```
`timescale 1ns / 1ps
//////////////////////////////////////////////
////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/11/2020 10:25:56 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, btnR, btnU, btnC,
    output [15:0] led,
    output [3:0] an,
output dp,
output [6:0] seg
);

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

//State Machine Wires
wire Show_Num;
wire R;
wire Run;
wire Match;
wire Score;
wire TwoSec;
wire FourSec;
wire Both;
wire Alt;
wire Go;
wire Stop;

//Counter Wires
wire [15:0] TC_Out;
wire [5:0] rnd6;
wire [15:0] Q16;
wire [7:0] top;
```

```
    wire [7:0] bot;
    wire [5:0] Random;

    // Unused Wires
    wire [1:0] LFSR_Unused;
    wire [1:0] Bit_Unused;

    //Top View
    Edge_Detector Go_Edge(.clk(clk),
.Btn(btnC), .out(Go));
    Edge_Detector Stop_Edge(.clk(clk),
.Btn(btnU), .out(Stop));

    lab5_clks slowit(.clkin(clkin),
.greset(btnR), .clk(clk), .digsel(digsel),
.qsec(qsec));

    State_Machine FSM(.Go(Go), .Stop(Stop),
.FourSecs(FourSec), .TwoSecs(TwoSec),
.Match(Match), .clk(clk), .ShowNum(Show_Num),
    .ResetTimer(R), .RunGame(Run),
.Scored(Score), .FlashBoth(Both),
.FlashAlt(Alt));

    // Generator
    Generator LFSR(.clk(clk), .run(~Show_Num),
.rnd({LFSR_Unused, rnd6}));
```

```
    //FDRE #(.INIT(1'b0)) LFSR_FF[5:0]
(.C({6{clk}}), .CE(~{6{Show_Num}}), .D(rnd6),
.Q(Random[5:0]));
    assign top = {1'b0, 1'b0, rnd6};

    // Game Counter
    counterUD16L Game(.clk(clk),
.Up(qsec&Run), .Dw(1'b0), .R(R|Q16[6]),
.din(16'b0), .Q(Q16));
    assign bot = {1'b0, 1'b0, Q16[5:0]};

    // Time Counter
//    Time_Counter Time(.clk(clk),
.CE(Show_Num|Both|Alt), .reset(Reset_Timer),
.Q(TC_Out));
    //wire Reset;
    //Edge_Detector Reset_Edge(.clk(clk),
.Btn(R), .out(Reset));
//    counter16UDL Time(.clk(clk), .CE(qsec),
.Reset(R), .Q(TC_Out));
    counterUD16L Time(.clk(clk), .Up(qsec),
.Dw(1'b0), .R(Go|Stop), .din(16'b0),
.Q(TC_Out));

//    input clk, Up, Dw, LD,
//    input [15:0] din,
//    output UTC, DTC,
//    output [15:0] Q

//    assign TwoSec =
~TC_Out[7]&~TC_Out[6]&~TC_Out[5]&~TC_Out[4]&~T
C_Out[3]&TC_Out[2]&~TC_Out[1]&~TC_Out[0];
//    assign FourSec =
~TC_Out[7]&~TC_Out[6]&~TC_Out[5]&~TC_Out[4]&TC
_Out[3]&TC_Out[2]&~TC_Out[1]&~TC_Out[0];
    assign TwoSec= TC_Out[3];
    assign FourSec= TC_Out[4];

    //assign Match= top&bot;
    assign Match = ~(rnd6[5] ^ Q16[5]) &
~(rnd6[4] ^ Q16[4]) & ~(rnd6[3] ^ Q16[3]) &
~(rnd6[2] ^ Q16[2]) & ~(rnd6[1] ^ Q16[1]) &
~(rnd6[0] ^ Q16[0]);
    //assign rnd6[5:0]&Q16[5:0];

    //Display Wires
    wire [3:0] ring;
    wire [3:0] Inputs;

    wire Scored;
    Edge_Detector LED(.clk(clk), .Btn(Score),
.out(Scored));
    LED_Shifter LEDs(.In(1'b1), .CE(Scored),
.R(btnR), .clk(clk), .out(led));
```

```
    Ring_Counter Ring(.start(digsel),
.clk(clk), .out(ring));
    Selector Select(.sel(ring), .N({top,bot}),
.H(Inputs));
    Segment_Display Display(.n(Inputs),
.sego(seg));

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

endmodule
```