《数字逻辑与数字系统》实验报告

学院 智算 年级 2021 级 班级 软工 3 班 姓名 陈昊昆 学号 3021001196

课程名称 数字逻辑与数字系统 实验日期 4.11 成绩_____

同组实验者_____

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。

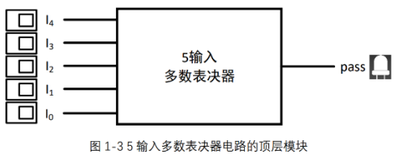


图 1-3 5 输入多数表决器电路的顶层模块

表 1-3 输入/输出端口

端口名	方向	宽度（位）	作用
I_4~I_0	输入	5	连接拨动开关 SW0~SW4，用于模拟五个裁判的输入。
led	输出	1	连接 LED 灯 LD0，用于显示是否成功。灯亮，表示成功；灯灭表示失败。

阶段 1：（基于集成电路模块）

1. 根据表 1-1 和 1-2，采用 SystemVerilog HDL 的行为建模方法，完成 74LS138 和 74LS139 两种译码器的设计。
2. 根据图 1-3 和表 1-4，基于 SystemVerilog HDL 的结构化建模方法，调用 74LS138 和 74LS139 两种译码器，以及若干基本逻辑门，完成 5 输入多数表 决器电路的设计（类似搭积木），并基于 Vivado 完成行为仿真、综合、实现、 生成比特流文件等操作，最终在远程 FPGA 硬件云平台上完成功能验证。

阶段 2：（基于行为建模）

1. 不使用 74LS138 和 74LS139 芯片，直接使用行为建模的方法完成 5 输入多数 表决器电路的设计，并基于 Vivado 完成电路的行为仿真、综合、实现、生成 比特流文件等操作，最终在远程 FPGA 硬件云平台上完成功能验证。

天津大学本科生实验报告专用纸

三．实验原理与步骤（注：步骤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。

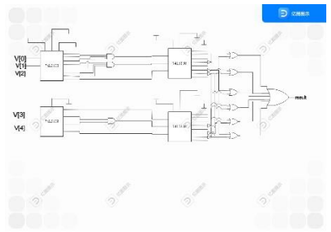
```
module dec_74LS138(
    input logic G, G2a, G2b,
    input logic [2 : 0]D,
    output logic [7 : 0]Y
    );

    always_comb begin
        case ({G, G2a, G2b, D})
            6'b100000:   Y = 8'b11111110;
            6'b100001:   Y = 8'b11111101;
            6'b100010:   Y = 8'b11111011;
            6'b100011:   Y = 8'b11110111;
            6'b100100:   Y = 8'b11101111;
            6'b100101:   Y = 8'b11011111;
            6'b100110:   Y = 8'b10111111;
            6'b100111:   Y = 8'b01111111;
            default :    Y = 8'b11111111;
        endcase
    end
endmodule

module dec_74LS139(
    input logic S,
    input logic [1 : 0]D,
    output logic [3 : 0]Y
    );

always_comb begin
    case ({S, D})
        3'b000:   Y = 4'b1110;
        3'b001:   Y = 4'b1101;
        3'b010:   Y = 4'b1011;
        3'b011:   Y = 4'b0111;
        default : Y = 4'b1111;
    endcase
end
endmodule
```

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 画图）。
1）将前三个裁判的投票输入 74LS138，将八个输出分成按得票总数 0、1、2、3 分成四类，并将同类结果相与（因为 0 表示有效）。将后两个裁判的投票输入 74LS139，做类似操作。
2）将前三个的投票得数为 1、2、3 的类输入一个 74LS138，再特定的输出为得到前三个投票的得数情况。后两个投票类似，将得票数 0、1、2 的类输入一个 74LS138。
3）然后综合两个 74LS138 的得票情况，枚举出所有通过的得票情况后相或，最后输出



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 logic [4 : 0]V,
    output logic result
    );
    logic G, G2a, G2b;
    logic [7 : 0] A, B, C;
    logic [3 : 0] D;
    logic A1, A2, A3;
    logic D0, D1, D2;
    logic R30, R31, R32, R12, R21, R22;

    assign G = 1;
    assign G2a = 0;
    assign G2b = 0;

    dec_74LS138 m1381 (G, G2a, G2b, {V[0], V[1], V[2]}, A);
    dec_74LS139 m139 (G2a, {V[3], V[4]}, D);

    assign A1 = A[1] & A[2] & A[4];
    assign A2 = A[3] & A[5] & A[6];
    assign A3 = A[7];

    assign D0 = D[0];
    assign D1 = D[1] & D[2];
    assign D2 = D[3];

    dec_74LS138 m1382 (G, G2a, G2b, {A1, A2, A3}, B);
    dec_74LS138 m1383 (G, G2a, G2b, {D0, D1, D2}, C);

    assign R32 = ~B[6] & ~C[6];
    assign R31 = ~B[6] & ~C[5];
    assign R30 = ~B[6] & ~C[3];
    assign R21 = ~B[5] & ~C[5];
    assign R22 = ~B[5] & ~C[6];
    assign R12 = ~B[3] & ~C[6];

    assign result = R32 | R31 | R30 | R21 | R22 | R12;


endmodule
```

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

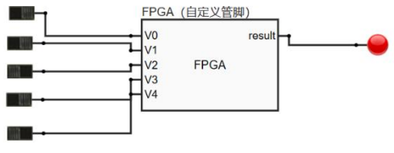
```
module voter5(
    input logic [4 : 0]V,
    output logic result
    );

    always_comb begin
        case (V)
        5'b00000: result = 1'b0;
        5'b00001: result = 1'b0;
        5'b00010: result = 1'b0;
        5'b00011: result = 1'b0;
        5'b00100: result = 1'b0;
        5'b00101: result = 1'b0;
        5'b00110: result = 1'b0;
        5'b00111: result = 1'b1;
        5'b01000: result = 1'b0;
        5'b01001: result = 1'b0;
        5'b01010: result = 1'b0;
        5'b01011: result = 1'b1;
        5'b01100: result = 1'b0;
        5'b01101: result = 1'b1;
        5'b01110: result = 1'b1;
        5'b01111: result = 1'b1;
        5'b10000: result = 1'b0;
        5'b10001: result = 1'b0;
```

```
            5'b10010: result = 1'b0;
            5'b10011: result = 1'b1;
            5'b10100: result = 1'b0;
            5'b10101: result = 1'b1;
            5'b10110: result = 1'b1;
            5'b10111: result = 1'b1;
            5'b11000: result = 1'b0;
            5'b11001: result = 1'b1;
            5'b11010: result = 1'b1;
            5'b11011: result = 1'b1;
            5'b11100: result = 1'b1;
            5'b11101: result = 1'b1;
            5'b11110: result = 1'b1;
            5'b11111: result = 1'b1;
            default : result = 1'b0;
            endcase
        end
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

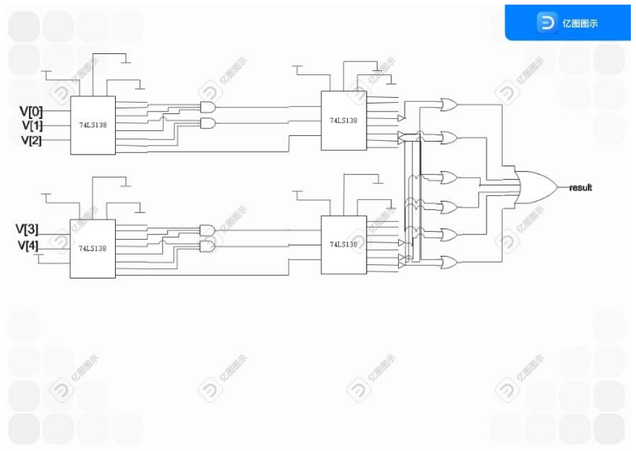
问题：实验阶段一中，如何将得票总数分类

解决方法：根据输出的位置，将相同的得票情况相与，得到票数相同的类，再进入 74LS138，得到结果，即前三（后二）投票的票数和。

六．附加题（若实验指导书无要求，则无需回答）

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。

假设有了第六个裁判，并该裁判永远给出通过



教师签字：

年　月　日