```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/11/2022 03:24:17 PM
// Design Name:
// Module Name: toplevel
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module toplevel(
    input clkin,
    input btnL,
    input btnU,
    input btnR,

    output [15:0]led,
    output [6:0]seg,
    output [3:0]an

    );
    wire clk,qsec,digsel;
    wire left,right;
    wire timeup,run,reset,load,idle;
    wire startleft,startright,exitleft,exitright;
    //line and file for clk is given to us
    lab6_clks slowit (.clkin(clkin), .greset(btnU), .clk(clk), .digsel(digsel),
.qsec(qsec));

    //syncronizing left and right before using them, same thing we did in lab5 for bt
    FDRE #(.INIT(1'b0)) sy2 (.C(clk), .CE(1'b1), .D(btnL), .Q(left));
    FDRE #(.INIT(1'b0)) sy1 (.C(clk), .CE(1'b1), .D(btnR), .Q(right));
    //assign led with butn
    //assign led[15]= left;
```

```
    //assign led[9] = right;
    assign led[15]= ~left;
    assign led[9] = ~right;

    wire [7:0]timer;
    wire [15:0]q;

    // wire [5:0] TimeCounter;
    //count6U timerCounter (.clk(clk), .Up(qsec), .Dw(1'b0), .LW(resetTime),
.d(6'b000000), .q(TimeCounterOut));
    // wire hack;
    //assign hack = TimeCounter[0]];

    // countU8L countF (.clk(clk) , .Up(run  & ~timeup & hack),.LW(reset),
.UTC(timeup), .d(4'b00000000), .q(timer[3:0]));
    countU8L countF (.clk(clk) , .Up(run  & ~timeup & qsec), .LW(reset),
.UTC(timeup), .d(8'b11000000), .q(timer));
    //below is done from [5:2], to slow down qsec to a full sec
    assign q[15:12]=timer[5:2];



    //stae machnie getting values
    SM(.clk(clk),.left(left),.right(right),.timesUp(timeup),.idle(idle),
.R({startleft,startright,exitleft,exitright}), .load(load),.run(run),.reset(reset));

    wire plus,minus,negative;
    assign minus = startleft & exitright;
    assign plus= startright & exitleft;

    negativeSign
count(.up(plus),.down(minus),.clk(clk),.q(q[7:0]),.reset(btnU),.neg(negative));

    //wires for sev seg display
    wire [3:0]sel;
    wire [3:0]con;
    wire [6:0]pseg;

    Ring_Counter rc (.Adv(digsel), .clk(clk), .out(sel));
    Selector select (.sel(sel), .N(q), .H(con));
    SevSeg hex (.n(con), .seg(pseg));

    //assign seg[6:0]= an[2] & pseg[6:0] | ~an[2] & ~pseg[6:0];
    //NO UNDERSTAND WHY THE ABOVE DOESNT WORK AND THE BELOW VERSION DOES
    // probably something to do with an[2] very strange
    assign seg[0]= an[2] & pseg[0] | ~an[2] & ~pseg[0];
    assign seg[1]= an[2] & pseg[1] | ~an[2] & ~pseg[1];
```

```
    assign seg[2]= an[2] & pseg[2] | ~an[2] & ~pseg[2];
    assign seg[3]= an[2] & pseg[3] | ~an[2] & ~pseg[3];
    assign seg[4]= an[2] & pseg[4] | ~an[2] & ~pseg[4];
    assign seg[5]= an[2] & pseg[5] | ~an[2] & ~pseg[5];
    assign seg[6]= an[2] & pseg[6] | ~an[2] & ~pseg[6];

    //each segment activation for the anoneds, active low
    assign an[0]= ~sel[0];
    assign an[1]= ~sel[1];
    assign an[2]= ~(sel[2] & negative);
    assign an[3]= ~(sel[3] & load);



endmodule
```