```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/04/2022 03:46:05 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 btnR,
    input btnC,
    input btnU,
    input clkin,

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

    wire clk, qsec, digsel;
    wire showNub;
    wire resetTime;
    wire runGame;
    wire scored;
    wire flashBoth;
    wire flashAlt;
    wire sticky;
    wire hack;
    //wee woo
    lab5_clks slowit (.clkin(clkin), .greset(btnR), .clk(clk), .digsel(digsel),
.qsec(qsec));
```

```
    //syncronizing the buttons bc i was told so by the lab manual
    wire syncC, syncU;
    FDRE #(.INIT(1'b0) ) sync0 (.C(clk), .CE(1'b1), .R(1'b0), .D(btnC), .Q(syncC));
    FDRE #(.INIT(1'b0) ) sync1 (.C(clk), .CE(1'b1), .R(1'b0), .D(btnU), .Q(syncU));

    wire [7:0] rand;
    RNG rng (.clk(clk),.Q(rand));

    wire [7:0] cap; //caputre numb rng

    //TEST
    FDRE #(.INIT(1'b0) ) capture[7:0](.C({8{syncC & sticky}}), .CE({8{1'b1}}),
.D(rand), .Q(cap));



    wire [5:0] gameCounterOut;
    wire [5:0] TimeCounterOut;
    count6U timerCounter (.clk(clk), .Up(qsec), .Dw(1'b0), .LW(resetTime),
.d(6'b000000), .q(TimeCounterOut));
    count6U gameCounter (.clk(clk), .Up(qsec & runGame), .Dw(1'b0), .LW(syncC &
sticky), .d(6'b000000), .q(gameCounterOut));
    wire match;

    assign match = (~(gameCounterOut[5] ^ cap[5]) & ~(gameCounterOut[4] ^ cap[4]) &
~(gameCounterOut[3] ^ cap[3]) & ~(gameCounterOut[2] ^ cap[2]) &~(gameCounterOut[1] ^
cap[1]) & ~(gameCounterOut[0] ^ cap[0]));



    wire twoSec, fourSec;

    assign twoSec = TimeCounterOut[3] & ~TimeCounterOut[2:0]; // (TimeCounterOut >=
4'b1000) ? 1'b1 : 1'b0;
    assign fourSec = TimeCounterOut[4] & ~TimeCounterOut[3:0]; //(TimeCounterOut >=
5'b10000) ? 1'b1 : 1'b0;



   stateMachine smachine (.clk(clk), .go(syncC), .stop(syncU), .fourSec(fourSec),
.twoSec(twoSec), .match(match), .showNub(showNub), .resetTime(resetTime),
.runGame(runGame), .scored(scored),
                           .flashBoth(flashBoth), .flashAlt(flashAlt),
.sticky(sticky));

   shifter16 shift (.clk(scored), .adv(1'b1), .out(led));
```

```
  wire [3:0] Rout;
  Ring_Counter ringCount (.clk(clk), .Adv(digsel), .out(Rout));

  wire [3:0] sevin;
  wire [15:0] sel;

  assign sel = {1'b0, 1'b0, cap[5:0], 1'b0, 1'b0, gameCounterOut};
  Selector select (.sel(Rout), .N(sel), .H(sevin));

  SevSeg display (.n(sevin), .seg(seg));


  assign dp =1'b1;
  assign hack = TimeCounterOut[0];
  // rout[0] flashalt flashsame
  assign an[0] = ~(Rout[0] & ((~flashAlt & ~flashBoth)  | (flashAlt & hack &
~fourSec) | (hack & ~fourSec & flashBoth)));
  assign an[1] = ~(Rout[1] & ((~flashAlt & ~flashBoth)  | (flashAlt & hack &
~fourSec) | (hack & ~fourSec & flashBoth)));
  assign an[2] = ~((Rout[2] & showNub) & ((~flashAlt & ~flashBoth) | (flashAlt &
~hack & ~fourSec) | (hack & ~fourSec & flashBoth)));
  assign an[3] = ~((Rout[3] & showNub) & ((~flashAlt & ~flashBoth) | (flashAlt &
~hack & ~fourSec) | (hack & ~fourSec & flashBoth)));
endmodule
```