Verilog期末设计

计算机学院 2211290 姚知言

**1 设计目标**

实现一个7路裁判打分电路，7个裁判各自在不同时刻打分，满分15分，输出平均整数得分，从第三个裁判给出分数开始，计算平均分时要去掉一个最高分和一个最低分。

**2 代码实现**

（1）judge7.v

`timescale 1ns / 1ps

module judge7(

输入时钟  
输入打分  
提交打分的按钮  
数码管输出  
选择输出到哪一个数码管

input clk,

input [3:0] scorein,

input submit,

output reg [7:0] seg,

output reg [1:0] sel

);

reg submit\_l;

记录上一个时钟周期的submit为submit\_l；submit\_e在提交打分按钮的下降沿触发

wire submit\_e = (submit\_l & ~submit);

reg [3:0] judgenum = 4'd0;//裁判总数

reg [4:0] scoremax = 5'd0;//最高分

reg [4:0] scoremin = 5'd16;//最低分

reg [6:0] scoresum = 5'd0;//总分

reg [15:0] clk\_div = 0;

wire slow\_clk = (clk\_div == 16'd50000);//分频器

wire [4:0] scoreavg = (judgenum >= 4'd3) ? ((scoresum - scoremax -scoremin )/( judgenum - 4'd2)):(judgenum == 4'd0) ? 5'd0 :(scoresum /judgenum);

wire [3:0] ten = scoreavg / 4'd10;

平均分的计算，在裁判超过三个的时候先减去最高和最低，再将分子减2

在1-2个时直接平均，在0个时输出0。

分别计算十位和个位用于数码管输出。

wire [3:0] one = scoreavg % 4'd10;

always @(posedge clk) begin

clk\_div <= clk\_div + 1;

if (clk\_div >= 16'd50000)

分频器，每50000个时钟周期更换一次数码管输出。

clk\_div <= 0;

submit\_l <= submit;

if (submit\_e) begin

if (judgenum < 3'd7) begin

judgenum <= judgenum + 1;

scoresum <= scoresum + scorein;

if (scoremax < scorein)

scoremax <= scorein;

if (scoremin > scorein)

对于每一次提交，若已经提交的裁判数量（judgenum）不足7个，则提交scorein，更新judgenum，scoremax和scoremin。

否则，此时提交认为是清零，将所有寄存器恢复到初始状态。

scoremin <= scorein;

end

else begin

judgenum <= 4'd0;

scoremax <= 5'd0;

scoremin <= 5'd16;

scoresum <= 5'd0;

end

end

if (slow\_clk) begin

if (sel == 2'b01) begin //个位

sel <= 2'b10;

case (one)

往复切换输出数码管个位和十位。根据0-9对应的数码管输出进行输出，若错误则输出字母E。

4'd0: seg <= 8'b0111\_1110; // 0

4'd1: seg <= 8'b0011\_0000; // 1

4'd2: seg <= 8'b0110\_1101; // 2

4'd3: seg <= 8'b0111\_1001; // 3

4'd4: seg <= 8'b0011\_0011; // 4

4'd5: seg <= 8'b0101\_1011; // 5

4'd6: seg <= 8'b0101\_1111; // 6

4'd7: seg <= 8'b0111\_0000; // 7

4'd8: seg <= 8'b0111\_1111; // 8

4'd9: seg <= 8'b0111\_1011; // 9

default: seg <= 8'b0100\_1111; // E

endcase

end else begin //十位

sel <= 2'b01;

case (ten)

4'd0: seg <= 8'b0111\_1110; // 0

4'd1: seg <= 8'b0011\_0000; // 1

4'd2: seg <= 8'b0110\_1101; // 2

4'd3: seg <= 8'b0111\_1001; // 3

4'd4: seg <= 8'b0011\_0011; // 4

4'd5: seg <= 8'b0101\_1011; // 5

4'd6: seg <= 8'b0101\_1111; // 6

4'd7: seg <= 8'b0111\_0000; // 7

4'd8: seg <= 8'b0111\_1111; // 8

4'd9: seg <= 8'b0111\_1011; // 9

default: seg <= 8'b0100\_1111; // E

endcase

end

end

end

endmodule

（2）约束文件mycon.xdc（实验箱）

//输入得分。用四个拨码开关表示8，4，2，1

set\_property PACKAGE\_PIN AC21 [get\_ports {scorein[3]}]

set\_property PACKAGE\_PIN AD24 [get\_ports {scorein[2]}]

set\_property PACKAGE\_PIN AC22 [get\_ports {scorein[1]}]

set\_property PACKAGE\_PIN AC23 [get\_ports {scorein[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[0]}]

//时钟

set\_property PACKAGE\_PIN AC19 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]

//提交信号，按钮

set\_property PACKAGE\_PIN Y3 [get\_ports submit]

set\_property IOSTANDARD LVCMOS33 [get\_ports submit]

//七段数码管

set\_property PACKAGE\_PIN C4 [get\_ports {seg[7]}]

set\_property PACKAGE\_PIN A2 [get\_ports {seg[6]}]

set\_property PACKAGE\_PIN D4 [get\_ports {seg[5]}]

set\_property PACKAGE\_PIN E5 [get\_ports {seg[4]}]

set\_property PACKAGE\_PIN B4 [get\_ports {seg[3]}]

set\_property PACKAGE\_PIN B2 [get\_ports {seg[2]}]

set\_property PACKAGE\_PIN E6 [get\_ports {seg[1]}]

set\_property PACKAGE\_PIN C3 [get\_ports {seg[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[0]}]

//七段数码管选通

set\_property PACKAGE\_PIN D3 [get\_ports {sel[1]}]

set\_property PACKAGE\_PIN D25 [get\_ports {sel[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel[0]}]

（3）约束文件mycon.xdc（口袋实验板）

//文件结构类似

set\_property PACKAGE\_PIN R2 [get\_ports {scorein[3]}]

set\_property PACKAGE\_PIN M4 [get\_ports {scorein[2]}]

set\_property PACKAGE\_PIN N4 [get\_ports {scorein[1]}]

set\_property PACKAGE\_PIN R1 [get\_ports {scorein[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {scorein[0]}]

set\_property PACKAGE\_PIN P17 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]

set\_property PACKAGE\_PIN R11 [get\_ports submit]

set\_property IOSTANDARD LVCMOS33 [get\_ports submit]

set\_property PACKAGE\_PIN D5 [get\_ports {seg[7]}]

set\_property PACKAGE\_PIN B2 [get\_ports {seg[0]}]

set\_property PACKAGE\_PIN B3 [get\_ports {seg[1]}]

set\_property PACKAGE\_PIN A1 [get\_ports {seg[2]}]

set\_property PACKAGE\_PIN B1 [get\_ports {seg[3]}]

set\_property PACKAGE\_PIN A3 [get\_ports {seg[4]}]

set\_property PACKAGE\_PIN A4 [get\_ports {seg[5]}]

set\_property PACKAGE\_PIN B4 [get\_ports {seg[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[0]}]

set\_property PACKAGE\_PIN C2 [get\_ports {sel[1]}]

set\_property PACKAGE\_PIN G2 [get\_ports {sel[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel[0]}]

（4）仿真文件tb.v

`timescale 1ns / 1ps

module tb();

reg clk;

reg [3:0] scorein;

reg submit;

wire [7:0] seg;

wire [1:0] sel;

judge7 uut(clk,scorein,submit,seg,sel);

initial begin

clk = 1'd0;

scorein = 4'd0;

submit = 1'd0;

因为数码管50000个时钟周期更新一次，所以设定250000个时钟周期提交一次scorein，以保证数码管可以充分更新

end

always #1 clk = ~clk;

always #500000 begin

scorein = $random % 4'd15 + 1;//每次更新置scorein为1-15的随机数

submit = 1'd1;

#3

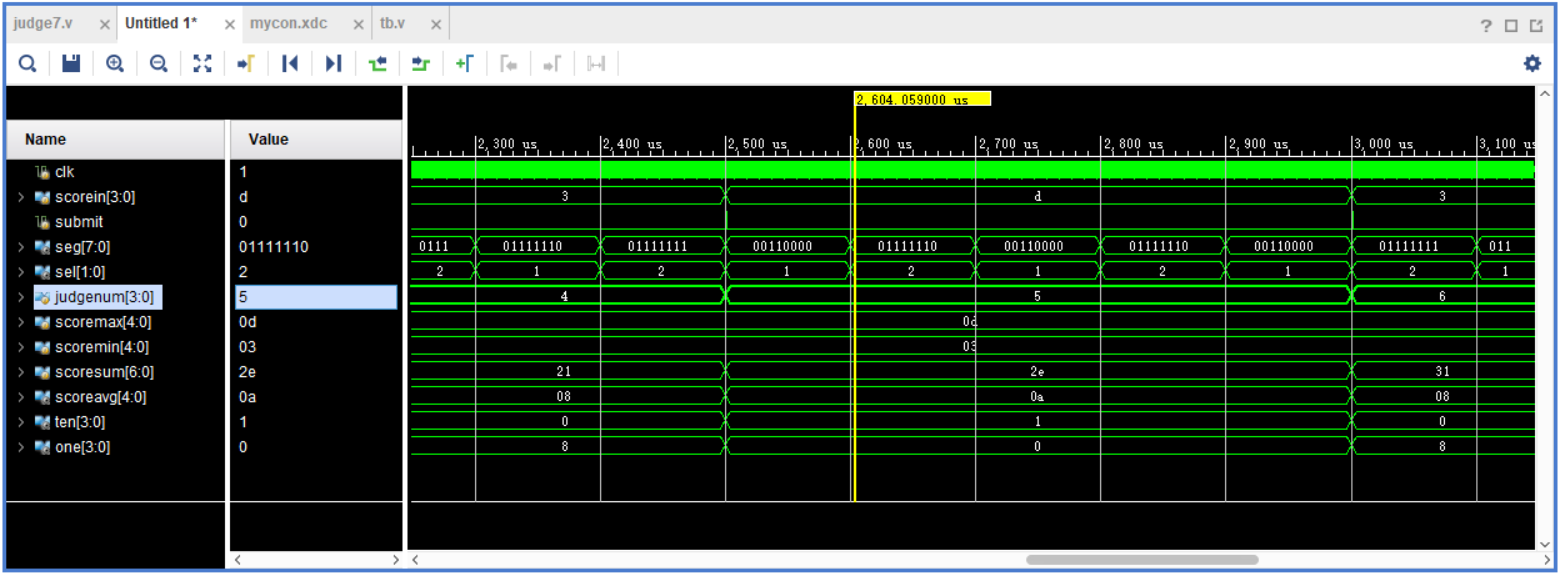
submit = 1'd0;

end

endmodule

**3 仿真验证**

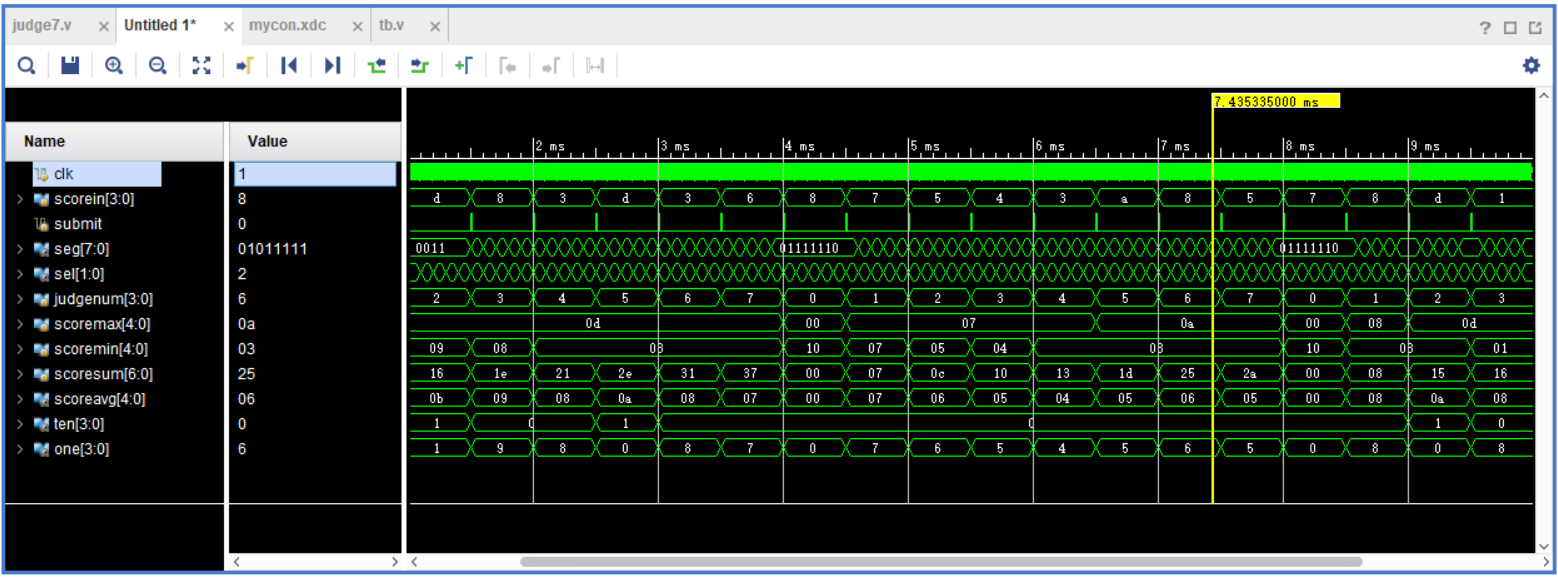
（1）细节计算和输出验证



可以看到，在探针位置，已经有5个裁判完成了打分，第五个裁判的打分为13（0xd），此时的总分是46（0x2e）（上一个总分为33（0x2a），增加13为46，正确），最高分是13（0xd），最低分是3（0x3），去掉最高分和最低分之后的总分是30（0x14），除以裁判总数3的结果应该为10（0xa），十位应该为1，个位应该为0。可以看到图中所有数字都正确更新。

当sel为1的时候，输出十位1对应的数码管，当sel为1的时候，输出个位0对应的数码管，也完全正确。

（2）总体逻辑验证



总体上可以看到，裁判器可以连续的完成多次裁判（judgnum一直在0-7之间更新）。

4-8ms是一次完整的裁判过程，在刚开始的时候个数据成功清零，随着七次裁判的输入（7，5，4，3，10，8，5），最高分和最低分都正常进行了更新，总分也正常进行了累加，平均分和输出也正常进行了计算。

**4 上箱验证**

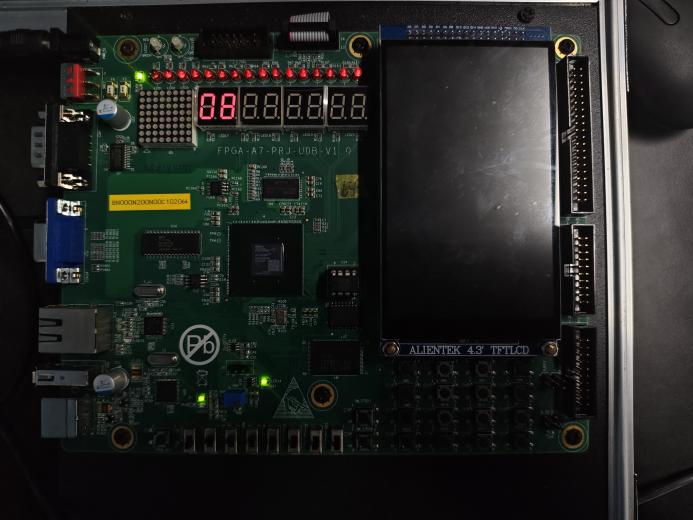
以依次提交8，9，10，11，12，13，14进行验证。

（1）初始上电状态



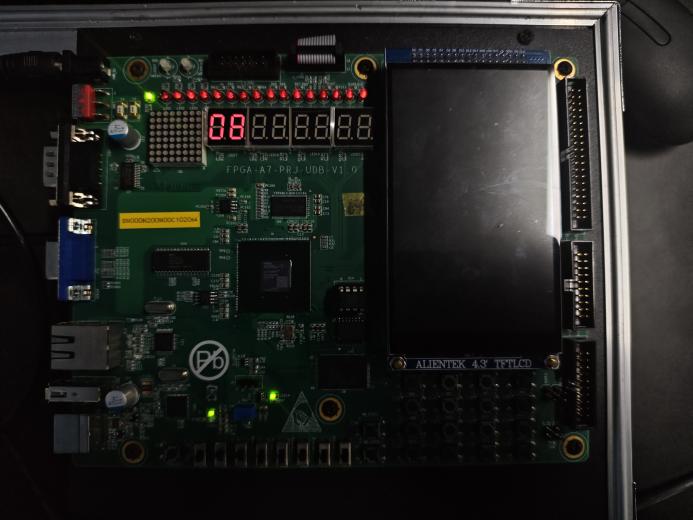
初始状态，平均分为0。

（2）第1个裁判：拨码开关输入8（1000），按动提交按钮



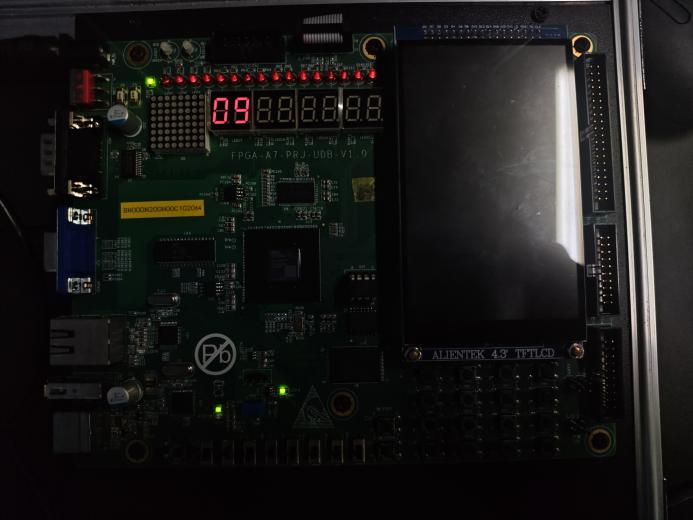
此时打分为8，共1个裁判，平均分8/1=8。

（3）第2个裁判：拨码开关输入9（1001），按动提交按钮



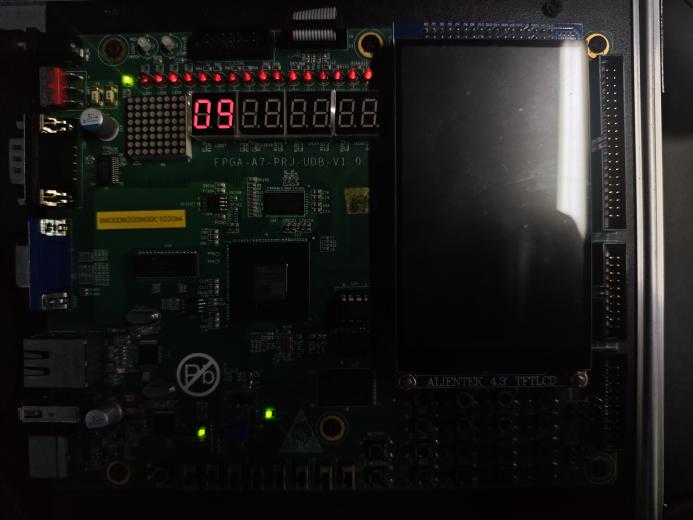
此时打分为8，9，共2个裁判，平均分17/2=8.5，向下取整为8。

（4）第3个裁判：拨码开关输入10（1010），按动提交按钮



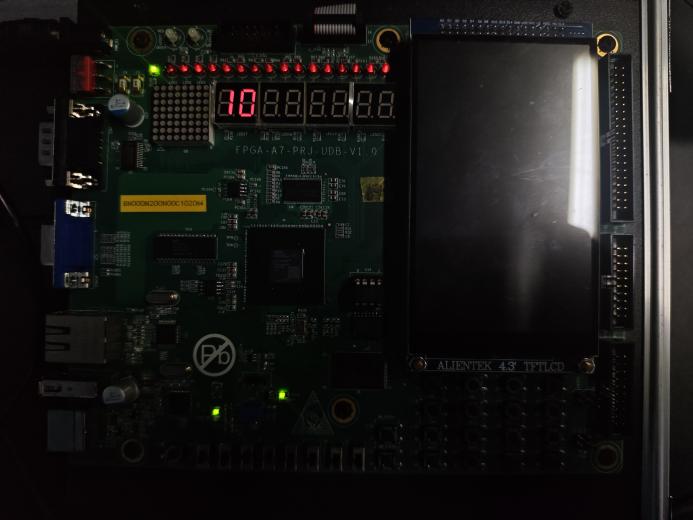
此时打分为8，9，10，去掉最低分8和最高分10，平均分9/1=9。

（5）第4个裁判：拨码开关输入11（1011），按动提交按钮



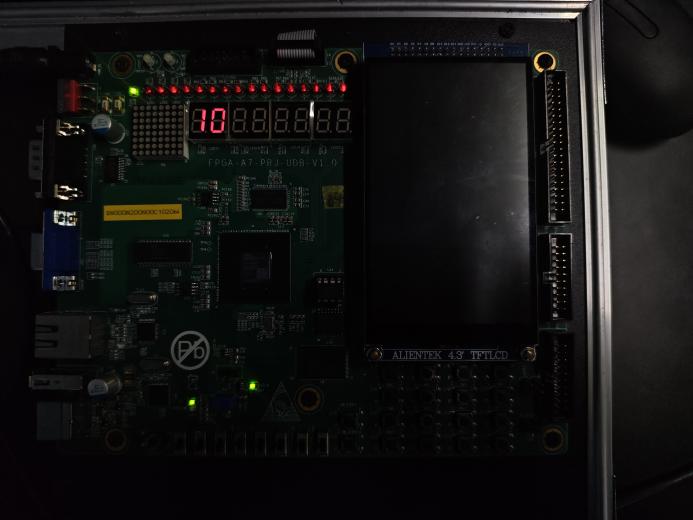
此时打分为8，9，10，11，去掉最低分8和最高分11，平均分19/2=9.5，向下取整9。

（6）第5个裁判：拨码开关输入12（1100），按动提交按钮



此时打分为8，9，10，11，12，去掉最低分8和最高分12，平均分30/3=10。

（7）第6个裁判：拨码开关输入13（1101），按动提交按钮



此时打分为8，9，10，11，12，13，去掉最低分8和最高分13，平均分42/4=10.5，向下取整10。

（8）第7个裁判：拨码开关输入14（1110），按动提交按钮



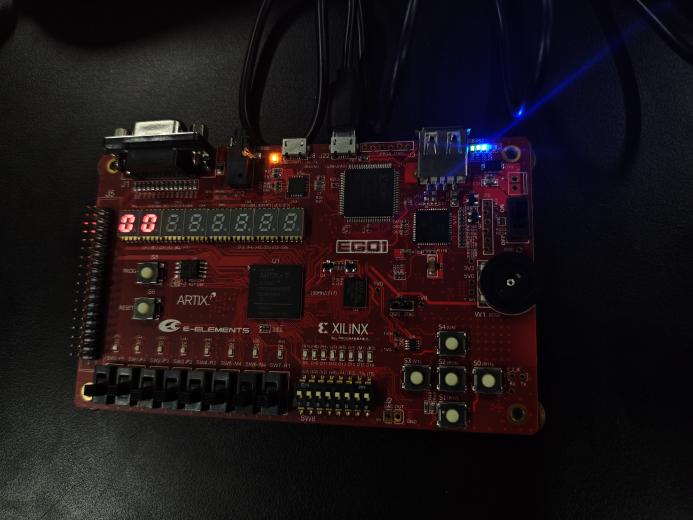
此时打分为8，9，10，11，12，13，14，去掉最低分8和最高分14，平均分55/5=11。

此时若再按动提交按钮，则将整体计数器清零，即回到初始状态。

**5 上板验证**

以依次提交12，9，1，6，8，12，10进行验证。

（1）初始上电状态



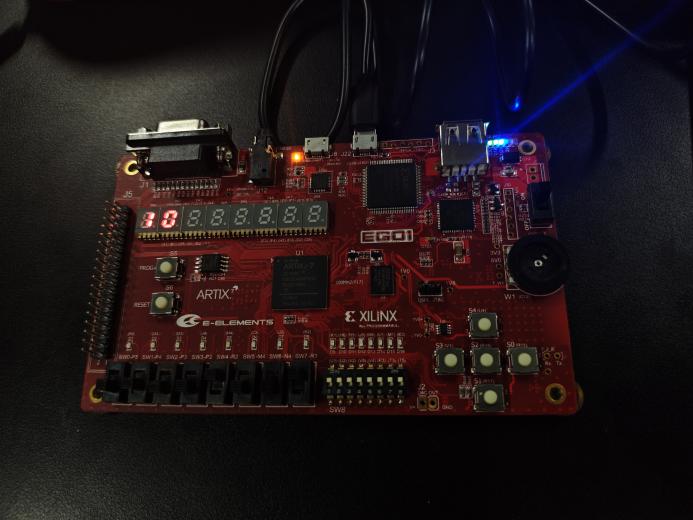
初始状态，平均分为0。

（2）第1个裁判：拨码开关输入12（1100），按动提交按钮



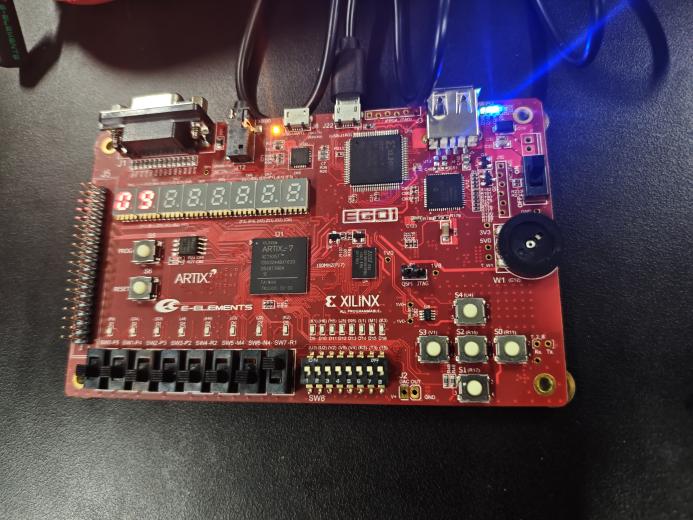
此时打分为12，共1个裁判，平均分12/1=12。

（3）第2个裁判：拨码开关输入9（1001），按动提交按钮



此时打分为12，9，共2个裁判，平均分21/2=10.5，向下取整为10。

（4）第3个裁判：拨码开关输入1（0001），按动提交按钮



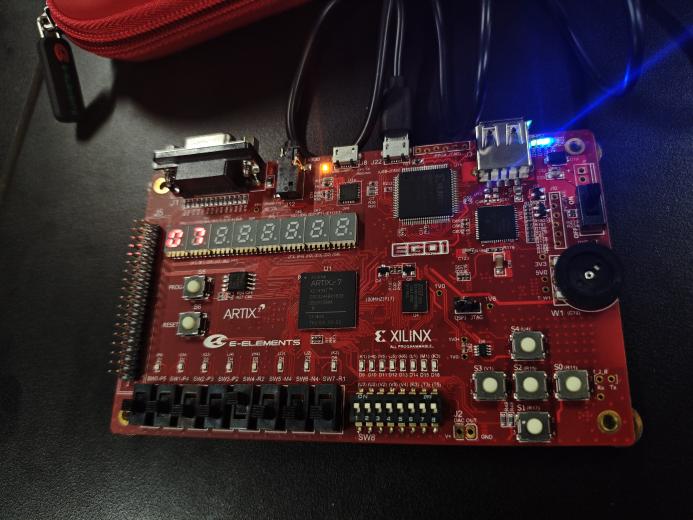
此时打分为12，9，1，去掉最低分1和最高分12，平均分9/1=9。（可以看到并不是直接平均）

（5）第4个裁判：拨码开关输入6（0110），按动提交按钮



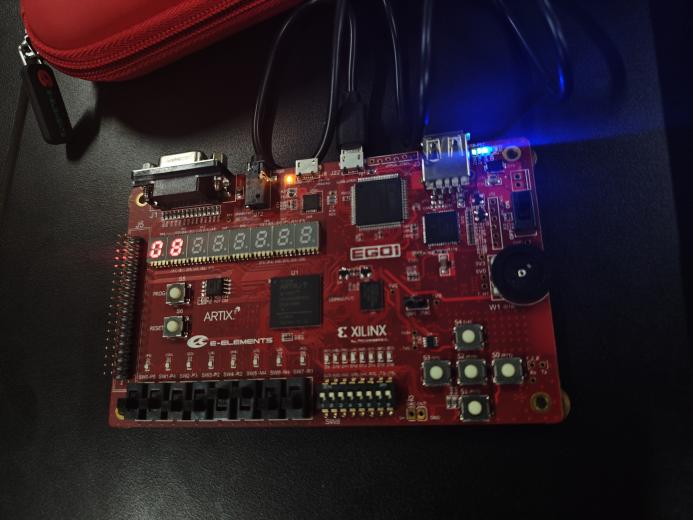
此时打分为12，9，1，6，去掉最低分1和最高分12，平均分15/2=7.5，向下取整7。

（6）第5个裁判：拨码开关输入8（1000），按动提交按钮



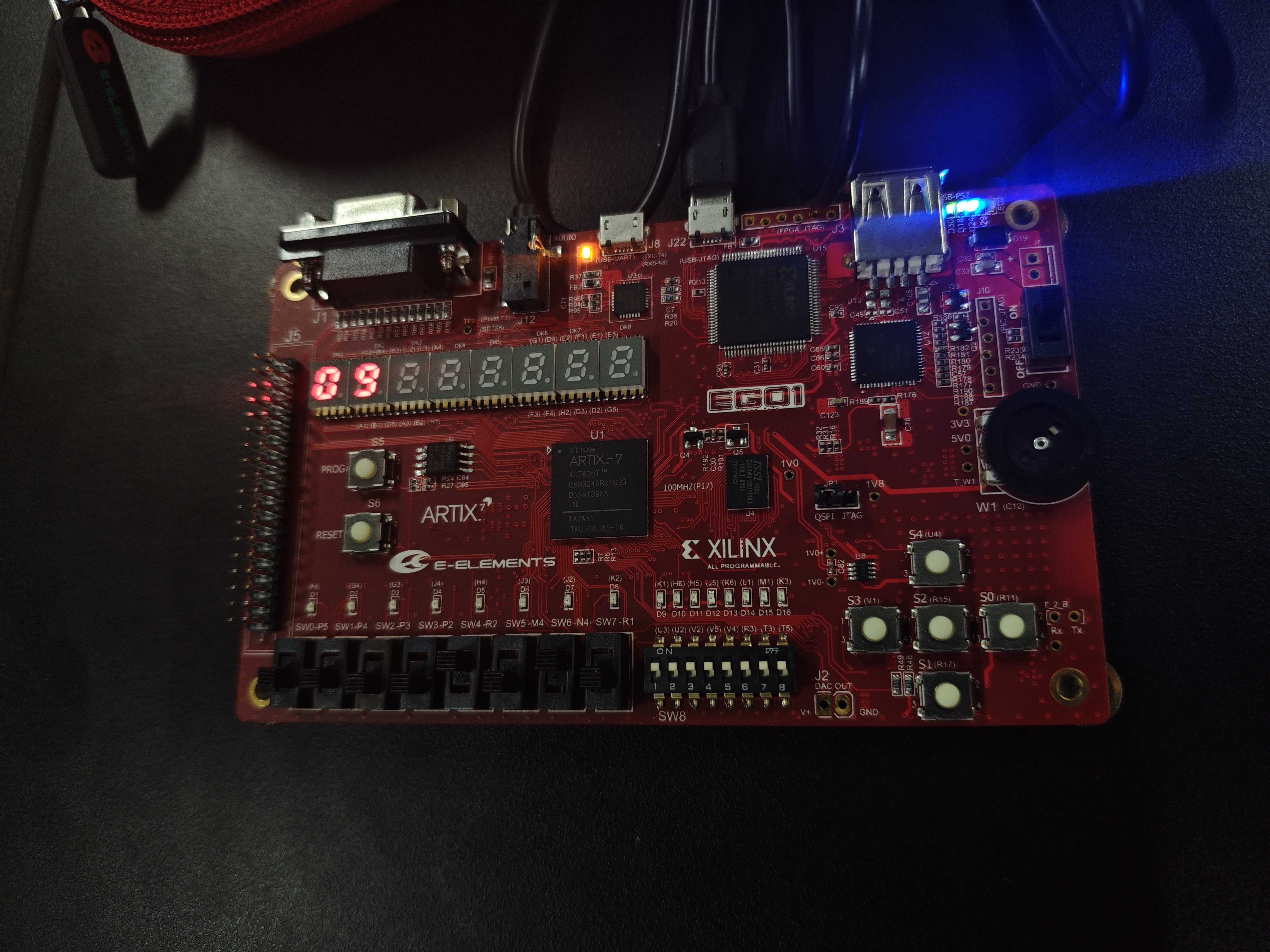
此时打分为12，9，1，6，8，去掉最低分1和最高分12，平均分23/3=7.67，向下取整7。

（7）第6个裁判：拨码开关输入12（1100），按动提交按钮



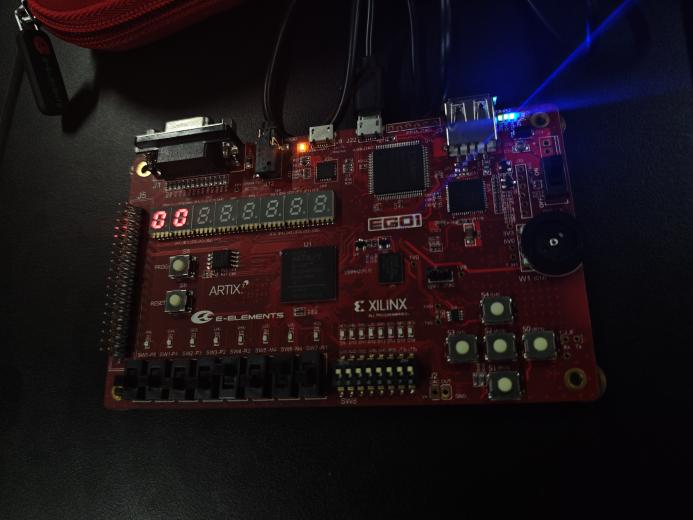
此时打分为12，9，1，6，8，12，去掉最低分8和最高分12（1个），平均分35/4=8.75，向下取整8。

（8）第7个裁判：拨码开关输入10（1010），按动提交按钮



此时打分为12，9，1，6，8，12，10，去掉最低分1和最高分12（1个），平均分45/5=9。

此时若再按动提交按钮，则将整体计数器清零，即回到初始状态。



**6 总结**

在本次实验中，我通过实现7路裁判打分电路，串联起本学习所学习的所有知识，更进一步的了解了Verilog语法。在实验中，我编写了以拨码开关控制打分，以按钮提交打分，以两位七段数码管输出平均得分。（向下取整，三人以上打分时去掉最高分和最低分。）在实验中，我成功通过了仿真验证，上箱验证，上板验证。并借助这个过程，对vivado和verilog语法，实验箱/实验板等有了较为深刻的认识。