

《数字逻辑与数字系统》实验报告

学院 智能与计算学部 年级 2022 级 班级 4 班

姓名 陆子毅 学号 3022206045

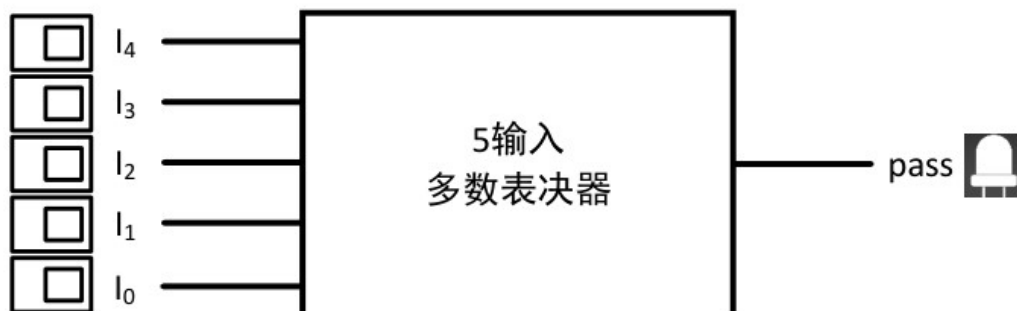
实验项目名称 多数表决器的设计与实现

一. 实验目的

1. 掌握基于 Vivado 的数字逻辑电路设计流程；
2. 熟练使用 SystemVerilog HDL 的行为建模方法对组合逻辑电路进行描述；
3. 熟练使用 SystemVerilog HDL 的结构建模方法对组合逻辑电路进行描述；
4. 掌握基于远程 FPGA 硬件云平台对数字逻辑电路进行功能验证的流程。

二. 实验内容

假如有五个举重裁判，举重选手完成比赛以后，当有多数裁判认定成功时，则成功；否则失败。本次实验请设计此举重裁决电路，即一个 5 输入的多数表决器。该电路的顶层模块如图 1-3 所示，输入/输出端口如表 1-3 所示。使用拨动开关来模拟裁判的裁定，使用 LED 灯来显示是否成功。



三. 实验原理与步骤（注：不用写工具的操作步骤，而是设计步骤）

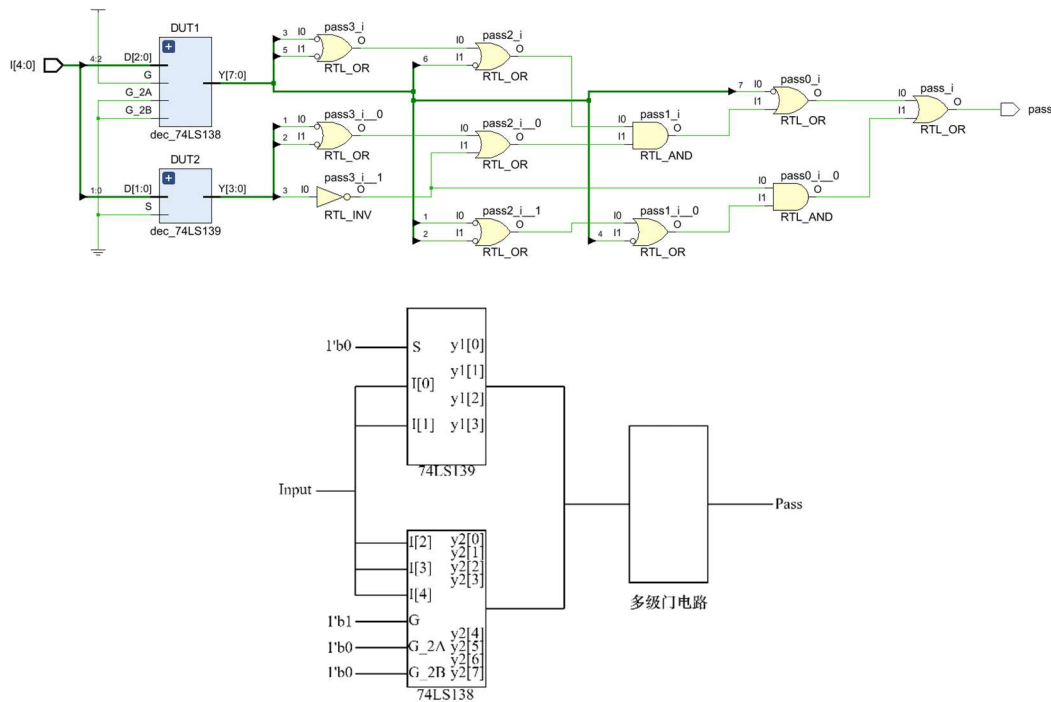
1. 写出 74LS138 和 74LS139 的行为建模的 SystemVerilog HDL 代码。

```
module dec_74LS138(  
    input G,  
    input G_2A,  
    input G_2B,  
    input [2:0] D,  
    output [7:0] Y  
);  
reg [7:0] y;  
always @* begin  
    if (G && ~G_2A && ~G_2B)  
        case (D)  
            3'b000: y = 8'b11111110;  
            3'b001: y = 8'b11111101;  
            3'b010: y = 8'b11111011;  
            3'b011: y = 8'b11110111;  
            3'b100: y = 8'b11101111;  
            3'b101: y = 8'b11011111;  
            3'b110: y = 8'b10111111;  
            3'b111: y = 8'b01111111;  
        endcase  
    end  
    assign Y = y;  
endmodule  
  
module dec_74LS139(  
    input S,  
    input [1:0] D,  
    output [3:0] Y  
);  
reg [3:0] y;  
always @* begin  
    if (~S)  
        case (D)  
            2'b00: y = 4'b1110;  
            2'b01: y = 4'b1101;  
            2'b10: y = 4'b1011;  
            2'b11: y = 4'b0111;  
        endcase  
    end  
    assign Y = y;  
endmodule
```

使用 MUX 多路选择器来实现两个模块。每种情况都对应一个最小项。

2. 给出基于 74LS138 和 74LS139 的 5 输入多数表决器的设计方案，画出原理图（采用 Visio 或其他工具画图）。

设计思路是分别用一个 74LS138 模块和一个 74LS139 模块来表示 0 号 1 号投票的具体情况 and 2 号 3 号 4 号投票的具体情况，最后将这些情况通过基础的门电路排列组合到一起，实现只要有三个人投票了灯就亮。



3. 写出 5 输入多数表决器的结构化建模的 SystemVerilog HDL 代码。

```

module voter5(
    input [4:0] I,
    output light,
);
    logic [7:0] y1;
    logic [3:0] y2;

    dec_74LS138 DUT1(.G(1'b1),.G_2A(1'b0),.G_2B(1'b0),.D(I[4:2]),.Y(y1));
    dec_74LS139 DUT2(.S(1'b0),.D(I[1:0]),.Y(y2));

    assign light=~y1[7]|(~y1[3]|~y1[5]|~y1[6])&(~y2[1]|~y2[2]|~y2[3])|~y2[3]&(~y1[1]|~y1[2]|~y1[4]);

endmodule

```

4. 给出基于行为建模的 5 输入多数表决的 SystemVerilog HDL 代码。

用循环遍历一遍所有投票人的选择，如果是 1，那么计数加一，最后将计数与 3 作比较，大于等于 3 就视为通过。

也可以使用多路选择器（MUX）来构建，不过至少需要写 32 行，代表投票的不同情况。

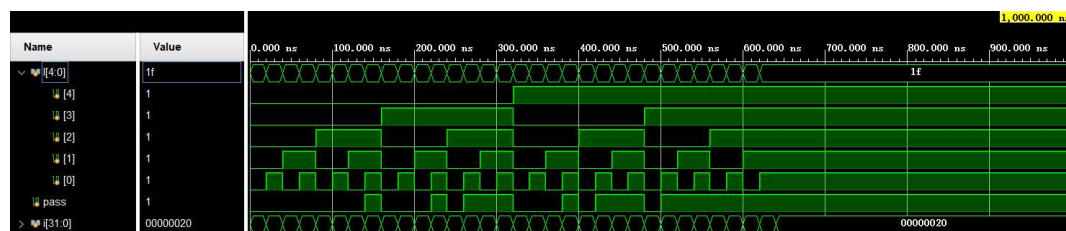
```
module voter5(
    input [4:0] I,
    output pass
);
    reg reg_pass;
    always @(*) begin
        // Count the number of ones in the input
        integer count = 0;
        for (int i = 0; i < 5; i = i + 1) begin
            if (I[i] == 1) begin
                count = count + 1;
            end
        end

        // Output high if majority of inputs are high
        if (count >= 3) begin
            reg_pass = 1;
        end else begin
            reg_pass = 0;
        end
    end

    assign pass=reg_pass;
endmodule
```

四. 仿真与实验结果（注：仿真需要给出波形图截图，截图要清晰，如果波形过长，可以分段截取；实验结果为远程 FPGA 硬件云平台的截图）

注：远程 FPGA 硬件云平台截图只需要一个测试激励即可



五. 实验中遇到的问题和解决办法

实验中遇到的主要问题是利用 74LS138 器件和 74LS139 器件去构建 5 选 1 的多数表决器。最朴素的想法是构建 32 个小项，每个小项都对应 5 名 voter 的一种投票情况。但是这样所使用的器件就很多。

后来想到用 74LS138 来表示后三个 voter 的投票情况，用 74LS139 来表示前两个 voter 的投票情况，最后将这两个投票情况用逻辑门来组合得到最后的正确结果，使用的器件也比较少了。

生成 bit stream 的时候找不到 bin 文件，反复生成了很多次，最后发现在设置中默认是不勾选生成 bin 文件的。

六. 附加题

1. 只采用 74LS138 译码器和一些基本逻辑门，是否也可以完成 5 输入多数表决器的设计？如果可以，请画出原理图。

用 38 译码器来表示 3 个投票人的投票情况，然后用两个门电路来表示里另外两个人的加起来的票数，最后通过多级门电路排列组合输出想要的票数（即大于等于 3 票）

