```
// CSE 100/L Spring 2021
// This is a testbench for the entire Lab 2 Project.
// If the top level module in your Lab 2 project is named "top_lab2"
// and you used the suggested names for its inputs/outputs then
// then it will run without modification.  Otherwise follow the instructions
// in the comments marked "TODO" to modify this testbench to conform to your project.

//Author: Martine Schlag, updated by Sagnik Nath
`timescale 1ns/1ps
// ns is the unit of measurement of delays and simulation time
//ps is the precision unit(specifies how delay values are rounded before being used
in the simulation)

module lab2_7Seg_testbench();

   wire [7:0] D7Seg3,D7Seg2,D7Seg1,D7Seg0;
   reg sw0, sw1, sw2, sw3, sw4, sw5, sw6;
   wire CA,CB,CC,CD,CE,CF,CG,DP,AN0,AN1,AN2,AN3;



//========================Start interface your toplevel here
=========================
// TODO: replace "top_lab2" with the name of your top level Lab 2 module.

   lab2_top UUT (
        .sw0(sw0), .sw1(sw1), .sw2(sw2),.sw3(sw3), .sw4(sw4), .sw5(sw5), .sw6(sw6),
        .CA(CA),.CB(CB),.CC(CC),.CD(CD),.CE(CE),.CF(CF),.CG(CG),.DP(DP),
        .AN0(AN0),.AN1(AN1),.AN2(AN2),.AN3(AN3)
        );
// TODO: In the three lines above, make sure the pin names match the names
// used for the inputs/outputs of your top level module.   For example, if you
// used "cin" rather than "sw0" in yout top level module, then replace ".sw0(sw0)"
with ".cin(sw0)"

//========================Stop interface your toplevel here =========================



 show_7segDisplay  showit
(.seg({CG,CF,CE,CD,CC,CB,CA}),.DP(DP),.AN0(AN0),.AN1(AN1),.AN2(AN2),.AN3(AN3),
               .D7Seg0(D7Seg0),.D7Seg1(D7Seg1),.D7Seg2(D7Seg2),.D7Seg3(D7Seg3));
    initial
    begin

        sw0=1'b0;
    sw1=1'b0;
    sw2=1'b0;
    sw3=1'b0;
    sw4=1'b0;
    sw5=1'b0;
    sw6=1'b0;
    // sum is 0
    //-------------  Current Time:  0ns
    #100;  //This advances time by 100 units (ns in this case)
    sw0 = 1'b1;
    // sum is 1
    // -------------  Current Time:  100ns
    #100;
    sw1 = 1'b1;
    // sum is 2
    // -------------  Current Time:  200ns
    #100;
    sw4 = 1'b1;
    // sum is 3
    // -------------  Current Time:  300ns
    #100;
    sw0 = 1'b0;
    sw5 = 1'b1;
    // sum is 4
    // -------------  Current Time:  400ns
    #100;
    sw0 = 1'b1;
    // sum is 5
    // -------------  Current Time:  500ns
    #100;
    sw1 = 1'b0;
    sw2 = 1'b1;
    // sum is 6
    // -------------  Current Time:  600ns
    #100;
    sw1 = 1'b1;
    // sum is 7
    // -------------  Current Time:  700ns
    #100;
    sw4 = 1'b0;
    sw5 = 1'b0;
    sw6 = 1'b1;
    // sum is 8
    // -------------  Current Time:  800ns
    #100;
    sw4 = 1'b1;
    // sum is 9
    // -------------  Current Time:  900ns
    #100;
    sw1 = 1'b0;
    sw2 = 1'b0;
    sw3 = 1'b1;
    // sum is 10
    // -------------  Current Time:  1000ns
    #100;
    sw0=1'b1;
    sw1=1'b1;
    sw2=1'b0;
    sw3=1'b1;
    sw4=1'b1;
    sw5=1'b0;
    sw6=1'b1;
    //sum is 11
    //---------------- Current time: 1100ns
    #100;
    sw0=1'b0;
    sw1=1'b1;
    sw2=1'b1;
    sw3=1'b1;
    sw4=1'b1;
    sw5=1'b0;
    sw6=1'b1;
//sum is 12
//------------------- Current time: 1200ns;
   #100;
   sw0=1'b1;
   sw1=1'b1;
   sw2=1'b1;
   sw3=1'b1;
   sw4=1'b1;
   sw5=1'b0;
   sw6=1'b1;
 //sum is 13
 //------------------- Current time: 1300ns
  #100;
  sw0=1'b1;
  sw1=1'b1;
  sw2=1'b1;
  sw3=1'b1;
  sw4=1'b0;
  sw5=1'b1;
  sw6=1'b1;
```

```
    //sum is 14
    //-------------------- Current time: 1400ns
      #100;
      sw0=1'b1;
      sw1=1'b1;
      sw2=1'b1;
      sw3=1'b1;
      sw4=1'b1;
      sw5=1'b1;
      sw6=1'b1;
      #100;
    //sum is 15
    //--------------------- Current time: 1500ns
    // complete this testbentch  so that all
    // 16 hex values are generated
    end
endmodule


//==============Do not edit below this line =========================================
module show_7segDisplay (
 input [6:0] seg,
 input DP,AN0,AN1,AN2,AN3,
 output reg [7:0] D7Seg0, D7Seg1, D7Seg2,D7Seg3);

 reg [7:0] val;

  always @(AN0 or val)
   begin
        if (AN0 == 0) D7Seg0 <= val;
        else if (AN0 == 1) D7Seg1 <= " ";
        else D7Seg0 <= 8'bX;   //  non-blocking assignment
   end

  always @(AN1 or val)
   begin
        if (AN1 == 0) D7Seg1 <= val;
        else if (AN1 == 1) D7Seg1 <= " ";
        else D7Seg1 <= 8'bX;   //  non-blocking assignment
   end

  always @(AN2 or val)
   begin
        if (AN2 == 0) D7Seg2 <= val;
        else if (AN2 == 1) D7Seg2 <= " ";
        else D7Seg2 <= 8'bX;   //  non-blocking assignment
 end

always @(AN3 or val)
 begin
      if (AN3 == 0) D7Seg3 <= val;
      else if (AN3 == 1) D7Seg3 <= " ";
      else D7Seg3 <= 8'bX;   //  non-blocking assignment
 end

  always @(seg)
  case (seg)
  7'b01111111:
       val = "-";
  7'b1111111:
       val = " ";
  7'b1000000:
       val = "0";
  7'b1111001:
       val = "1";
  7'b0100100:
       val = "2";
  7'b0110000:
       val = "3";
  7'b0011001:
       val = "4";
  7'b0010010:
       val = "5";
  7'b0000010:
       val = "6";
  7'b1111000:
       val = "7";
  7'b0000000:
       val = "8";
  7'b0011000:
       val = "9";
  7'b0001000:
       val = "A";
  7'b0000011:
       val = "B";
  7'b1000110:
       val = "C";
  7'b0100001:
       val = "D";
  7'b0000110:
       val = "E";
  7'b0001110:
        val = "F";
    default:
        val = 8'bX;
    endcase
endmodule
```