

1. 实验内容

- 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

2. 实验环境

- PC 一台
- Ripes
- Rars

3. 实验过程

(1)理解并仿真 RIPES 示例汇编程序

加载 Ripes 示例汇编程序 (Console Printing)-> 选择单周期 CPU 数据通路 -> 单步执行程序 -> 观察数据通路控制信号和寄存器内容的变化

- 示例一为计算两个复数相乘的程序
- 示例二为控制台输出的程序
- 示例三为计算阶乘的程序

(2)设计汇编程序, 验证 6 条指令功能

Rars 软件设计汇编程序 -> 单步运行程序 -> 人工检查 -> 生成 COE 文件

```
.data
testlw:    .word    10, 15
testsw:    .word    0
testbeq:   .word    0
testjal:   .word    0

.text
main:
    la a1, testlw
    lw a2, 4(a1)
    addi a2, x0, 5
    sw a2, 0(a1)
    add a2, a2, a2
```

```

sw a2, 0(a1)
add t0, x0, a2
addi t1, x0, 7
beq t0, a2, next
addi t1, t1, 7
next:
jal test
li a7, 10
ecall
test:
la a1, testjal
addi t1, t1, 7
sw t1, 0(a1)
jr x1

```

该程序取出 15 给 a2,再将 a2 利用立即数加为 5,并将其存储到 a1 的位置,再将 a2 和自身相加,再存到 a1 的位置,接着将 t0 a2 设置成相等的值,测试 beq 功能,最后测试 jal,将 t1 存储到对应位置

程序运行最后的数据映像:

| Reg | Address | Code | Basic | Source |
|-----|------------|------------------|------------------|--------|
| 0 | 0x00000000 | lui a2, 15 | lui a2, 15 | |
| 1 | 0x00000004 | addi a2, a2, 5 | addi a2, a2, 5 | |
| 2 | 0x00000008 | addi t0, x0, 7 | addi t0, x0, 7 | |
| 3 | 0x0000000c | beq t0, a2, next | beq t0, a2, next | |
| 4 | 0x00000010 | addi t1, t0, 7 | addi t1, t0, 7 | |
| 5 | 0x00000014 | jal test | jal test | |
| 6 | 0x00000018 | li a7, 10 | li a7, 10 | |
| 7 | 0x0000001c | ecall | ecall | |
| 8 | 0x00000020 | la a1, testjal | la a1, testjal | |
| 9 | 0x00000024 | sw t1, 0(a1) | sw t1, 0(a1) | |
| 10 | 0x00000028 | jr x1 | jr x1 | |

| Name | Number | Value |
|------|--------|-------|
| x0 | 0 | 0 |
| x1 | 1 | 12345 |
| x2 | 2 | 12345 |
| x3 | 3 | 6144 |
| x4 | 4 | 0 |
| x5 | 5 | 10 |
| x6 | 6 | 18 |
| x7 | 7 | 0 |
| x8 | 8 | 0 |
| x9 | 9 | 0 |
| x10 | 10 | 0 |
| x11 | 11 | 12 |
| x12 | 12 | 15 |
| x13 | 13 | 14 |
| x14 | 14 | 0 |
| x15 | 15 | 0 |
| x16 | 16 | 0 |
| x17 | 17 | 10 |
| x18 | 18 | 0 |
| x19 | 19 | 0 |
| x20 | 20 | 0 |
| x21 | 21 | 0 |
| x22 | 22 | 0 |
| x23 | 23 | 0 |
| x24 | 24 | 0 |
| x25 | 25 | 0 |
| x26 | 26 | 0 |
| x27 | 27 | 0 |
| x28 | 28 | 0 |
| x29 | 29 | 0 |
| x30 | 30 | 0 |
| x31 | 31 | 12345 |

生成 coe 文件如下所示:

| lab3 > data.coe | lab3 > text.coe |
|-------------------------------------|-------------------------------------|
| 1 memory_initialization_radix = 16; | 1 memory_initialization_radix = 16; |
| 2 memory_initialization_vector = | 2 memory_initialization_vector = |
| 3 0000000a | 3 fffffd597 |
| 4 0000000f | 4 00058593 |
| 5 00000000 | 5 0045a603 |
| 6 00000000 | 6 00500613 |
| 7 00000000 | 7 00c5a023 |
| 8 00000000 | 8 00c60633 |
| 9 00000000 | 9 00c5a023 |
| 10 00000000 | 10 00c002b3 |
| 11 00000000 | 11 00700313 |
| 12 00000000 | 12 00c28463 |
| 13 00000000 | 13 00730313 |
| 14 00000000 | 14 00c000ef |
| 15 00000000 | 15 00a00893 |
| 16 00000000 | 16 00000073 |
| 17 00000000 | 17 fffffd597 |
| 18 00000000 | 18 fd858593 |
| 19 00000000 | 19 00730313 |
| 20 00000000 | 20 0065a023 |
| 21 00000000 | 21 00000067 |
| 22 00000000 | |

(3) Rars 软件设计汇编程序,实现计算斐波那契—卢卡斯数列(数列前两项为 1, 2),并生成 COE 文件

代码如下:

```
.data
first:      .word      1
second:     .word      2
goal:       .word      20
outans:     .word      0

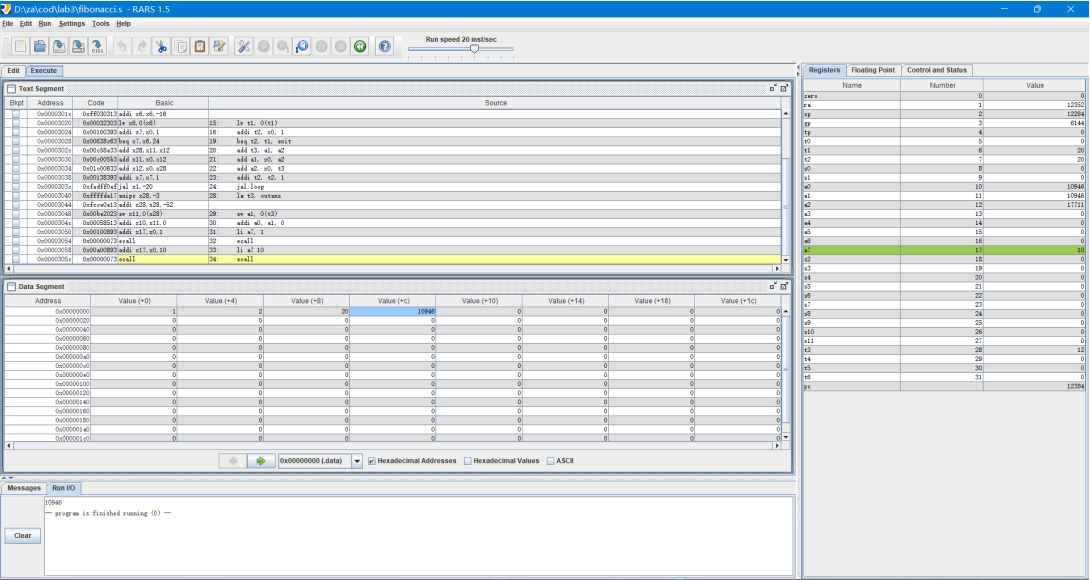
.text
main:
    la a1, first
    lw a1, 0(a1)
    la a2, second
    lw a2, 0(a2)
    la t1, goal
    lw t1, 0(t1)
    addi t2, x0, 1

loop:
    beq t2, t1, exit
    add t3, a1, a2
    add a1, x0, a2
    add a2, x0, t3
    addi t2, t2, 1
    jal loop

exit:
    la t3, outans
    sw a1, 0(t3)
    addi a0, a1, 0
    li a7, 1
    ecall
    li a7 10
    ecall
```

该程序通过 label:goal取得需要计算的第 goal 项元素,并将计算得到的结果,储存在outans处

测试goal=20 时, console 的输出:



计算可以发现第 20 项的值与该汇编程序计算出的答案相同 coe 文件:

```
text_fib.coe
1 memory_initialization_radix = 16;
2 memory_initialization_vector =
3 ffffd597
4 00058593
5 0005a583
6 ffffd617
7 ff860613
8 00062603
9 ffffd317
10 ff030313
11 00032303
12 00100393
13 00638c63
14 00c58e33
15 00c005b3
16 01c00633
17 00138393
18 fedff0ef
19 ffffd617
20 fcce0e13
21 00be2023
22 00058513
23 00100893
24 00000073
25 00a00893
26 00000073
27

data_fib.coe
1 memory_initialization_radix = 16;
2 memory_initialization_vector =
3 00000001
4 00000002
5 0000000a
6 00000059
7 00000000
8 00000000
9 00000000
10 00000000
11 00000000
12 00000000
13 00000000
14 00000000
15 00000000
16 00000000
17 00000000
18 00000000
19 00000000
20 00000000
21 00000000
22 00000000
23 00000000
24 00000000
25 00000000
26 00000000
27 00000000
28 00000000
29 00000000
30 00000000
31 00000000
32 00000000
33 00000000
34 00000000
35 00000000
```

4. 实验收获

- 学习了 RISC-V 汇编程序的书写
- 学习了仿真软件 Ripes 和 Rars 的使用
- 对单周期 CPU 有了更深的认识