```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/27/2021 01:05:50 PM
// Design Name:
// Module Name: lab4_simulation
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module lab4_simulation();
    reg [15:0] sw = 16'b1111111111111001;
    reg btnR, btnU, btnD, btnL, btnC, clkin;
    wire [6:0] seg;
    wire [3:0] an;
    wire dp;
    wire [15:0] led;

    lab4_top UUT (.clkin(clkin), .btnR(btnR), .btnU(btnU), .btnD(btnD), .btnC(btnC),
    .btnL(btnL), .sw(sw), .seg(seg), .dp(dp), .an(an), .led(led));

    //wire [7:0] D7Seg3,D7Seg2,D7Seg1,D7Seg0; // Change the Radix of these signals
to ASCII
    //show_7segDisplay  showit (.seg(seg),.dp(dp),.an(an),
    // .D7Seg0(D7Seg0),.D7Seg1(D7Seg1),.D7Seg2(D7Seg2),.D7Seg3(D7Seg3));

    parameter PERIOD = 10;
    parameter real DUTY_CYCLE = 0.5;
    parameter OFFSET = 2;

    initial    // Clock process for clkin
    begin
        #OFFSET
          clkin = 1'b1;
```

```
    forever
    begin
        #(PERIOD-(PERIOD*DUTY_CYCLE)) clkin = ~clkin;
    end
end
initial
begin
  // add your stimuli here
  // to set signal foo to value 0 use
  // foo = 1'b0;
  // to set signal foo to value 1 use
  // foo = 1'b1;
  //always advance time my multiples of 100ns
  // to advance time by 100ns use the following line
  btnU = 1'b0;
  btnR = 1'b0;
  btnL = 1'b0;
  btnC = 1'b0;
  btnD = 1'b0;
  #1000;
  btnL = 1'b1;  //Loading FFF9.
  #100;
  btnL = 1'b0;
  btnC = 1'b1;
  #100; //BtnC held for multiple cycles.
  #100;
  #100;
  #100;
  #100;
  btnC = 1'b0;
  #100; //It's now FFFC.
  btnU = 1'b1;
  #100; //FFFD
  btnU = 1'b0;
  #100;
  btnU = 1'b1;
  #100; //FFFE
  btnU = 1'b0;
  #100;
  btnU = 1'b1;
  #100; //FFFF
  btnU = 1'b0;
  #100;
  btnU = 1'b1;
  #100; //0000
  btnU = 1'b0;
```

```
    #100;
    btnD = 1'b1;
    #100; //FFFF
    #100; //BtnD held for multiple cycles.
    #100;
    btnD = 1'b0;
    #100;
    btnD = 1'b1;
    btnU = 1'b1;
    #100; //FFFF
    #100; //BtnU AND BtnD held for multiple cycles.
    #100;
    btnD = 1'b0;
    btnU = 1'b0;
    #100; //FFFF
    btnU = 1'b1;
    #100; //0000
    #100; //BtnU held for multiple cycles.
    #100;
    btnU = 1'b0;
    #100;
  end


endmodule
```