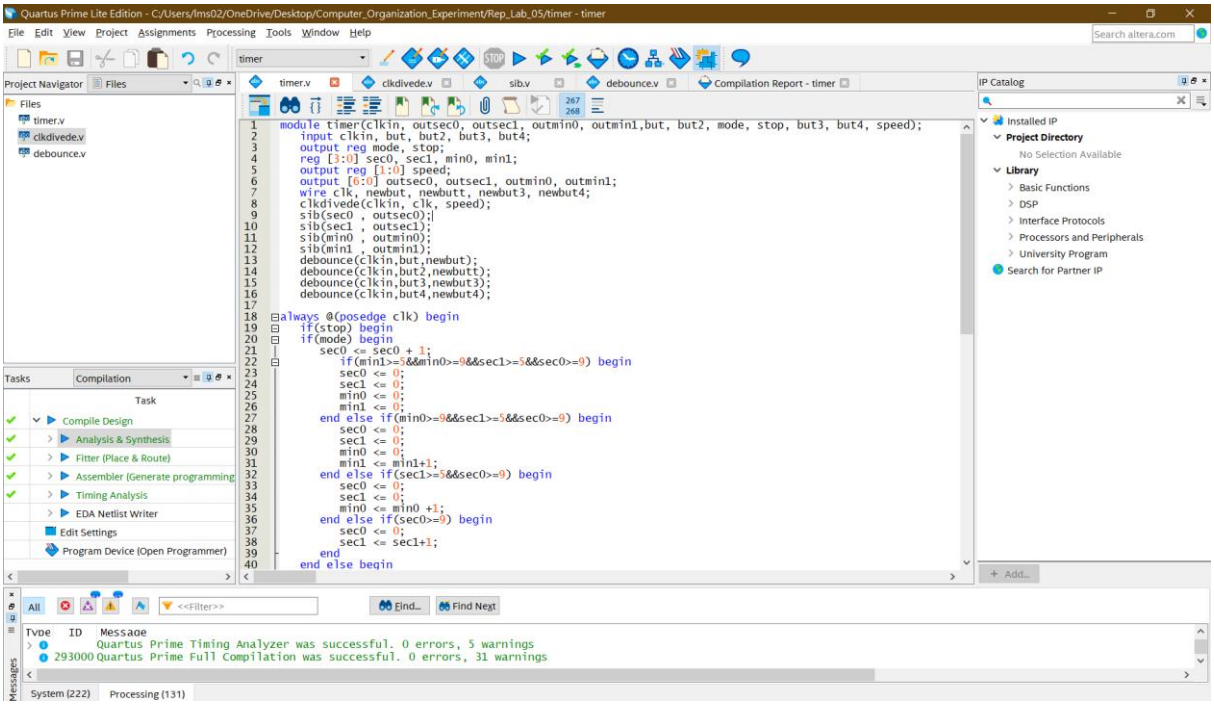
Lab 5 Sequence and Control; Digital clock design

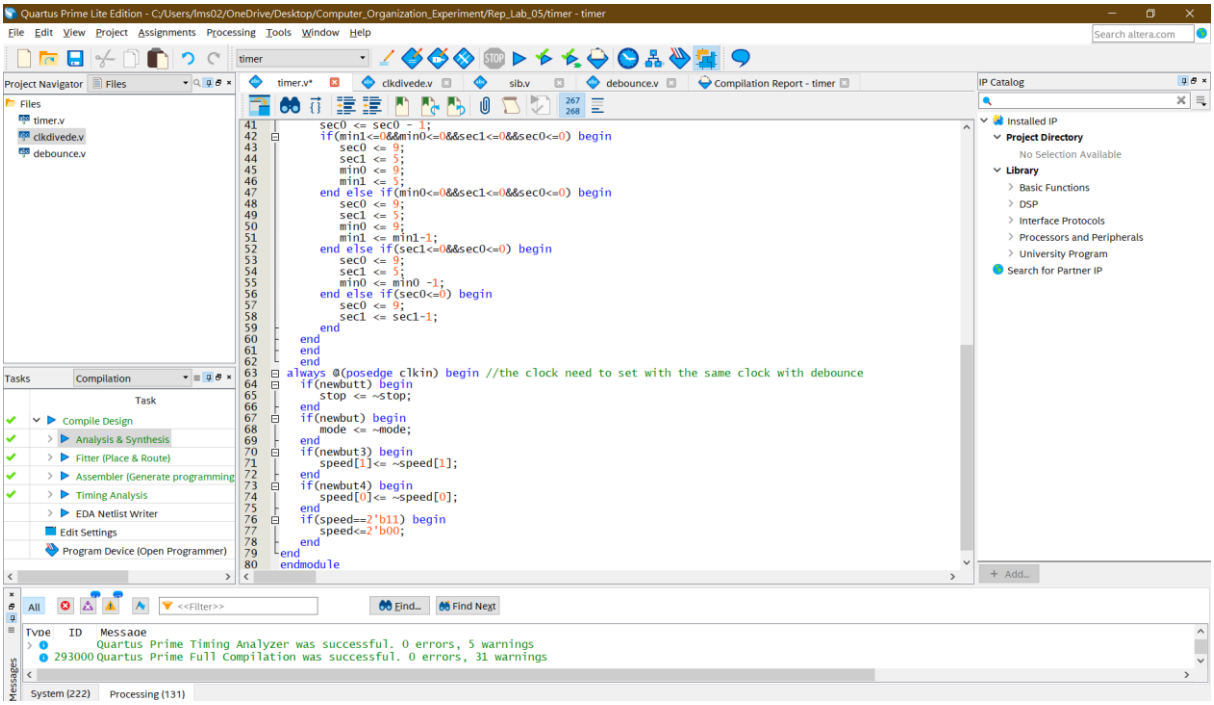
Description:

In this lab, we are going to make a timer, which has a lot of function to make it count faster, slower, count up, and count down. Furthermore, we also need a button to make the timer stop and start.

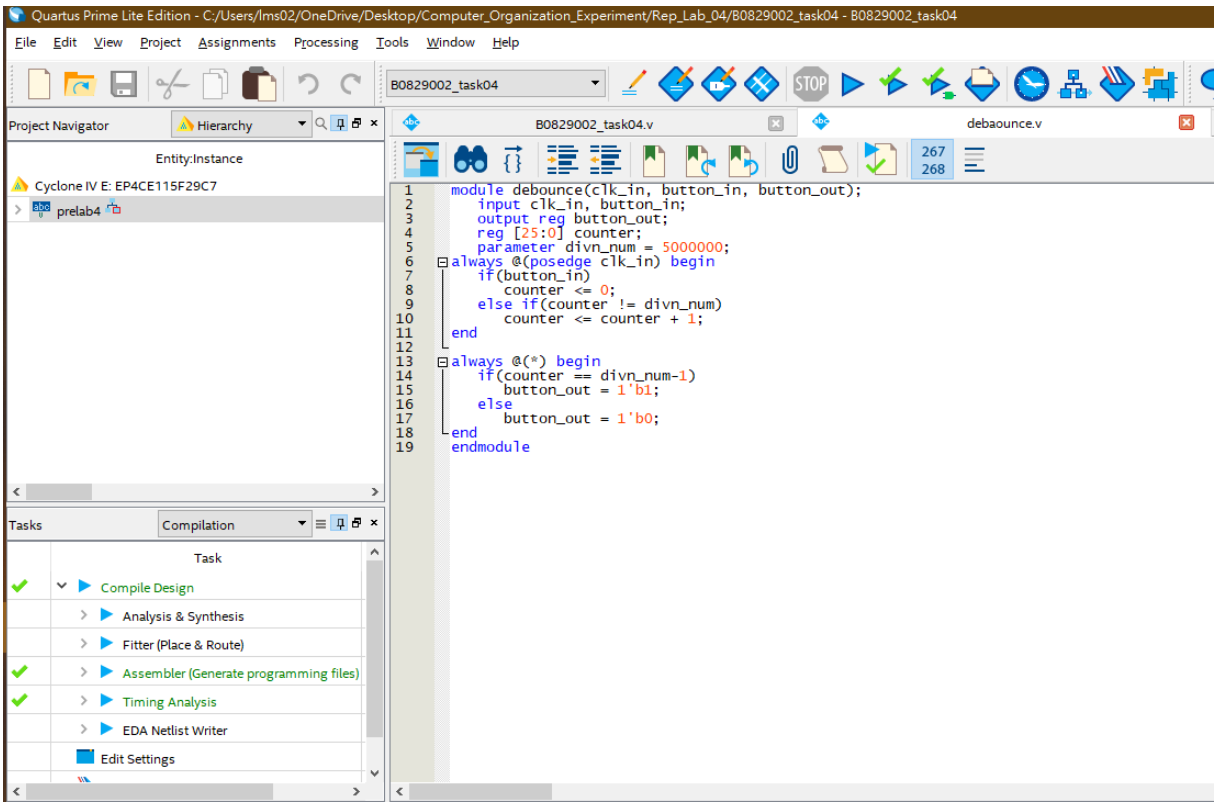
Code:

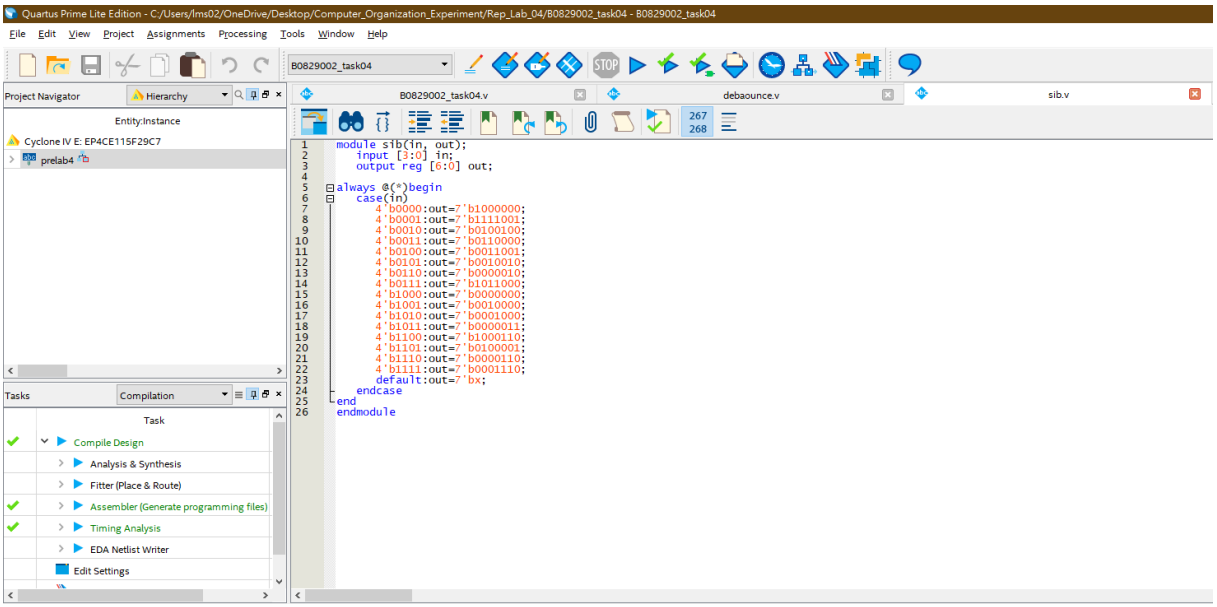


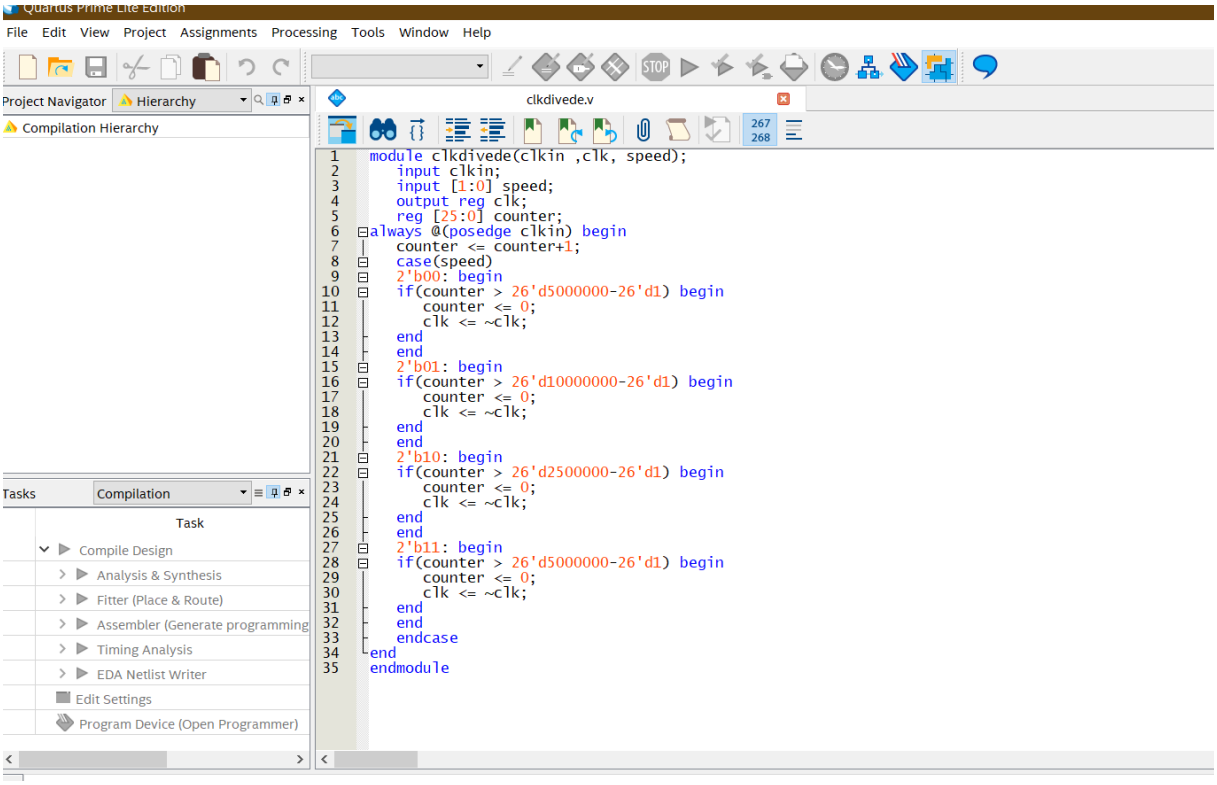
```
module timer(clkin, outsec0, outsec1, outmin0, outmin1,but, but2, mode, stop, but3, but4, speed);
   input clkin, but, but2, but3, but4;
   output reg mode, stop;
   reg [3:0] sec0, sec1, min0, min1;
   output reg [1:0] speed;
   output [6:0] outsec0, outsec1, outmin0, outmin1;
   wire clk, newbut, newbutt, newbut3, newbut4;
   clkdivede(clkin, clk, speed);
   sib(sec0 , outsec0);
   sib(sec1 , outsec1);
   sib(min0 , outmin0);
   sib(min1 , outmin1);
   debounce(clkin,but,newbut);
   debounce(clkin,but2,newbutt);
   debounce(clkin,but3,newbut3);
   debounce(clkin,but4,newbut4);

always @(posedge clk) begin
   if(stop) begin
   if(mode) begin
      sec0 <= sec0 + 1;
         if(min1>=5&&min0>=9&&sec1>=5&&sec0>=9) begin
            sec0 <= 0;
            sec1 <= 0;
            min0 <= 0;
            min1 <= 0;
         end else if(min0>=9&&sec1>=5&&sec0>=9) begin
            sec0 <= 0;
            sec1 <= 0;
            min0 <= 0;
            min1 <= min1+1;
         end else if(sec1>=5&&sec0>=9) begin
            sec0 <= 0;
            sec1 <= 0;
            min0 <= min0 +1;
         end else if(sec0>=9) begin
            sec0 <= 0;
            sec1 <= sec1+1;
      end
   end else begin
```



```
      sec0 <= sec0 - 1;
      if(min1<=0&&min0<=0&&sec1<=0&&sec0<=0) begin
         sec0 <= 9;
         sec1 <= 5;
         min0 <= 9;
         min1 <= 5;
      end else if(min0<=0&&sec1<=0&&sec0<=0) begin
         sec0 <= 9;
         sec1 <= 5;
         min0 <= 9;
         min1 <= min1-1;
      end else if(sec1<=0&&sec0<=0) begin
         sec0 <= 9;
         sec1 <= 5;
         min0 <= min0 -1;
      end else if(sec0<=0) begin
         sec0 <= 9;
         sec1 <= sec1-1;
      end
   end
   end
   end
always @(posedge clkin) begin //the clock need to set with the same clock with debounce
   if(newbutt) begin
      stop <= ~stop;
   end
   if(newbut) begin
      mode <= ~mode;
   end
   if(newbut3) begin
      speed[1]<= ~speed[1];
   end
   if(newbut4) begin
      speed[0]<= ~speed[0];
   end
   if(speed==2'b11) begin
      speed<=2'b00;
   end
end
endmodule
```


Quartus Prime Lite Edition - C:/Users/lms02/OneDrive/Desktop/Computer_Organization_Experiment/Rep_Lab_04/B0829002_task04 - B0829002_task04
File Edit View Project Assignments Processing Tools Window Help
B0829002_task04
Project Navigator
Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
prelab4
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Timing Analysis
EDA Netlist Writer
Edit Settings
B0829002_task04.v
debaounce.v
267
268
module debounce(clk_in, button_in, button_out);
input clk_in, button_in;
output reg button_out;
reg [25:0] counter;
parameter divn_num = 5000000;
always @(posedge clk_in) begin
if(button_in)
counter <= 0;
else if(counter != divn_num)
counter <= counter + 1;
end
always @(*) begin
if(counter == divn_num-1)
button_out = 1'b1;
else
button_out = 1'b0;
end
endmodule

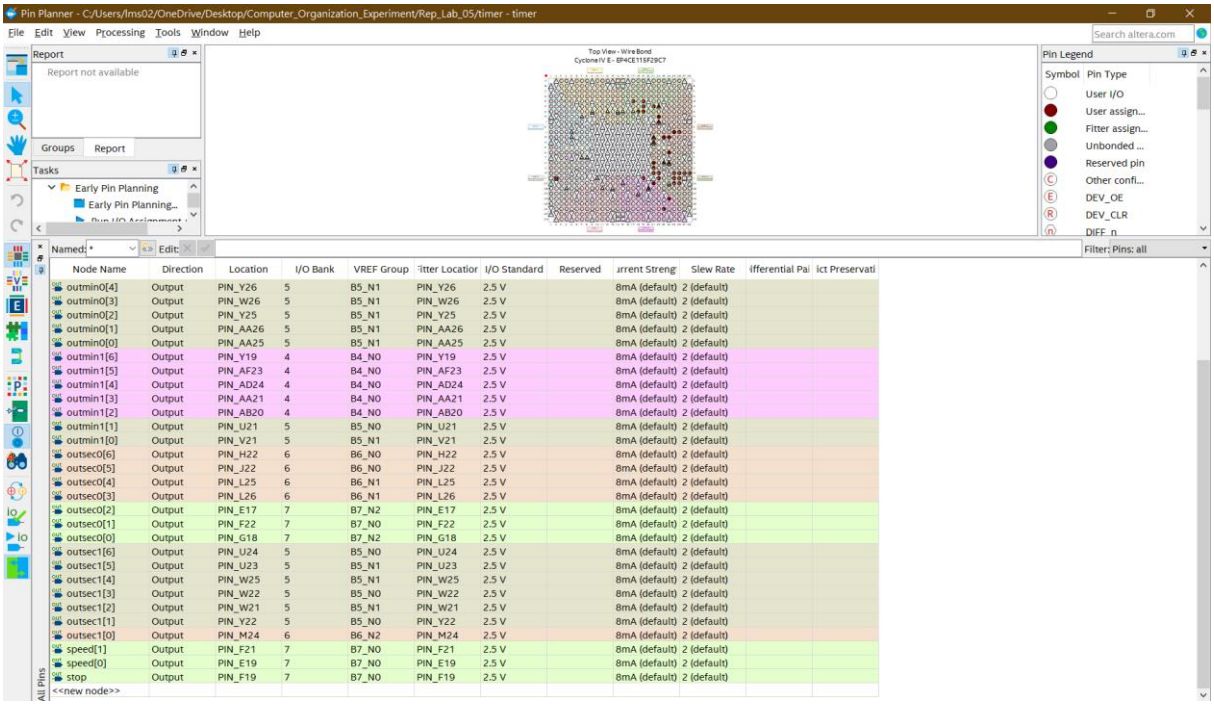

Quartus Prime Lite Edition - C:/Users/lms02/OneDrive/Desktop/Computer_Organization_Experiment/Rep_Lab_04/B0829002_task04 - B0829002_task04
File Edit View Project Assignments Processing Tools Window Help
B0829002_task04
Project Navigator
Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
prelab4
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Timing Analysis
EDA Netlist Writer
Edit Settings
B0829002_task04.v
debaounce.v
sib.v
267
268
module sib(in, out);
input [3:0] in;
output reg [6:0] out;
always @(*)begin
case(in)
4'b0000:out=7'b1000000;
4'b0001:out=7'b1111001;
4'b0010:out=7'b0100100;
4'b0011:out=7'b0110000;
4'b0100:out=7'b0011001;
4'b0101:out=7'b0010010;
4'b0110:out=7'b0000010;
4'b0111:out=7'b1011000;
4'b1000:out=7'b0000000;
4'b1001:out=7'b0010000;
4'b1010:out=7'b0001000;
4'b1011:out=7'b0000011;
4'b1100:out=7'b1000110;
4'b1101:out=7'b0100001;
4'b1110:out=7'b0000110;
4'b1111:out=7'b0001110;
default:out=7'bx;
endcase
end
endmodule



RTL Design:

RTL.pdf

Pin planer:



Explanation:

For count up, we just determine count to 9+1 or 5+1 need the back signal reset to zero and the next signal add 1. For counting down, we use another method to determine when it count to 0-1 and the back signal will reset to 9 or 5. And the front signal will minus 1.

If we need to use start and stop, it is easy to use control unit to achieve the action. We can use a if-else determination when stop==1, then the timer stop to count and stop ==0, the timer start counting.

Discussion for difficulties and problems:

Last but not least, when we face to the speed of counting, we can divide the clock trigger to achieve the faster 2x or 0.5x times (using n-bit counter or module-n counter), then we also using the control unit (if-else statement) to control the state of speed.

The most difficult problem I met is, when I using the different clock trigger for state determination and changing, it will happen that the trigger of clock for state changing and debounce.v is different, thus the state will not change. It also lead to most of control unit can not take action because when the clock trigger the state of button has changed and there is no clock trigger or different signal yet.

Therefore, the solution is using the same clock for all of the non-blocking function.