

计算机组成原理 实验报告

实验题目：汇编程序设计

学生姓名：阿非提

学生学号：PB20111633

完成日期：2022.4.12

实验目的

- 熟悉RISC-V汇编指令的格式
- 熟悉CPU仿真软件Ripes，理解汇编指令执行的基本原理（数据通路和控制器的协调工作过程）
- 熟悉汇编程序的基本结构，掌握简单汇编程序的设计
- 掌握汇编仿真软件RARS(RISC-V Assembler & Runtime Simulator)的使用方法，会用该软件进行汇编程序的仿真、调试以及生成CPU测试需要的指令和数据文件（COE）
- 理解CPU调试模块PDU的使用方法

实验平台

- Ripes
- RARS

实验练习

- 题目1

指令

```
# values may be printed to the console
```

```
.data
str:      .string      "A string"
newline:  .string      "\n"
delimiter: .string      ", "
```

```
.text
# ----- String printing -----
    la a0, str
    li a7, 4
    ecall

    jal printNewline
```

将字符串str的首地址存在a0里，并令a7=4，调用ecall。ecall将会把存放在a0的内容根据参数a7输出到终端。例如：

a7=1, a0以整数输出；

a7=2, a0以浮点数输出；

a7=4, 存放在a0地址的内容以字符串输出；

a7=11, a0作为ASCII码以字符形式输出；

```
# ===== Helper routines =====  
printNewline:  
    la a0, newline  
    li a7, 4  
    ecall  
    jr x1
```

printNewline 将换行

```
# ----- Integer printing -----  
# Print numbers in the range [-10:10]  
    li a0, -10  
    li a1, 10  
    li a2, 1  
    jal loopPrint  
  
    jal printNewline
```

将a0 = -10 , a1 = 10, a2 = 1 作为参数调用loopPrint,

```
loopPrint:  
    mv t0 a0  
    mv t1 a1  
loop:  
    # Print value in a0 as specified by argument a2  
    mv a0 t0  
    mv a7 a2  
    ecall  
    # Print a delimiter between the numbers  
    li a7, 4  
    la a0, delimiter
```

```

    ecall
    # Increment
    addi t0, t0, 1
    ble  t0, t1, loop
    jr x1

```

loopPrint 先将a0, a1的值分别保存在t0, t1, 再通过a0 = t0; a7 = a2 = 1 作为参数调用ecall将t0的内容以整数的形式输出到终端中, 并在调用ecall输出', '。使t0 = t0 + 1, 如t0小于等于t1则继续执行上述步骤。此时终端上会有 -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10。

```

# ----- Float printing -----
# Print an approximation of Pi (3.14159265359)
    li a0, 0x40490FDB
    li a7, 2
    ecall

    jal printNewline

```

输出浮点数 π

```

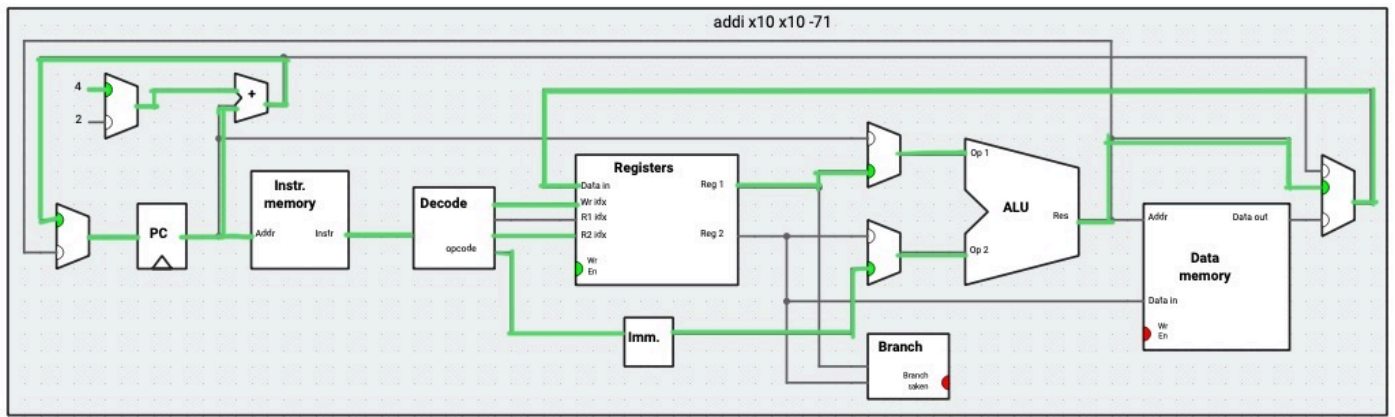
# ----- ASCII character printing -----
# Print ASCII characters in the range [33:53]
    li a0, 33
    li a1, 53
    li a2, 11
    jal loopPrint

# Finish execution
    jal exit

```

将ascii 码 在 33 到 53 的字符以字符输出。

以 addi x10, x10, -71 指令为例, 数据通路如下图所示。



• 题目2

`.data`

`Data: .word 0x17`

`.text`

`#sw, lw, addi`

`la a0, Data`

`lw a1, 0(a0)`

`addi a1, a1, -10`

`sw a1, 0(a0)`

`#add, jal, beq`

`li a0, 23`

`li a1, 13`

`li a2, 10`

`add a4, a1, a2`

`beq a0, a4, GoTo`

`GoTo:`

`jal Exit`

`Exit:`

`a0` = `Data` 的指向的地址

`a1` = `a0` 地址中的内容 = 17

`a1` = `a1` - 10 = 13

`a0` 地址中的内容 = 17

`a0` = 23

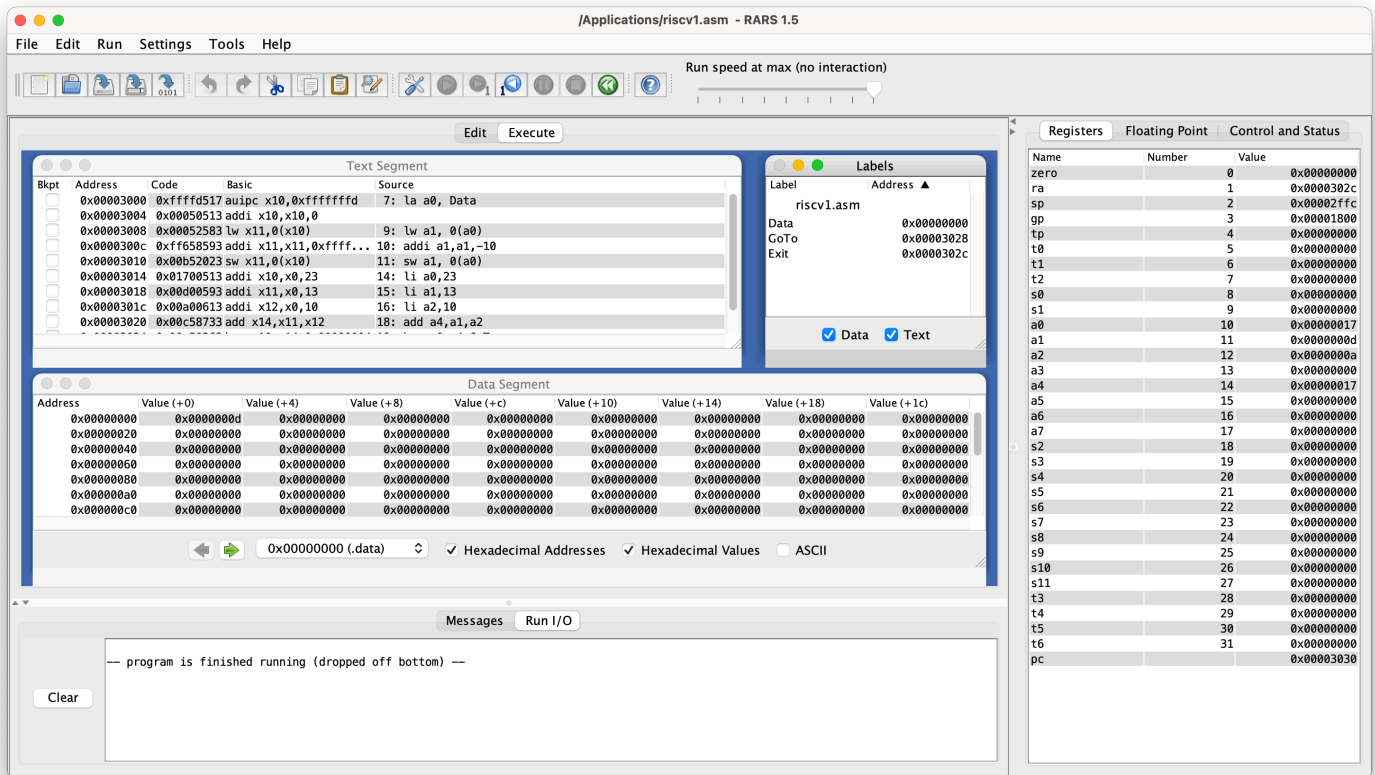
`a1` = 13

`a2` = 10

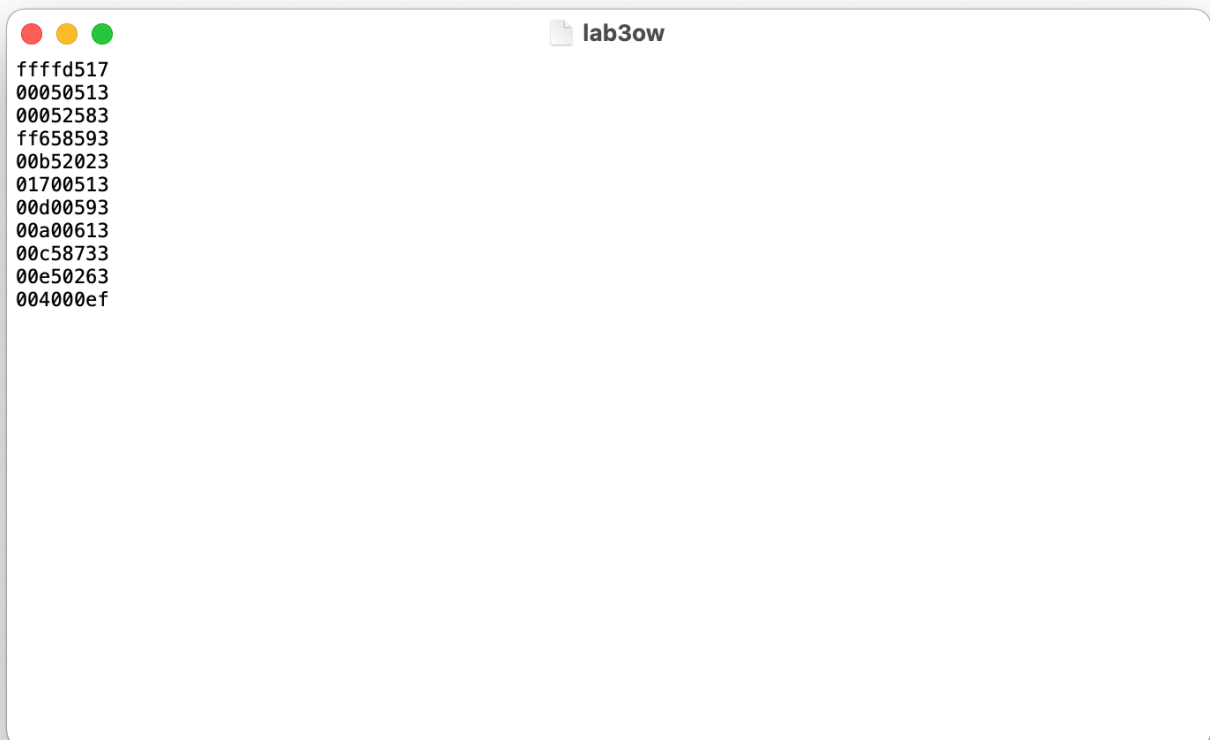
`a4` = `a1` + `a2` = 23

If (`a0` == `a4`)

GoTo



生成COE文件



• 题目3

指令:

```
.data
    PROMPT: .string "input n:"
    RESULT: .string "f(n)="
.text
    la a0,PROMPT
    li a7,4
    ecall
    li a7,5
    ecall

    li t0,1
    li t1,2
    mv t2,a0
    li t3,3

LOOP:   blt a0,t3,PRINT
        add t2,t1,t0
        add t0,t1,zero
        add t1,t2,zero
        addi a0,a0,-1
        jal LOOP

PRINT:
    la a0,RESULT
    li a7,4
    ecall
    mv a0,t2
    li a7,1
    ecall
```

生产COE文件



总结与思考

- 本次试验难易程偏低。
- 本次试验任务量少。