实验名称：石头剪刀布 姓名： 学号：

装

订

线

浙江大学

本科实验报告

姓名：

学院：生物医学工程与仪器科学学院

系：生物医学工程系

专业：生物医学工程

学号：

指导教师：余锋

实验名称：石头剪刀布 姓名： 学号：

专业：生物医学工程
姓名：
学号：
日期：2025.3.18
地点：教 6－204

浙江大学 实验报告

课程名称：硬件描述语言 指导老师：余锋 实验名称：HDL 实验二

一、实验要求

二、实验代码与注释

三、仿真结果与分析

四、讨论与心得

装 订 线

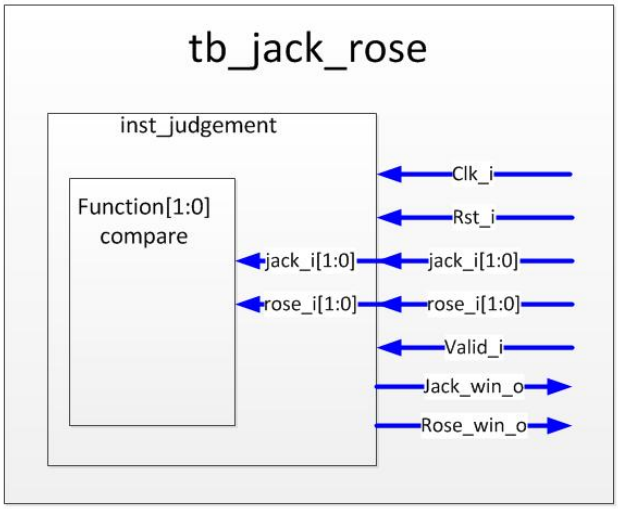
一、实验要求

模拟实现剪刀石头布的 HDL 语言设计。

提供输入：clk_i 时钟信号，100MHz
rst_i 复位信号，高电平复位
jack_i 表示 jack 的输入状态
rose_i 表示 rose 的输入状态
valid_i 使能信号，表示输入输出有效；

要求输出：jack_win_o 当 win_valid_o 有效时，表示 jack 获胜
rose_win_o 当 win_valid_o 有效时，表示 rose 获胜
win_valid_o 判决结果有效

实验名称：___石头剪刀布___ 姓名：__ 学号：



1. Function 本身作为返回值，不需要 output 接口；

2. 胜、负、平与其他，至少三种状态，function 返回至少为两位[1:0]；

3. 输入至少有剪刀、石头、布三种手势，还可能有其他非法手势，至少有 3-4 种状态，故输入为两位[1:0]；

4. 剪刀、石头、布的参数值分别为 2’b10，2’b00，2’b01，逻辑关系与数值大小不相关；

5. 输入组合可能有 3*3 共 9 种，或者 4*4 共 16 种（考虑非法手势），考虑到其中手势相同的情况可归为 1 种，非法手势可归为一种，可简化为 8 种情况，再将非法手势与手势相同归为一类列入 else 部分，可简化为 7 种情况；

6. 要求输入不同的状态（三种或者四种），输出胜负结果。（只输出每一局的胜负结果即可，如果感兴趣可以拓展设计输出三局两胜的结果，不作要求）。

二、实验代码与注释

信号名	方向	位宽	说明
clk_i	输入	1	100MHz 时钟信号
rst_i	输入	1	高电平复位信号

实验名称：___石头剪刀布___ 姓名：__ 学号：

jack_i	输入	2	Jack 的手势输入（2'b00-10）
rose_i	输入	2	Rose 的手势输入（2'b00-10）
valid_i	输入	1	输入使能信号
jack_win_o	输出	1	Jack 获胜指示（valid 有效时）
rose_win_o	输出	1	Rose 获胜指示（valid 有效时）
win_valid_o	输出	1	判决结果有效信号

装

订

线

1、判决模块

```
// judgement.v
module judgement(
input clk_i,
input rst_i,
input [1:0] jack_i,
input [1:0] rose_i,
input valid_i,
output reg jack_win_o,
output reg rose_win_o,
output reg win_valid_o
);

reg [1:0] compare_result;
```

实验名称：___石头剪刀布___ 姓名：__ 学号：

装订线

```
parameter ROCK = 2'b00,
          PAPER = 2'b01,
          SCISSORS = 2'b10;
function [1:0] compare(
input [1:0] a,
input [1:0] b
);
parameter WIN_A = 2'b01;
parameter WIN_B = 2'b10;
parameter DRAW = 2'b00;
parameter INVALID = 2'b11;
if(a == b)
    compare = DRAW;
else if(a > 2'b10 || b > 2'b10)
    compare = INVALID;
else
case({a, b})
    {ROCK, SCISSORS}: compare = WIN_A;
    {SCISSORS, PAPER}: compare = WIN_A;
    {PAPER, ROCK}: compare = WIN_A;
    {SCISSORS, ROCK}: compare = WIN_B;
    {PAPER, SCISSORS}: compare = WIN_B;
    {ROCK, PAPER}: compare = WIN_B;
    default: compare = INVALID;
endcase
endfunction
always @(posedge clk_i or posedge rst_i) begin
if(rst_i) begin
    compare_result <= 2'b00;
    jack_win_o <= 1'b0;
    rose_win_o <= 1'b0;
    win_valid_o <= 1'b0;
end
else begin
    if(valid_i) begin
        compare_result <= compare(jack_i, rose_i);
        win_valid_o <= 1'b1;
        case(compare(jack_i, rose_i))
            2'b01: begin
                jack_win_o <= 1'b1;
                rose_win_o <= 1'b0;
            end
            2'b10: begin
                jack_win_o <= 1'b0;
```

实验名称：___石头剪刀布___ 姓名：__ 学号：

```
                rose_win_o <= 1'b1;
            end
            default: begin
                jack_win_o <= 1'b0;
                rose_win_o <= 1'b0;
            end
        endcase
    end
    else begin
        win_valid_o <= 1'b0;
    end
end
end
Endmodule
```

此处我们遵循实验要求，定义一个名为 compare 的函数，它接收两个 2 位输入 a 和 b，返回一个 2 位的比较结果。函数内部定义参数：

- WIN_A = 2'b01: 表示 a 获胜
- WIN_B = 2'b10: 表示 b 获胜
- DRAW = 2'b00: 表示平局
- INVALID = 2'b11: 表示非法输入

如果 a 和 b 相等，返回 DRAW（平局）；

如果 a 或 b 大于 2'b10（即非法手势 2’ b11），返回 INVALID（非法输入）。

对于其他情况，使用 case 语句根据不同的手势组合判断胜负：

- 当{a, b}为{ROCK, SCISSORS}、{SCISSORS, PAPER}、{PAPER, ROCK}时，返回 WIN_A（a 获胜）
- 当{a, b}为{SCISSORS, ROCK}、{PAPER, SCISSORS}、{ROCK, PAPER}时，返回 WIN_B（b 获胜）
- 其他情况返回 INVALID

装 订 线

实验名称：___石头剪刀布___ 姓名：__ 学号：

在主程序部分，使用 always 块，在时钟上升沿或复位信号上升沿触发。

如果复位信号 rst_i 为高电平，则将 compare_result 置为 2'b0，并将 jack_win_o、rose_win_o、win_valid_o 都置为 0。

在复位信号为低电平的前提下，如果使能信号 valid_i 为高电平，就调用 compare 函数，将比较结果存储到 compare_result 中并将 win_valid_o 置为 1 表示判决输出有效。

然后根据 compare 函数的返回结果，更新 jack_win_o 和 rose_win_o:

如果返回 2'b01（Jack 获胜），将 jack_win_o 置为 1，rose_win_o 置为 0;

如果返回 2'b10（Rose 获胜），将 jack_win_o 置为 0，rose_win_o 置为 1;

其他情况（平局或非法输入），将 jack_win_o 和 rose_win_o 都置为 0。

如果使能信号 valid_i 为低电平，就将 win_valid_o 置为 0。

装

订

线

2、测试模块

```
// judgement_tb.v
`timescale 1ns / 1ps

module judgement_tb;
reg clk;
reg rst;
reg [1:0] jack;
reg [1:0] rose;
reg valid;
wire jack_win;
wire rose_win;
wire win_valid;
parameter ROCK = 2'b00,
          PAPER = 2'b01,
          SCISSORS = 2'b10;
judgement u_judgement(
    .clk_i(clk),
```

实验名称：___石头剪刀布___ 姓名：__ 学号：

装

订

线

```
    .rst_i(rst),
    .jack_i(jack),
    .rose_i(rose),
    .valid_i(valid),
    .jack_win_o(jack_win),
    .rose_win_o(rose_win),
    .win_valid_o(win_valid)
);
initial begin
    clk = 0;
    forever #5 clk = ~clk;
end
task initialize;
begin
    rst = 1;
    jack = 2'b00;
    rose = 2'b00;
    valid = 0;
    #20;
    rst = 0;
end
endtask
task test_case(input [1:0] j, input [1:0] r);
begin
    jack = j;
    rose = r;
    valid = 1;
    #10;
    valid = 0;
    #10;
end
endtask
initial begin
    $dumpfile("./build/judgement_tb.vcd");
    $dumpvars(0, judgement_tb);

    initialize;

    test_case(ROCK, ROCK);
    test_case(PAPER, PAPER);
    test_case(SCISSORS, SCISSORS);

    test_case(ROCK, SCISSORS);
    test_case(SCISSORS, PAPER);
```

实验名称：___石头剪刀布___ 姓名：__ 学号：

```
    test_case(PAPER, ROCK);

    test_case(SCISSORS, ROCK);
    test_case(PAPER, SCISSORS);
    test_case(ROCK, PAPER);

    test_case(2'b11, ROCK);
    test_case(ROCK, 2'b11);

    #50;
    $finish;
end
initial begin
    $monitor("Time=%0t: jack=%b, rose=%b, valid=%b => jack_win=%b,
rose_win=%b, win_valid=%b",
             $time, jack, rose, valid, jack_win, rose_win, win_valid);
end
endmodule
```

装订线

初始化定义完成之后，我们使用 initial 块执行测试流程：

- 打开波形文件 judgement_tb.vcd，用于记录仿真波形
- 记录 judgement_tb 模块的所有变量到波形文件中
- 复位操作：将 rst 置为 1，保持 20ns 后将 rst 置为 0
- 调用 test_case 任务进行不同场景的测试：
    - 测试平局场景：分别传入(ROCK, ROCK)、(PAPER, PAPER)、(SCISSORS, SCISSORS)
    - 测试 Jack 获胜场景：分别传入(ROCK, SCISSORS)、(SCISSORS, PAPER)、(PAPER, ROCK)
    - 测试 Rose 获胜场景：分别传入(SCISSORS, ROCK)、(PAPER, SCISSORS)、(ROCK, PAPER)
    - 测试非法输入场景：分别传入(2'b11, ROCK)、(ROCK, 2'b11)
- 结束仿真

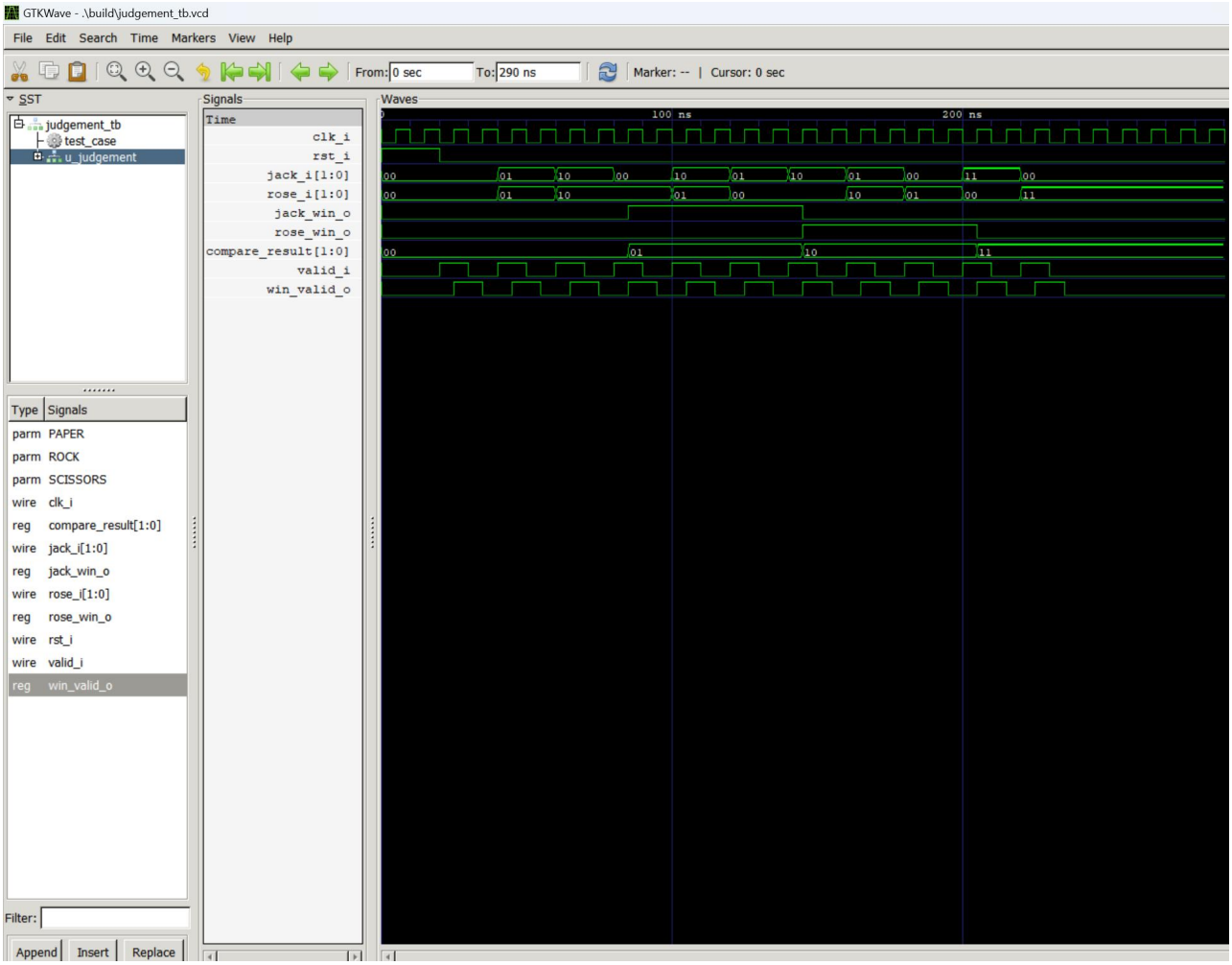
test_case 的任务具体接收两个 2 位输入 j 和 r，将 j 赋值给 jack，r 赋值给 rose。最后将 valid 置为 1，保持 10ns，以示输入输出有效。然后将 valid 置为 0，保持 10ns。

最后，使用$monitor 系统任务，监控并打印当前时间、Jack 的输入、Rose 的输入、使能信号、判决结果有效信号、Jack 获胜信号和 Rose 获胜信号。

实验名称：石头剪刀布 姓名：__ 学号：

三、仿真结果与分析

本次实验选择采用 iverilog + gtkwave 进行仿真波形分析。



测试场景	Jack 输入	Rose 输入	胜负结果	jack_win_o, rose_win_o, win_valid_o
平局-石头	ROCK (2'b00)	ROCK (2'b00)	平局	0, 0, 1
平局-布	PAPER (2'b01)	PAPER (2'b01)	平局	0, 0, 1

装 订 线

实验名称：___石头剪刀布___ 姓名：__ 学号：

平局-剪刀	SCISSORS (2'b10)	SCISSORS (2'b10)	平局	0, 0, 1
Jack 胜-石头胜剪刀	ROCK (2'b00)	SCISSORS (2'b10)	Jack 胜	1, 0, 1
Jack 胜-剪刀胜布	SCISSORS (2'b10)	PAPER (2'b01)	Jack 胜	1, 0, 1
Jack 胜-布胜石头	PAPER (2'b01)	ROCK (2'b00)	Jack 胜	1, 0, 1
Rose 胜-石头胜剪刀	SCISSORS (2'b10)	ROCK (2'b00)	Rose 胜	0, 1, 1
Rose 胜-剪刀胜布	PAPER (2'b01)	SCISSORS (2'b10)	Rose 胜	0, 1, 1
Rose 胜-布胜石头	ROCK (2'b00)	PAPER (2'b01)	Rose 胜	0, 1, 1
无效输入-高位非法	2'b11	ROCK (2'b00)	无效	0, 0, 0
无效输入-低位非法	ROCK (2'b00)	2'b11	无效	0, 0, 0

装 订 线

四、讨论与心得

在 HDL 设计中，时序逻辑（如时钟驱动的寄存器更新）与组合逻辑（如函数计算）的协同至关重要。本次实验中，compare 函数作为组合逻辑完成胜负判断，而 win_valid_o 和胜负信号的更新需严格遵循时钟同步（时序逻辑）。实验中需考虑所有可能的输入组合（包括非法手势）。通过以下策略提升鲁棒性：

1、**参数化设计**：将合法手势（ROCK/PAPER/SCISSORS）定义为参数，增强代码可读性；

实验名称：___石头剪刀布___ 姓名：__ 学号：

2、**默认处理**：非法输入统一归类为 INVALID 状态，并在输出时置 win_valid_o=1 但胜负信号为 0。

通过遍历所有合法与非法组合的测试用例（如 test_case(ROCK, 2'b11)），我验证了设计的完备性，且仿真波形直观反映了时序逻辑的正确性。经过本次实验，我不仅掌握了 HDL 模块化设计的基本方法，更深刻体会到硬件设计中“明确时序”和“覆盖边界”的重要性。这些经验为后续复杂数字系统的设计奠定了坚实基础。