```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: 
// Engineer: 
// 
// Create Date: 11/30/2020 05:44:46 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 clkin, btnC, btnL, btnD, btnR,
    input [15:0] sw,
    output [15:0] led,
    output [6:0] seg,
    output [3:0] an,
    output dp
    );
    
    //CLK Global
    wire clk, digsel, qsec;
    lab6_clks global_clk(.clkin(clkin), .greset(btnR), .qsec(qsec), .clk(clk), 
.digsel(digsel));
    
    
    //Edge Detection
    wire Clear, Test;
    Edge_Detector Button_C(.clk(clk), .Btn(btnC), .out(Clear));
    Edge_Detector Button_L(.clk(clk), .Btn(btnL), .out(Test));
    
    
    //FSM
    wire ledinput, lednprime, ledprime, flash, ShowIn, ShowDiv, prime, nprime, start;
    State_Machine FSM(.clk(clk), .btnC(Clear), .btnL(Test), .btnD(btnD),
.Prime(prime), .NPrime(nprime), .LED_Input(ledinput),
    .LED_nprime(lednprime), .LED_prime(ledprime), .Show_In(ShowIn),
.Show_Div(ShowDiv), .Flash(flash), .Start(start));


    // Input Module
    wire [15:0] Q;
    counterUD16L input_mode(.clk(clk), .din(sw), .LD(ShowIn), .Q(Q));


    //Divider
    wire [15:0] divisor;
    wire [15:0] quotient;
    Divider Dividermod(.clk(clk), .Start(start), .Input(Q), .Q_out(quotient),
.Clear(ShowIn), .Div_out(divisor), .NPrime(nprime), .Prime(prime));


    //LED Control
    Led_Selector leds(.clk(clk), .qsec(qsec), .in(ledinput), .prime(ledprime),
.nprime(lednprime&~ShowDiv), .div(ShowDiv&lednprime), .working(start), .ledin(led));

    //Blinker
    wire [15:0] TC_Out;
    counterUD16L Time(.clk(clk), .Up(qsec), .Q(TC_Out));


    //Display Module
    wire [3:0] ring;
    wire [3:0] Inputs;
    wire [15:0] Out;

    Out_Selector out(.In(Q), .Div(divisor),
.main((ShowIn|ledprime|lednprime)&~ShowDiv), .div(ShowDiv&lednprime), .out(Out));
//    assign Out =
(Q[15:0]&{16{ShowIn}})|(Q[15:0]&{16{ledprime}})|(Q[15:0]&{16{lednprime}})|(divisor[1
5:0]&{16{ShowDiv}});
    Ring_Counter Ring(.start(digsel), .clk(clk), .out(ring));
    Selector Select(.sel(ring), .N(Out), .H(Inputs));
    Segment_Display Display(.n(Inputs), .sego(seg));

    assign an[0] = ~(ring[0]&(~flash|~TC_Out[0]));
    assign an[1] = ~(ring[1]&(~flash|~TC_Out[0]));
    assign an[2] = ~(ring[2]&(~flash|~TC_Out[0]));
    assign an[3] = ~(ring[3]&(~flash|~TC_Out[0]));
    assign dp = 1'b1;
```

```
endmodule
```