

# 中国科学技术大学计算机学院

## 《计算机组成原理》实验报告



实验题目：LabH3\_CPU 测试及汇编程序设计

学生姓名：钟书锐

学生学号：PB19000362

完成日期：2021.3.31

计算机实验教学中心制

2020 年 09 月

### 【实验题目】

LabH3\_CPU 测试及汇编程序设计

### 【实验目的】

1. 掌握 Rars 的使用
2. 掌握 Ripes-v2.1.0-win-x86\_64 的使用
3. 熟悉汇编指令

### 【实验环境】

硬件：

处理器：i7-10750H @ 2.60GHz 六核

显卡：RTX2060(6GB)

操作系统：

WINDOWS10 家庭中文版

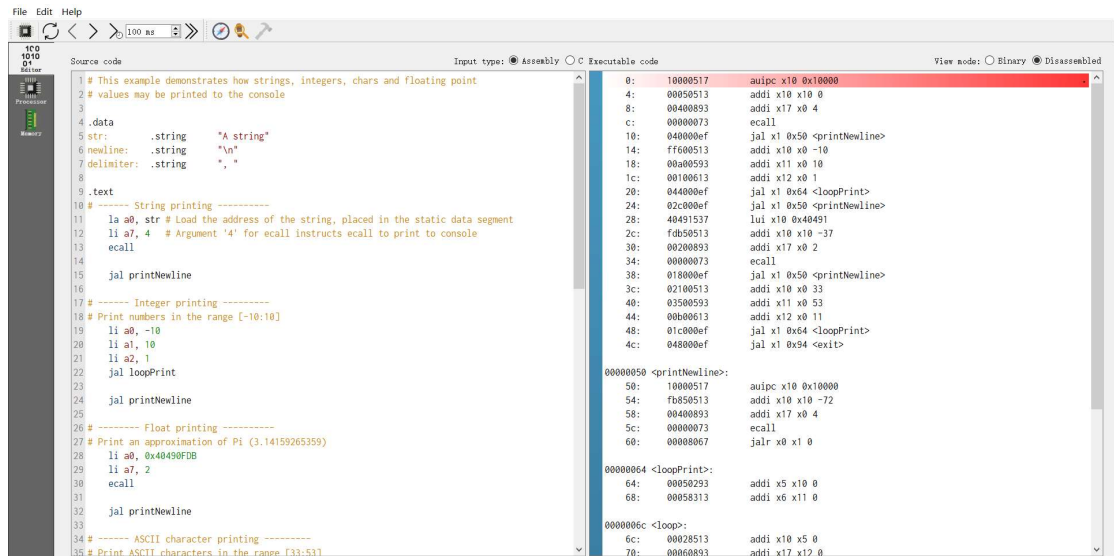
软件：

Rars

Ripes-v2.1.0-win-x86\_64

## 【实验内容】

### 1. 仿真 RIPES 示例汇编程序 (Console Printing)

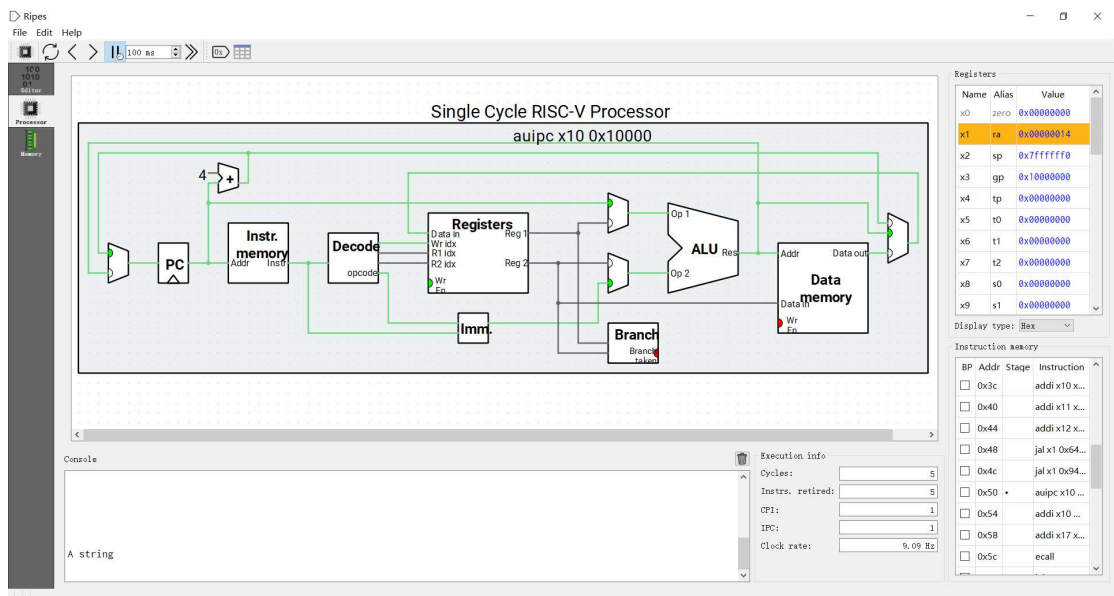


The screenshot shows the Ripes IDE with the source code on the left and the assembly output on the right. The source code is a RISC-V assembly program that demonstrates various data types and control flow. The assembly output shows the corresponding machine code instructions.

```
1 # This example demonstrates how strings, integers, chars and floating point
2 # values may be printed to the console
3
4 .data
5 str: .string "A string"
6 newline: .string "\n"
7 delimiter: .string ", "
8
9 .text
10 # ----- String printing -----
11 la a0, str # Load the address of the string, placed in the static data segment
12 li a7, 4 # Argument '4' for ecall instructs ecall to print to console
13 ecall
14
15 jal printNewline
16
17 # ----- Integer printing -----
18 # Print numbers in the range [-10:10]
19 li a0, -10
20 li a1, 10
21 li a2, 1
22 jal loopPrint
23
24 jal printNewline
25
26 # ----- Float printing -----
27 # Print an approximation of Pi (3.14159265359)
28 li a0, 0x40490FDB
29 li a7, 2
30 ecall
31
32 jal printNewline
33
34 # ----- ASCII character printing -----
35 # Print ASCII characters in the range f'33-53'
```

The assembly output shows the following instructions:

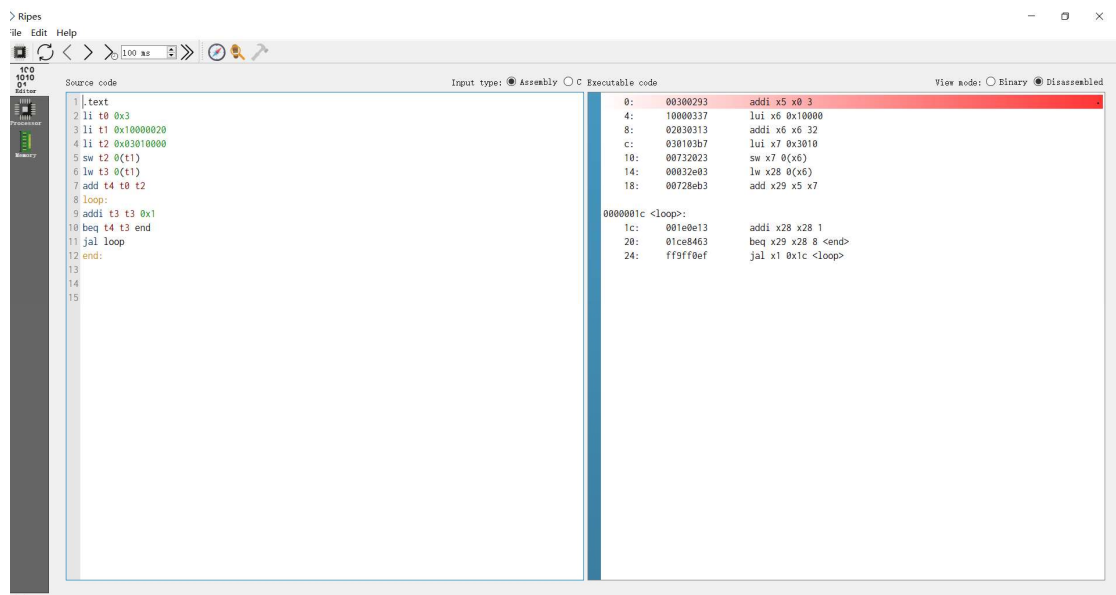
```
0: 10000517 auipc x10, 0x10000
4: 00050513 addi x10, x10, 0
8: 00400893 addi x17, x0, 4
c: 00000073 ecall
10: 040000ef jal x1, 0x50 <printNewline>
14: ff600513 addi x10, x0, -10
18: 00a00593 addi x11, x0, 10
1c: 00100613 addi x12, x0, 1
20: 044000ef jal x1, 0x64 <loopPrint>
24: 02c000ef jal x1, 0x50 <printNewline>
28: 40491537 lui x10, 0x40491
2c: fdb50513 addi x10, x10, -37
30: 00200893 addi x17, x0, 2
34: 00000073 ecall
38: 018000ef jal x1, 0x50 <printNewline>
3c: 02100513 addi x10, x0, 33
40: 03500593 addi x11, x0, 53
44: 00b00613 addi x12, x0, 11
48: 01c000ef jal x1, 0x64 <loopPrint>
4c: 048000ef jal x1, 0x94 <exit>
00000050 <printNewline>:
50: 10000517 auipc x10, 0x10000
54: fb50513 addi x10, x10, -72
58: 00400893 addi x17, x0, 4
5c: 00000073 ecall
60: 00000067 jalr x0, x1, 0
00000054 <loopPrint>:
64: 00052233 addi x5, x10, 0
68: 00058313 addi x6, x11, 0
0000006c <loop>:
6c: 00028513 addi x10, x5, 0
70: 00b60893 addi x17, x12, 0
```



## 2. 人工检查 6 条指令功能

### 核心代码

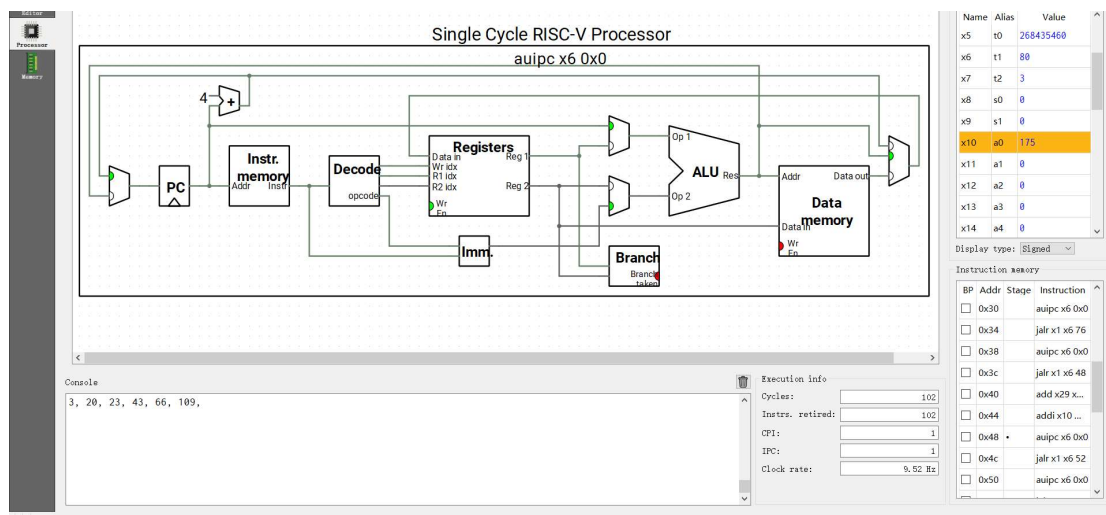
```
1 .text
2 li t0 0x3
3 li t1 0x10000020
4 li t2 0x03010000
5 sw t2 0(t1)
6 lw t3 0(t1)
7 add t4 t0 t2
8 loop:
9 addi t3 t3 0x1
10 beq t4 t3 end
11 jal loop
12 end:
```



### 3. 汇编程序计算斐波那契—卢卡斯数列

#### 核心代码

```
1  .data
2  number1: .byte 3
3  number2: .byte 20
4  delimiter: .string  ", "
5  .text
6  la t0 number1
7  lw t2 0(t0)
8  la t0 number2
9  lw t3 0(t0)
10 addi a0 t2 0
11 call print2
12 call print1
13 addi a0 t3 0
14 call print2
15 call print1
16 add t4 t2 t3
17 loop:
18 addi a0 t4 0
19 call print2
20 call print1
21 addi t5 t4 0
22 add t4 t3 t4
23 addi t3 t5 0
24 j loop
```



输出到 console

## 【实验总结】

复习了 RISC-V 汇编