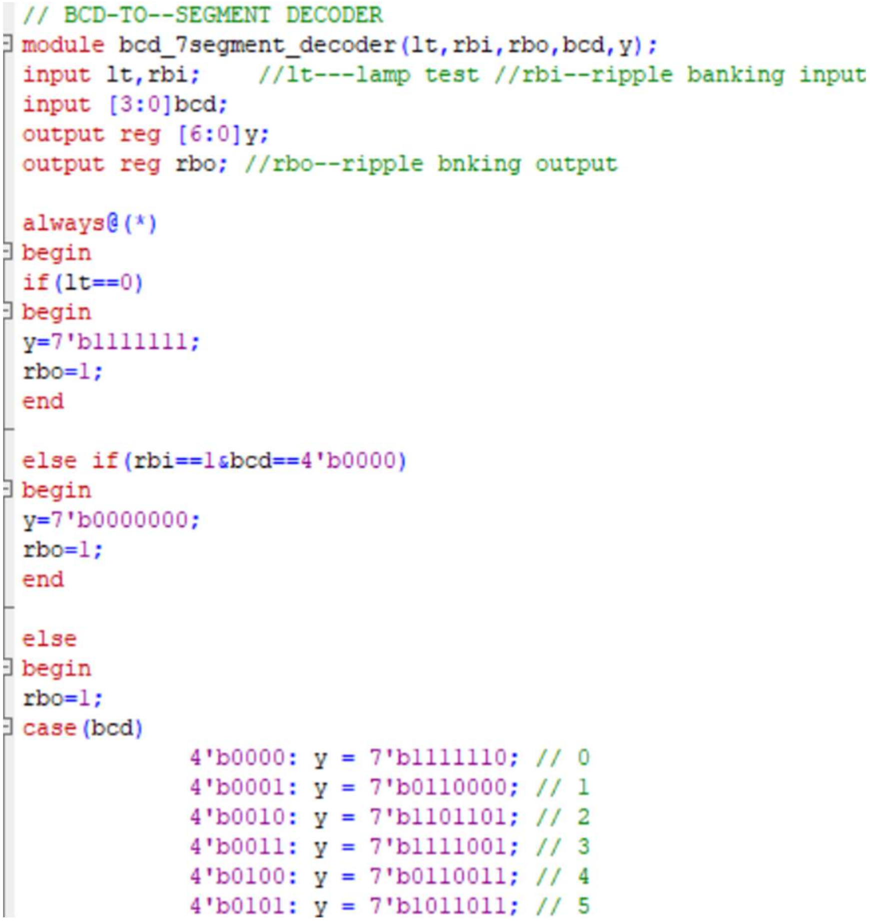
#50DAYS CHALLENGE

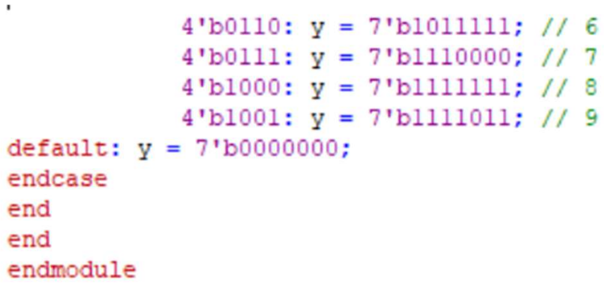
DAY 11

BCD TO SEGMENT DECODER DISPLAY

a)Design Code

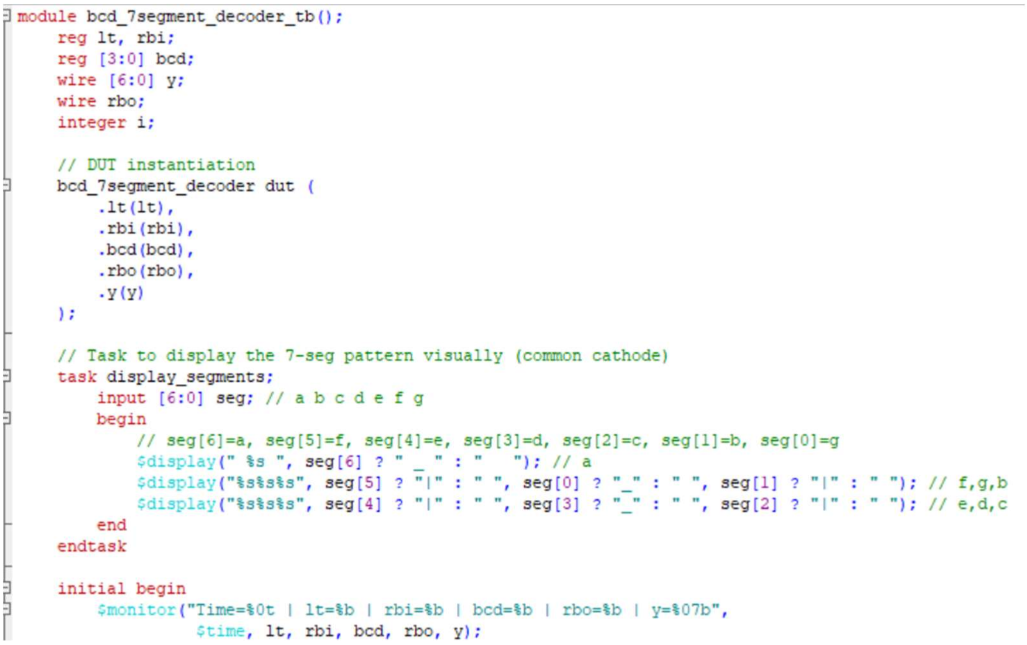


```
// BCD-TO--SEGMENT DECODER
module bcd_7segment_decoder(lt,rbi,rbo,bcd,y);
input lt,rbi;    //lt---lamp test //rbi--ripple banking input
input [3:0]bcd;
output reg [6:0]y;
output reg rbo; //rbo--ripple bnking output

always@(*)
begin
if(lt==0)
begin
y=7'b1111111;
rbo=1;
end

else if(rbi==1&bcd==4'b0000)
begin
y=7'b0000000;
rbo=1;
end

else
begin
rbo=1;
case(bcd)
            4'b0000: y = 7'b1111110; // 0
            4'b0001: y = 7'b0110000; // 1
            4'b0010: y = 7'b1101101; // 2
            4'b0011: y = 7'b1111001; // 3
            4'b0100: y = 7'b0110011; // 4
            4'b0101: y = 7'b1011011; // 5
```

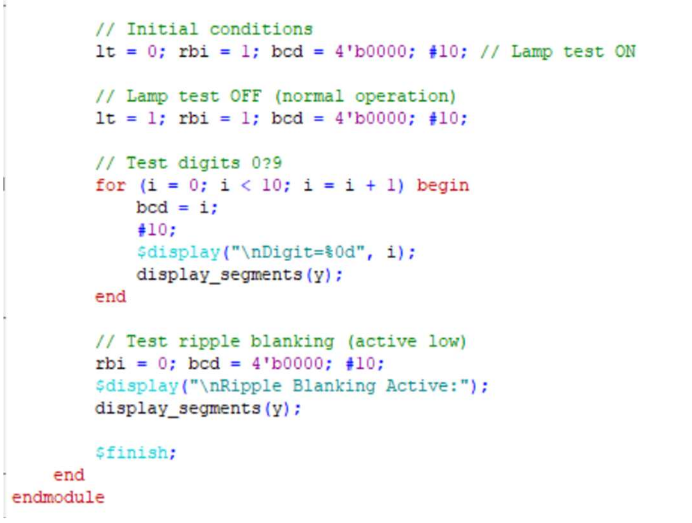


```
            4'b0110: y = 7'b1011111; // 6
            4'b0111: y = 7'b1110000; // 7
            4'b1000: y = 7'b1111111; // 8
            4'b1001: y = 7'b1111011; // 9
default: y = 7'b0000000;
endcase
end
end
endmodule
```

b)Testbench

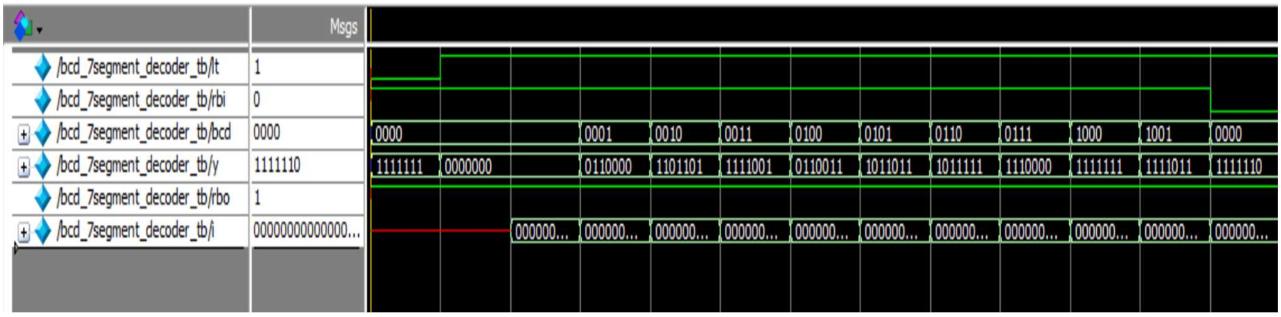


```
module bcd_7segment_decoder_tb();
    reg lt, rbi;
    reg [3:0] bcd;
    wire [6:0] y;
    wire rbo;
    integer i;

    // DUT instantiation
    bcd_7segment_decoder dut (
        .lt(lt),
        .rbi(rbi),
        .bcd(bcd),
        .rbo(rbo),
        .y(y)
    );

    // Task to display the 7-seg pattern visually (common cathode)
    task display_segments;
        input [6:0] seg; // a b c d e f g
        begin
            // seg[6]=a, seg[5]=f, seg[4]=e, seg[3]=d, seg[2]=c, seg[1]=b, seg[0]=g
            $display(" %s ", seg[6] ? " _ " : "   "); // a
            $display("%s%s%s", seg[5] ? "|" : " ", seg[0] ? "_" : " ", seg[1] ? "|" : " "); // f,g,b
            $display("%s%s%s", seg[4] ? "|" : " ", seg[3] ? "_" : " ", seg[2] ? "|" : " "); // e,d,c
        end
    endtask

    initial begin
        $monitor("Time=%0t | lt=%b | rbi=%b | bcd=%b | rbo=%b | y=%07b",
                 $time, lt, rbi, bcd, rbo, y);
```

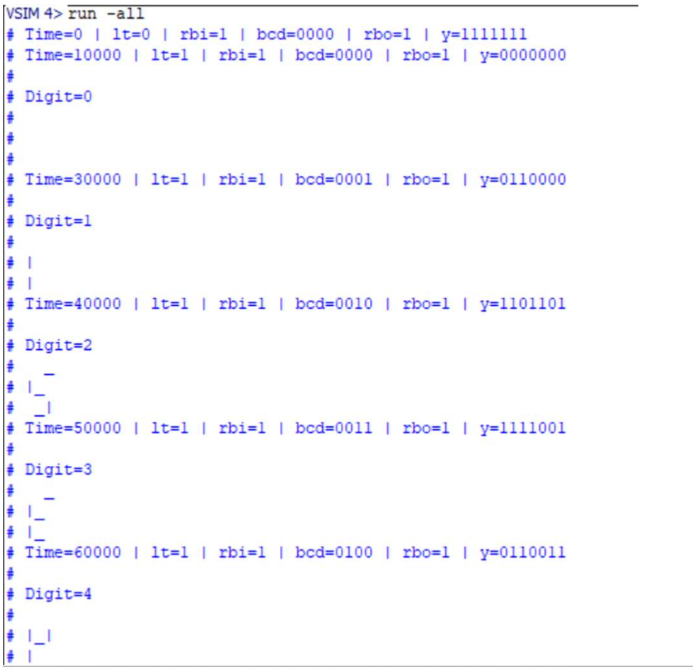


```
        // Initial conditions
        lt = 0; rbi = 1; bcd = 4'b0000; #10; // Lamp test ON

        // Lamp test OFF (normal operation)
        lt = 1; rbi = 1; bcd = 4'b0000; #10;

        // Test digits 0?9
        for (i = 0; i < 10; i = i + 1) begin
            bcd = i;
            #10;
            $display("\nDigit=%0d", i);
            display_segments(y);
        end

        // Test ripple blanking (active low)
        rbi = 0; bcd = 4'b0000; #10;
        $display("\nRipple Blanking Active:");
        display_segments(y);

        $finish;
    end
endmodule
```

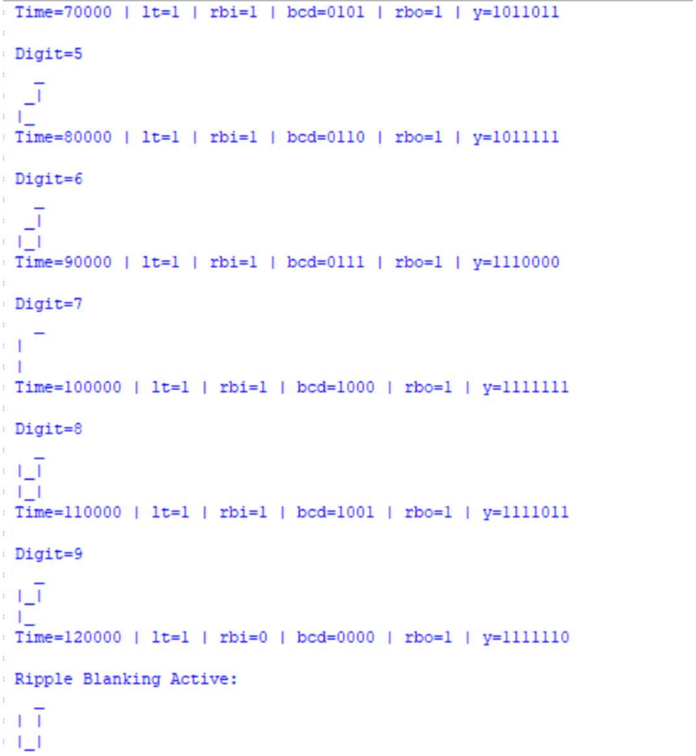
c)Waveform



d)Transcript



```
VSIM 4> run -all
# Time=0 | lt=0 | rbi=1 | bcd=0000 | rbo=1 | y=1111111
# Time=10000 | lt=1 | rbi=1 | bcd=0000 | rbo=1 | y=0000000
#
# Digit=0
#
#
#
# Time=30000 | lt=1 | rbi=1 | bcd=0001 | rbo=1 | y=0110000
#
# Digit=1
#
# |
# |
# Time=40000 | lt=1 | rbi=1 | bcd=0010 | rbo=1 | y=1101101
#
# Digit=2
#   _
# |_
#  _|
# Time=50000 | lt=1 | rbi=1 | bcd=0011 | rbo=1 | y=1111001
#
# Digit=3
#   _
# |_
# |_
# Time=60000 | lt=1 | rbi=1 | bcd=0100 | rbo=1 | y=0110011
#
# Digit=4
#
# |_|
# |
```



```
Time=70000 | lt=1 | rbi=1 | bcd=0101 | rbo=1 | y=1011011

Digit=5

  _
 _|
|_
Time=80000 | lt=1 | rbi=1 | bcd=0110 | rbo=1 | y=1011111

Digit=6

  _
 _|
|_|
Time=90000 | lt=1 | rbi=1 | bcd=0111 | rbo=1 | y=1110000

Digit=7

  _
|
|
Time=100000 | lt=1 | rbi=1 | bcd=1000 | rbo=1 | y=1111111

Digit=8

  _
|_|
|_|
Time=110000 | lt=1 | rbi=1 | bcd=1001 | rbo=1 | y=1111011

Digit=9

  _
|_|
|_
Time=120000 | lt=1 | rbi=0 | bcd=0000 | rbo=1 | y=1111110

Ripple Blanking Active:

  _
| |
|_|
```