计算机组成原理实验报告

1. **实验题目**

利用RISC-V指令系统实现冒泡排序算法

1. **实验目的**

基于RV32IM汇编，设计一个冒泡排序程序，并用Ripes工具进行调试执行，测量冒泡排序程序的执行时间

1. **实验平台**

Ripes工具

1. **实验过程**
2. 设计思路

临时变量i从1曾至数组长度n-1，用两个寄存器分别从数组元素对应地址中读取array[i-1]和array[i]进行比较，若比较结果为逆序，则将两个元素交换。设置一个交换标志flag，初始默认值为0，若发生交换，将flag设置为1。I的值增加至n-1后，进行判断，若发生过交换，即flag==1则将i的值恢复为1，flag的值重新赋为0，重新进行循环操作。

1. 代码解释

排序算法的实现主要分为以下几个部分

▶输出output：对数组元素进行输出，展示算法实现结果

▶外循环outerloop：对临时变量，标志变量进行初始化

▶内循环innerloop：取数组元素并比较，依据比较结果进入交换部分或者跳过交换部分

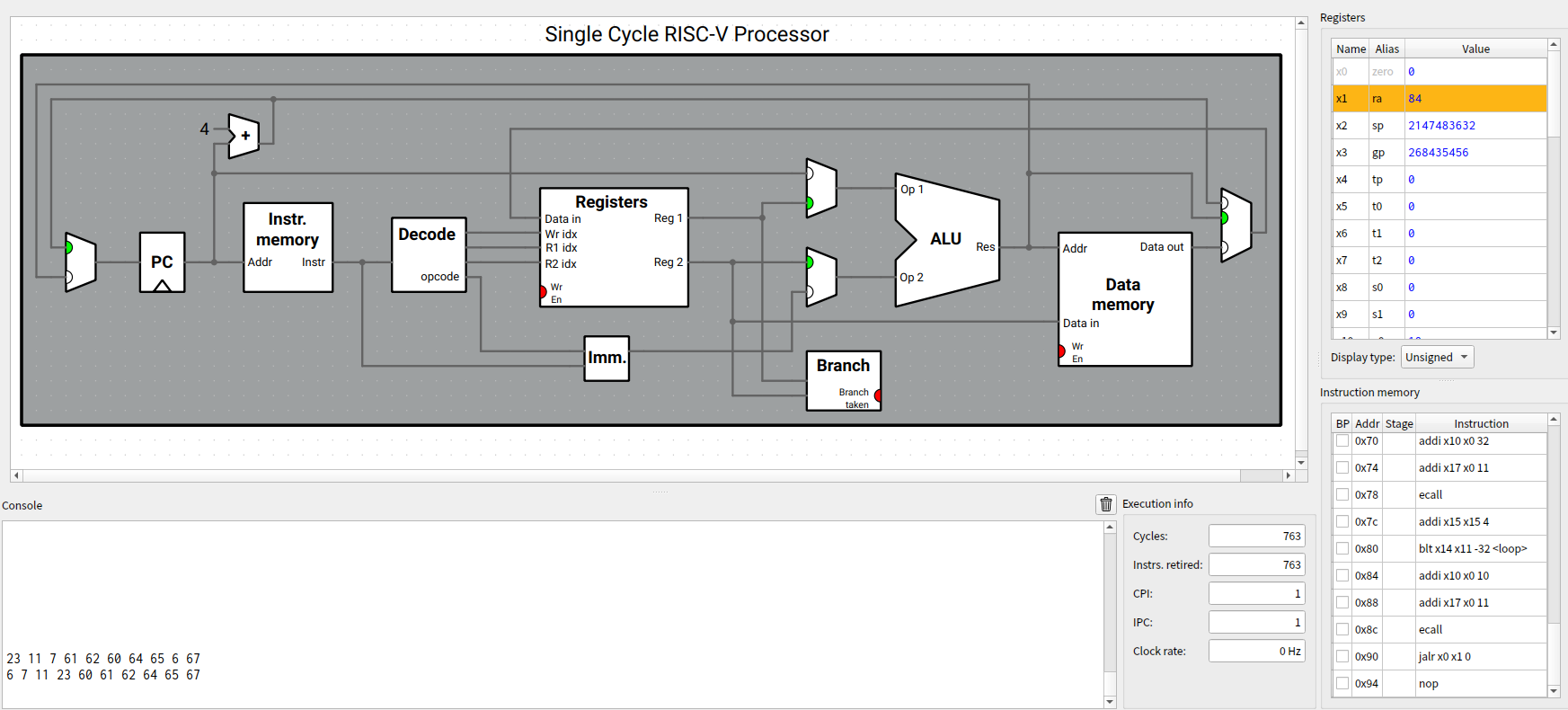
▶交换swap：交换两个元素的值，并且值标志变量的值

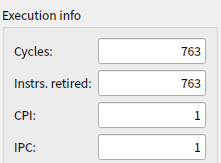
▶跳过交换skip-swap：对临时变量进行增加，依据运行结果判断是否继续内循环或外循环

1. 代码展示
2. .data
4. n: .word 10
5. array: .word 23,11,7,61,62,60,64,65,6,67
7. .text
9. lw a1 n                        #a1 = n
10. la a3 array                    #a3 -> array[0]
11. li a4  0                       #a4 = i,now a4 = 0
12. mv a5 a3                       #a5 -> array[0] now
14. jal x1 output                  #Output array  first
16. outerloop:
17. li a4 1                        #i = 1
18. mv a5 a3                       #a5 -> array[0] now
19. mv a6 zero                     #a6:swap flag: 1 **else**: 0,now flag = 0
21. innerloop:
22. lw a2 0(a5)                    #a2 = array[i - 1]
23. lw s2 4(a5)                    #a3 = array[i]
24. bge s2 a2 skip\_swap            #**if** array[i] >= array[i - 1] skip swap
26. swap:
27. sw a2 4(a5)                    #array[i] = a2
28. sw s2 0(a5)                    #array[i - 1] = a3
29. addi a6 zero 1                 #flag = 1
31. skip\_swap:
32. addi a5 a5 4                   #a5 point to array[i - 1 + 1]
33. addi a4 a4 1                   #i = i + 1
34. blt a4 a1 innerloop            #**if**(i < n) cotinue
35. bne a6 zero outerloop          #**if** flag == 1 occurs **continue** outerloop
36. jal x1 output                  #output array after sort
37. j Exit                         #Exit
39. output:
40. li a4  0                       #a4 = i,now a4 = 0
41. mv a5 a3                       #a5 -> array[0] now
42. loop:
43. addi a4 a4 1                   #i = i + 1
44. lw a0 0(a5)                    #a0 = a[i - 1]
45. li a7 1
46. ecall                          #output a[i - 1]
47. li a0 32
48. li a7 11
49. ecall                          #output ' '
50. addi a5 a5 4
51. blt a4 a1 loop                 #**if**(i < n) **continue**
52. li a0 10
53. li a7 11
54. ecall                          #output '\n'
55. ret
57. Exit:
58. nop                            #EXIT
59. **实验结果**

▶测试数据1：23 11 7 61 62 60 64 65 6 67

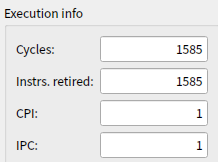






▶测试数据2：0 0 0 78 65 43 44 5 9 1 1 -2 43 15 24 666





1. **总结反思**

初次使用汇编语言实现已经学习过的算法，这个过程中我还是遇到了许多困难，例如对指令没有熟练掌握，对汇编程序的结构还不是很熟悉，这次实验是一个学习的过程，也是一个克服定式思维的过程，我需要跳出以前编写C语言时的思维模式来思考问题，来实现算法。因此可以用收获颇丰来形容此次实验的过程。