Clock Gen for 1 sec

```
if(rst)                                 //rst used to RESET time to 00:00:00

clk<= 1’b0;

else if(cnt == 32’d49999999)

begin

clk<= ~clk;

cnt<= 32’d0;

end

else

cnt<= cnt+1’b1;
```

ALARM TIME SET

```
//For incrementing hour :

if(inc_hr==1’b1)//If inc_hr push button (M18) is pressed

begin

if(outh==6’d24)

alarmh<=6’d0; //alarmh are alarm registers to stores the  alarm values

else

alarmh<=alarmh+1’b1;

end

//B. For incrementing minute :

if(inc_min==1’b1)// If inc_min push button (P17) is pressed

begin

if(outm==6’d60)
```

```
alarmm<=6’d0; //alarmm are alarm registers to stores the  alarm values
else
alarmm<=alarmm+1’b1;
end
//C. For decrementing hour:
if(dec_hr==1’b1)// If dec_hr push button (M17)is pressed
begin
if(outh==6’d24)
alarmh<=6’d0;
else
alarmh<=alarmh-1’b1;
end
//D. For decrementing minute :
if(dec_min==1’b1)// If dec_min push button (P18)is pressed
begin
if(outm==6’d60)
alarmm<=6’d0;
else
alarmm<=alarmm-1’b1;
end
CURRENT TIME SET
//A. For incrementing hour :
if(inc_hr==1’b1)// If inc_hr push button (M18) is pressed
```

```
begin

if(outh==6’d24)

outh<=6’d0; // outh are registers to stores the current time values

else

outh<=outh+1’b1;

end

//B. For incrementing minute :

if(inc_min==1’b1) // If inc_min push button (P17) is pressed

begin

if(outm==6’d60)

outm<=6’d0; // outm are registers to stores the current time values

else

outm<=outm+1’b1;

end

//C. For decrementing hour :

if(dec_hr==1’b1)// If dec_hr push button (M17)is pressed

begin

if(outh==6’d24)

outh<=6’d0;

else

outh<=outh-1’b1;

end

//For decrementing minute :
```

```
if(dec_min==1’b1)// If dec_min push button (P18)is pressed

begin

if(outm==6’d60)

outm<=6’d0;

else

outm<= outm-1’b1;

end
```

REAL TIME

```
if(outs!=6’d59)

outs<=outs+1’b1;

else

begin

outs<=6’d0;

outm<=outm+1’b1;

end

if(outm==6’d59)

begin

outm<=6’d0;

outh<=outh+1’b1;

end

if(outh==6’d24)

outh<=6’d0;

end
```

ALARM ON

```
always @ (*)

begin

alarm_out<= 1’b0;

if ((alarmh == outh) && (alarmm == outm) && (alarm_ON == 1))

alarm_out<= 1’b1;

end

if(rst)

begin

Disp_Val<= Zero;

Disp_Seg<= 8’d0;

end

if(CntRec == 16’d10922)

begin

if(!alarm_set) begin

Disp_Val<= outsegs1;

Disp_Seg<= 8’b1111_1110; end

else begin

Disp_Val<= outsegs1_a;

Disp_Seg<= 8’b1111_1110; end

end

if(CntRec == 16’d21844)

begin
```

```
if(!alarm_set) begin

Disp_Val<= outsegs2;

Disp_Seg<= 8’b1111_1101; end

else begin

Disp_Val<= outsegs2_a;

Disp_Seg<= 8’b1111_1101; end

end

if(CntRec == 16’d32766)

begin

if(!alarm_set) begin

Disp_Val<= outsegm1;

Disp_Seg<= 8’b1111_1011; end

else begin

Disp_Val<= outsegm1_a;

Disp_Seg<= 8’b1111_1011; end

end

if(CntRec == 16’d43688)

begin

if(!alarm_set) begin

Disp_Val<= outsegm2;

Disp_Seg<= 8’b1111_0111; end

else begin

Disp_Val<= outsegm2_a;
```

```
Disp_Seg<= 8'b1111_0111; end

end

if(CntRec == 16'd54610)

begin

if(!alarm_set) begin

Disp_Val<= outsegh1;

Disp_Seg<= 8'b1110_1111; end

else begin

Disp_Val<= outsegh1_a;

Disp_Seg<= 8'b1110_1111; end

end

if(CntRec == 16'd65532)

begin

if(!alarm_set) begin

Disp_Val<= outsegh2;

Disp_Seg<= 8'b1101_1111; end

else begin

Disp_Val<= outsegh2_a;

Disp_Seg<= 8'b1101_1111; end

End
```

SEVEN SEG DISPLAY

```
input [3:0] bcd;

output [7:0] outseg;
```

```
case(bcd)

5’h00: outseg<= 8’b00000011;

5’h01: outseg<= 8’b10011111;

5’h02: outseg<= 8’b00100101;

5’h03: outseg<= 8’b00001101;

5’h04: outseg<= 8’b10011001;

5’h05: outseg<= 8’b01001001;

5’h06: outseg<= 8’b01000001;

5’h07: outseg<= 8’b00011111;

5’h08: outseg<= 8’b00000001;

5’h09: outseg<= 8’b00001001;

default :outseg<= 8’b00000010;

Endcase

input [5:0] bin;

output [3:0] bcd1;

output [3:0] bcd0;

case (bin)

6’d0 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0000; end

6’d1 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0001; end

6’d2 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0010; end

6’d3 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0011; end

6’d4 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0100; end

6’d5 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0101; end
```

```
6’d6 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0110; end

6’d7 : begin bcd1 <= 4’b0000; bcd0 <= 4’b0111; end

6’d8 : begin bcd1 <= 4’b0000; bcd0 <= 4’b1000; end

6’d9 : begin bcd1 <= 4’b0000; bcd0 <= 4’b1001; end

6’d10 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0000; end

6’d11 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0001; end

6’d12 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0010; end

6’d13 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0011; end

6’d14 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0100; end

6’d15 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0101; end

6’d16 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0110; end

6’d17 : begin bcd1 <= 4’b0001; bcd0 <= 4’b0111; end

6’d18 : begin bcd1 <= 4’b0001; bcd0 <= 4’b1000; end

6’d19 : begin bcd1 <= 4’b0001; bcd0 <= 4’b1001; end

6’d20 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0000; end

6’d21 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0001; end

6’d22 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0010; end

6’d23 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0011; end

6’d24 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0100; end

6’d25 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0101; end

6’d26 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0110; end

6’d27 : begin bcd1 <= 4’b0010; bcd0 <= 4’b0111; end

6’d28 : begin bcd1 <= 4’b0010; bcd0 <= 4’b1000; end
```

```
6’d29 : begin bcd1 <= 4’b0010; bcd0 <= 4’b1001; end

6’d30 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0000; end

6’d31 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0001; end

6’d32 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0010; end

6’d33 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0011; end

6’d34 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0100; end

6’d35 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0101; end

6’d36 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0110; end

6’d37 : begin bcd1 <= 4’b0011; bcd0 <= 4’b0111; end

6’d38 : begin bcd1 <= 4’b0011; bcd0 <= 4’b1000; end

6’d39 : begin bcd1 <= 4’b0011; bcd0 <= 4’b1001; end

6’d40 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0000; end

6’d41 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0001; end

6’d42 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0010; end

6’d43 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0011; end

6’d44 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0100; end

6’d45 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0101; end

6’d46 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0110; end

6’d47 : begin bcd1 <= 4’b0100; bcd0 <= 4’b0111; end

6’d48 : begin bcd1 <= 4’b0100; bcd0 <= 4’b1000; end

6’d49 : begin bcd1 <= 4’b0100; bcd0 <= 4’b1001; end

6’d50 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0000; end

6’d51 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0001; end
```

```
6’d52 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0010; end

6’d53 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0011; end

6’d54 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0100; end

6’d55 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0101; end

6’d56 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0110; end

6’d57 : begin bcd1 <= 4’b0101; bcd0 <= 4’b0111; end

6’d58 : begin bcd1 <= 4’b0101; bcd0 <= 4’b1000; end

6’d59 : begin bcd1 <= 4’b0101; bcd0 <= 4’b1001; end

6’d60 : begin bcd1 <= 4’b0110; bcd0 <= 4’b0000; end

default:begin bcd1 <= 4’b0000; bcd0 <= 4’b0000; end

endcase
```