RISC-V 汇编练习

# 一、学习资料

**RISC-V手册：**

http://riscvbook.com/chinese/RISC-V-Reader-Chinese-v2p1.pdf

RISC-V手册的作者将这本薄薄的书作为RISC-V的介绍和参考资料，供有兴趣编写RISC-V代码的学生和嵌入式系统程序员使用。假设读者事先已经了解过至少一个指令集。通过阅读此手册可以很快的对RISC-V指令集有一个清晰的认识。

**Computer Organization and Design RISC-V Edition: The Hardware Software Interface：**http://home.ustc.edu.cn/~louwenqi/reference\_books\_tools/Computer%20Organization%20and%20Design%20RISC-V%20edition.pdf

《计算机组成与设计：硬件/软件接口》的最新版本，采用了RISC-V指令集作为讲解，同时中文译本也已经出版。

**RISC-V cheatsheet**：https://metalcode.eu/2019-12-06-rv32i.html

RV32I指令集的cheatsheet，在编写汇编时可以很方便的查阅具体指令的作用与格式。

## 二、练习题

首先学习《RISC-V 手册》第二章“RV32I”23~33页，以及第三章“RISC-V汇编语言”41~50页，主要掌握内存读取、算术运算指令。

然后编写一个RISC-V汇编程序，其功能是将数组内的元素求和，对应功能的C函数如下。假定数组array的首地址已经存放于a0寄存器中，数组长度len存放于a1寄存器中，变量sum使用a2寄存器，最后将返回结果放置回a0寄存器中并使用ret指令返回。

```c

int calSum(int \*array , int len){

int sum = 0;

for(int i = 0 ; i < len ; i++)

sum += array[i];

return sum;

}

```

汇编程序要求：代码按照如下框架编写

calSum:

初始化 # i=0

Loop: 条件判断，不成立则跳转到Exit

循环体

循环变量调整 #i++

Exit:

返回 sum值

2. 访问数组元素

对于数组元素的读取，可以通过`lw`指令来完成。在读取之前首先要获得需要读取元素的内存地址，在这里可以通过`slli`指令和`add`指令来完成。

对于数组下标为index的元素，其地址为array + index \* sizeof(element)，对于int类型来说sizeof(int) = 4。因此在这里需要首先使用移位指令将index左移两位以实现index\*4，并且将其与数组首地址相加得到元素的地址。最后通过`lw`指令读取对应的数值到寄存器中。

3 main函数代码（.text）框架

为了能使用仿真器运行自己写的求和代码并检查结果是否正常，需要将程序编写完整。首先需要初始化数组中的数据，并且将函数的参数保存到a0和a1寄存器中，然后使用call指令调用我们编写的求和函数。在仿真器中运行，最终在a0寄存器中查看计算结果是否符合预期。

只需要将我们前面编写的循环实现求和的代码复制到下面calSum对应的位置后即可。

1. .data
2. array: .word 1,2,3,4,5,6,7,8,9,10   #初始化数组
4. .text
5. la a0 , array           #a0 = array
6. li a1 , 10              #a1 = 10 (len)
7. call calSum             #调用calsum函数
8. j STOP                  #跳转到最后，以表示结束
10. #int calSum(int \*array ,int len);
11. calSum:
12. #将编写的代码复制于此
14. STOP:
15. nop #使用Ripes仿真时，需要加上这条nop指令

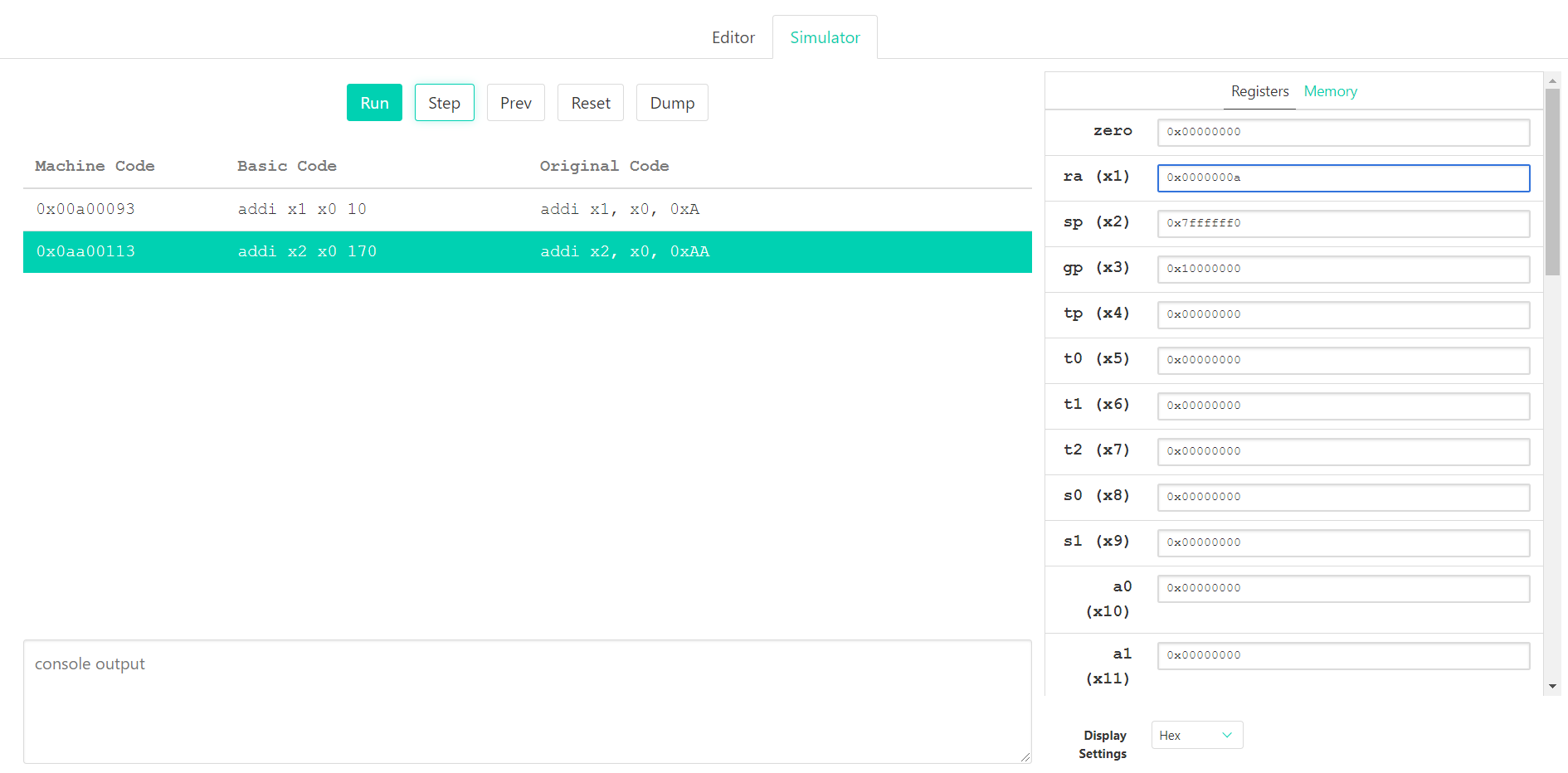
## 二、仿真运行

可以使用网络在线的仿真器，无须安装软件。也可以使用Ripes仿真软件，解压后即可运行。

1. 在RISC-V仿真器venus中运行汇编代码

可在仿真环境上运行我们编写的RISC-V汇编代码，对其进行功能的验证。在这里可以使用RISC-V在线仿真器。url: <https://www.kvakil.me/venus/>

可使用上面准备好的完整代码框架进行测试。通过仿真器，可以将我们自行编写的RISC-V汇编代码粘贴到Editor上，随后在Simulator中运行我们编写的汇编代码。在仿真执行的过程当中可以选择Step单步执行，并在这个过程中观察各个寄存器（Registers）中数值以及内存（Memory）数据的变化。



注意区分RISC-V的伪指令和真实汇编指令的不同，它们在仿真器Simulator中分别是Original  
Code和Basic Code。例如la指令是伪指令，会被拆分成两条basic code。

2. 在RISC-V仿真器Ripes中运行汇编代码

请见“Ripes仿真器的使用.docx”文档，进行安装和运行。

## 四、提交内容

1 sum函数的汇编代码 （60分）

2 运行结果截图，并说明：1）array在内存中的地址和数据；2）运行结束时sum对应的a2寄存器的值（40分）