```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/27/2024 01:52:44 PM
// Design Name:
// Module Name: lab6_top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module lab6_top(
    input clkin,
    input btnC,
    input btnU,
    input btnL,
    input btnR,
    input btnD,
    input [15:0] sw,
    output Hsync,
    output Vsync,
    output [3:0] vgaRed,
    output [3:0] vgaGreen,
    output [3:0] vgaBlue,
    output [15:0] led,
    output [3:0] an,
    output [6:0] seg
    );

    wire clk, digsel;
    labVGA_clks clkOUT (.clkin(clkin), .greset(btnD), .clk(clk), .digsel(digsel));

    wire [15:0] H, V;
    pixelAddress pixel(.clk(clk), .H(H), .V(V));
    wire Hs, Vs;
    VGASyncs sync (.H(H), .V(V), .Hsync(Hs), .Vsync(Vs));
    FDRE #(.INIT(1'b1)) HsyncFF (.C(clk), .CE(1'b1), .R(1'b0), .D(Hs), .Q(Hsync)) ;
//make Hsync and Vsync synchronous
    FDRE #(.INIT(1'b1)) VsyncFF (.C(clk), .CE(1'b1), .R(1'b0), .D(Vs), .Q(Vsync)) ;
    wire [9:0] iceHit, iceColor;
    wire [3:0] r, g, b;
    wire frame, currColor;
    gameRGB game (.btnC(btnC), .btnU(btnU), .btnL(btnL), .btnR(btnR), .sw(sw),
.frame(frame), .clk(clk), .red(r),.green(g),.blue(b), .Hpos(H), .Vpos(V), .leds(led)
, .iceHit(iceHit), .iceColor(iceColor));
    FDRE #(.INIT(1'b0)) Red [3:0](.C({4{clk}}), .CE({4{1'b1}}), .R(4'b0), .D(r),
.Q(vgaRed));
    FDRE #(.INIT(1'b0)) Green [3:0](.C({4{clk}}), .CE({4{1'b1}}), .R(4'b0), .D(g),
.Q(vgaGreen));
    FDRE #(.INIT(1'b0)) Blue [3:0](.C({4{clk}}), .CE({4{1'b1}}), .R(4'b0), .D(b),
.Q(vgaBlue));

    wire incScore;
    edgedetector frames (.clk(clk), .x(Vsync), .y(frame));
    edgedetector hit (.clk(clk), .x(|iceHit), .y(incScore));

    chooseColor Color (.iceHit(iceHit), .iceColor(iceColor), .currColor(currColor));
    wire [7:0] scoreRaw, scoreToDisplay;
    counterUD16L score (.clk(clk), .UD(currColor), .CE(incScore), .LD(1'b0),
.Din(1'b0), .Q(scoreRaw), .UTC(), .DTC());


    SignChanger covertToTwos (.a(scoreRaw), .sign(scoreRaw[7]), .d(scoreToDisplay));

    wire [3:0] sel, selOUT;
    wire [6:0] hexOUT;
    RC1bit ring (.clk(clk), .advance(digsel), .Q(sel));

    selector select (.N(scoreToDisplay), .sel(sel), .H(selOUT));

    hex7seg segments (.n(selOUT), .seg(hexOUT));

    assign seg = (hexOUT & {7{~sel[2]}}) | (~hexOUT & {7{sel[2]}});
    assign an[0] = ~sel[0];
    assign an[1] = ~sel[1];
    assign an[2] = ~(sel[2] & scoreRaw[7]);
    assign an[3] = 1;
endmodule
```