COD LAB6 实验报告

姓名：刘芷辰

学号：PB21111728

日期：2023.6.14

1. 实验题目

流水线CPU指令集拓展

2. 实验目标

- 理解流水线CPU的结构和工作原理
- 熟练掌握流水线CPU数据通路和控制器的设计和描述方法
- 理解流水线CPU的调试方法
- 了解riscv指令集各指令的执行方式

3. 实验内容

3.1 指令集拓展（主要修改部分）

> R型

- 代码

```
7'b0110011: begin
            jal = 0;
            jalr = 0;
            br_type = 0;
            rf_we = 1;
            rf_wd_sel = 2'b00;
            alu_src1_sel = 0;
            alu_src2_sel = 0;
            case(inst[14:12])
                3'b000: begin
                    if(inst[30] == 0) alu_func = 0; //add
```

```
                    else alu_func = 1; //sub
                        end
                3'b111: alu_func = 5; //and
                3'b110: alu_func = 6; //or
                3'b100: alu_func = 7; //xor
                3'b010: alu_func = 4; //slt
                3'b011: alu_func = 3; //sltu
                3'b001: alu_func = 9; //sll
                3'b101: begin
                    if(inst[30] == 0) alu_func = 8; //srl
                    else alu_func = 10; //sra
                        end
                default: alu_func = 15;
                endcase
            mem_we = 0;
            imm_type = 3'b000;
            if(inst[19:15] != 5'h0) rf_re0 = 1'b1;
            else rf_re0 = 1'b0;
            if(inst[24:20] != 5'h0) rf_re1 = 1'b1;
            else rf_re1 = 1'b0;
        end

```

- 主要区别是根据inst[14:12]选择不同的alu_func

> I型

- 代码

```
7'b0010011: begin
            jal = 0;
            jalr = 0;
            br_type = 0;
            rf_we = 1;
            rf_wd_sel = 2'b00;
            alu_src1_sel = 0;
            alu_src2_sel = 1;
            case(inst[14:12])
                3'b000: alu_func = 0; //addi
                3'b111: alu_func = 5; //andi
                3'b110: alu_func = 6; //ori
                3'b100: alu_func = 7; //xori
```

```
                3'b010: alu_func = 4; //slti
                3'b011: alu_func = 3; //sltiu
                3'b001: alu_func = 9; //slli
                3'b101: begin
                        if(inst[30] == 0) alu_func = 8; //srli
                        else alu_func = 10; //srai
                        end
                default: alu_func = 15;
            endcase
            mem_we = 0;
            if((inst[14:12] == 3'b101) & (inst[31:25] == 7'b0100000))
                imm_type = 3'b110;
            else
                imm_type = 3'b001;
            if(inst[19:15] != 5'h0) rf_re0 = 1'b1;
            else rf_re0 = 1'b0;
            rf_re1 = 1'b0;
        end

```

- 主要区别是根据inst[14:12]选择不同的alu_func，并且对于imm_type，srai需要特殊处理为3‘b110

> B型

- 代码

```
7'b1100011: begin
            jal = 0;
            jalr = 0;
            case(inst[14:12])
                3'b000: br_type = 2; //beq
                3'b100: br_type = 1; //blt
                3'b001: br_type = 3; //bne
                3'b101: br_type = 4; //bge
                3'b110: br_type = 5; //bltu
                3'b111: br_type = 6; //bgeu
                default:br_type = 0;
            endcase
            rf_we = 0;
            rf_wd_sel = 2'b11;
            alu_src1_sel = 1;
            alu_src2_sel = 1;
            alu_func = 0;
```

```
            mem_we = 0;
            imm_type = 3'b010;
            if(inst[19:15] != 5'h0) rf_re0 = 1'b1;
            else rf_re0 = 1'b0;
            if(inst[24:20] != 5'h0) rf_re1 = 1'b1;
            else rf_re1 = 1'b0;
        end

```

```
module Branch(
    input [31:0]op1,
    input [31:0]op2,
    input [2:0]br_type,
    output reg br
);

    always@(*)
    begin
        case(br_type)
            3'b001: begin           //blt
                if($signed(op1) < $signed(op2))
                    br = 1;
                else
                    br = 0;
            end
            3'b010: begin           //beq
                if(op1 == op2)
                    br = 1;
                else
                    br = 0;
            end
            3'b011: begin           //bne
                if(op1 != op2)
                    br = 1;
                else
                    br = 0;
            end
            3'b100: begin       //bge
                if($signed(op1) >= $signed(op2))
                    br = 1;
                else
                    br = 0;
            end
            3'b101: begin       //bltu
                if(op1 < op2)

```

```
                    br = 1;
                else
                    br = 0;
            end
            3'b110: begin
                if(op1 >= op2)//bgeu
                    br = 1;
                else
                    br = 0;
            end
            default: br = 0;
        endcase
    end

endmodule
```

- 主要区别是根据inst[14:12]选择不同的br_type,然后在branch模块中根据br_type进行不同的比较来确定br，需要注意的是由于br_type种类变多，在lab5中的两位应该改为3位，并同时修改CPU模块中相应的位宽

3.2 实验结果

> test

- 烧写

  已在线下检查

4. 总结

更加了解了计算机体系结构，对riscv指令集的架构理解更加深入