```verilog
`timescale 1ns / 1ps


module Top_Level(
    input btnU, btnD, btnC, btnL, btnR,      // btnC is global reset (not btnR)
    input [15:0] sw,
    input clkin,

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

    output Hsync, Vsync,
    output [3:0] vgaBlue,
    output [3:0] vgaGreen,
    output [3:0] vgaRed
    );

    wire clk, digsel;
    labVGA_clks not_so_slow (.clkin(clkin), .greset(btnC), .clk(clk), .digsel(digsel)


    wire btnU_Click, btnD_Click, btnC_Click, btnL_Click, btnR_Click;
    FDRE #(.INIT(1'b0)) buttonU (.C(clk), .R(1'b0), .CE(1'b1), .D(btnU),
.Q(btnU_Click));
    FDRE #(.INIT(1'b0)) buttonD (.C(clk), .R(1'b0), .CE(1'b1), .D(btnD),
.Q(btnD_Click));
    FDRE #(.INIT(1'b0)) buttonC (.C(clk), .R(1'b0), .CE(1'b1), .D(btnC),
.Q(btnC_Click));
    FDRE #(.INIT(1'b0)) buttonL (.C(clk), .R(1'b0), .CE(1'b1), .D(btnL),
.Q(btnL_Click));
    FDRE #(.INIT(1'b0)) buttonR (.C(clk), .R(1'b0), .CE(1'b1), .D(btnR),
.Q(btnR_Click));


 ///////////////////////////////                 Pixel Address Stuff
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
    wire [15:0] Vrow, Hcol;
    wire [3:0] Red, Green, Blue;
    wire Active_Region;
    Pixels PixelAddress (.clk(clk), .Vert(Vrow), .Horiz(Hcol), .Hsync(Hsync),
.Vsync(Vsync), .Active_Region(Active_Region));
///////////////////////////////////////////////////////////////////////////////////
/////////
```

```
/////////////////////////////////                    Border Stuff
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
    wire border;
    Border TheBorder (.clk(clk), .GreenWall(border), .Horizontal(Hcol),
.Vertical(Vrow));

/////////////////////////////////                    Frame
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
    wire frame;
    Edge_Detector TheFrame (.clk(clk), .btnU(Vsync), .Edge_out(frame));




/////////////////////////////////                    Game State Machine    (Chill -> Go -> Play
-> Win -> Lose)          \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
            // Inputs
                wire TimeUp, //(8 Seconds)
                    GameCountdown,
                    Win;

            // Outputs
                wire Chill, Go, Play,
                    ShowMem,
                    ShowCountdown,
                    CountingDown,
                    LDTime,
                    FlashTime,
                    FlashBorder;
                wire [4:0] NS, PS;



        // -> // 1 second counter
    wire Second1_out, Second1_dtc;
    wire [15:0] Second1;
    Counter_16Bit AsingleSecond (.clk(clk), .UP(1'b0), .DW(frame), .LD(Second1_dtc),
.sw(16'd60), .Q(Second1), .UTC(), .DTC(Second1_dtc));

        // -> // 8 second timer at the start
    wire Second8, Second8_out, Second8_dtc;
    Counter_16Bit Counter16bit (.clk(clk), .UP(1'b0), .DW(frame), .LD(CountingDown),
.sw(16'd480), .Q(Second8_out), .UTC(), .DTC(Second8_dtc));

        // -> // Game Timer
```

```verilog
        wire [15:0] GameCount_out, TimeAnode;
    Counter_16Bit GameCountdownTimer (.clk(clk), .UP(1'b0), .DW((Second1 == 16'd0) &
Play), .LD(Go), .sw({8'd0, sw[15:8]}), .Q(TimeAnode), .UTC(), .DTC(GameCountdown));


    GameState GameStateMachine (.clk(clk),
                                .btnU(btnU),
                                // .TimeUp(Second8_out),
                                .TimeUp(Second8_dtc),
                                .GameCountdown(GameCountdown),
                                .Win(Win),

                                .Go(Go),
                                .Play(Play),
                                .ShowMem(ShowMem),
                                .ShowCountdown(ShowCountdown),
                                .CountingDown(CountingDown),
                                .LDTime(LDTime),
                                .FlashTime(FlashTime),
                                .FlashBorder(FlashBorder),
//                                  .NS(NS),
                                .PS(PS)
                                );


/////////////////////////////////                Ball Makin Stuff
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
    wire BallRed  [3:0];
    wire BallBlue [3:0];
    wire MemRed   [7:0];
    wire MemBlue  [7:0];
    wire [15:0] Ballx [7:0];
    wire [15:0] Bally [7:0];


    // Ball 1 Red
    Balls_Moving MoveBall_1 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                .MemRed(MemRed[0]), .MemBlue(MemBlue[0]),
.BallRed(1'b1), .BallBlue(1'b0), .MakeBall(BallRed[0]),
                                .BallStartx(16'd150), .BallStarty(16'd150),
                                .ShowMem(ShowMem), .Ballx(Ballx[0]),
.Bally(Bally[0]));
    // Ball 2 Red
    Balls_Moving MoveBall_2 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
```

```verilog
                                                .MemRed(MemRed[1]), .MemBlue(MemBlue[1]),
.BallRed(1'b1), .BallBlue(1'b0), .MakeBall(BallRed[1]),
                                                .BallStartx(16'd154), .BallStarty(16'd75),
                                                .ShowMem(ShowMem), .Ballx(Ballx[1]),
.Bally(Bally[1]));
    // Ball 3 Red
    Balls_Moving MoveBall_3 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                                .MemRed(MemRed[2]), .MemBlue(MemBlue[2]),
.BallRed(1'b1), .BallBlue(1'b0), .MakeBall(BallRed[2]),
                                                .BallStartx(16'd333), .BallStarty(16'd410),
                                                .ShowMem(ShowMem), .Ballx(Ballx[2]),
.Bally(Bally[2]));
    // Ball 4 Red
    Balls_Moving MoveBall_4 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                                .MemRed(MemRed[3]), .MemBlue(MemBlue[3]),
.BallRed(1'b1), .BallBlue(1'b0), .MakeBall(BallRed[3]),
                                                .BallStartx(16'd500), .BallStarty(16'd500),
                                                .ShowMem(ShowMem), .Ballx(Ballx[3]),
.Bally(Bally[3]));

    // Ball 5 Blue
    Balls_Moving MoveBall_5 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                                .MemRed(MemRed[4]), .MemBlue(MemBlue[4]),
.BallRed(1'b0), .BallBlue(1'b1), .MakeBall(BallBlue[0]),
                                                .BallStartx(16'd500), .BallStarty(16'd100),
                                                .ShowMem(ShowMem), .Ballx(Ballx[4]),
.Bally(Bally[4]));
    // Ball 6 Blue
    Balls_Moving MoveBall_6 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                                .MemRed(MemRed[5]), .MemBlue(MemBlue[5]),
.BallRed(1'b0), .BallBlue(1'b1), .MakeBall(BallBlue[1]),
                                                .BallStartx(16'd222), .BallStarty(16'd64),
                                                .ShowMem(ShowMem), .Ballx(Ballx[5]),
.Bally(Bally[5]));
    // Ball 7 Blue
    Balls_Moving MoveBall_7 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                                .MemRed(MemRed[6]), .MemBlue(MemBlue[6]),
.BallRed(1'b0), .BallBlue(1'b1), .MakeBall(BallBlue[2]),
                                                .BallStartx(16'd95), .BallStarty(16'd95),
                                                .ShowMem(ShowMem), .Ballx(Ballx[6]),
.Bally(Bally[6]));
```

```verilog
    // Ball 8 Blue
    Balls_Moving MoveBall_8 (.clk(clk), .btnU(btnU), .btnR(btnR_Click),
.btnL(btnL_Click), .Hpixel(Hcol), .Vpixel(Vrow), .frame(frame),
                                .MemRed(MemRed[7]), .MemBlue(MemBlue[7]),
.BallRed(1'b0), .BallBlue(1'b1), .MakeBall(BallBlue[3]),
                                .BallStartx(16'd444), .BallStarty(16'd444),
                                .ShowMem(ShowMem), .Ballx(Ballx[7]),
.Bally(Bally[7]));


/////////////////////////////////        Win state
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\


        // -> // Win State
                // Red Left & Blue Right
    assign Win = (       (Ballx[0] >= 16'd316) & (Ballx[1] >= 16'd316) & (Ballx[2] >=
16'd316) & (Ballx[3] >= 16'd316) &
                        (Ballx[4] <= 16'd323) & (Ballx[5] <= 16'd323) &
(Ballx[6] <= 16'd323) & (Ballx[7] <= 16'd323) ) |
                // Red Right & Blue Left
                (       (Ballx[0] <= 16'd323) & (Ballx[1] <= 16'd323) & (Ballx[2] <=
16'd323) & (Ballx[3] <= 16'd323) &
                        (Ballx[4] >= 16'd316) & (Ballx[5] >= 16'd316) &
(Ballx[6] >= 16'd316) & (Ballx[7] >= 16'd316) );

/////////////////////////////////        VGA colors
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
            // assign vgaGreen = Green & (4'b1111 & {4{border}} & (Second1[4] &
FlashBorder | ~FlashBorder));
    assign vgaGreen = {4{border & Active_Region & (Second1[4] & FlashBorder |
~FlashBorder) & (~PS[0] | ~PS[1]) }};
////////////////////////////////////////////////////////////////////////////////////
/////////

    assign vgaRed = {4{Active_Region & (BallRed[0]  | BallRed[1]  | BallRed[2]  |
BallRed[3]
            | MemRed[0] | MemRed[1] | MemRed[2] | MemRed[3] | MemRed[4] | MemRed[5]
| MemRed[6] | MemRed[7])
////////////////////////////////////////////////////////////////////////////////////
/////////
            }};

    assign vgaBlue = {4{Active_Region & (BallBlue[0] | BallBlue[1] | BallBlue[2] |
BallBlue[3]
        | MemBlue[0] | MemBlue[1] | MemBlue[2] | MemBlue[3] | MemBlue[4] |
MemBlue[5] | MemBlue[6] | MemBlue[7])
////////////////////////////////////////////////////////////////////////////////////
```

```verilog
/////////
        }};


    wire [3:0] Ring_out;
    Ring_Counter ringcounter (.Advance(digsel), .clk(clk), .Ring_out(Ring_out));
    wire [3:0] Sel_out_hex;
    Selector selector (.N({TimeAnode[7:0], 8'd0}), .sel(Ring_out), .H(Sel_out_hex));
    hex7seg Hex7Display (.n(Sel_out_hex), .seg(seg));

    assign an[3] = ~(~PS[0] & Ring_out[3]) & ~(FlashTime & (Second1[4]));
    assign an[2] = ~(~PS[0] & Ring_out[2]) & ~(FlashTime & (Second1[4]));

    assign led[15:8] = sw[15:8];
    assign led[4:0]  = PS[4:0];
    assign dp = 1'b1;

endmodule
```