T.C.
YEDİTEPE UNIVERSITY

ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT

EE 241

Digital Circuits

Basys3 Ping-Pong Game

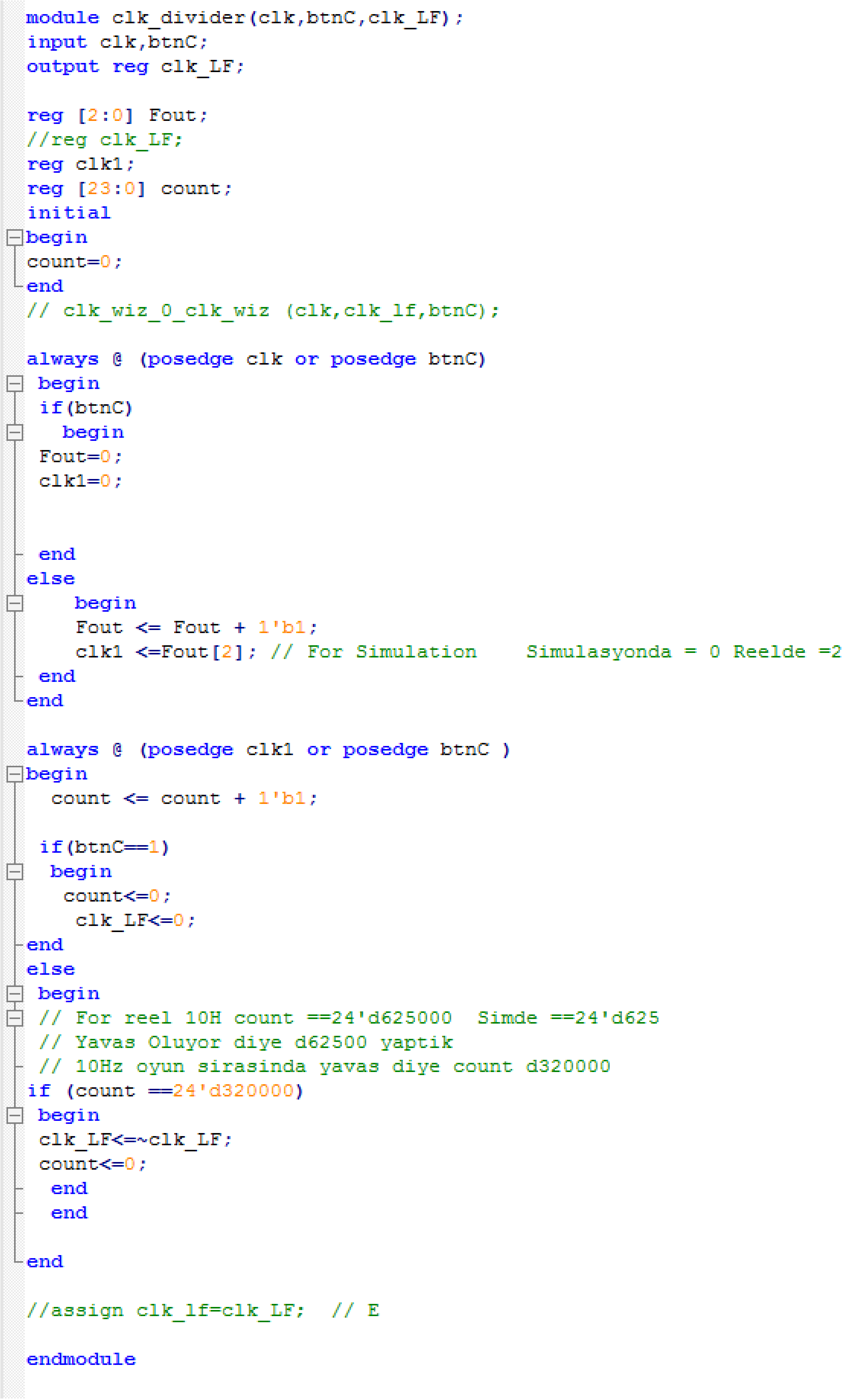
By

Mahmud Sami Ünlü

1) Modules of Game

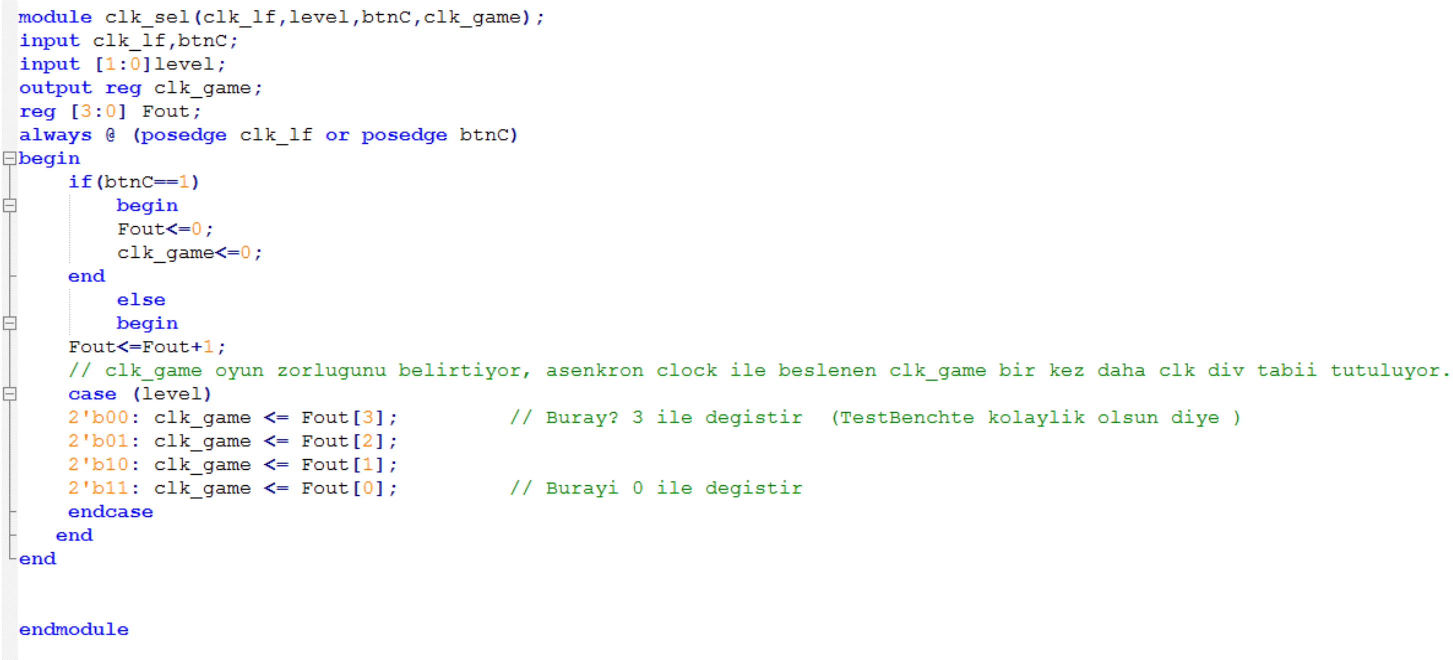
- **clk_divider.v** This module div to 100MHz clock To 10 Hz
- **clk_sel.v** This module select clock via game difficulty level.
- **lamp_handball.v** This module calculate score, simultaneously ball working prprincipal.
- **updown_counter.v** This module evulate game difficulty using BTN_U and BTN_D.
- **tb_for_all.v** This module test the topmodule.
- **edge_detector.v** This module get hold of the input rasing side.
- **seg7_decoder.v** This module driver for 7 segment display.
- **topmodule.v** This module include all function and work together.

clk_ divider.v



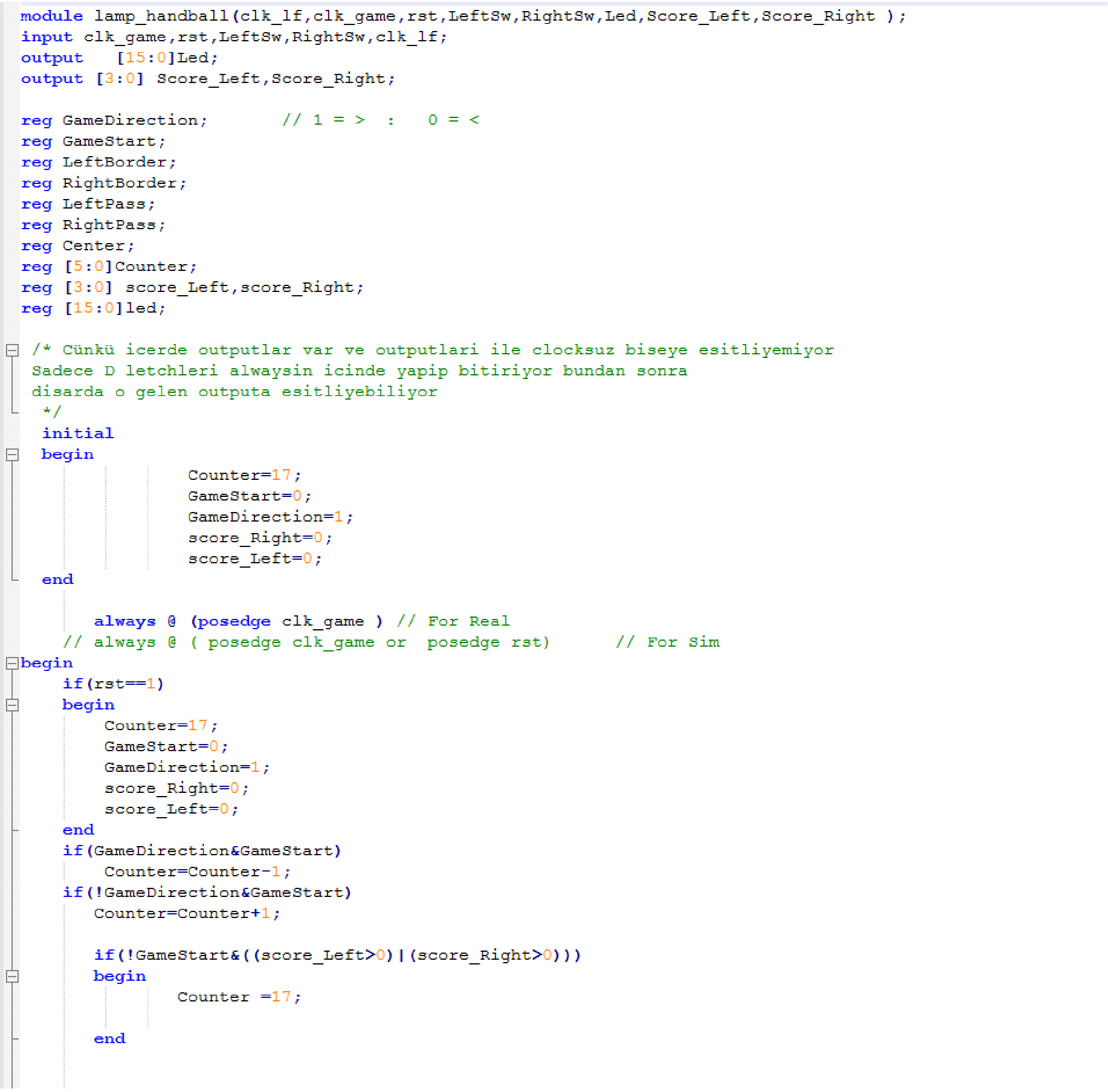
```
module clk_divider(clk,btnC,clk_LF);
input clk,btnC;
output reg clk_LF;

reg [2:0] Fout;
//reg clk_LF;
reg clk1;
reg [23:0] count;
initial
begin
count=0;
end
// clk_wiz_0_clk_wiz (clk,clk_lf,btnC);

always @ (posedge clk or posedge btnC)
 begin
 if(btnC)
   begin
 Fout=0;
 clk1=0;


 end
else
    begin
    Fout <= Fout + 1'b1;
    clk1 <=Fout[2]; // For Simulation    Simulasyonda = 0 Reelde =2
 end
end

always @ (posedge clk1 or posedge btnC )
begin
  count <= count + 1'b1;

 if(btnC==1)
  begin
   count<=0;
    clk_LF<=0;
end
else
 begin
 // For reel 10H count ==24'd625000  Simde ==24'd625
 // Yavas Oluyor diye d62500 yaptik
 // 10Hz oyun sirasinda yavas diye count d320000
if (count ==24'd320000)
 begin
 clk_LF<=~clk_LF;
 count<=0;
  end
  end

end

//assign clk_lf=clk_LF;  // E

endmodule
```

clk_sel.v

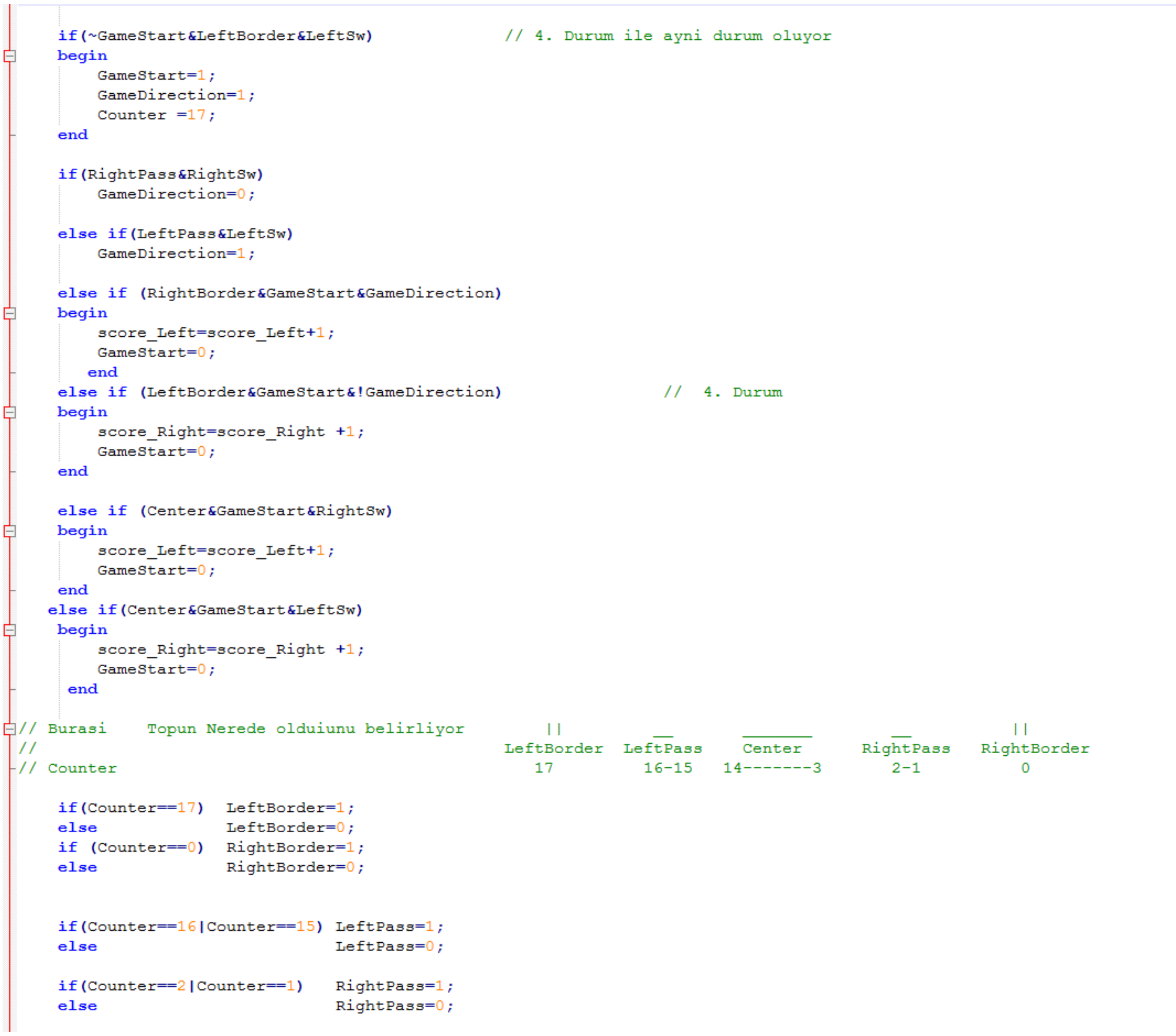


```verilog
module clk_sel(clk_lf,level,btnC,clk_game);
input clk_lf,btnC;
input [1:0]level;
output reg clk_game;
reg [3:0] Fout;
always @ (posedge clk_lf or posedge btnC)
begin
    if(btnC==1)
        begin
        Fout<=0;
        clk_game<=0;
    end
        else
        begin
    Fout<=Fout+1;
    // clk_game oyun zorlugunu belirtiyor, asenkron clock ile beslenen clk_game bir kez daha clk div tabii tutuluyor.
    case (level)
    2'b00: clk_game <= Fout[3];         // Buray? 3 ile degistir  (TestBenchte kolaylik olsun diye )
    2'b01: clk_game <= Fout[2];
    2'b10: clk_game <= Fout[1];
    2'b11: clk_game <= Fout[0];         // Burayi 0 ile degistir
    endcase
   end
end


endmodule
```

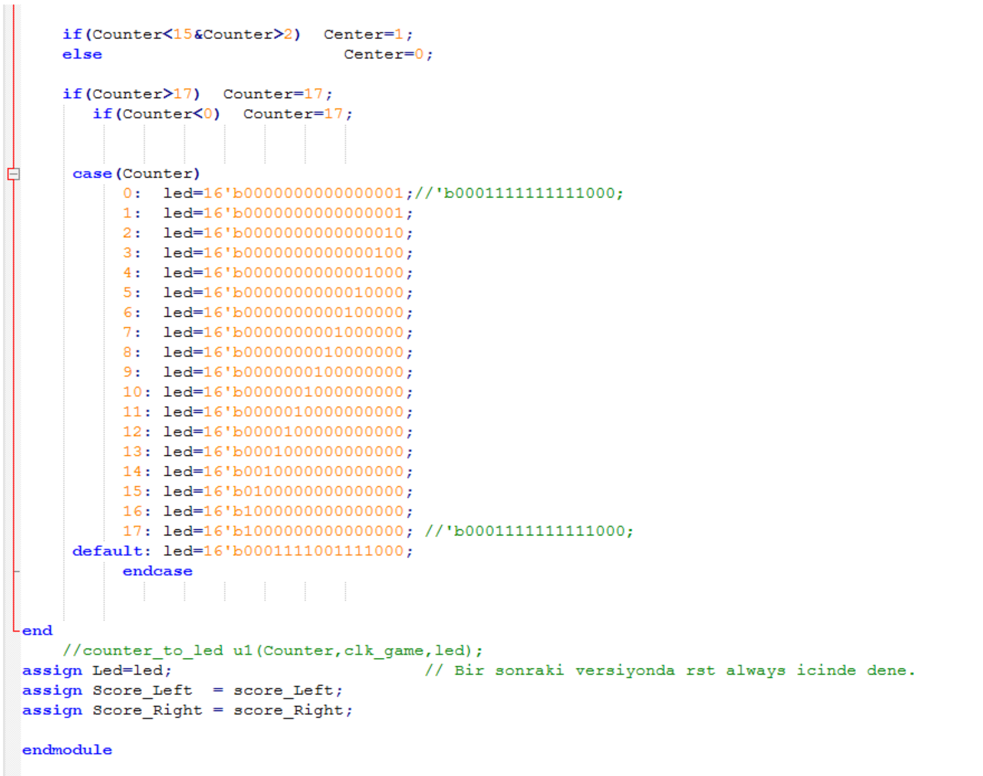
lamp_handball.v



```
module lamp_handball(clk_lf,clk_game,rst,LeftSw,RightSw,Led,Score_Left,Score_Right );
input clk_game,rst,LeftSw,RightSw,clk_lf;
output   [15:0]Led;
output [3:0] Score_Left,Score_Right;

reg GameDirection;       // 1 = >  :   0 = <
reg GameStart;
reg LeftBorder;
reg RightBorder;
reg LeftPass;
reg RightPass;
reg Center;
reg [5:0]Counter;
reg [3:0] score_Left,score_Right;
reg [15:0]led;

 /* Cünkü icerde outputlar var ve outputlari ile clocksuz biseye esitliyemiyor
 Sadece D letchleri alwaysin icinde yapip bitiriyor bundan sonra
 disarda o gelen outputa esitliyebiliyor
  */
  initial
  begin
                Counter=17;
                GameStart=0;
                GameDirection=1;
                score_Right=0;
                score_Left=0;
  end

       always @ (posedge clk_game ) // For Real
    // always @ ( posedge clk_game or  posedge rst)      // For Sim
begin
    if(rst==1)
    begin
        Counter=17;
        GameStart=0;
        GameDirection=1;
        score_Right=0;
        score_Left=0;
    end
    if(GameDirection&GameStart)
        Counter=Counter-1;
    if(!GameDirection&GameStart)
       Counter=Counter+1;

       if(!GameStart&((score_Left>0)|(score_Right>0)))
       begin
               Counter =17;

       end
```

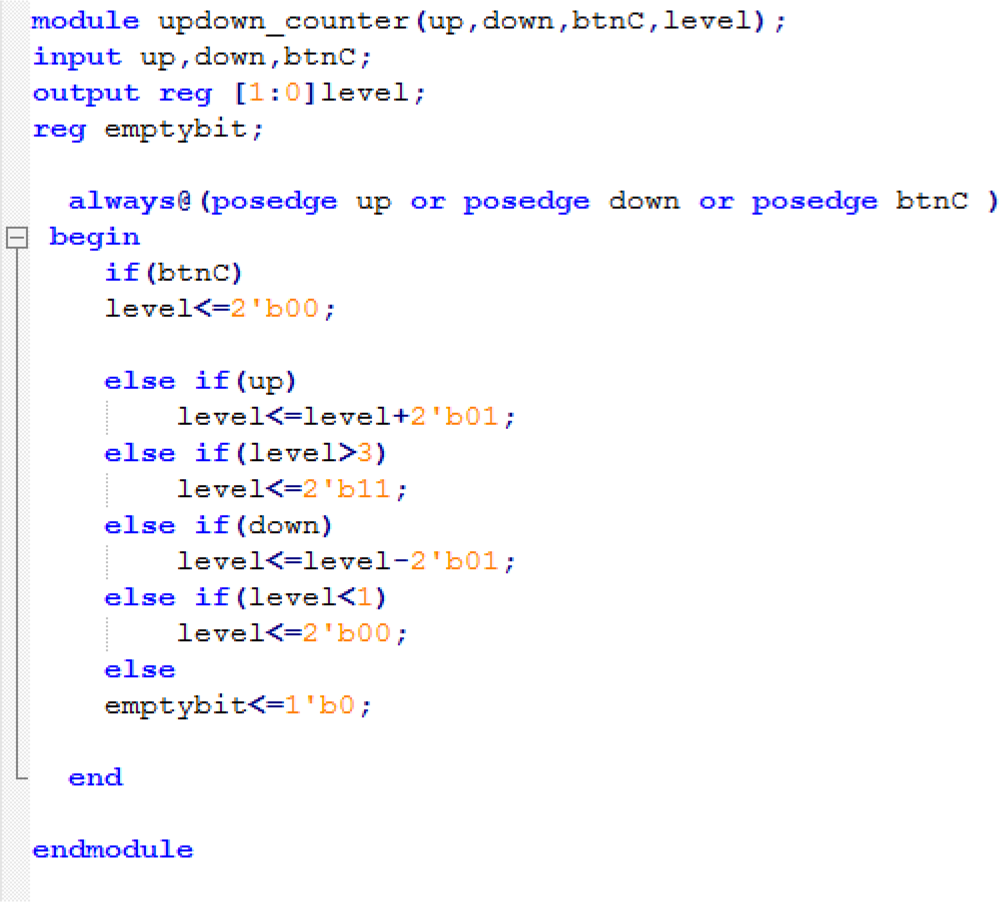


```
    if(~GameStart&LeftBorder&LeftSw)             // 4. Durum ile ayni durum oluyor
    begin
        GameStart=1;
        GameDirection=1;
        Counter =17;
    end

    if(RightPass&RightSw)
        GameDirection=0;

    else if(LeftPass&LeftSw)
        GameDirection=1;

    else if (RightBorder&GameStart&GameDirection)
    begin
        score_Left=score_Left+1;
        GameStart=0;
       end
    else if (LeftBorder&GameStart&!GameDirection)                // 4. Durum
    begin
        score_Right=score_Right +1;
        GameStart=0;
    end

    else if (Center&GameStart&RightSw)
    begin
        score_Left=score_Left+1;
        GameStart=0;
    end
   else if(Center&GameStart&LeftSw)
    begin
        score_Right=score_Right +1;
        GameStart=0;
     end
// Burasi    Topun Nerede olduiunu belirliyor        ||         __       _______        __          ||
//                                               LeftBorder  LeftPass    Center      RightPass   RightBorder
// Counter                                          17         16-15   14-------3       2-1          0

    if(Counter==17)  LeftBorder=1;
    else             LeftBorder=0;
    if (Counter==0)  RightBorder=1;
    else             RightBorder=0;


    if(Counter==16|Counter==15) LeftPass=1;
    else                        LeftPass=0;

    if(Counter==2|Counter==1)   RightPass=1;
    else                        RightPass=0;
```



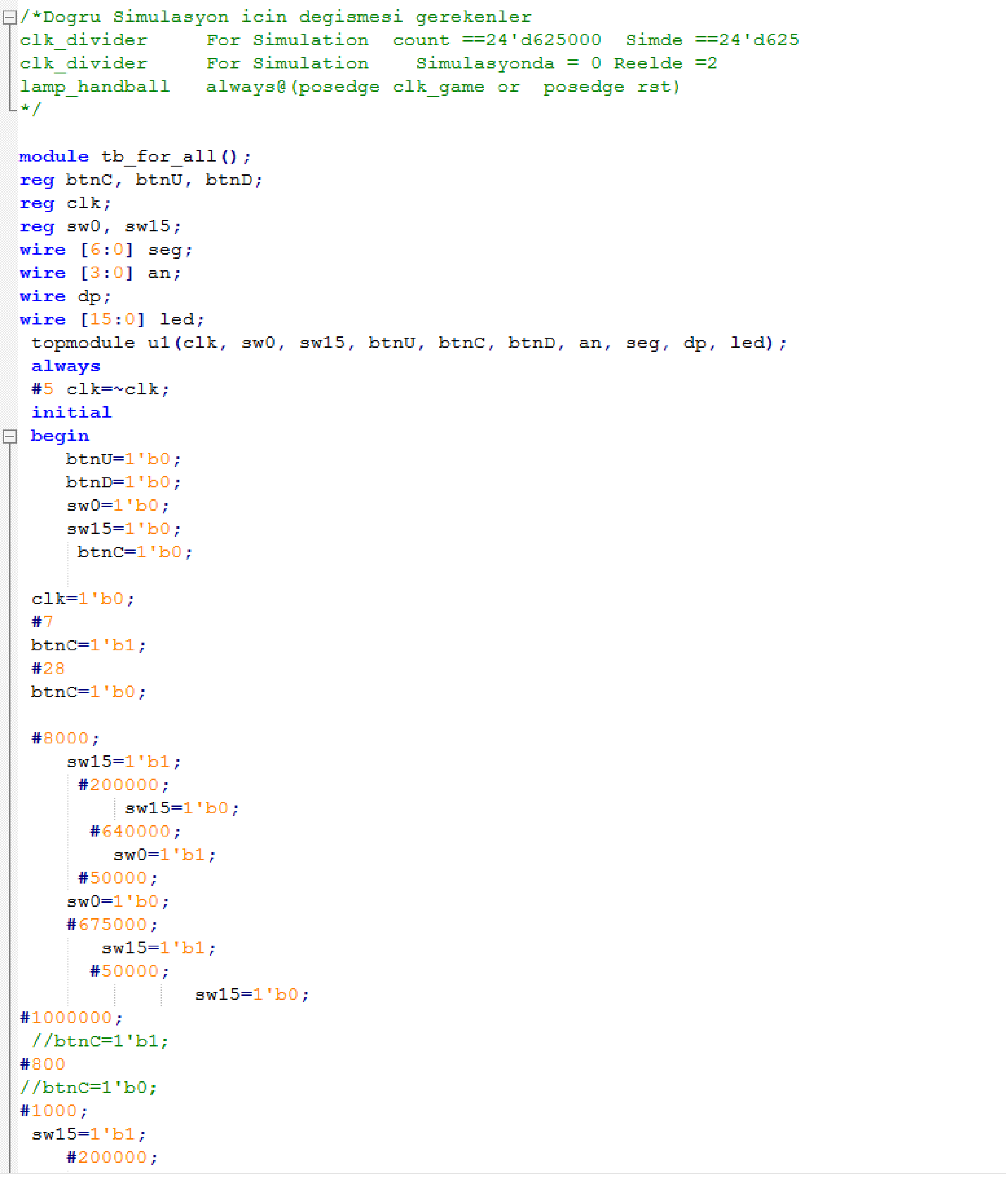
```
    if(Counter<15&Counter>2)  Center=1;
    else                        Center=0;

    if(Counter>17)  Counter=17;
       if(Counter<0)  Counter=17;


     case(Counter)
          0:  led=16'b0000000000000001;//'b0001111111111000;
          1:  led=16'b0000000000000001;
          2:  led=16'b0000000000000010;
          3:  led=16'b0000000000000100;
          4:  led=16'b0000000000001000;
          5:  led=16'b0000000000010000;
          6:  led=16'b0000000000100000;
          7:  led=16'b0000000001000000;
          8:  led=16'b0000000010000000;
          9:  led=16'b0000000100000000;
          10: led=16'b0000001000000000;
          11: led=16'b0000010000000000;
          12: led=16'b0000100000000000;
          13: led=16'b0001000000000000;
          14: led=16'b0010000000000000;
          15: led=16'b0100000000000000;
          16: led=16'b1000000000000000;
          17: led=16'b1000000000000000; //'b0001111111111000;
     default: led=16'b0001111001111000;
          endcase


end
    //counter_to_led u1(Counter,clk_game,led);
assign Led=led;                          // Bir sonraki versiyonda rst always icinde dene.
assign Score_Left  = score_Left;
assign Score_Right = score_Right;

endmodule
```

updown_counter.v

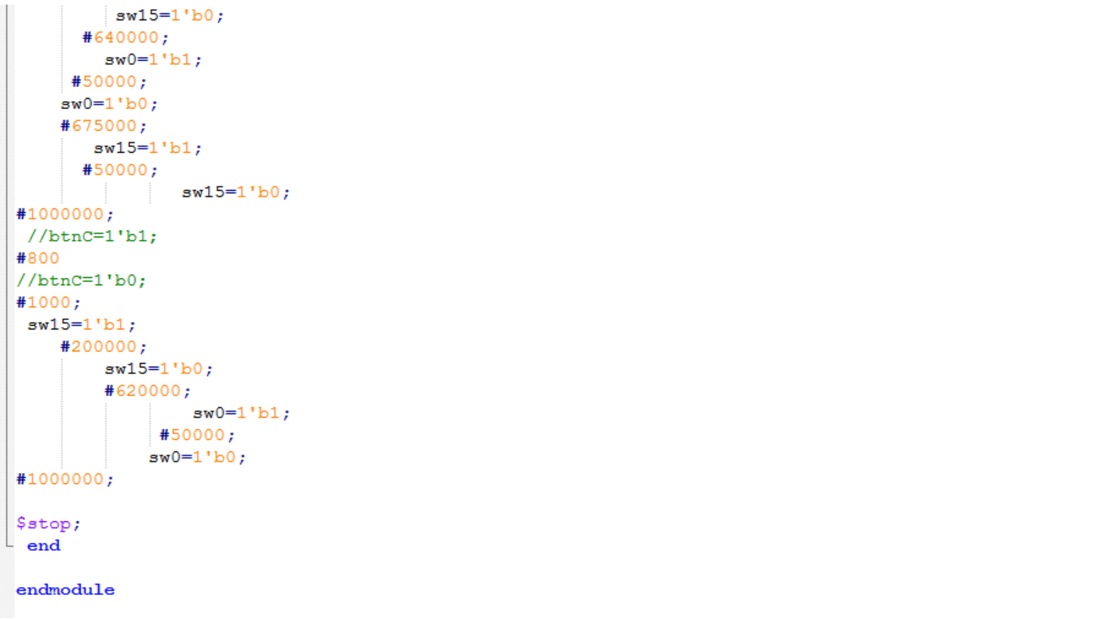


```
module updown_counter(up,down,btnC,level);
input up,down,btnC;
output reg [1:0]level;
reg emptybit;

  always@(posedge up or posedge down or posedge btnC )
 begin
    if(btnC)
    level<=2'b00;

    else if(up)
        level<=level+2'b01;
    else if(level>3)
        level<=2'b11;
    else if(down)
        level<=level-2'b01;
    else if(level<1)
        level<=2'b00;
    else
    emptybit<=1'b0;

  end

endmodule
```

tb_for_all.v



```
/*Dogru Simulasyon icin degismesi gerekenler
clk_divider      For Simulation  count ==24'd625000  Simde ==24'd625
clk_divider      For Simulation    Simulasyonda = 0 Reelde =2
lamp_handball    always@(posedge clk_game or  posedge rst)
*/

module tb_for_all();
reg btnC, btnU, btnD;
reg clk;
reg sw0, sw15;
wire [6:0] seg;
wire [3:0] an;
wire dp;
wire [15:0] led;
 topmodule u1(clk, sw0, sw15, btnU, btnC, btnD, an, seg, dp, led);
 always
 #5 clk=~clk;
 initial
 begin
    btnU=1'b0;
    btnD=1'b0;
    sw0=1'b0;
    sw15=1'b0;
     btnC=1'b0;

 clk=1'b0;
 #7
 btnC=1'b1;
 #28
 btnC=1'b0;

 #8000;
    sw15=1'b1;
     #200000;
         sw15=1'b0;
      #640000;
        sw0=1'b1;
     #50000;
    sw0=1'b0;
    #675000;
       sw15=1'b1;
      #50000;
               sw15=1'b0;
#1000000;
 //btnC=1'b1;
#800
//btnC=1'b0;
#1000;
 sw15=1'b1;
    #200000;
```



```
         sw15=1'b0;
     #640000;
        sw0=1'b1;
    #50000;
   sw0=1'b0;
   #675000;
      sw15=1'b1;
     #50000;
              sw15=1'b0;
#1000000;
 //btnC=1'b1;
#800
//btnC=1'b0;
#1000;
 sw15=1'b1;
   #200000;
       sw15=1'b0;
       #620000;
               sw0=1'b1;
            #50000;
           sw0=1'b0;
#1000000;

$stop;
 end

endmodule
```