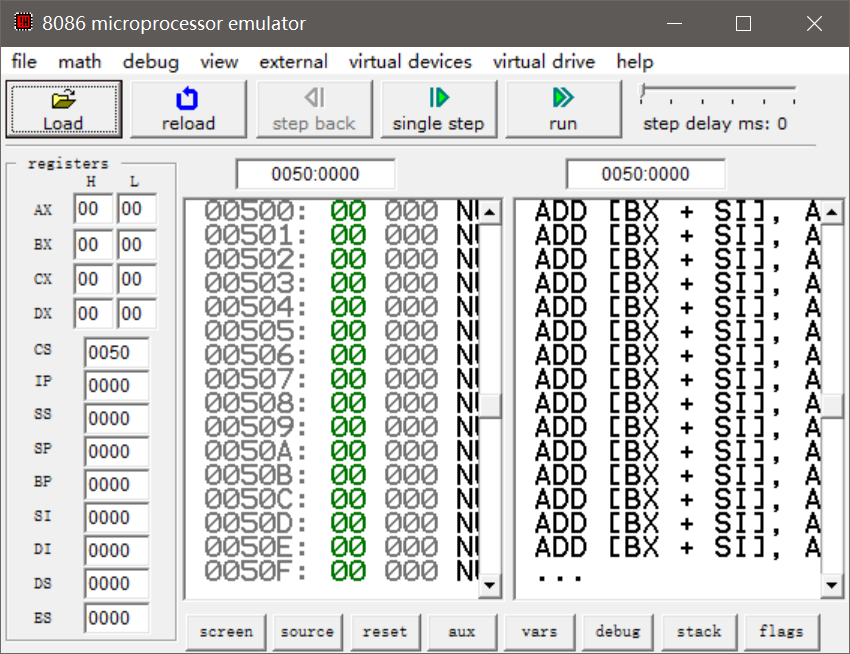
◎COM模板：适用于简单且不需分段的程序，所有内容均放在代码段中，程序代码默认从 ORG 0100H 开始。

◎EXE模板：适用于需分段的复杂程序，内容按代码段、数据段、堆栈段划分。需要注意的是，采用该模板时，用户不可将代码段人为地设置为ORG 0100H，而应由编译器自动完成空间分配。

◎BIN模板：二进制文件，适用于所有用户定义结构类型。

◎BOOT模板：适用于在软盘中创建文件。

此外，可以选择 empty workspace 选项打开一个空白文档。

点击compile或者emulate进入主控界面：

◎点击 run 按钮程序开始进行，再次点击 run 可以停止运行。

◎点击 reload 可以进行重载，从头开始执行程序。

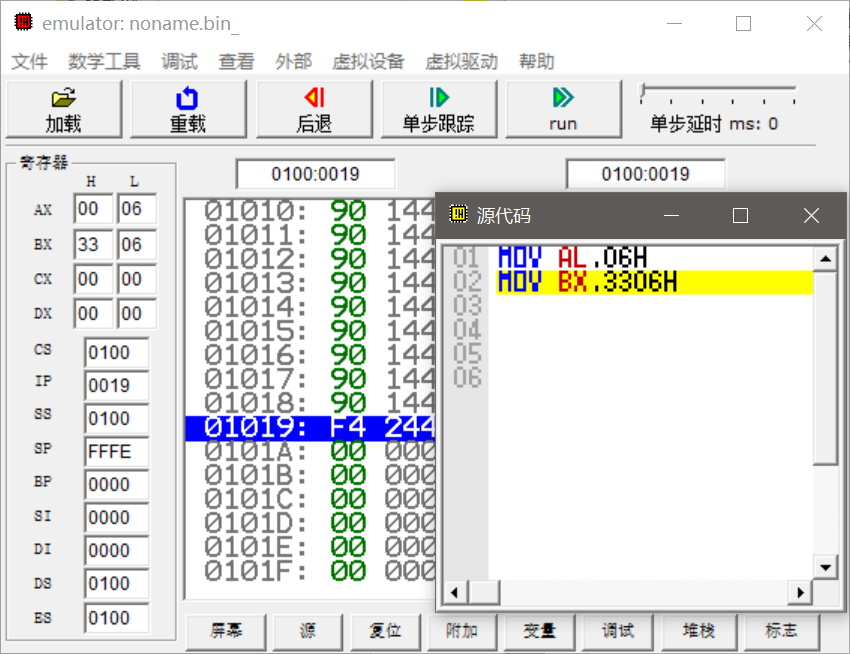
◎点击 single step可以单步调试。

◎点击 step back可以退回到上一条指令

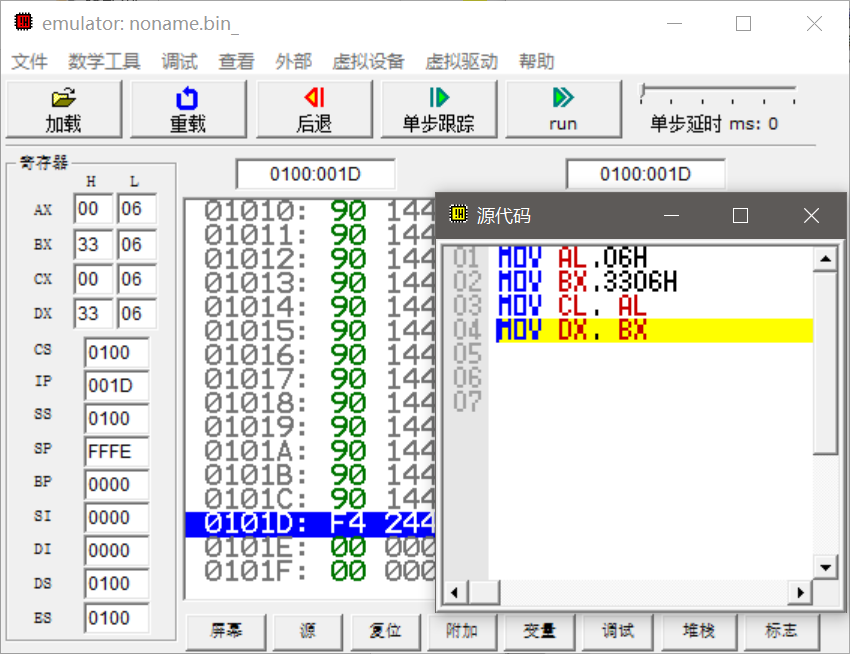
◎左边registers是寄存器栏，可以通过寄存器中值的变化，观察每一个step的代码执行结果。

2．8086寻址方式

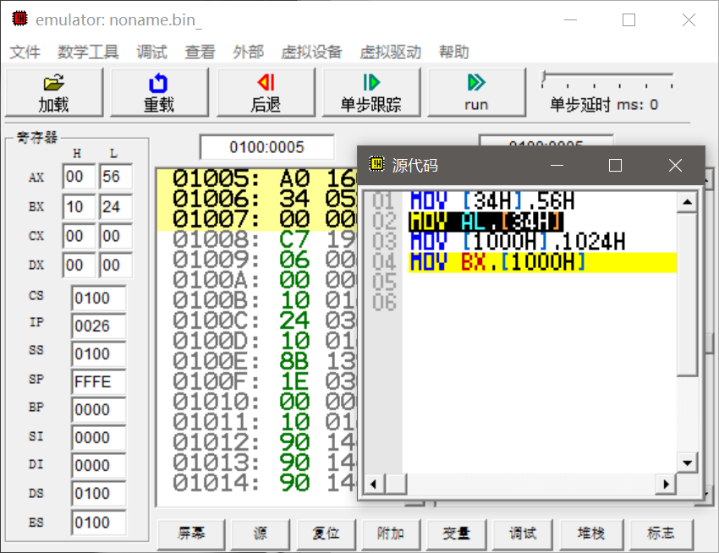
（1）立即数寻址（Immediate Addressing）



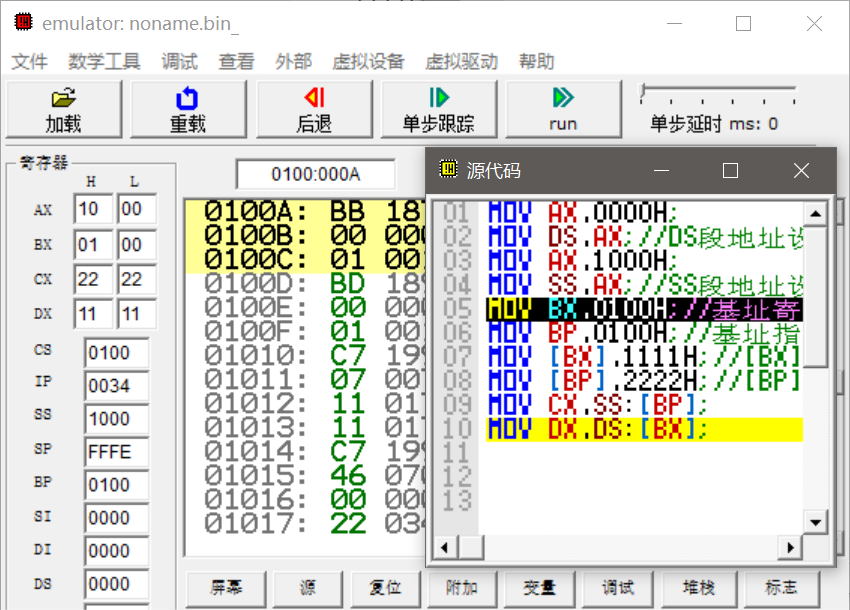
（2）寄存器寻址（Register Addressing）

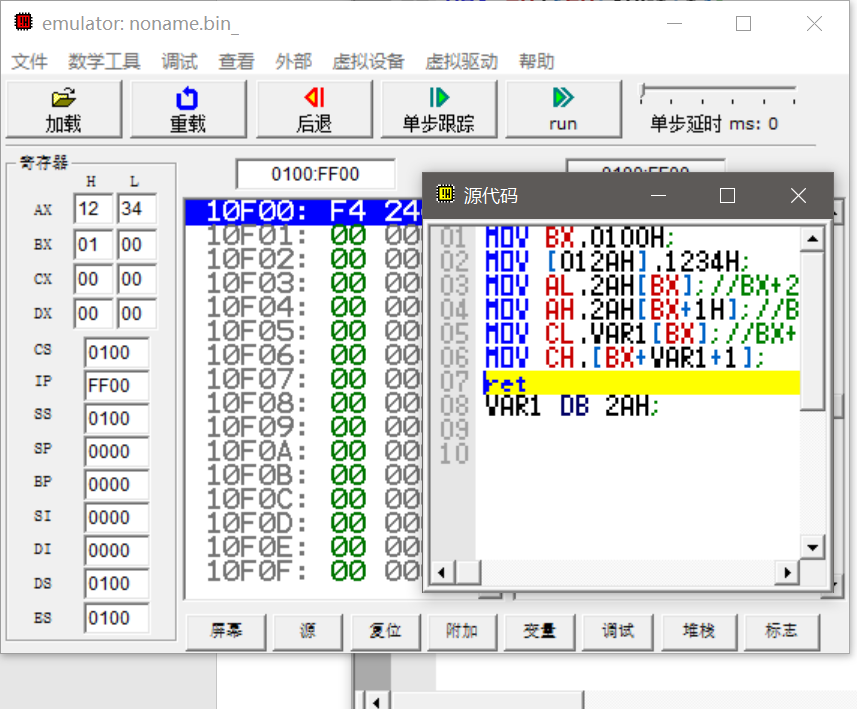


（3）直接寻址

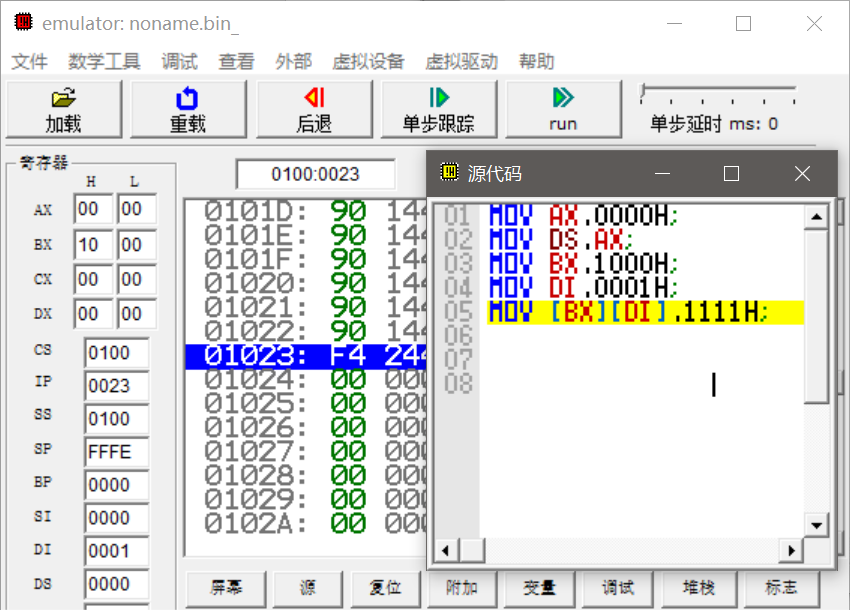


（4）间接寻址

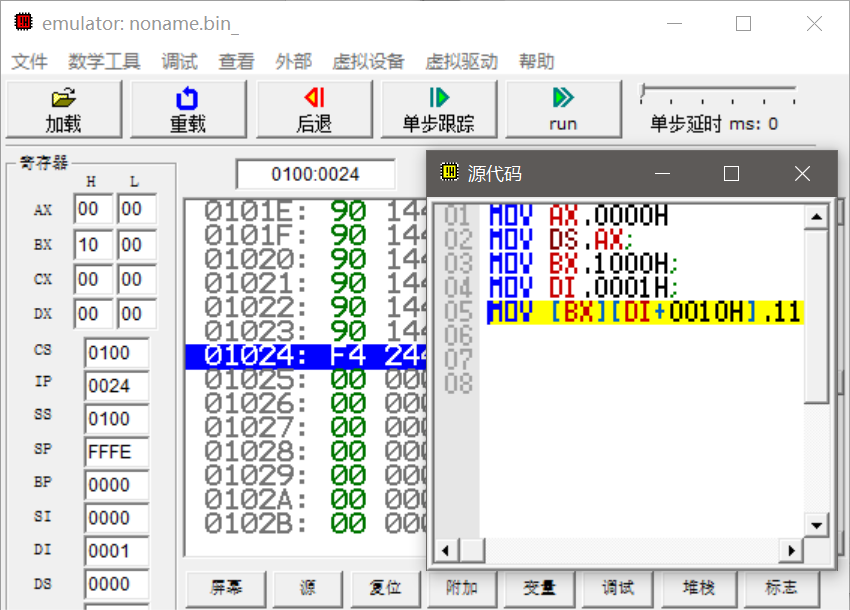


（5）相对寻址

（6）基址变址寻址

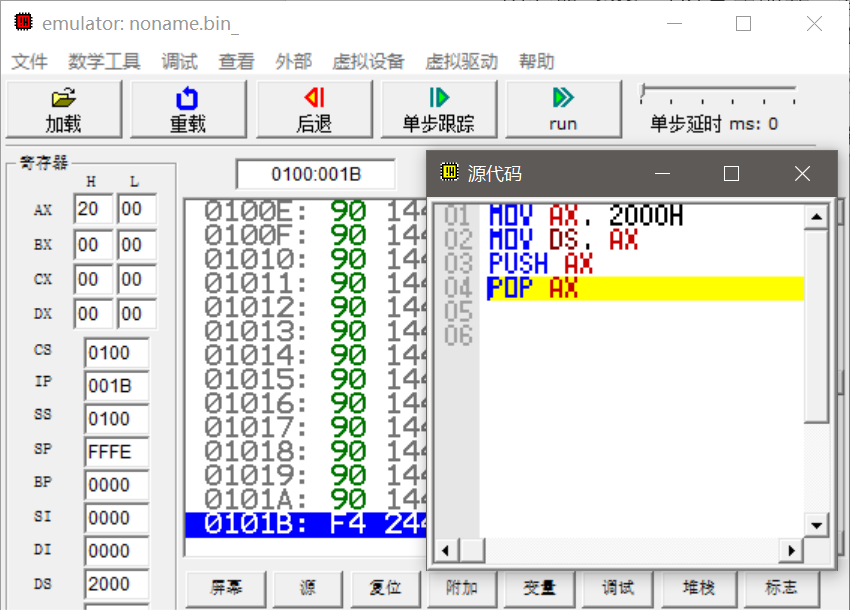


（7）基址变址相对寻址

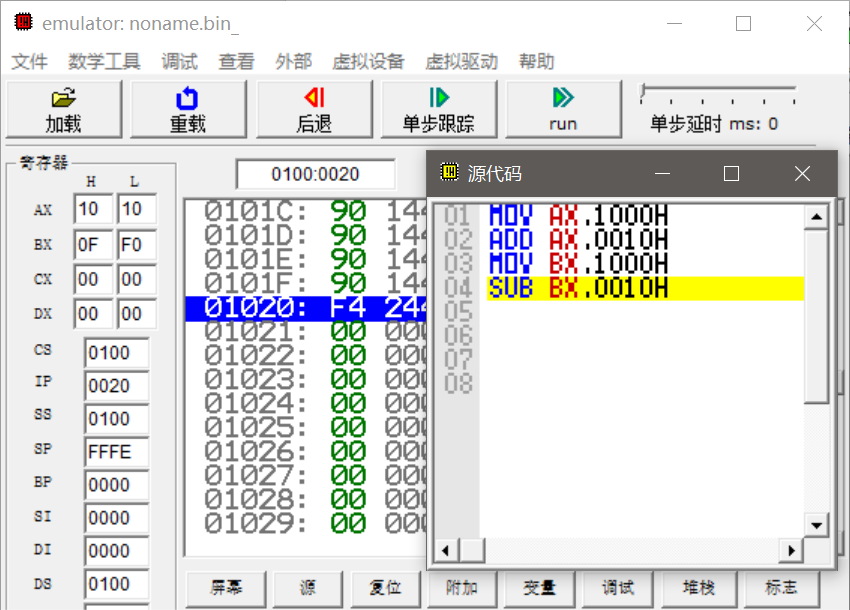


2.8086指令

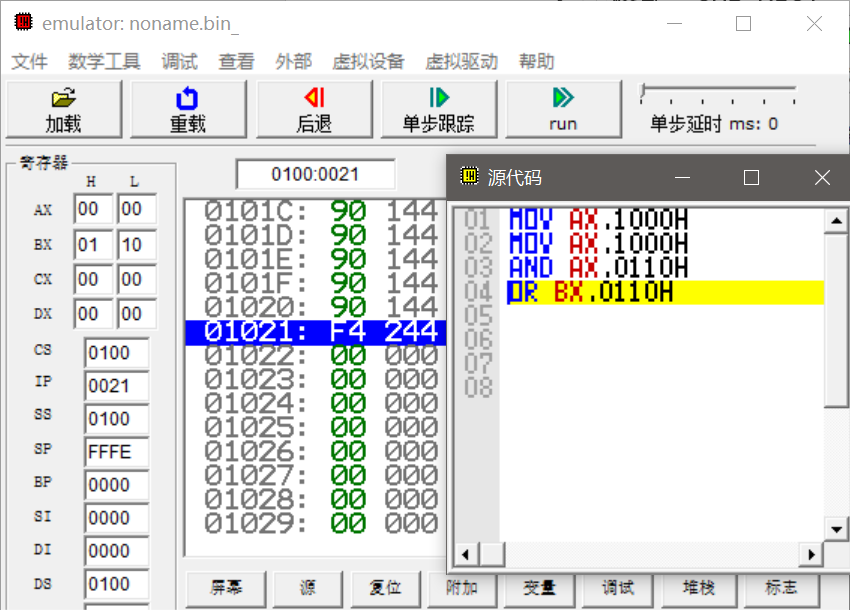
（1）数据传送类指令



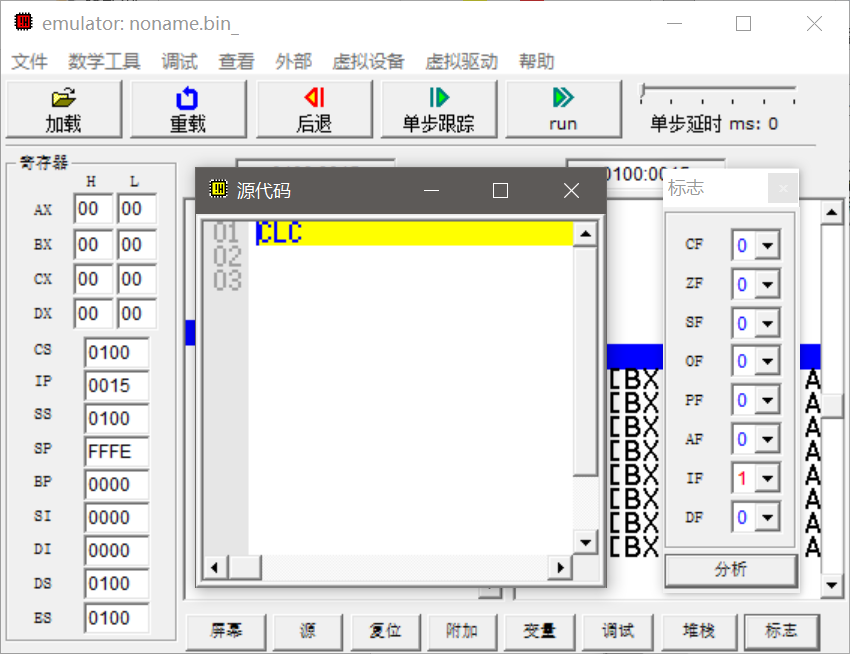
（2）算术运算类指令

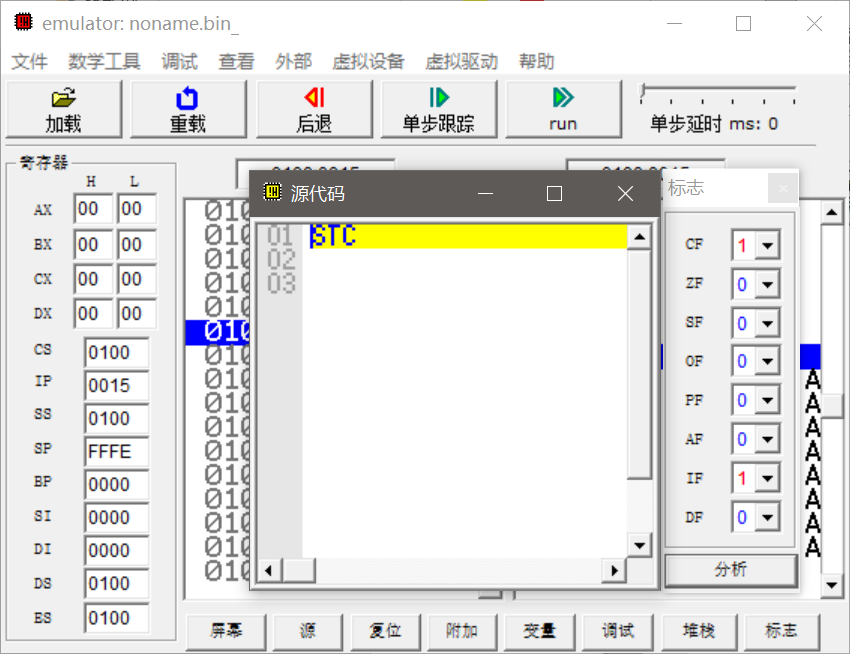


（3）逻辑运算类指令



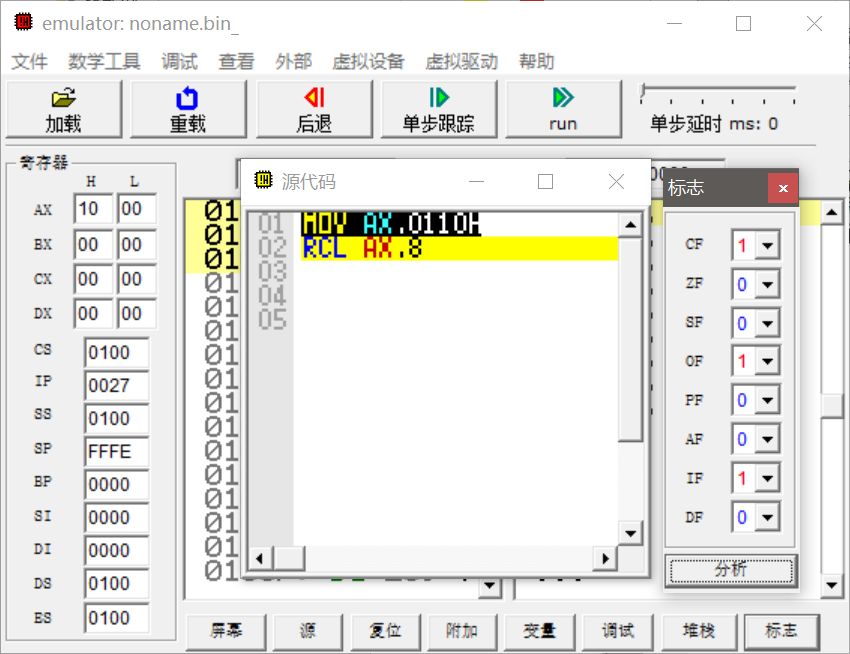
（4）标志处理和CPU控制类指令





（5）移位和循环移位类指令





7.编写程序



1. **实验心得**

通过这次实验，我对汇编语言有了一定的理解，对上课所讲的寻址方式也有了更深层次的认识。这次实验锻炼了我的动手能力，也让我熟悉了8086的基本用法，收获满满。