

```
timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/18/2020 07:43:26 PM
// Design Name:
// Module Name: Top_Level
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
module Top_Level(
    input clk, btnU, btnL, btnR,
    output [15:0] led,
    output [3:0] an,
    output [6:0] seg,
    output dp
);

    // Timing Wires
    wire clk;
    wire digsel;
    wire qsec;

    lab6_clks slowit(.clk(clk), .greset(btnU), .clk(clk), .digsel(digsel),
.qsec(qsec));

    // Edge Detector Wires
    wire Left;
    wire Right;

    //Edge_Detector Go_Edge(.clk(clk), .Btn(btnL), .out(Left));
    //Edge_Detector Stop_Edge(.clk(clk), .Btn(btnR), .out(Right));
    assign Left = btnR;
    assign Right = btnL;
```

```

// Test
//wire [1:0] test;
//assign led[1] = Left;
//assign led[0] = Right;
//assign test[0] = Left;
//assign test[1] = Right;

// State Machine Wires
wire Up;
wire Dw;
wire R;
wire t1;

    State_Machine FSM(.clk(clk), .btnL(Left), .btnR(Right), .Up(Up), .Dw(Dw),
.L_LED(led[15]), .R_LED(led[9]), .R(R), .T_On(t1));

// Counter Wires
wire [7:0] P;
wire [7:0] N;
wire [7:0] Q;
wire Neg;
//wire [7:0] Unused1;

//8bit counter
Counter8UDL TurkeyCount8(.clk(clk), .Up(Up), .Dw(Dw), .din(8'b0), .Q(P),
.Z_out(Neg));
    Bit8_Adder TwosComp(.In(~P), .Out(N));
    MUX2_1x8 NegSwitch(.in0(P), .in1(N), .sel(Neg), .out(Q));

// Timer Wires
wire [7:0] Timer;
//wire [15:4] Unused2;
//4bit counter
//    wire [3:0] timer;
//    assign timer = {Timer[3:0]};
    Counter8UDL Timer_Display8(.clk(clk), .Up(qsec&t1&(~Timer[3:0])), .R(R),
.Q(Timer));

// Display Wires
wire [3:0] ring;
wire [3:0] Neg_D;
wire [3:0] Inputs;
assign Neg_D = 4'b0;

wire [15:0] main;

```

```

    assign main = {Timer[3:0], Neg_D[3:0], Q[7:0]};
    //assign seg[6] = Neg;

    Ring_Counter Ring(.start(digsel), .clk(clk), .out(ring));
    Selector Select(.sel(ring), .N(main), .H(Inputs));
    Segment_Display Display(.n(Inputs), .sego(seg), .neg(Neg&ring[2]),
.state(ring[2]));

    assign an[0] = ~(ring[0]);
    assign an[1] = ~(ring[1]);
    assign an[2] = ~(ring[2]&Neg);
    assign an[3] = ~(ring[3]&t1);
    assign dp = 1'b1;

endmodule

```