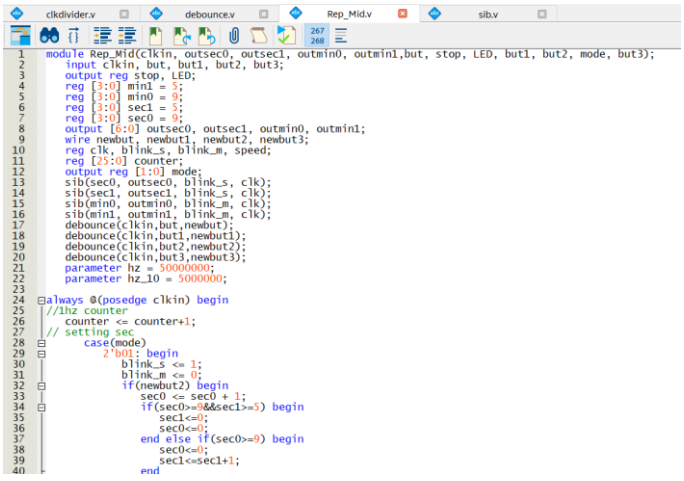
Rep_Mid_B0829002

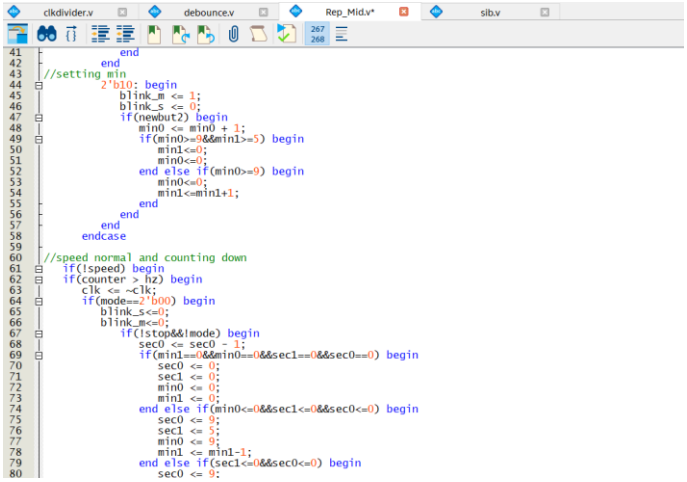
In this task, we are going to a count down counter with setting mode, blink, stop, and speed-up mode.

1. When the stop button was pressed, the counter will be stopped.
2. There are 2 button for setting clock, one is for change resume counting, setting second, setting minute. The other one is for increment the setting number, once pressed will add 1.
3. We can use speed-up button to make speed become 10x faster and pressed it again will become back to normal speed.
4. When setting sec, setting minute, and count to 00:00, this clock will blink.

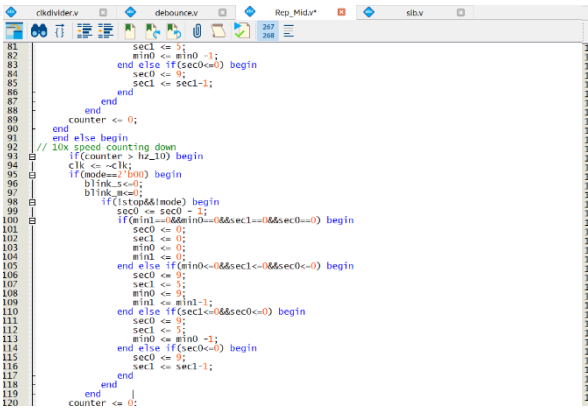
Code:



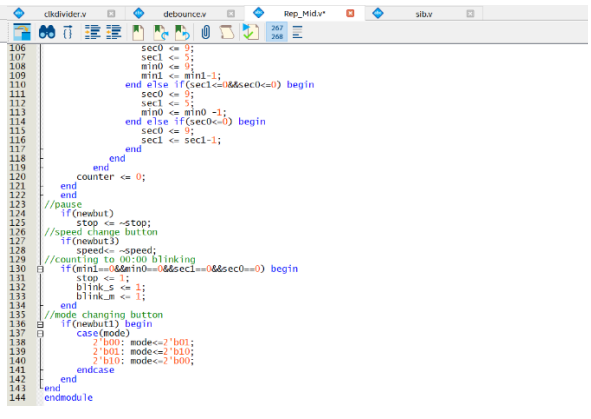
```
module Rep_Mid(clkin, outsec0, outsec1, outmin0, outmin1,but, stop, LED, but1, but2, mode, but3);
    input clkin, but, but1, but2, but3;
    output reg stop, LED;
    reg [3:0] min1 = 5;
    reg [3:0] min0 = 9;
    reg [3:0] sec1 = 5;
    reg [3:0] sec0 = 9;
    output [6:0] outsec0, outsec1, outmin0, outmin1;
    wire newbut, newbut1, newbut2, newbut3;
    reg clk, blink_s, blink_m, speed;
    reg [25:0] counter;
    output reg [1:0] mode;
    sib(sec0, outsec0, blink_s, clk);
    sib(sec1, outsec1, blink_s, clk);
    sib(min0, outmin0, blink_m, clk);
    sib(min1, outmin1, blink_m, clk);
    debounce(clkin,but,newbut);
    debounce(clkin,but1,newbut1);
    debounce(clkin,but2,newbut2);
    debounce(clkin,but3,newbut3);
    parameter hz = 50000000;
    parameter hz_10 = 5000000;

always @(posedge clkin) begin
 //1hz counter
    counter <= counter+1;
// setting sec
      case(mode)
        2'b01: begin
           blink_s <= 1;
           blink_m <= 0;
           if(newbut2) begin
              sec0 <= sec0 + 1;
              if(sec0>=9&&sec1>=5) begin
                 sec1<=0;
                 sec0<=0;
              end else if(sec0>=9) begin
                 sec0<=0;
                 sec1<=sec1+1;
              end
```



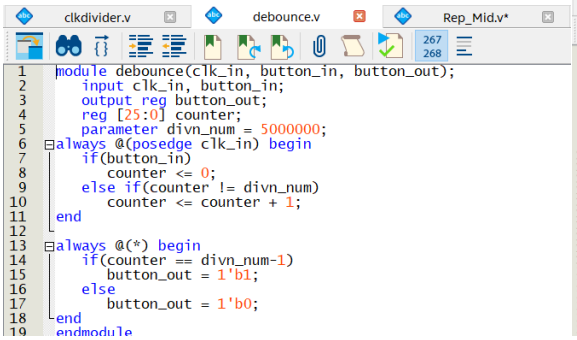
```
           end
        end
//setting min
        2'b10: begin
           blink_m <= 1;
           blink_s <= 0;
           if(newbut2) begin
              min0 <= min0 + 1;
              if(min0>=9&&min1>=5) begin
                 min1<=0;
                 min0<=0;
              end else if(min0>=9) begin
                 min0<=0;
                 min1<=min1+1;
              end
           end
        end
      endcase

//speed normal and counting down
  if(!speed) begin
  if(counter > hz) begin
     clk <= ~clk;
     if(mode==2'b00) begin
        blink_s<=0;
        blink_m<=0;
           if(!stop&&!mode) begin
              sec0 <= sec0 - 1;
              if(min1==0&&min0==0&&sec1==0&&sec0==0) begin
                 sec0 <= 0;
                 sec1 <= 0;
                 min0 <= 0;
                 min1 <= 0;
              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;
```

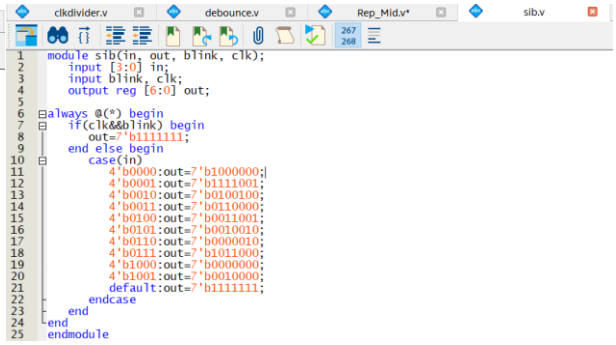


```
                 sec1 <= 5;
                 min0 <= min0 -1;
              end else if(sec0<=0) begin
                 sec0 <= 9;
                 sec1 <= sec1-1;
              end
           end
        end
        counter <= 0;
  end
  end else begin
// 10x speed counting down
     if(counter > hz_10) begin
     clk <= ~clk;
     if(mode==2'b00) begin
        blink_s<=0;
        blink_m<=0;
           if(!stop&&!mode) begin
              sec0 <= sec0 - 1;
              if(min1==0&&min0==0&&sec1==0&&sec0==0) begin
                 sec0 <= 0;
                 sec1 <= 0;
                 min0 <= 0;
                 min1 <= 0;
              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
        counter <= 0;
```



```
                 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
        counter <= 0;
  end
  end
//pause
    if(newbut)
       stop <= ~stop;
//speed change button
    if(newbut3)
       speed<= ~speed;
//counting to 00:00 blinking
  if(min1==0&&min0==0&&sec1==0&&sec0==0) begin
       stop <= 1;
       blink_s <= 1;
       blink_m <= 1;
  end
//mode changing button
  if(newbut1) begin
     case(mode)
        2'b00: mode<=2'b01;
        2'b01: mode<=2'b10;
        2'b10: mode<=2'b00;
     endcase
  end
end
endmodule
```





Difference:

In my opinion, the most difficult part is trying to combine the 2 different speed count-down counter with blinking function. Because we need to use the same clock to make sure debounce button can work successfully, thus we must to write a mux to select the parameter for different speed counter. The second hard module is blink mode. We can easily use the counter complement to make the seven segment blinking. And when the seven segment blinking, the signal will also be sometime all shine, sometimes shine the number.

Discussion:

After this task, I learned that we cannot modify value in different module and how we can make seven segment blinking. And because this count-down counter has a lot of different mode, we need to make the coding style clean and module working logic.