

```

timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/10/2021 02:03:17 PM
// Design Name:
// Module Name: lab5_simulation
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
module lab5_simulation();
    reg Up = 1'b0;
    reg btnR = 1'b0;
    reg clkIn;
    reg [15:0] sw;
    wire [15:0] led;
    wire [3:0] an;
    wire [6:0] seg;
    wire dp;

    lab5_top UUT (.btnU(Up), .sw(sw), .clkIn(clkIn), .btnR(btnR), .led(led),
.an(an), .dp(dp), .seg(seg));

    //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)) clk_in = ~clk_in;
    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
    sw = 16'b0000000000000000;
    #1200; //Initial. State 0, new game/game over.
    Up = 1'b1;
    #200; //State 1, something should be loaded into the counter.
    Up = 1'b0;
    #200; //State 2, should be counting down now.
    #600;
    sw[0] = 1'b1;
    #200; //A flips too early, causing B to score!
    sw[0] = 1'b0;
    #200;
    Up = 1'b1;
    #200;
    Up = 1'b0;
    #200;
    sw[15] = 1'b1;
    #200; //B flips too early, causing A to score!
    sw[15] = 1'b0;
    #200;
    Up = 1'b1;
    #200;
    Up = 1'b0;
    #200;
    sw[0] = 1'b1;
    sw[15] = 1'b1;
    #200; //A and B flip too early, causing neither of them to score!
    sw[0] = 1'b0;
    sw[15] = 1'b0;
    #200;
    Up = 1'b1;
    #200;

```

```
Up = 1'b0;
#100000;
sw[0] = 1'b1;
#200; //A scores!
sw[0] = 1'b0;
#200;
Up = 1'b1;
#200;
Up = 1'b0;
#100000;
sw[15] = 1'b1;
#200; //B scores!
sw[15] = 1'b0;
#200;
Up = 1'b1;
#200;
Up = 1'b0;
#100000;
sw[0] = 1'b1;
sw[15] = 1'b1;
#200; //A and B both score!
sw[0] = 1'b0;
sw[15] = 1'b0;
//sw[0] = 1'b1;
#200; //A scores!
end
```

endmodule