

Lab3

实验目的与内容

- 本次实验，我们使用了汇编语言设计了自己的斐波那契数列计算
- 掌握汇编实验的设计方法，学会导出相关的COE文件，方便后续的实验
- 完善移位寄存器，学会PDU调式方法

逻辑设计

计算斐波那契-卢卡斯数列

我们的设计思路是设计一个循环，每次循环的时候 `t0` 减一，那么当 `t0` 等于零的时候退出循环

```
EXIT:
li t4, 0
li t1, 1
sw t1, 0(t4)
addi t4, t4, 4
addi t0, t0, -1
li t2, 1
sw t2, 0(t4)
addi t4, t4, 4
addi t0, t0, -1

LOOP:
add t3, t2, t1
sw t3, 0(t4)
addi t4, t4, 4
addi t0, t0, -1
beq t0, x0, END

add t1, t2, t3
sw t1, 0(t4)
addi t4, t4, 4
addi t0, t0, -1
beq t0, x0, END

add t2, t3, t1
sw t2, 0(t4)
addi t4, t4, 4
addi t0, t0, -1
beq t0, x0, END
j LOOP
```

我们使用三个寄存器 `t1`, `t2`, `t3`，分别在一个循环里面实现 `t1=t2+t3`, `t3=t2+t1` 和 `t2=t1+t3`。

并且我们完成了附加实验的第一题，我们按照PPT上的流程图，循环读取内存中的值，并且通过合适的处理手段进行进位加减

```

# 读取n存储到t0中
li t0, 0
li t2, 0x0A
li a0, 0x7f00

LOOP1:
lw t0, 0(a0)
beq t0, x0, LOOP1

addi a0, a0, 4
lw t0, 0(a0)
addi t0, t0, -48
addi a0, a0, -4

LOOP2:
lw t1, 0(a0)
beq t1, x0, LOOP2

addi a0, a0, 4
lw t1, 0(a0)
beq t1, t2, EXIT

addi t1, t1, -48
slli t0, t0, 4
add t0, t0, t1
addi a0, a0, -4

LOOP3:
lw t1, 0(a0)
beq t1, x0, LOOP3

```

PDU

该模块非常简单，我们仅需要补全移位寄存器的模块即可

```

`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
);

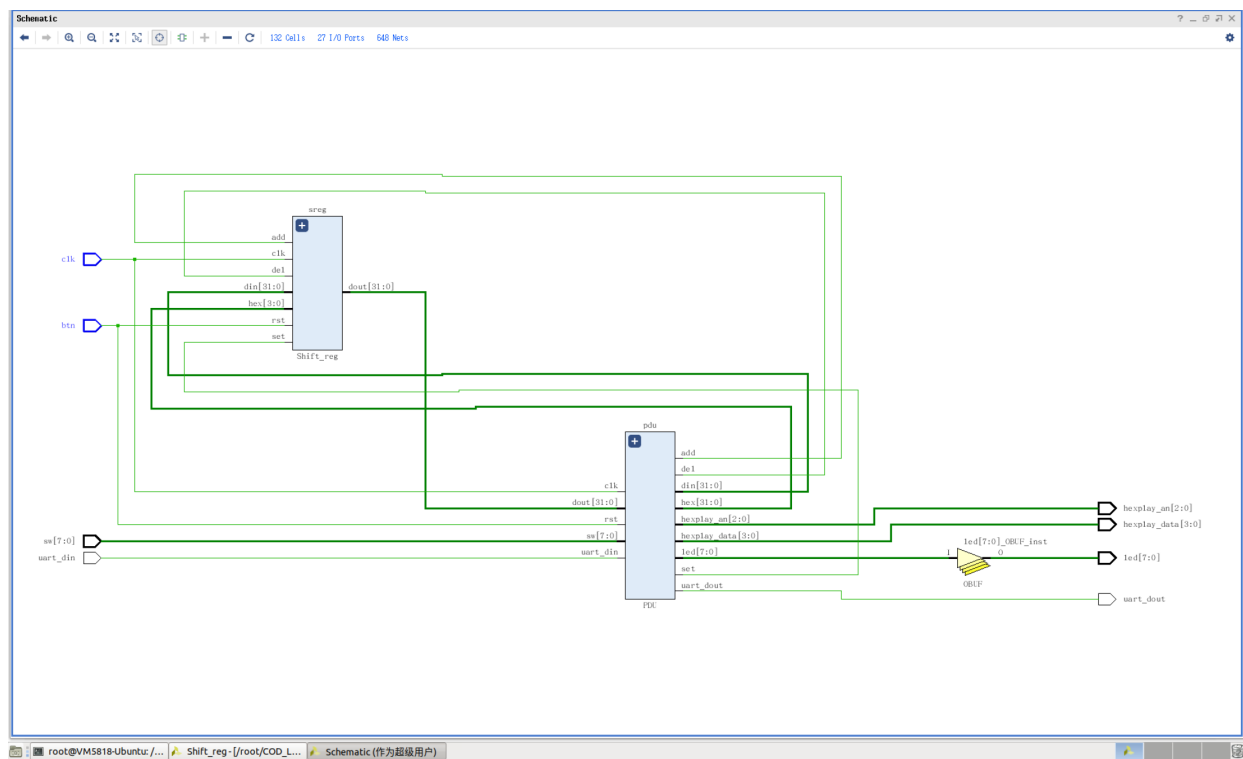
```

```

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

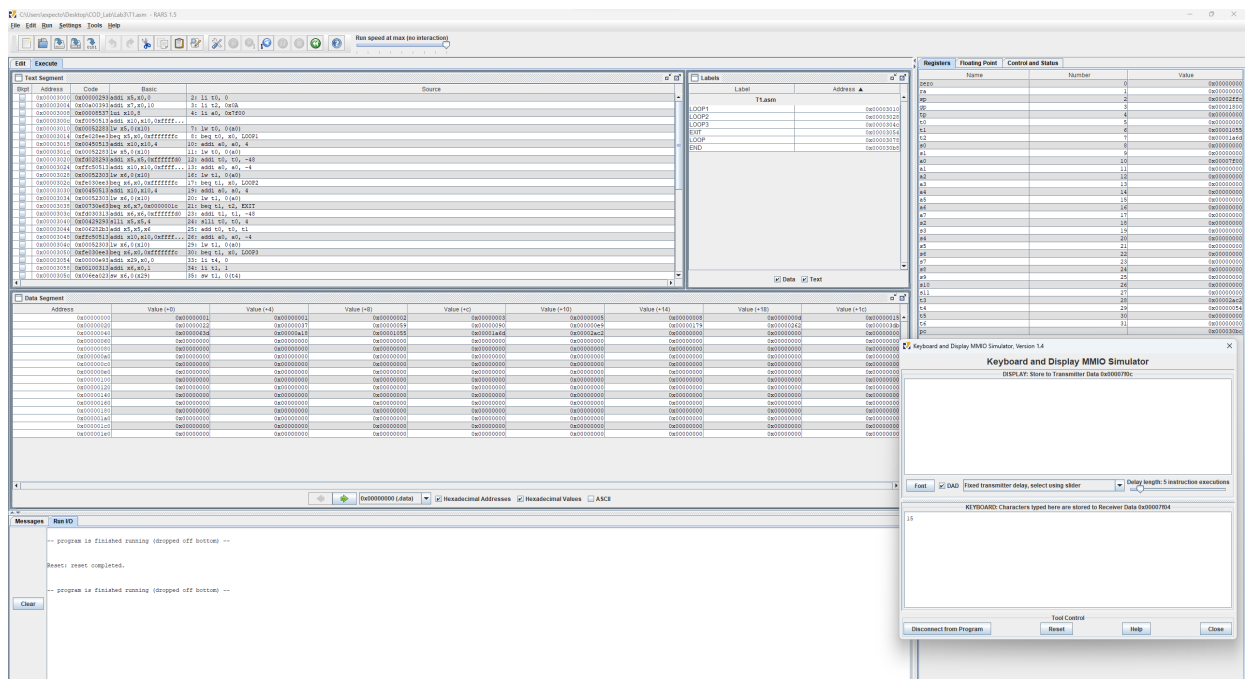
```

电路设计与分析



实验结果

斐波那契数列



可以从图中看出，我们输入了15_{HEX}，结果输出正确

PDU

该模块正确性已经在实验检查的时候向助教说明，在此就不再赘述

总结

- 本次实验收获很多，主要是学会了如何写汇编程序并且输出COE文件
- 本次实验对PDU模块也有一些初步的了解