实验二、智力抢答器

一、实验目的

1.学习VerilogHDL基本语法；

2.巩固VIVADO环境下VerilogHDL编程设计的基础；

3.掌握数字系统中按键消抖，定时计数的设计要素及编写；

4.掌握Verilog中模块化调用的方法；

5.对有限状态机（FSM）做初步了解；

6.实现对EGO1开发板四位动态数码管的控制。

二、实验内容

设计一台智能电子抢答器并在EGO1开发板上实现，编号为1~4的选手在规定时间（10s）内按键抢答，抢中时锁定选手编号并显示，其他无效。主持人按开始和清零。当有人违规抢答时有提醒报警功能。

三、实验要求

1.抢答器可容纳4组参赛者抢答，每组设一个抢答按钮；

2.主持人按下开始键后显示抢答倒计时，此时选手可进行抢答；

3.若选手在主持人按下之前进行抢答，则系统显示该选手违规；

4.系统具有第一抢答信号的鉴别和锁存功能。即系统只显示首先抢答选手的组号；

5.若倒计时结束仍无人抢答，主持人再按一次则系统进入起始态。

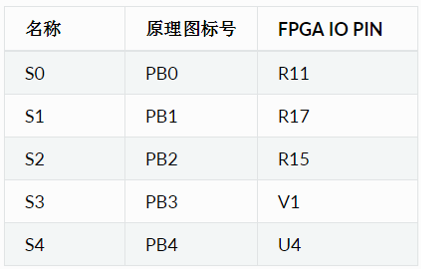
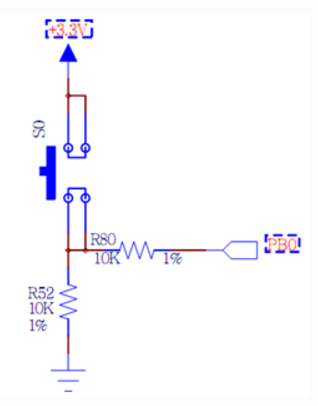
四、实验原理

1.按键处理

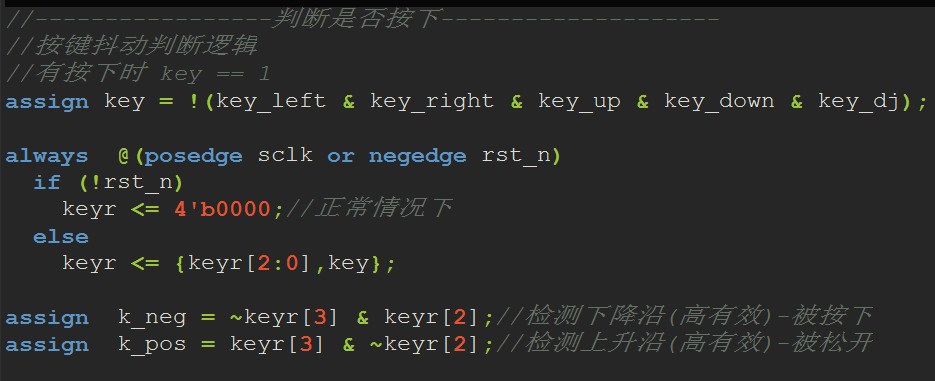
在实际应用中，很大一部分的按键是机械按键。在机械按键的触点闭合和断开时，都会产生抖动，为了保证系统能正确识别按键的开关，就必须对按键的抖动进行处理。在处理按键抖动的程序中，必须同时考虑消除闭合和断开两种情况下的抖动。

常用处理方式：系统时钟打两拍。

五个通用按键，默认为低电平，按键按下时输出高电平。管脚约束情况见右下图



一种有效的处理方式：



2.抢答器主控

当有抢答信号输入系统后，系统对最先抢到的选手号码进行编码锁存，并将这个号码显示输出。选手抢答时间为10s，所以需要定时器。

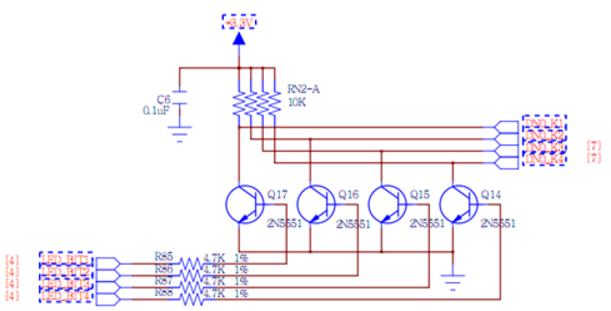
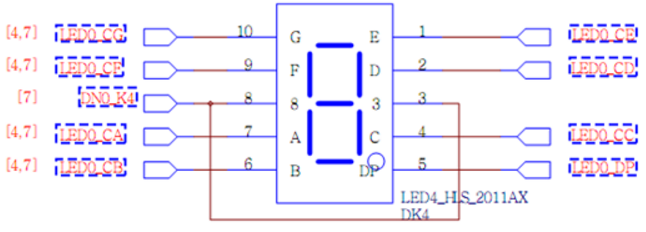
主持人按键有清零和开始功能。当第一次按下时为开始，系统进入10s倒计时，选手开始按键抢答。如果没有选手抢答，则系统倒计时到0。两种情况下，主持人按键清零准备下一轮抢答。

根据实验要求分析，系统共有六种状态，分别是初始态IDLE，等待主持人按下开始ST\_PRE，抢答态ST\_AHE，倒计时态ST\_WAT，有人回答态ST\_REC，无人回答态ST\_NON。根据分析画出状态迁移图。Btn\_state即为按键状态.

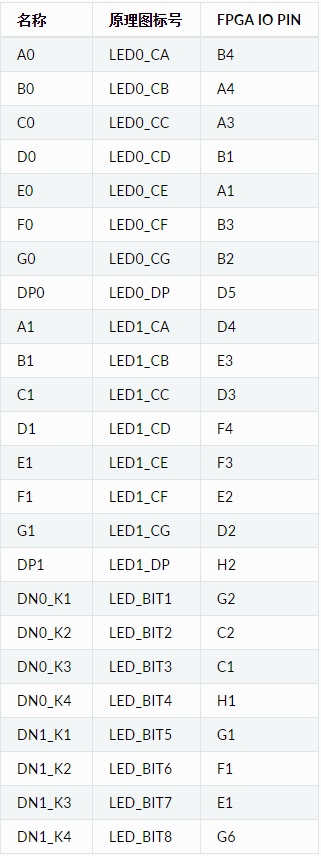


3.驱动数码管显示

EGO1开发板的数码管为共阴极数码管，即公共级输入低电平。共阴管由三极管驱动，FPGA需要提供正向信号。同时段选连接高电平，数码管上的对应位置才可以被点亮。因此，FPGA输出有效的片选信号和段选信号都应该是高电平。



动态显示的特点是将所有位数码管的段选线并联在一起，由位选线控制是哪一位数码管有效。点亮数码管采用动态扫描显示。所谓动态扫描显示即轮流向各位数码管送出字形码和相应的位选，只要扫描显示速度够快，利用发光管的余辉和人眼视觉暂留作用，使人的感觉好像各位数码管同时都在显示。动态扫描显示时刷新频率最好大于50HZ,即没显示一轮的时间不超过20ms，每个数码管显示时间不能太长也不能太短，时间太长会影响刷新率，导致总体显示呈现闪烁的现象，时间太短发光二极管的电流导通时间也就短，会影响总体的显示亮度。一般控制在1ms左右最佳。



开发板资料详见：http://e-elements.readthedocs.io/zh/ego1\_v2.1/EGo1.html#fpga

一个数码管动态显示的例子程序

module scan\_smg(

input wire sclk ,

input wire rst\_n ,

output reg [7:0] sm\_seg ,//段选

output reg [7:0] sm\_bit //位选

);

reg [15:0] cnt\_scan;//扫描频率计数器

reg [4:0] dataout\_buf;

always @(posedge sclk or negedge rst\_n)

if (!rst\_n)

cnt\_scan <= 0;

else begin

cnt\_scan <= cnt\_scan + 1;

end

always @(posedge sclk or negedge rst\_n)

if (!rst\_n)

sm\_bit <= 8'hff;

else begin

case(cnt\_scan[15:13])//根据扫描频率选择位选（确定哪个数码管亮）

3'b000: sm\_bit <= 8'b0000\_0001;

3'b001: sm\_bit <= 8'b0000\_0010;

3'b010: sm\_bit <= 8'b0000\_0100;

3'b011: sm\_bit <= 8'b0000\_1000;

3'b100: sm\_bit <= 8'b0001\_0000;

3'b101: sm\_bit <= 8'b0010\_0000;

3'b110: sm\_bit <= 8'b0100\_0000;

3'b111: sm\_bit <= 8'b1000\_0000;

default:sm\_bit <= 8'b0000\_0000;

endcase

end

always @(posedge sclk or negedge rst\_n)

if (!rst\_n)

dataout\_buf <= 0;

else begin

case(sm\_bit)//确定当前数码管显示数字

8'b0000\_0001: dataout\_buf <= 0;

8'b0000\_0010: dataout\_buf <= 2;

8'b0000\_0100: dataout\_buf <= 3;

8'b0000\_1000: dataout\_buf <= 4;

8'b0001\_0000: dataout\_buf <= 5;

8'b0010\_0000: dataout\_buf <= 3;

8'b0100\_0000: dataout\_buf <= 6;

8'b1000\_0000: dataout\_buf <= 7;

default: dataout\_buf <= 4'ha;

endcase

end

//数码管映射

always @(posedge sclk or negedge rst\_n)

if (!rst\_n)begin

sm\_seg <= ~(8'h88);//复位显示A

end

else begin//共阴极--高电平选中

case(dataout\_buf)//数码管显示

4'h0:sm\_seg <= ~(8'hc0);//0

4'h1:sm\_seg <= ~(8'hf9);//1

4'h2:sm\_seg <= ~(8'ha4);//2

4'h3:sm\_seg <= ~(8'hb0);//3

4'h4:sm\_seg <= ~(8'h99);//4

4'h5:sm\_seg <= ~(8'h92);//5

4'h6:sm\_seg <= ~(8'h82);//6

4'h7:sm\_seg <= ~(8'hf8);//7

4'h8:sm\_seg <= ~(8'h80);//8

4'h9:sm\_seg <= ~(8'h90);//9

4'ha:sm\_seg <= ~(8'h88);//a

4'hb:sm\_seg <= ~(8'h83);//b

4'hc:sm\_seg <= ~(8'hc6);//c

4'hd:sm\_seg <= ~(8'ha1);//d

4'he:sm\_seg <= ~(8'h86);//e

4'hf:sm\_seg <= ~(8'h8e);//f

default:sm\_seg<= 8'h80;//.

endcase

end

endmodule

五、实验结果

