

# Lab3 汇编程序设计

PB21020651 武宇星

## 实验目的

理解 RISC-V 常用 32 位整数指令功能

熟悉 RISC-V 汇编仿真软件 RARS, 掌握程序调试的基本方法

掌握 RISC-V 简单汇编程序设计, 以及存储器初始化文件(COE)的生成方法

理解 CPU 调试模块 PDU 的使用方法

## 实验环境

1.Vivado 2021.1

2.Microsoft Visual Studio Code

3.FPGAOL

## 实验过程

首先配置 java 环境添加到环境变量, 然后利用软件编写 FIB 的汇编实现:

```
.text
li t0, 15
li t1, 1
li t2, 1
li t3, 1
li t4, 0
loop:
sw t1, 0(t4)
addi t4, t4, 4
add t5, t1, t2
addi t1, t2, 0
addi t2, t5, 0
addi t3, t3, 1
bge t0, t3, loop
exit:
```

T4 记录写入地址的偏移量, t3 记录循环次数, t1t2t5 用来计算新的 FIB 一项

T0 存入 15 运行结果如下:

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x00000000	0x00000001	0x00000001	0x00000002	0x00000003	0x00000005	0x00000008	0x0000000d	0x00000015
0x00000020	0x00000022	0x00000037	0x00000059	0x00000090	0x000000a9	0x00000179	0x00000262	0x00000000

移位寄存器设计如下：

```
`timescale 1ns / 1ps

module Shift_reg(
    input rst,
    input clk,          // Work at 100MHz clock

    input [31:0] din,    // Data input
    input [3:0] hex,     // Hexadecimal code for the switches
    input add,           // Add signal
    input del,           // Delete signal
    input set,           // Set signal

    output reg [31:0] dout // Data output
);

always @(posedge clk or posedge rst) begin
    if (rst) begin
        dout <= 32'b0;
    end
    else if (set) begin
        dout <= din;
    end
    else if (add) begin
        dout[31:0] <= {dout[27:0], hex[3:0]};
    end
    else if (del) begin
        dout <= dout >> 4;
    end
end
endmodule
```

利用位拼接完成左移功能。