`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 15.10.2018 14:59:23

// Design Name:

// Module Name: sec

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 14.10.2018 14:10:41

// Design Name:

// Module Name: sec

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module clk\_divider(clk\_out,clk);

input clk;

output reg clk\_out;

integer counter;

initial

begin

counter=0;

clk\_out=0;

end

always@(posedge clk)

begin

if (counter<50000000)

counter=counter+1;

else

begin

counter=0;

clk\_out=~clk\_out;

end

end

endmodule

module sec(clk,rst,user,sec,min,hour,m,h);

input clk;

input user;

input rst;

input [5:0]m;

input [4:0]h;

output reg [5:0]sec;

output reg [5:0]min;

output reg [4:0]hour;

wire clk\_out;

clk\_divider c1(clk\_out,clk);

//initial

//begin

//min<= m;

//hour<= h;

//end

always @(posedge clk\_out)

begin

if (rst == 1)

begin

sec= 6'd0;

min=6'd0;

hour=5'd0;

end

else if(user==1)

begin

min= m;

hour= h;

if(sec != 6'd60)

begin

sec= sec + 6'b000001;

if(sec == 6'd60)

begin

sec=6'd0;

min= min + 6'd1;

if(min == 6'd60)

begin

min=6'd0;

hour= hour + 5'd1;

if(hour == 5'd24)

hour= 5'd0;

end

end

end

end

else

begin

if(sec != 6'd60)

begin

sec= sec + 6'b000001;

if(sec == 6'd60)

begin

sec=6'd0;

min= min + 6'd1;

if(min == 6'd60)

begin

min=6'd0;

hour= hour + 5'd1;

if(hour == 5'd24)

hour= 5'd0;

end

end

end

end

end

endmodule

module display(an,seg,num,clk);

input clk;

input [63:0]num;

output [7:0]an;

output [7:0]seg;

reg [19:0]counter;

always@(posedge clk)

counter=counter+1;

assign an = (counter[19:17]==3'b000)?8'hFE:

(counter[19:17]==3'b001)?8'hFD:

(counter[19:17]==3'b010)?8'hFB:

(counter[19:17]==3'b011)?8'hF7:

(counter[19:17]==3'b100)?8'hEF:

(counter[19:17]==3'b101)?8'hDF:

(counter[19:17]==3'b110)?8'hBF:

8'h7F;

assign seg = (counter[19:17]==3'b000)?num[7:0]:

(counter[19:17]==3'b001)?num[15:8]:

(counter[19:17]==3'b010)?num[23:16]:

(counter[19:17]==3'b011)?num[31:24]:

(counter[19:17]==3'b100)?num[39:32]:

(counter[19:17]==3'b101)?num[47:40]:

(counter[19:17]==3'b110)?num[55:48]:

num[63:56];

endmodule

module segdisplay(seg,an,u,clk,rst,min,hour);

input rst;

input u;

input [5:0]min;

input [4:0]hour;

input clk;

output [7:0]seg,an;

wire [5:0]P;

wire [5:0]Q;

wire [4:0]R;

wire [7:0]x[9:0];

reg [7:0]k;

reg [7:0]l;

reg [7:0]m;

reg [63:0]num;

assign x[0]=8'hC0;

assign x[1]=8'hF9;

assign x[2]=8'hA4;

assign x[3]=8'hB0;

assign x[4]=8'h99;

assign x[5]=8'h92;

assign x[6]=8'h82;

assign x[7]=8'hF8;

assign x[8]=8'h80;

assign x[9]=8'h98;

sec a1(clk,rst,u,P,Q,R,min,hour);

always @(posedge clk)

begin

k={2'b00,P};

l={2'b00,Q};

m={2'b00,R};

num={x[m/10],x[m%10],8'hFF,x[l/10],x[l%10],8'hFF,x[k/10],x[k%10]};

end

display d1(an,seg,num,clk);

endmodule

module toptff(sw,clk,seg,an);

input [12:0]sw;

input clk;

output [7:0]seg,an;

segdisplay s(seg,an,sw[0],clk,sw[1],sw[7:2],sw[12:8]);

endmodule