电子秒表实现00分00.00秒到59分59.99秒的计时（vivado,verilog语言）(上)

原创

初升的太阳LX 于 2019-07-17 19:16:15 发布 9555 收藏 96
分类专栏： 学校学习实践 文章标签： 数电实验

本文链接：https://blog.csdn.net/weixin_43977564/article/details/96343437

学校学习实践 专栏收录该内容
1 篇文章 0 订阅
订阅专栏

本实验项目为数电实验期末验收课题，需要结合前几次数电实验内容自行实现电子秒表。因为在做的过程中比较容易实现，因此我就简要地向大家描述一下我的思路过程或者说是实现内容。前几次的数电实验内容也在本博客中有所展现，希望大家能够从我的博客中对电子秒表实现有一个了解吧。大佬请略过！

一、实现内容：

（1）计时范围：00.00.00到59.59.99

（2）能够完成复位、启动、暂停功能

（6）用6位七段数码管显示读数

二、注意事项

（1）时钟分频：秒表精度为0.01秒，我们使用的FPGA提供50MHZ信号，因此要用到一个500000HZ的分频器。

（2）实现小数点显示：

首先我们知道数码管里有七个led灯，用7bit,但是我们要用八位，在实现小数点的数码管中，将最高位置为1即可。

（3）源码：

顶层文件：

```
module final_top(
    input clk_50M,
    input CLR_L,
    input start_stop,
    output [7:0] seg,
    output [5:0] dig
    );
  //分频部分
 wire clk_s;

 clk_div u1(.clk_in(clk_50M),.clk_out(clk_s));

 //计数器部分

      wire[3:0] Q_0,Q_1,Q_2,Q_3,Q_4,Q_5;// 计数器的输出
         wire cy_0,cy_1,cy_2,cy_3,cy_4,cy_5;  //进位信号

             modu10_counter u2(.clk(clk_s),.clr(CLR_L),.EN(start_stop),.cy(cy_0),.Q(Q_0));
                modu10_counter u3(.clk(clk_s),.clr(CLR_L),.EN(cy_0),.cy(cy_1),.Q(Q_1));
                modu10_counter u4(.clk(clk_s),.clr(CLR_L),.EN(cy_1),.cy(cy_2),.Q(Q_2));
                modu6_counter u5(.clk(clk_s),.clr(CLR_L),.EN(cy_2),.cy(cy_3),.Q(Q_3));
                modu10_counter u6(.clk(clk_s),.clr(CLR_L),.EN(cy_3),.cy(cy_4),.Q(Q_4));
                modu6_counter u7(.clk(clk_s),.clr(CLR_L),.EN(cy_4),.cy(cy_5),.Q(Q_5));

         //调用3位数码管显示模块
          wire[3:0] disp_data_right0,disp_data_right1,disp_data_right2,disp_data_right3,disp_data_right4,di
          assign disp_data_right0=Q_0;
          assign disp_data_right1=Q_1;
          assign disp_data_right2=Q_2;
          assign disp_data_right3=Q_3;
          assign disp_data_right4=Q_4;
          assign disp_data_right5=Q_5;
          dynamic_led6 u8 (
                           .disp_data_right0(disp_data_right0),
                           .disp_data_right1(disp_data_right1),
                           .disp_data_right2(disp_data_right2),
                           .disp_data_right3(disp_data_right3),
                           .disp_data_right4(disp_data_right4),
                           .disp_data_right5(disp_data_right5),
                           .clk(clk_50M),
                           .seg(seg),
                           .dig(dig)
                           );
endmodule
```

时钟分频模块：

```
module clk_div(clk_in,clk_out);
 input clk_in;
 output reg clk_out=0;//用reg后面always中需要改变数值
  reg[24:0] clk_div_cnt=0;
//分频为100Hz的信号
        always @ (posedge clk_in)
        begin
            if (clk_div_cnt==249999)
            begin
                clk_out=~clk_out;
                clk_div_cnt=0;
            end
            else
                clk_div_cnt=clk_div_cnt+1;
       end
 endmodule
```

计数器模块：用两种计数模块，一种模10，一种模6。总共六个计数器，1，2，3，5这四个数码管用模10；4，6这两个数码管用模6。

```
module modu10_counter(clk,clr,EN,Q,cy);
 input clk,clr;
 input EN;               //使能信号
 output cy;              //计数器进位输出
 output reg [3:0] Q;     // 计数器的输出

         always @(posedge clk or negedge clr)  //异步清零
              begin
                  if (~clr)       //清零有效
                    begin
                     Q<=0;
                    end          //完成清零操作，计数器输出为0
                  else if(EN==1)  //使能有效
                    begin
                     if (Q==9)    //计数+1，若低位已经到最大数9
                      begin
                        Q<=0;      //输出跳转到最小数0
                        end
                     else Q<=Q+1;       //若输出未到最大数，则只加1
                     end
              end
               //计到最大数9，同时使能有效，输出Cy为1
              assign cy=((EN==1)&&(Q==9))?1'b1:1'b0;
endmodule
```

```
module modu6_counter(clk,clr,EN,Q,cy);
 input clk,clr;
 input EN;               //使能信号
 output cy;              //计数器进位输出
 output reg [3:0] Q;     // 计数器的输出

         always @(posedge clk or negedge clr)  //异步清零
              begin
                  if (~clr)       //清零有效
                    begin
                     Q<=0;
                    end          //完成清零操作，计数器输出为0
                  else if(EN==1)  //使能有效
                    begin
                     if (Q==5)    //计数+1，若低位已经到最大数9
                      begin
                        Q<=0;      //输出跳转到最小数0
                        end
                     else Q<=Q+1;       //若输出未到最大数，则只加1
                     end
              end
               //计到最大数9，同时使能有效，输出Cy为1
              assign cy=((EN==1)&&(Q==5))?1'b1:1'b0;
endmodule
```

时钟动态显示模块：

```
module dynamic_led6(
input [3:0]disp_data_right0,
input [3:0]disp_data_right1,
input [3:0]disp_data_right2,
input [3:0]disp_data_right3,
input [3:0]disp_data_right4,
input [3:0]disp_data_right5,
input clk,
output  reg  [7:0] seg,
output  reg  [5:0] dig
 );

 //分频为1KHz
 reg[24:0] clk_div_cnt=0;
 reg clk_div=0;
 always @ (posedge clk)
 begin
  if (clk_div_cnt==24999)
  begin
   clk_div=~clk_div;
   clk_div_cnt=0;
  end
  else
      clk_div_cnt=clk_div_cnt+1;
 end
 //6进制计数器
 reg [2:0] num=0;
 always @ (posedge clk_div)
 begin
  if (num>=5)
```

```
   num=0;
  else
   num=num+1;
 end

 //译码器
 always @ (num)
 begin
  case(num)
  0:dig=6'b111110;
  1:dig=6'b111101;
  2:dig=6'b111011;
  3:dig=6'b110111;
        4:dig=6'b101111;
        5:dig=6'b011111;
  default: dig=0;
  endcase
 end

 //选择器，确定显示数据
 reg [3:0] disp_data;
 always @ (num)
 begin
  case(num)
  0:disp_data=disp_data_right0;
  1:disp_data=disp_data_right1;
  2:disp_data=disp_data_right2;
  3:disp_data=disp_data_right3;
        4:disp_data=disp_data_right4;
        5:disp_data=disp_data_right5;
  default: disp_data=0;
  endcase
 end
 //显示译码器
 always@(disp_data)
 begin
  case(disp_data)
  4'h0: seg=8'h3f;// DP,GFEDCBA
  4'h1: seg=8'h06;
  4'h2: seg=8'h5b;
  4'h3: seg=8'h4f;
  4'h4: seg=8'h66;
  4'h5: seg=8'h6d;
  4'h6: seg=8'h7d;
  4'h7: seg=8'h07;
  4'h8: seg=8'h7f;
  4'h9: seg=8'h6f;
  4'ha: seg=8'h77;
  4'hb: seg=8'h7c;
  4'hc: seg=8'h39;
  4'hd: seg=8'h5e;
  4'he: seg=8'h79;
  4'hf: seg=8'h71;
  default: seg=0;
  endcase
 end

endmodule
```

因为有注释，看代码应该不是问题

要进行实现，我们还需要约束文件，因为我马上要吃饭了，下篇博客发约束文件。