

《数字逻辑与数字系统》实验报告

学院 智算学部 年级 2019 级 班级 计科一班 姓名 李润泽 学号 3019244266

课程名称 数字逻辑与数字系统 实验日期 2021.04.14 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。

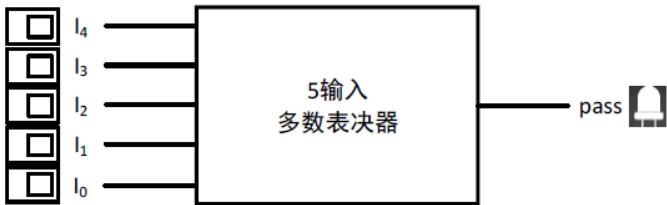


图 1-3 5 输入多数表决器电路的顶层模块

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。

天津大学本科生实验报告专用纸

5 输入多数表决器电路的真值表如下表所示。

输入					输出
I ₄	I ₃	I ₂	I ₁	I ₀	pass
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

本实验分为两阶段任务，每个阶段均是完成一个 5 输入多数表决器的设计，但采用的设计方法不同。具体实验内容如下所示：

阶段 1：（基于集成电路模块）

1、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、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。

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）

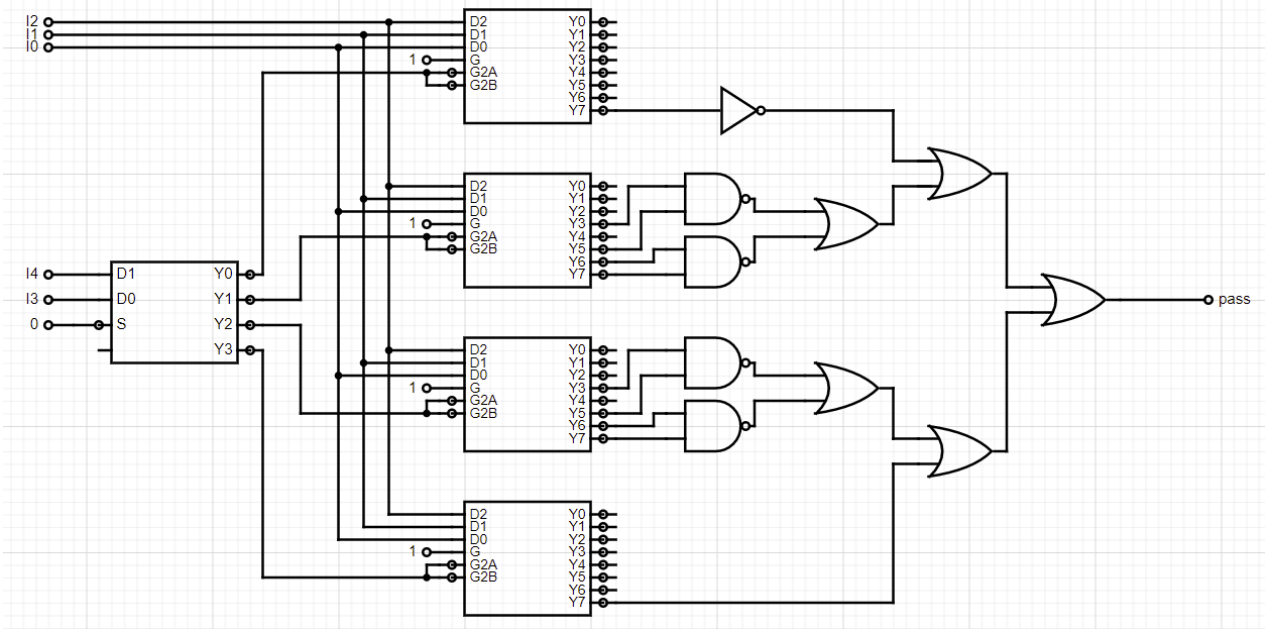
1. 写出 74LS138 和 74LS139 的行为建模的 SystemVerilog HDL 代码。

```
module dec_74LS138(  
    input [2:0] D,  
    [2:0] G,  
    output logic [7:0] Y  
);  
always_comb begin  
    if( G == 3'b100)begin  
        case(D)  
            3'b000 : Y = 8'b11111110 ;  
            3'b001 : Y = 8'b11111101 ;  
            3'b010 : Y = 8'b111111011 ;  
            3'b011 : Y = 8'b111110111 ;  
            3'b100 : Y = 8'b11101111 ;  
            3'b101 : Y = 8'b11011111 ;  
            3'b110 : Y = 8'b10111111 ;  
            3'b111 : Y = 8'b01111111 ;  
        endcase  
    end  
    else begin Y = 8'b11111111 ; end  
end  
endmodule
```

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

```
always_comb begin  
    if( s == 1'b0)begin  
        case(D)  
            2'b00 : Y = 4'b11110 ;  
            2'b01 : Y = 4'b11101 ;  
            2'b10 : Y = 4'b10111 ;  
            2'b11 : Y = 4'b01111 ;  
        endcase  
    end  
    else begin Y = 4'b11111 ; end  
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）。



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 logic pass  
);  
    logic [3:0] Y1;  
    logic [7:0] Y2;  
    logic [7:0] Y3;  
    logic [7:0] Y4;
```

```
logic [7:0] Y5;

dec_74LS139 test(I[4:3],1'b0,Y1);
dec_74LS138 a(.D(I[2:0]),.G({1'B1,Y1[0],Y1[0]}),.Y(Y2));
dec_74LS138 b(.D(I[2:0]),.G({1'B1,Y1[1],Y1[1]}),.Y(Y3));
dec_74LS138 c(.D(I[2:0]),.G({1'B1,Y1[2],Y1[2]}),.Y(Y4));
dec_74LS138 d(.D(I[2:0]),.G({1'B1,Y1[3],Y1[3]}),.Y(Y5));

always_comb begin
if(Y5 != 8'b11111111) begin
    if (Y5 == 8'b11111110) begin
        pass = 1'b0;
    end
    else begin pass = 1'b1; end
end

if(Y4 !=8'b11111111) begin
    if (Y4 >= 8'b11111011| Y4==8'b11101111) begin
        pass = 1'b0;
    end
    else begin pass = 1'b1; end
end

if(Y3 !=8'b11111111) begin
    if (Y3 >= 8'b11111011| Y3==8'b11101111) begin
        pass = 1'b0;
    end
    else begin pass = 1'b1;end
end

if(Y2 !=8'b11111111) begin
    if (Y2 == 8'b01111111) begin
        pass = 1'b1;
    end
    else begin pass = 1'b0;end
end

end
endmodule
```

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

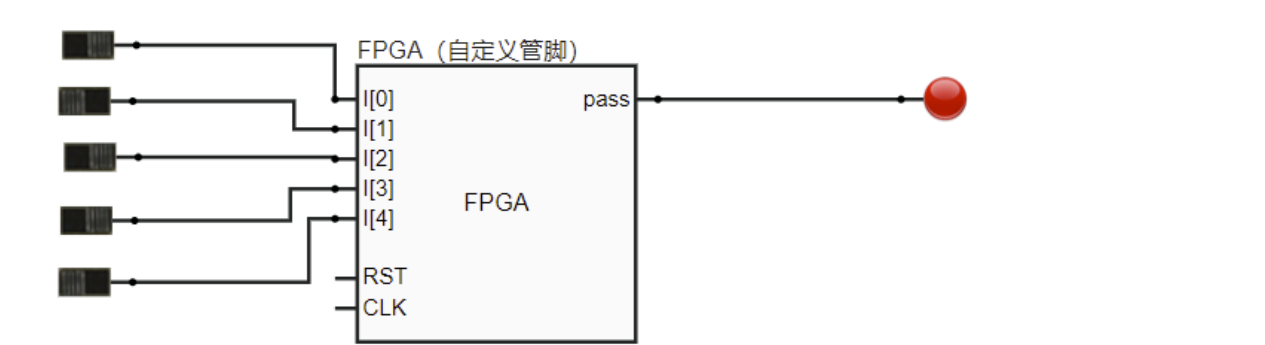
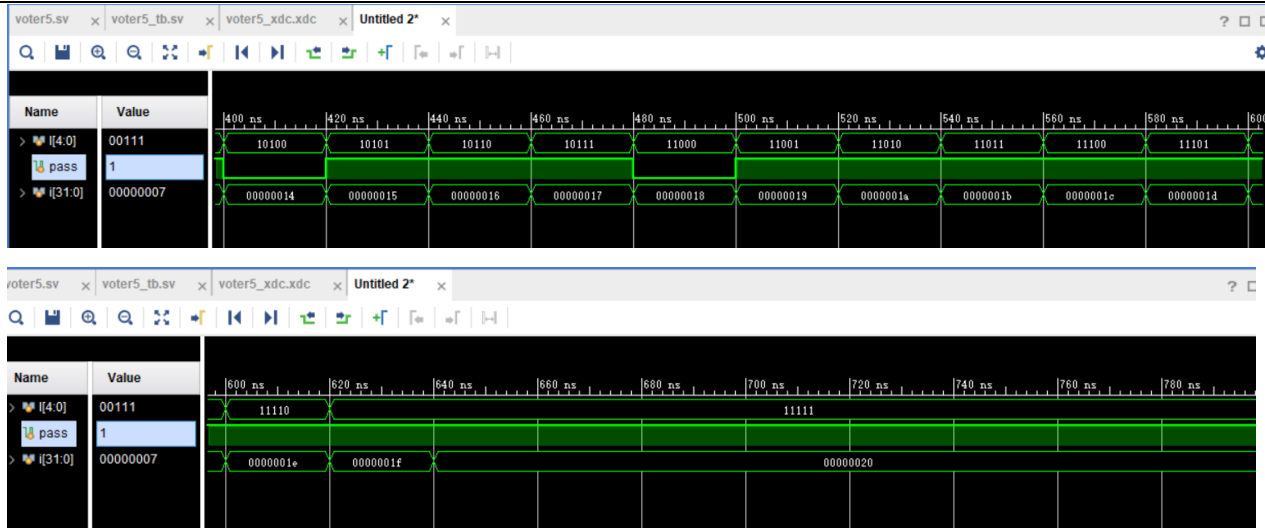
```
module voter5(
    input [4:0] I,
    output logic pass
);

    int counter = 0;
    always_comb begin
        for( int i = 0; i < 5 ;i = i+1) begin
            if ( I[i] == 1'b1) begin
                counter = counter + 1;
            end
        end
        if (counter>=3) begin pass = 1'b1 ; end
        else begin pass = 1'b0 ; end
        counter = 0;
    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

1、Vivado 无法生成比特流：

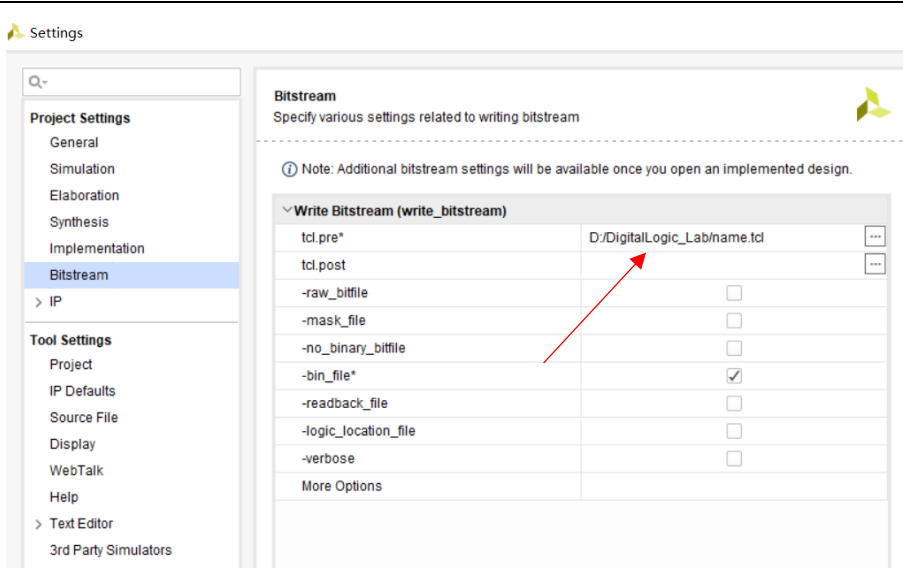
在进行 Vivado 实验中，其中一个步骤：Generate Bitstream 时发生报错，其中，重点句如下：

NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs Tcl command), add this command to a .tcl file and add that file as a pre-hook for write_bitstream step for the implementation run.

在进行网络查询之后，我发现缺少一个文件，因此，我们可以将如下三行代码：

```
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property SEVERITY {Warning} [get_drc_checks UCIO-1]
set_property SEVERITY {Warning} [get_drc_checks RTSTAT-1]
```

写到一个.tcl 文件中并保存，再将保存文件导入，便可正确生成比特流，操作如下：



2、虚拟平台上无法正确运行

实验过程中，虽然我的程序在 Vivado 上正确运行，但在虚拟平台上却无法正确运行，通过对课程实验指导的查询，我发现我的工程中，约束文件与虚拟平台的端口没有绑定正确，导致实验出现错误。

例如，输出管脚中：

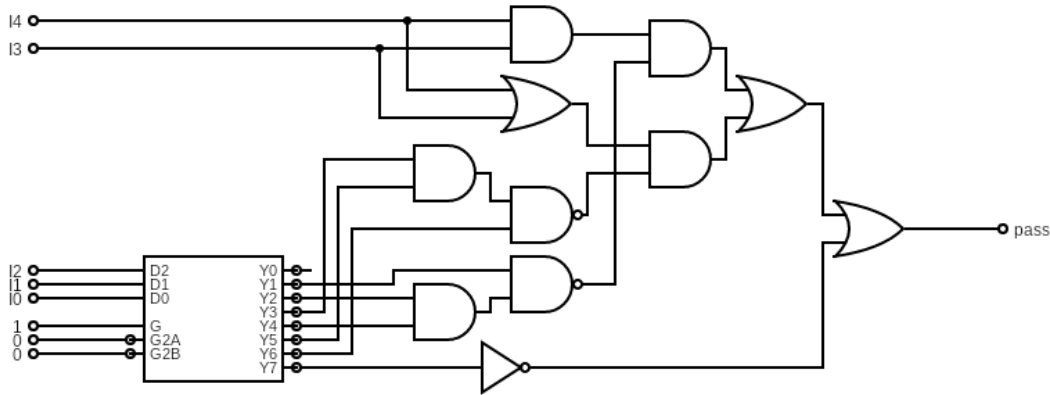
#Constraints of output ports pass

```
set_property -dict { PACKAGE_PIN K13 IOSTANDARD LVCMOS33 } [get_ports { pass }];
```

这句话代表在虚拟平台中，输出管脚对应的编号应该为 4（远程硬件平台管脚对应关系.xlsx）。在这种情况下，虚拟平台才可以正确运行。

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。



		<p>教师签字：</p> <p>年 月 日</p>
--	--	---------------------------