

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：LabH4\_CPU 单周期设计

学生姓名：钟书锐

学生学号：PB19000362

完成日期：2021.5.13

计算机实验教学中心制

2020 年 09 月

## 【实验题目】

LabH4\_CPU 单周期设计

## 【实验目的】

- 理解 CPU 的结构和工作原理
- 掌握单周期 CPU 的设计和调试方法
- 熟练掌握数据通路和控制器的设计和描述方法

## 【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

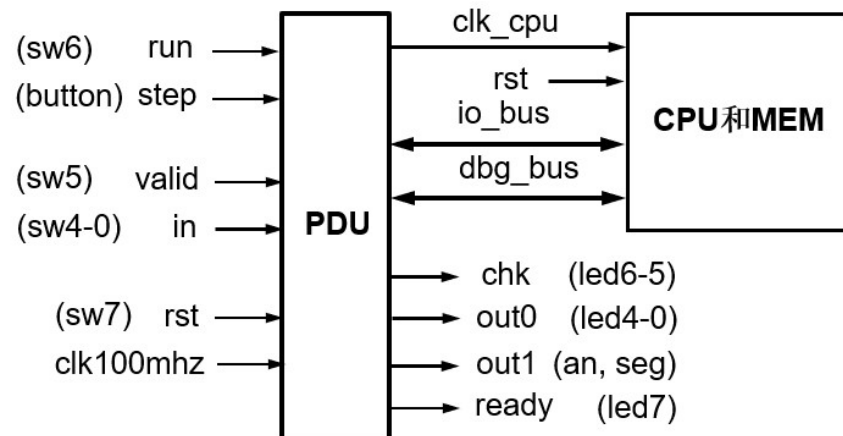
软件：

Vivado

## 【实验内容】

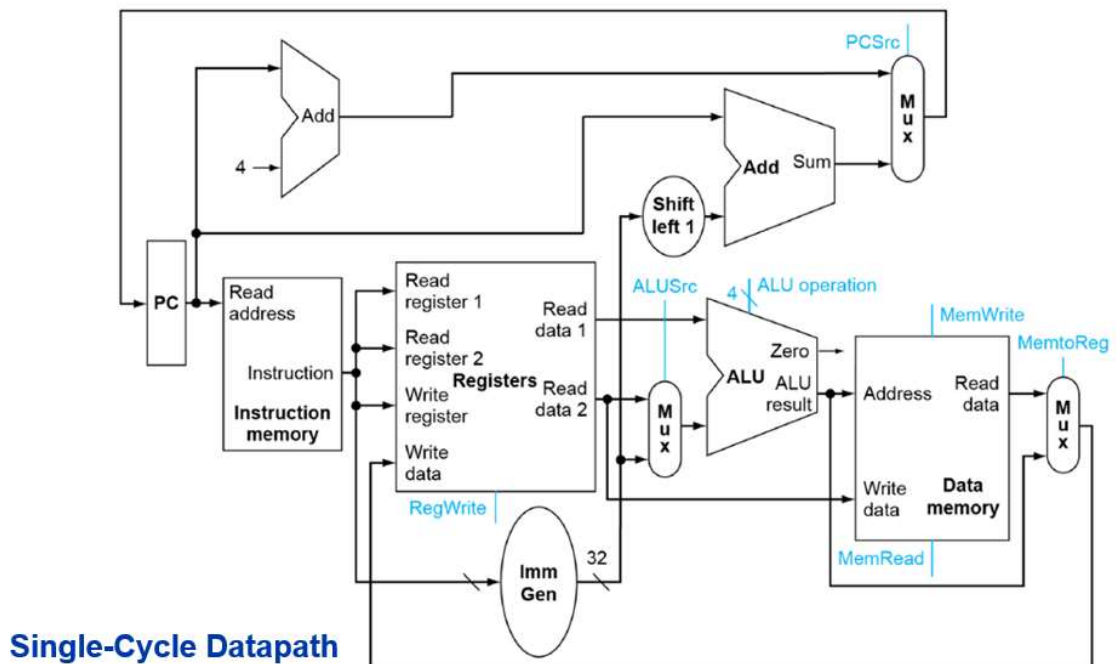
### 设计实现单周期 RISC-V CPU

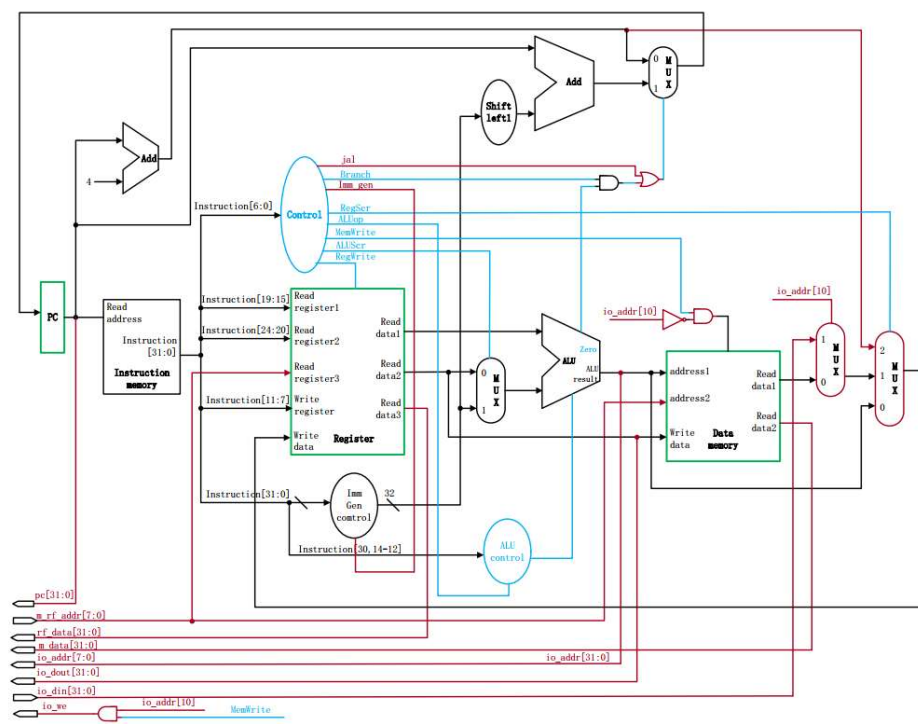
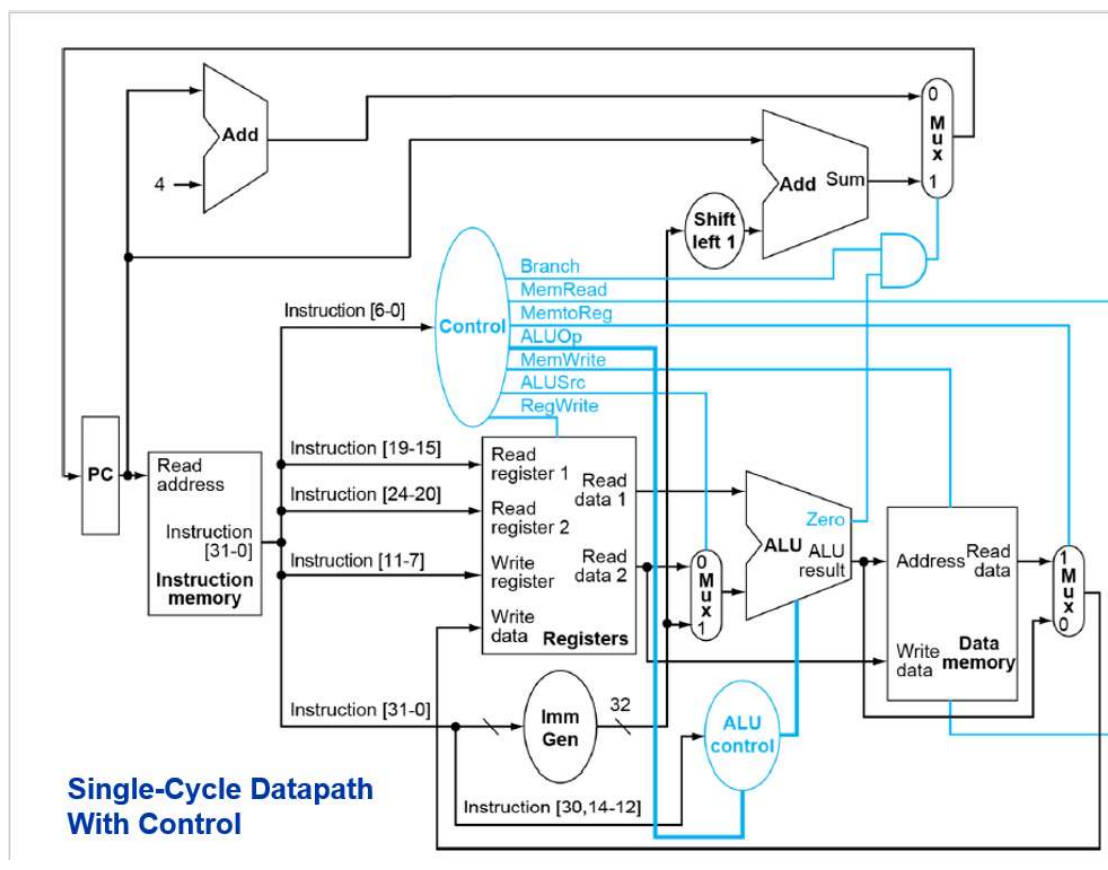
- 设计实现单周期RISC-V CPU，具有如下6条指令：
  - `add`, `addi`, `lw`, `sw`, `beq`, `jal`



### 1.数据通路

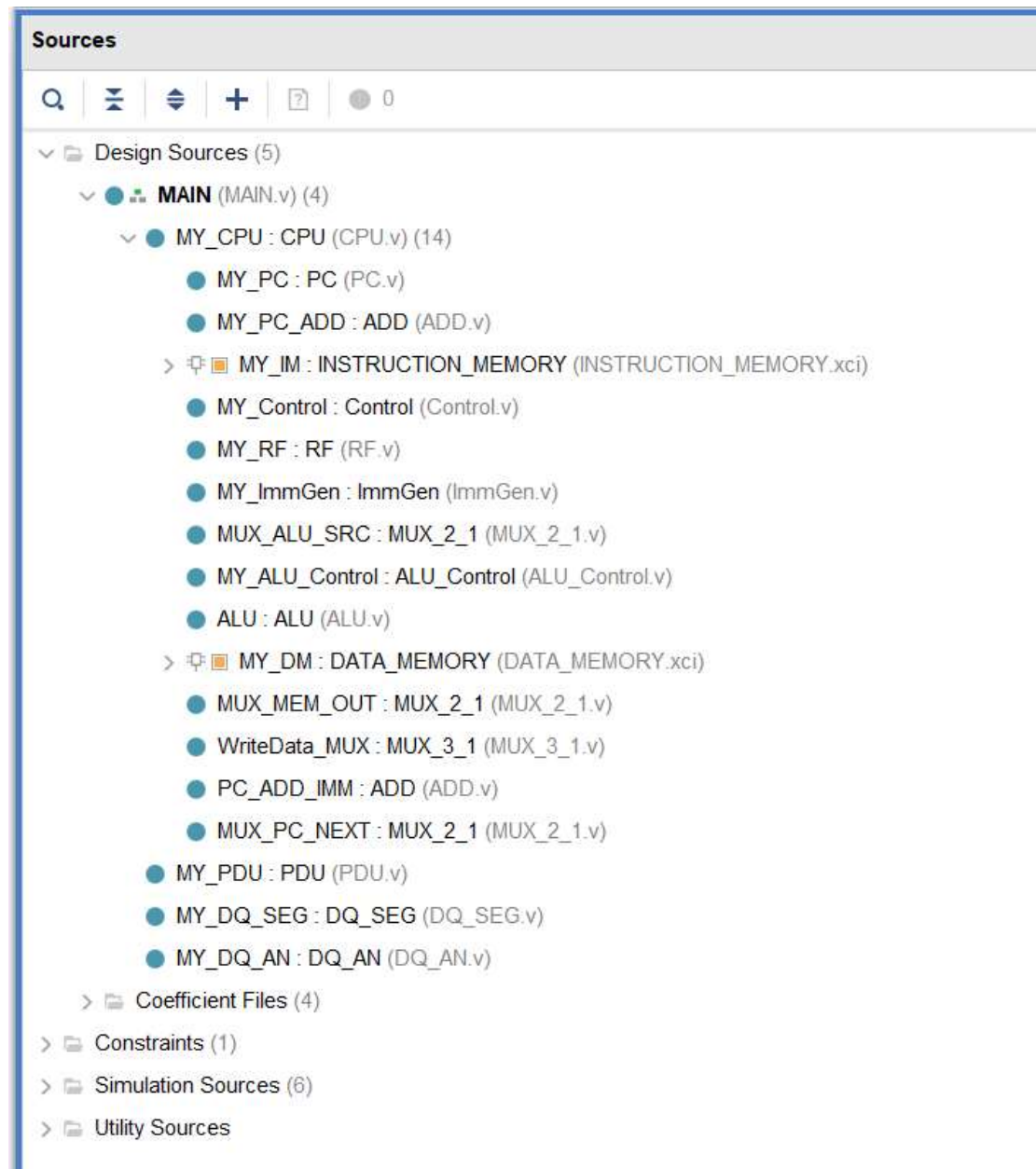
## 单周期CPU数据通路



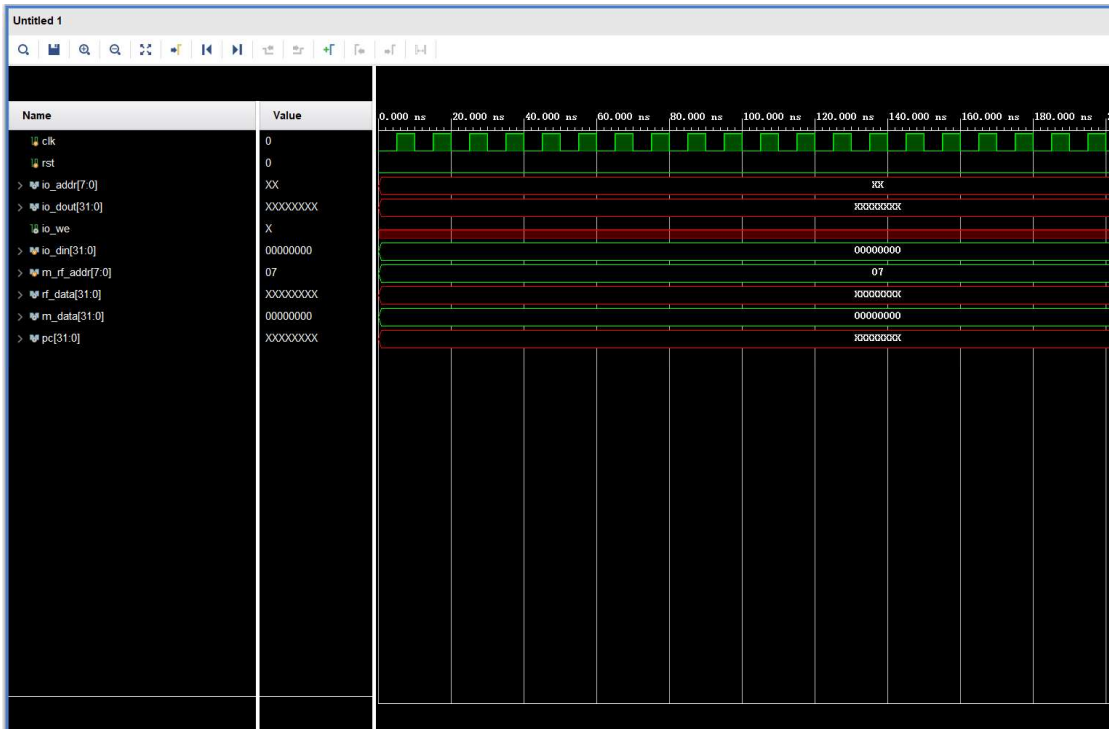


## 2.设计逻辑

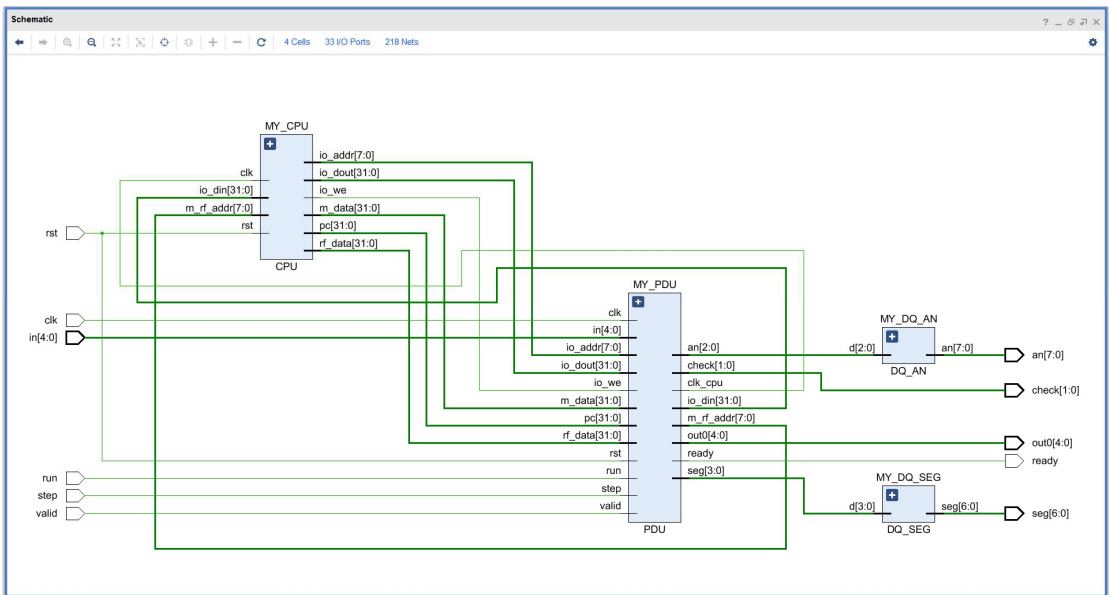
- PDU (Processor Debug Unit)负责处理处理器调试
- CPU 负责运算
- MAIN 负责调用 PDU 模块与 CPU 模块



### 3.仿真测试



### 4.RTL 电路



## 5.核心代码

```
1 module MAIN(  
2     input clk,  
3     input rst,  
4  
5     //选择CPU工作方式;  
6     input run,  
7     input step,  
8     input valid,  
9     input [4:0] in,  
10  
11     //输出led和seg  
12     output [1:0] check,           //led6-5:查看类型  
13     output [4:0] out0,           //led4-0  
14     output [7:0] an,             //8个数码管  
15     output [6:0] seg,            //FPGA数码管显示,ps:此2项在FPGA online有区别  
16     output ready                 //led7  
17 );
```

```
rsity > usc > data > ComputerScience > ComputerDesign > lab > ComputerDesignLab > lab4 > sources > CPU.v  
module CPU(  
    input clk,  
    input rst,  
  
    //IO_BUS  
    output [7:0] io_addr,           //led和seg的地址  
    output [31:0] io_dout,         //输出led和seg的数据  
    output io_we,                  //输出led和seg数据时的使能信号  
    input [31:0] io_din,           //来自sw的输入数据  
  
    //Debug_BUS  
    input [7:0] m_rf_addr,         //存储器(MEM)或寄存器堆(RF)的调试读口地址  
    output [31:0] rf_data,         //从RF读取的数据  
    output [31:0] m_data,         //从MEM读取的数据  
    output [31:0] pc               //PC的内容  
);
```

```
1 module PC(  
2     input clk,rst,  
3     input [31:0]Next_PC,  
4     output reg [31:0]PC  
5 );  
6 always @(posedge clk, posedge rst)  
7 begin  
8     if(rst)  
9         PC <= 0;  
10    else  
11        PC <= Next_PC;  
12    end  
13  
14 endmodule
```

```

module Control(
    input  wire [6:0] opcode,
    output reg jal,
    output reg Branch,
    output reg [1:0] Imm_gen,
    output reg [1:0] RegScr,
    output reg [1:0] ALUop,
    output reg MemWrite,
    output reg ALUSrc,
    output reg RegWrite
);

```

```

module ALU_Control(
    input [1:0] ALUop,
    input [6:0] func7,
    input [2:0] func3,
    output reg [2:0] ALU_ctl_input
);
    always @(*)
    begin
        case (ALUop)
            2'b01: ALU_ctl_input=3'd1;
            default: ALU_ctl_input=3'd0;
        endcase
    end
endmodule

```

```

module ImmGen (
    input [1:0] Imm_gen,
    input [31:0] instruct,
    output reg [31:0] Imm
);
    always @(*) begin
        case(Imm_gen)
            2'b00: //addi lw
            begin
                Imm[11:0] = instruct[31:20];
                if(instruct[31] == 1)
                    Imm[31:12] = 20'hffffff;
                else Imm[31:12] = 20'h000000;
            end

            2'b01: //sw
            begin
                Imm[11:5] = instruct[31:25];
                Imm[4:0] = instruct[11:7];
                if(instruct[31] == 1)
                    Imm[31:12] = 20'hffffff;
                else Imm[31:12] = 20'h000000;
            end
        endcase
    end
endmodule

```



6.资源分析

Hierarchy

Summary

▼ Slice Logic

▼ Slice LUTs (1%)

▼ LUT as Memory (2%)

LUT as Distributed

LUT as Logic (1%)

F8 Muxes (<1%)

utilization\_1

Hierarchy

Name	Slice LUTs (63400)	Slice Registers (126800)	F7 Muxes (31700)	F8 Muxes (15850)	Slice (15850)	LUT as Logic (63400)	LUT as Memory (19000)	Bonded IOB (210)	BUFGCTRL (32)
MAIN	707	108	196	32	214	251	456	33	2
MY_CPU (CPU)	694	32	196	32	201	238	456	0	0
MY_PDU (PDU)	13	76	0	0	32	13	0	0	0

7.时间性能分析

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 7.566 ns	Worst Hold Slack (WHS): 0.191 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 25	Total Number of Endpoints: 25	Total Number of Endpoints: 77

All user specified timing constraints are met.

8.结果分析

在 FPGA 开发板上能成功运行 fib 的 coe 文件读取输入，产生输出。

【实验总结】

深入了解 riscv cpu 单周期设计