

# 汇编语言模拟器实验大纲

了解汇编指令和汇编集成开发环境，利用 emu8086 观察执行汇编指令时寄存器的变化。

## 实验目标

- 了解汇编指令和汇编集成开发环境。
- 理解汇编指令执行时寄存器的变化。

## 实验材料

- 汇编语言编译器（emu8086）。
- 汇编代码示例（加法程序）。

## 实验步骤

### 第一课时：实验

1. 安装 emu8086（15 分钟）
  - 确保所有学生安装好 emu8086。
2. 编写汇编程序（15 分钟）
  - 指导学生编写简单的汇编程序（如两个数的加法）

```
org 100h

; 代码部分
mov ax, 5      ; 将 5 加载到 AX 寄存器
mov bx, 10     ; 将 10 加载到 BX 寄存器
add ax, bx     ; 将 AX 和 BX 的值相加，结果存储在 AX 中

ret
```

- 确保每个学生的代码可以在 emu8086 中正确运行。
3. 运行程序并记录实验结果（15 分钟）
    - 将编写的程序加载到 emu8086 中，准备执行。
    - 每执行一条指令，记录以下信息：
      1. **AX、BX 寄存器**：观察其值的变化。
    - 具体指令分析：
      1. **mov ax, 5:**
        1. AX 寄存器更新为 5。

2. **mov bx, 10:**

1. BX 寄存器更新为 10。

3. **add ax, bx:**

1. 执行后, AX 寄存器的值更新为 15。

## 实验评估

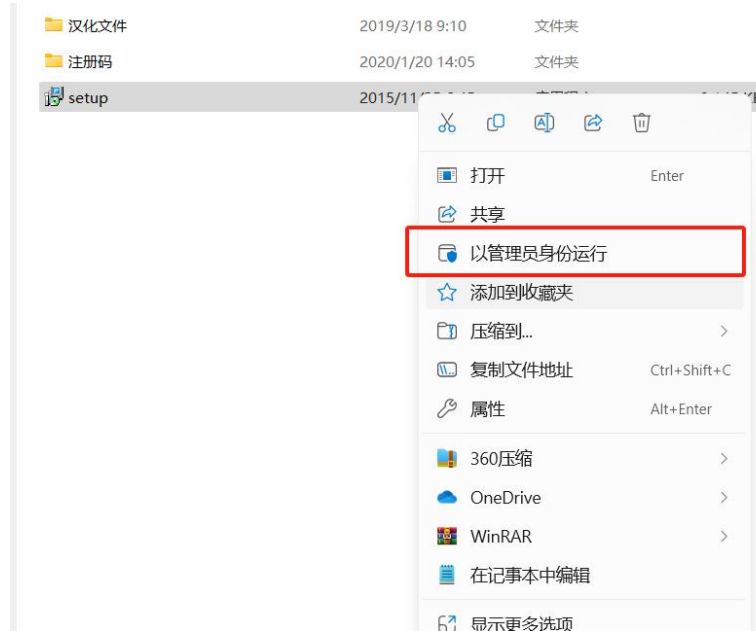
- 学生是否能正确安装 emu8086 并执行汇编指令。
- 学生是否能正确观察和理解寄存器的变化。
- 学生在讨论中的参与度和对实验内容的理解深度。

通过这个实验, 学生能够更直观地理解寄存器在执行汇编指令时的变化。

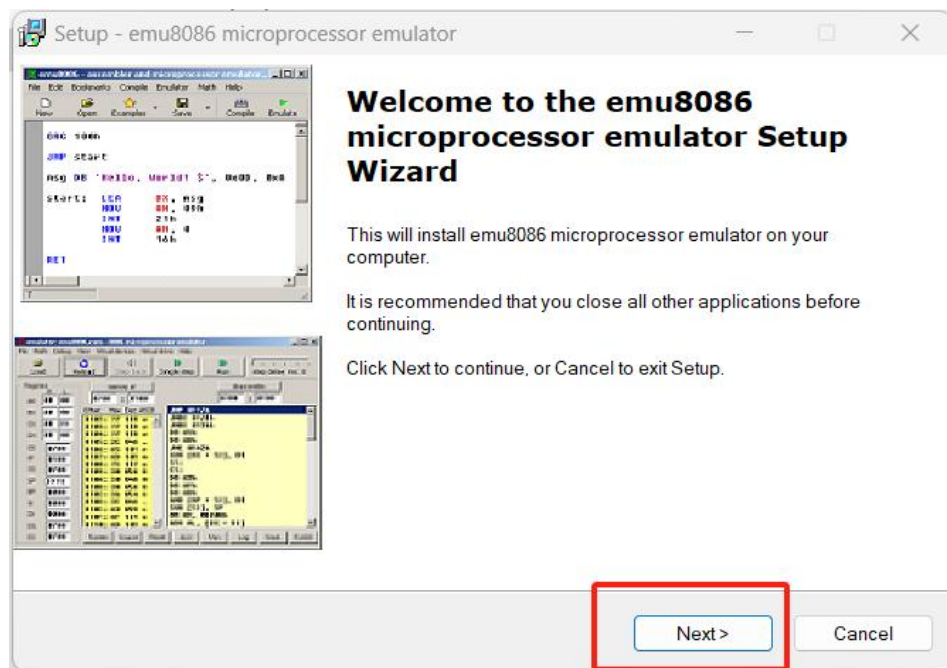
# 汇编语言模拟器实验步骤

## 1. 安装 emu8086

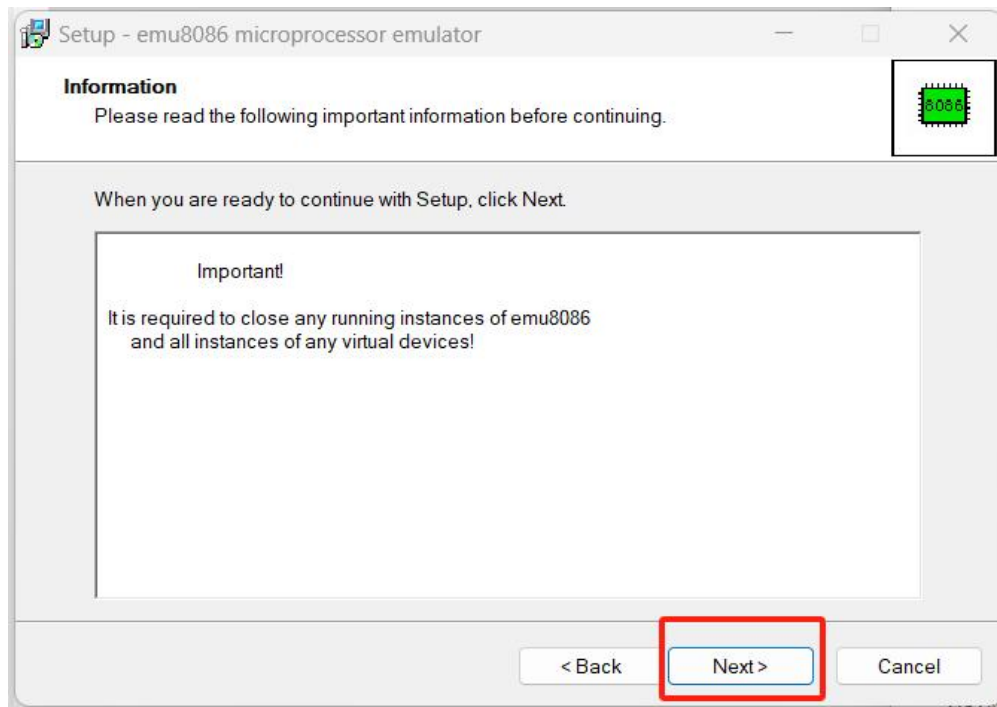
打开文件夹，右键点击 setup.exe，以管理员身份运行 setup.exe



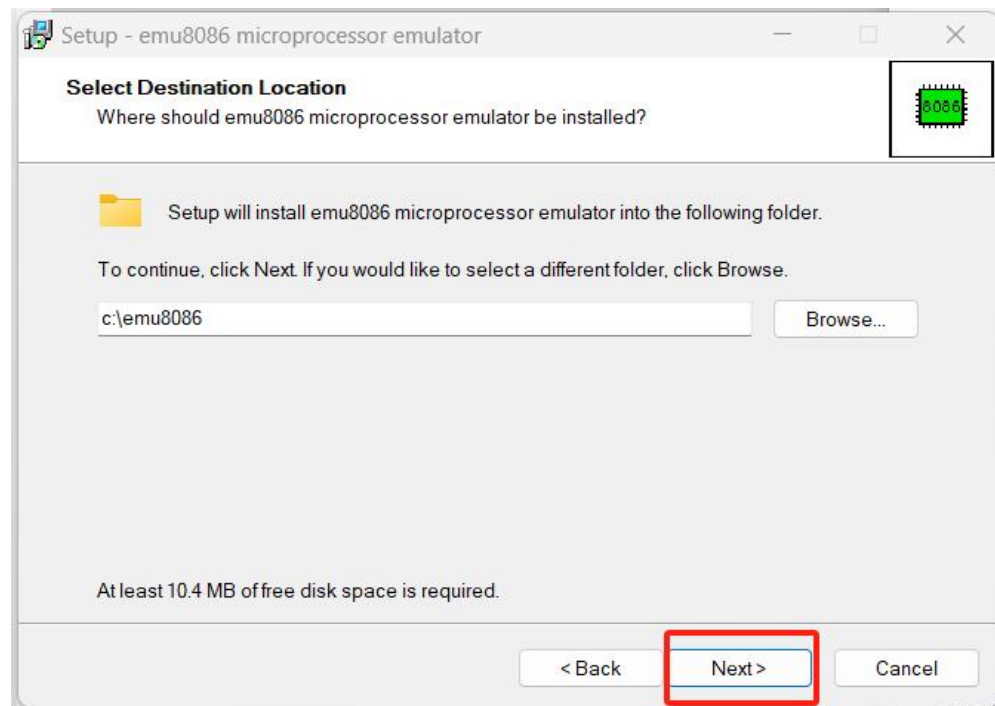
点击 next



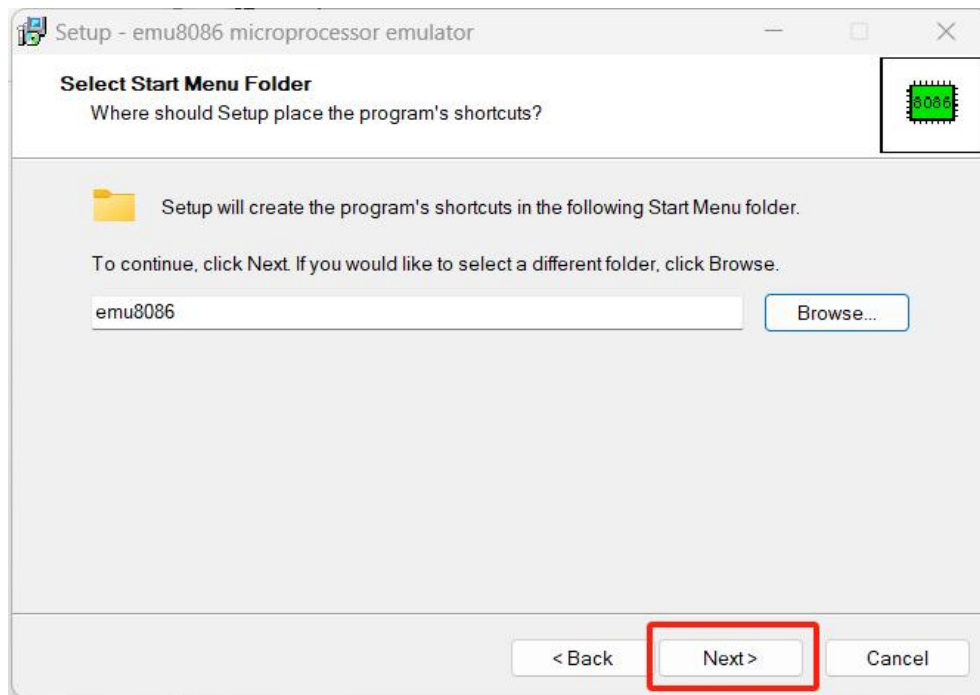
继续点击 next



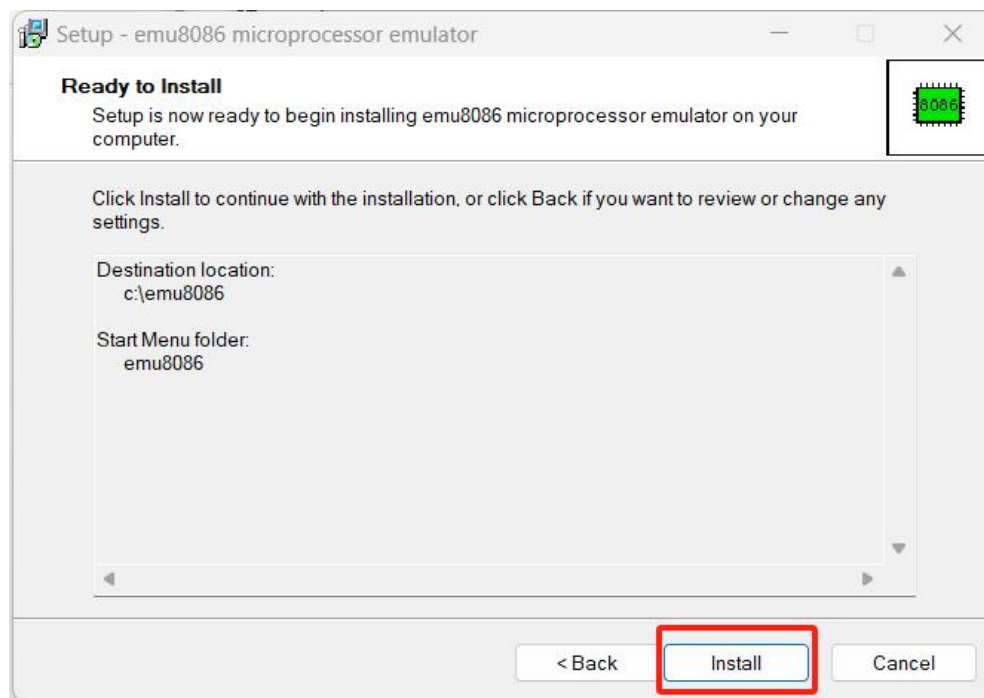
修改安装路径，之后点击 next



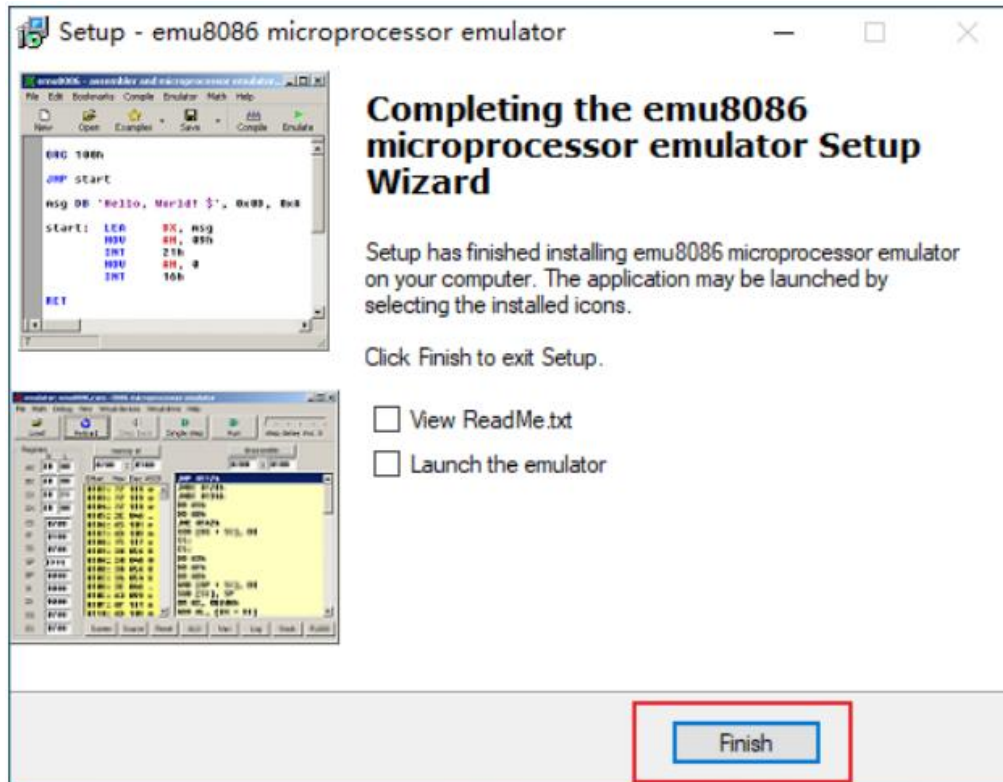
这一步是填写程序快捷方式存储的文件夹名称，默认就好，直接点击 Next



点击 install，进行安装

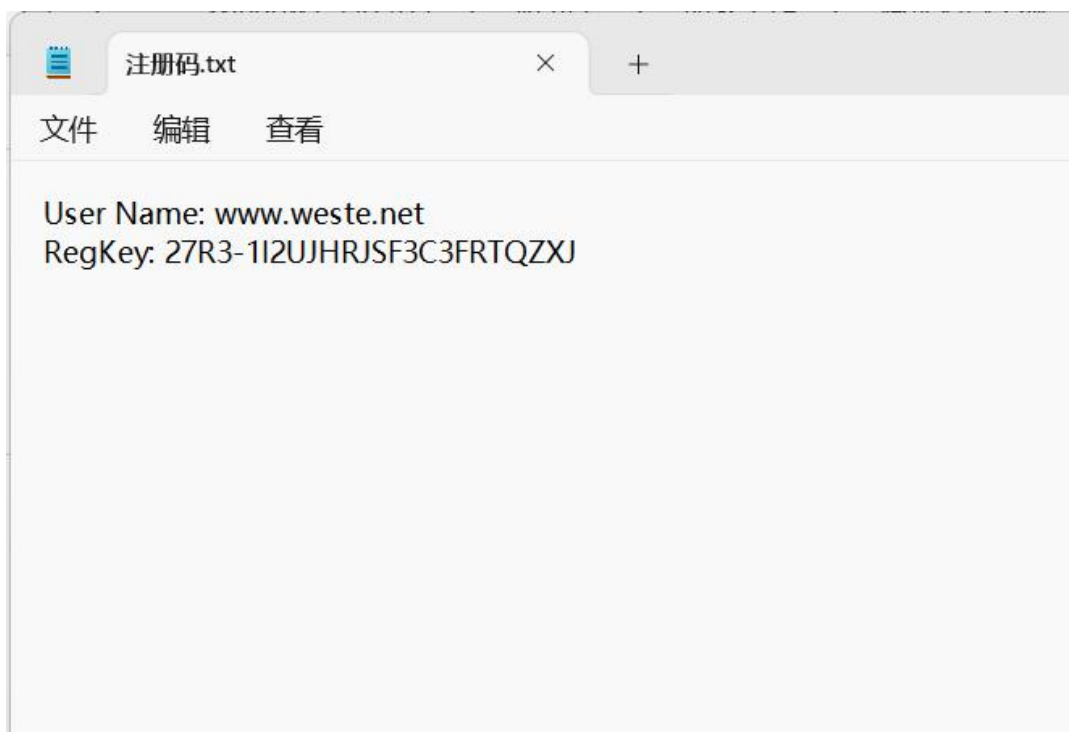


两项都不勾选，点击 finish 结束安装。



## 2. 破解

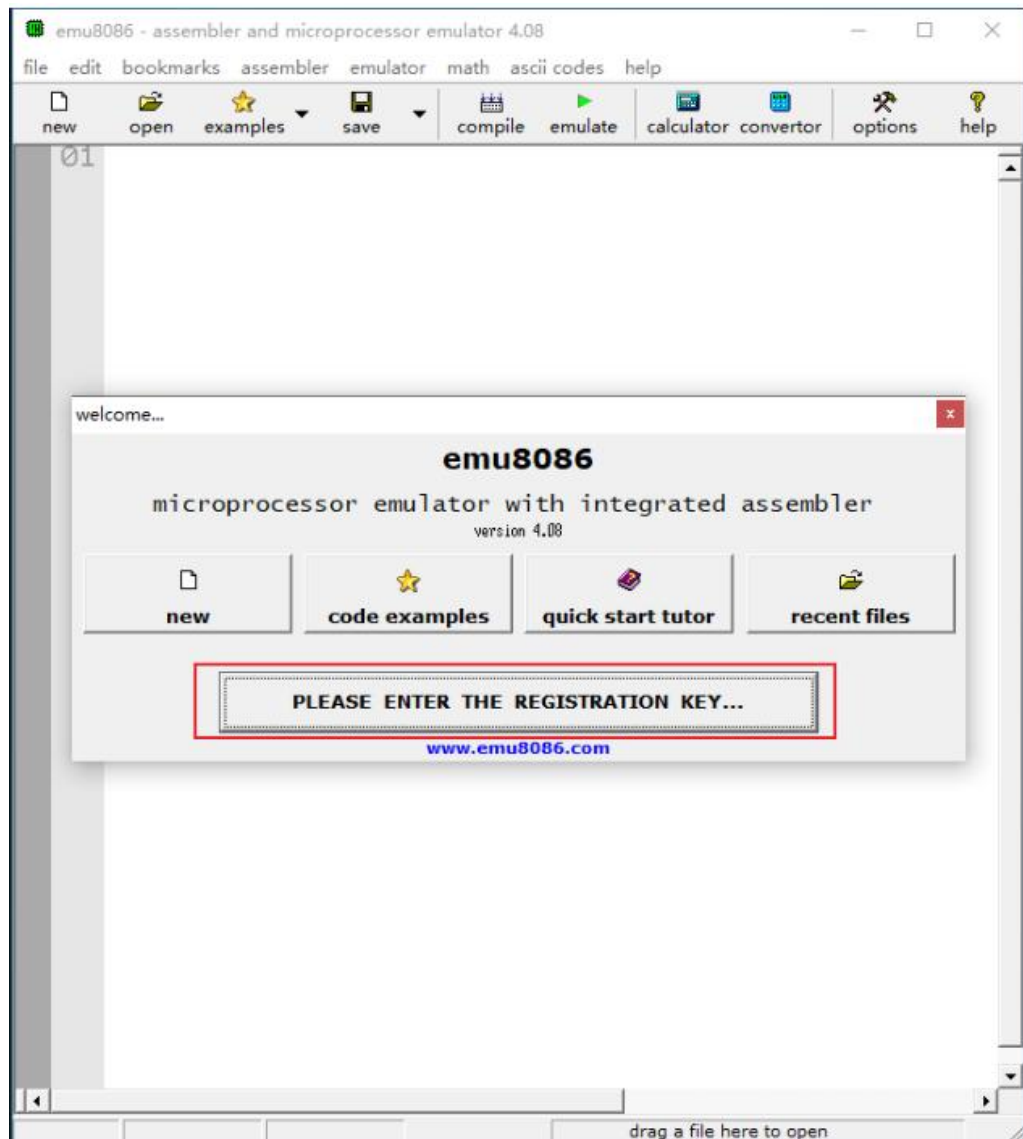
打开注册码.txt



User Name: www.weste.net

RegKey: 27R3-1I2UJHRJSF3C3FRTQZXJ

打开 emu8086，点击 PLEASE ENTER THE REGISTRATION KEY



输入用户名和 RegKey，点击 OK

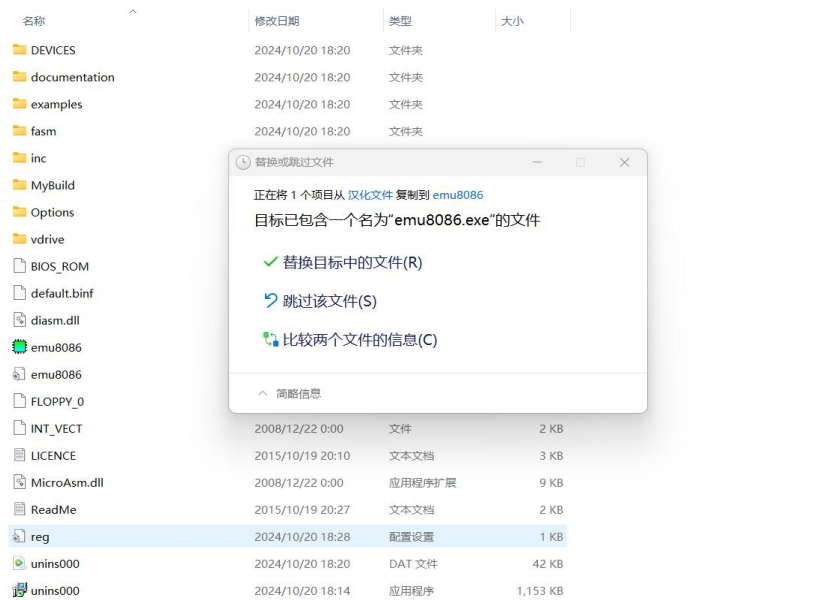


### 3. 汉化

关闭 emu8086，打开汉化文件文件夹，复制文件夹下的 emu8086.exe

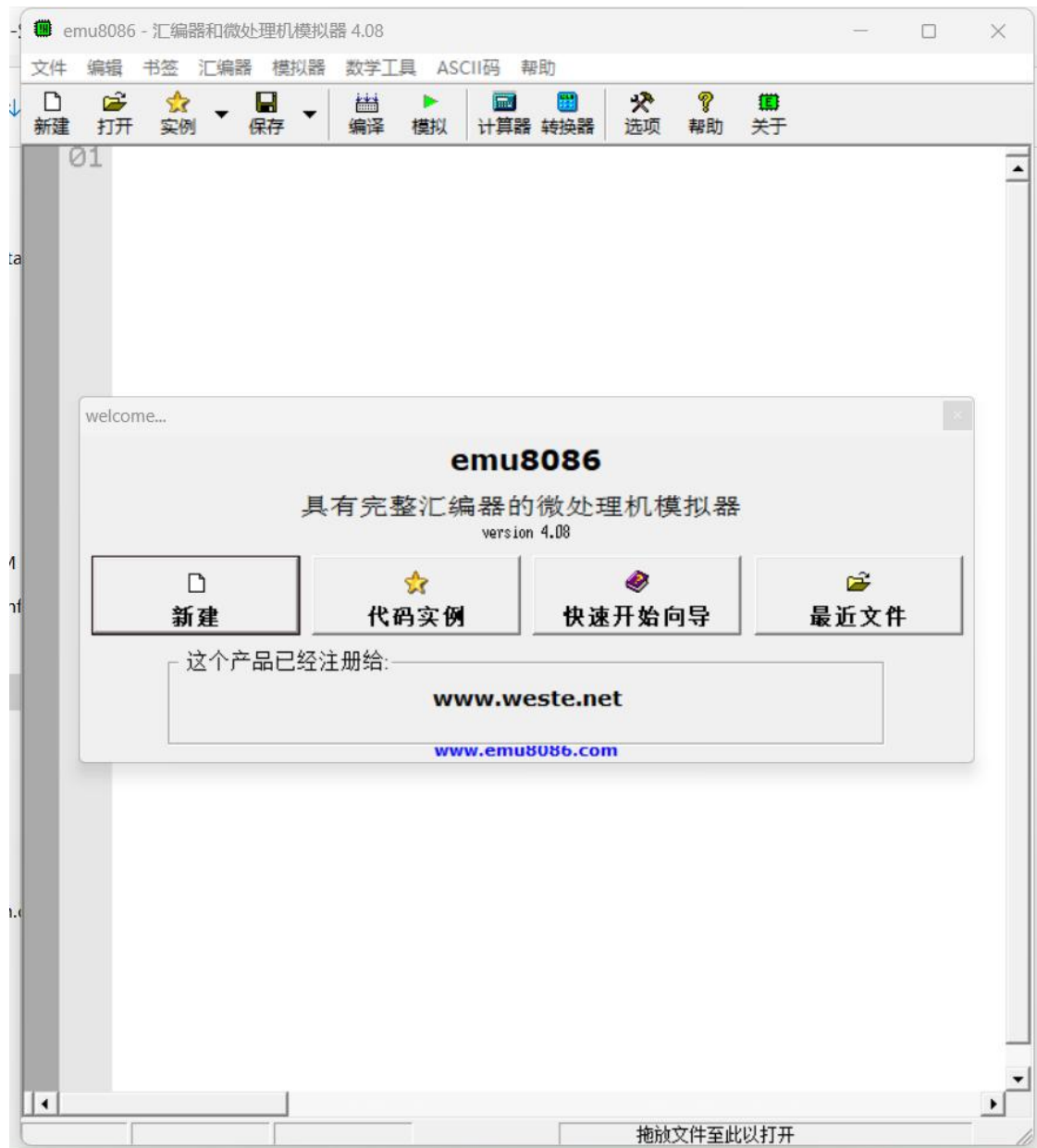


复制汉化文件目录下的 emu8086.exe 到安装目录下，替换安装目录下的 emu8086.exe



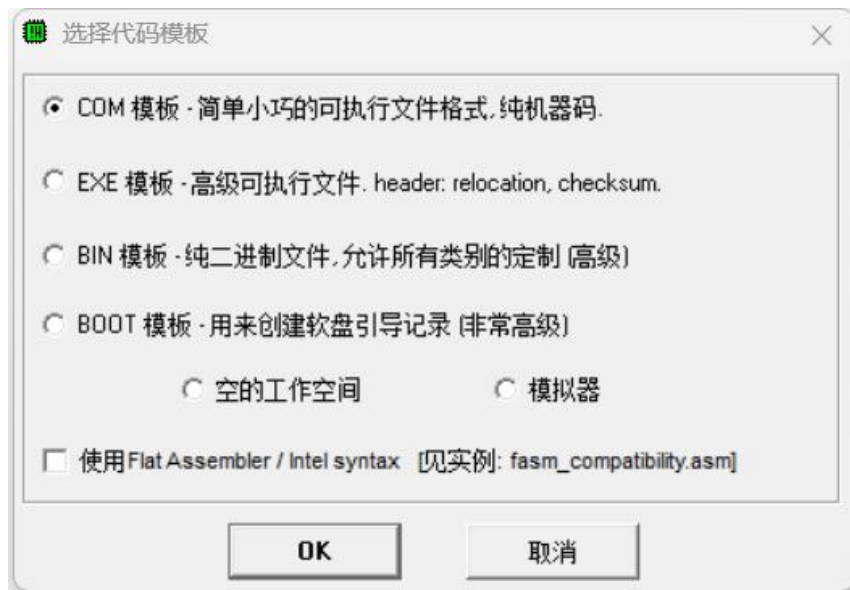


重新打开 emu8086，显示中文，汉化成功



#### 4. 执行代码

点击新建，选择 com 模板



代码示例如下：

```
emu8086 - 汇编器和微处理器模拟器 4.08
文件 编辑 书签 汇编器 模拟器 数学工具 ASCII码 帮助
新建 打开 实例 保存 编译 模拟 计算器 转换器 选项 帮助 关于

01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04
05 org 100h
06
07 ; add your code here
08 MOV AX, 5 ; 将5加载到AX寄存器
09 MOV BX, 10 ; 将10加载到BX寄存器
10 ADD AX, BX ; AX = AX + BX
11
12
13 ret
14
15
16
17
18
```

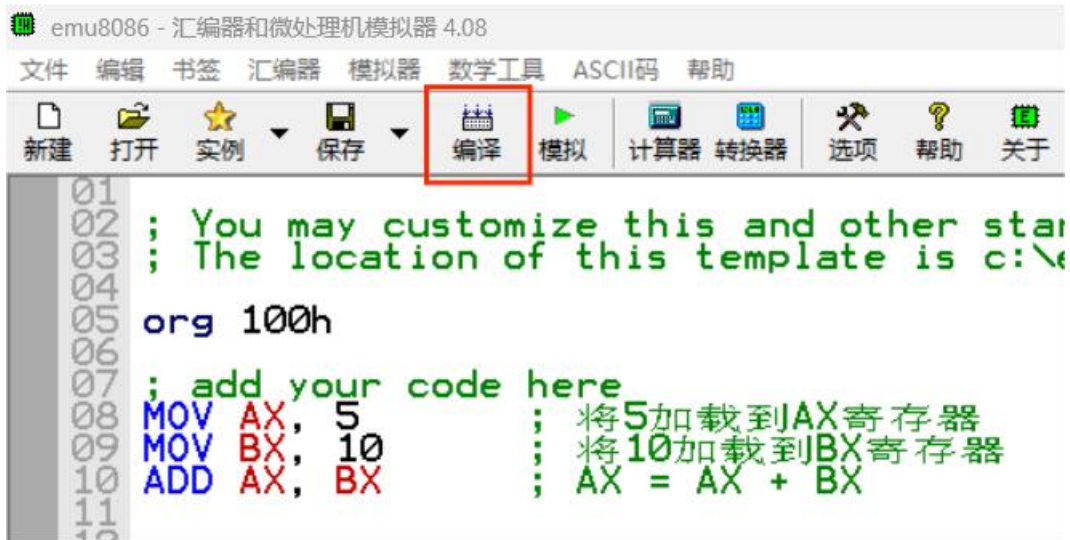
MOV AX, 5 ;将 5 加载到 AX 寄存器

MOV BX, 10 ;将 10 加载到 BX 寄存器

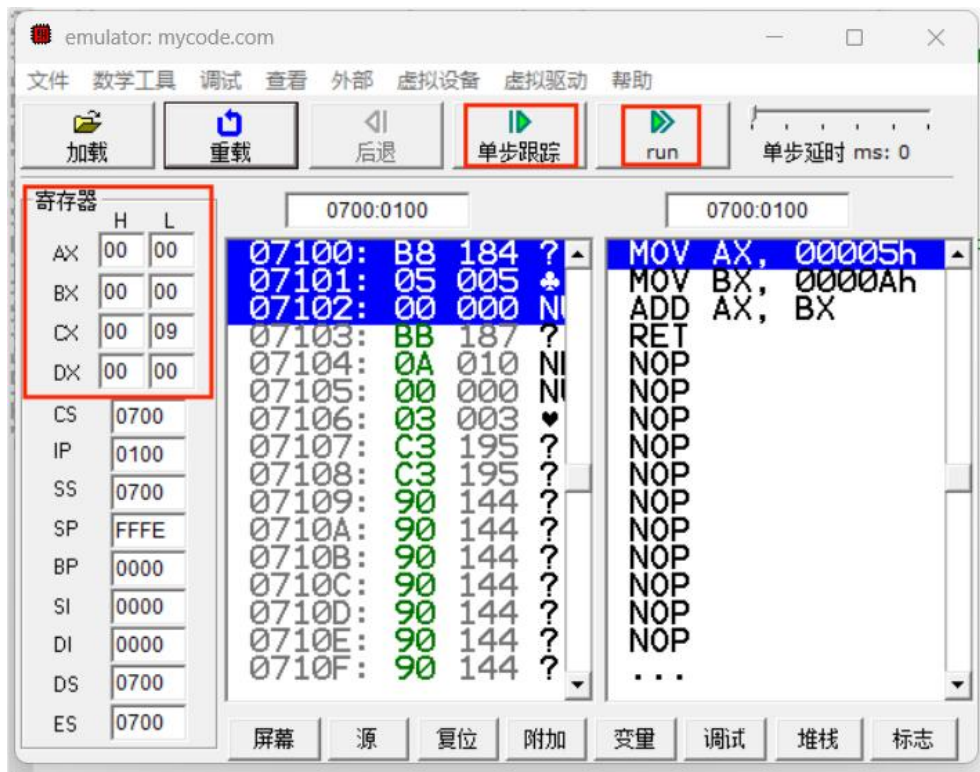
ADD AX,BX ;将 AX 和 BX 的值相加，结果存储在 AX 中

## 5. 执行结果示例

(1) 进行程序“编译”



(2) 程序编译结束没执行时寄存器状态如下



(3) 请点击“RUN”执行程序，观察寄存器变化

(4) 请点击“单步跟踪”逐步执行指令，观察寄存器变化

执行第一句指令，寄存器变化如下：

执行第二句指令，寄存器变化如下：

执行第三句指令，寄存器变化如下：