RISC-V仿真器的使用

# Ripes安装

Ripes是一个基于RISC-V指令集的虚拟计算机体系机构模拟器，我们可以在其中运行我们编写的RISC-V汇编程序。并且在这个过程中可以观察流水线各个阶段的执行过程，以及寄存器、内存和缓存中的数据情况。支持Windows、mac、Linux，使用很方便。

从GitHub：<https://github.com/mortbopet/Ripes> 下载软件包，解压后即可运行，无需安装。如果使用汇编语言,则可以直接进行仿真运行；如果希望使用C语言来编写程序，还需要通过点击菜单Eidt->Settings，对编译器的路径进行设置，如图 0‑1所示。

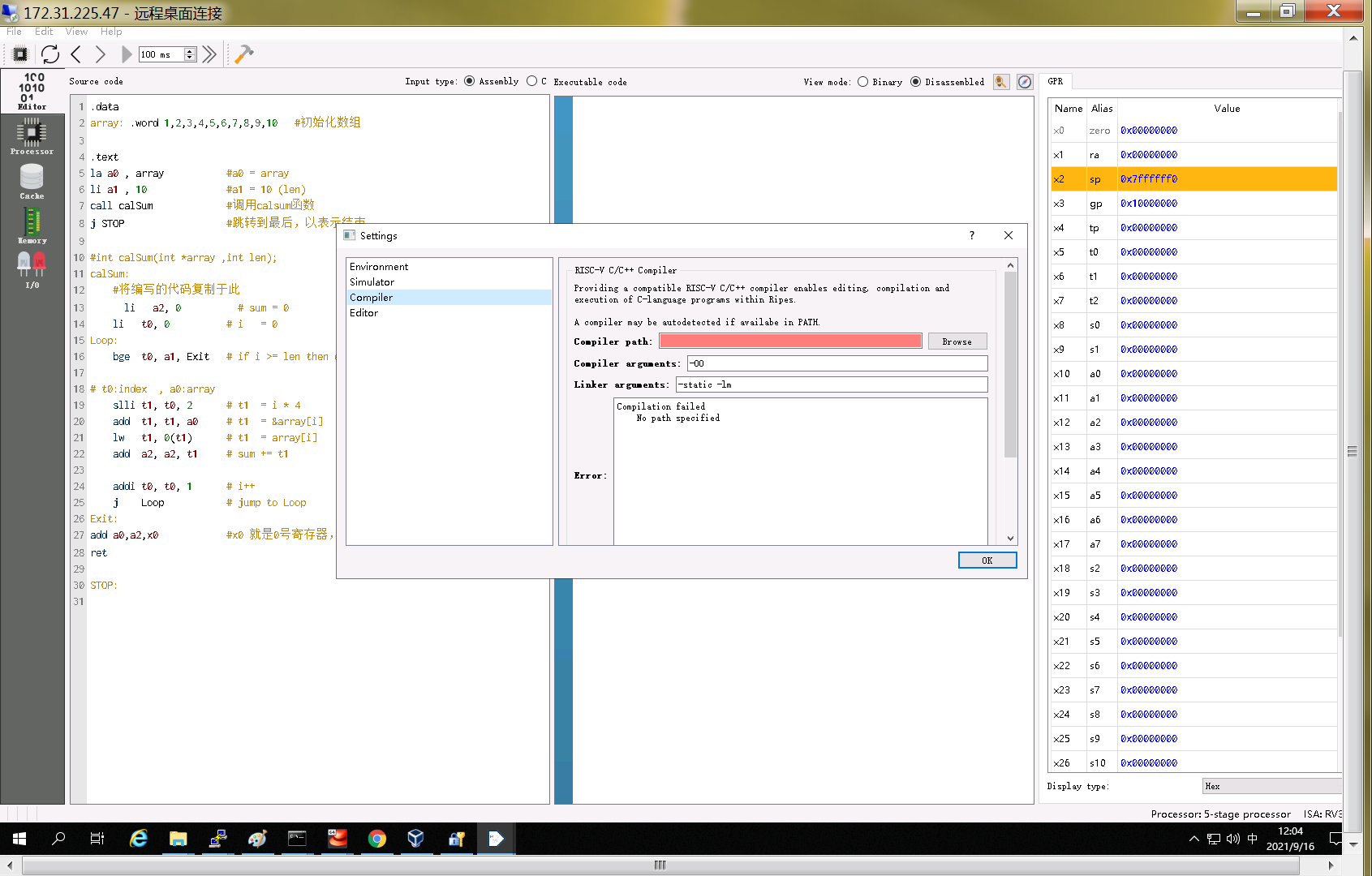


图 ‑1 C语言编译器的安装路径设置

如果希望使用C语言来编写程序，而你的系统中还没有安装RISC-V 的C编译器，那么还要先完成相应的编译器安装，具体见下面步骤。

# C编译器安装

本次试验无需C语言编译器

# 仿真测试过程

我们可以在editor界面中输入自己编写的汇编代码，如图 0‑1，随后运行即可。可以选择单步执行或者全部执行，全部执行的过程中可以调整延迟以便于观察，在右方可以看到各个寄存器在执行过程中的数值。

图形用户界面, 应用程序

描述已自动生成

图 ‑1 输入汇编指令

在Processor部分可以查看指令执行过程中数据通路的情况。

图示, 示意图

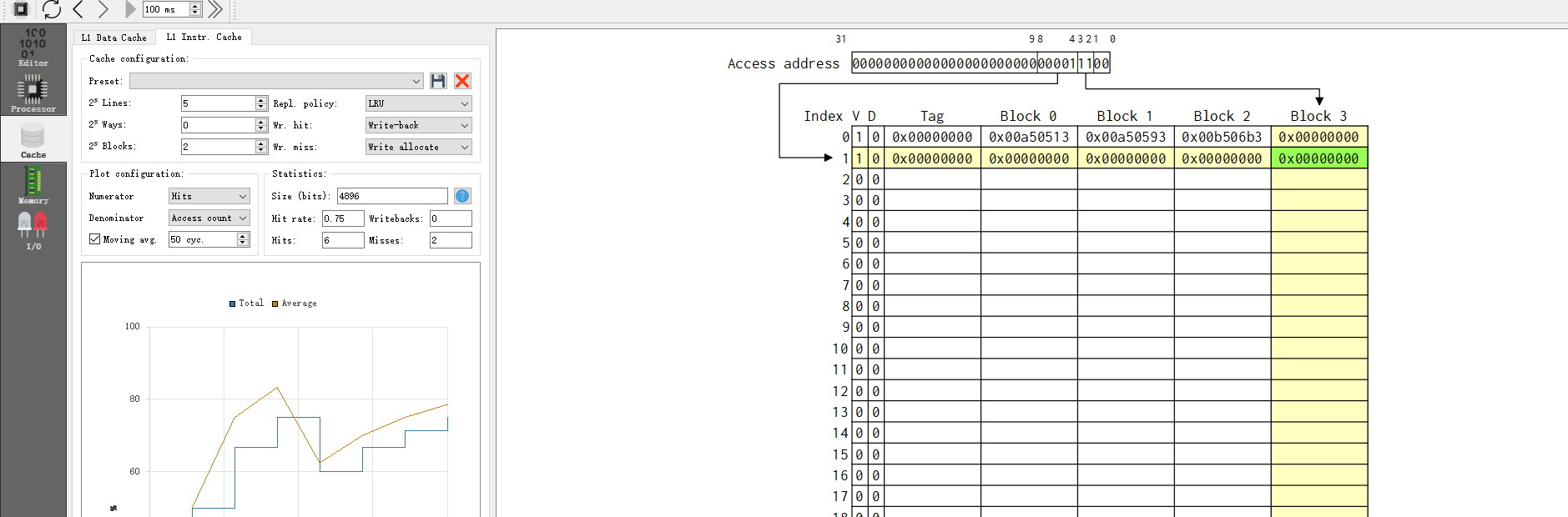
描述已自动生成

同时可以设置成更为具体的显示

图示

描述已自动生成

在cache部分可以设置cache的大小、相联度以及置换策略。并在程序执行过程中观察cache中的数据。



若想测试我们编写的求和代码，可以使用如下的代码框架，在运行完成之后通过检查对应寄存器中的数值验证程序的执行是否符合预期。

1. .data
2. array: .word 1,2,3,4,5,6,7,8,9,10 #初始化数组
3. .text
4. la a0 , array #a0 = array
5. li a1 , 10  #a1 = 10 (len)
6. call calSum  #调用calsum函数
7. j STOP
9. #int calSum(int \*array ,int len);
10. calSum:
11. #将你编写的代码复制于此
12. STOP:
13. nop #空指令,即什么都不做(相当于addi x0 , 0)