

```
`timescale 1ns / 1ps
///////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/23/2022 05:08:12 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 btnR,
    input btnU,
    input btnD,
    input btnC,
    input [15:0] sw,

    output [15:0] led,
    output [3:0] vgaRed,
    output [3:0] vgaBlue,
    output [3:0] vgaGreen,
    output Hsync,
    output dp,
    output [3:0] an,
    output [6:0] seg,
    output Vsync
);
    wire clk, digsel;
    wire [3:0] red, blue, green;
    wire [10:0] Hpixel;
    wire [10:0] Vpixel;
```

```

wire new_frame; //high 3 times in a cycle

lab7_clks not_so_slow (.clkin(clkin), .greset(btnR), .clk(clk), .digsel(digsel));

vga_setup screen (.clock(clk), .vgaRed(red), .vgaBlue(blue), .vgaGreen(green),
                  .Hsync(Hsync), .Vsync(Vsync), .Hpixel(Hpixel),
.Vpixel(Vpixel), .new_frame(new_frame));

wire [3:0] wet;
wire [3:0] frog;
wire [3:0] cactus, cactus2, cactus3;
wire up, down, res, pause , blink , score, resetTime, blinkscore; //outputs for sm
wire [15:0] frogsition;
wire syncU, syncD, syncL, syncR, syncC;
wire [7:0] rand;
wire [15:0] xpos1,xpos2,xpos3;
wire [10:0] ypos1,ypos2,ypos3;

synchronizer syncer (.clock(clk), .d({btnU, btnD, btnL, btnR, btnC}), .q({syncU,
syncD, syncL, syncR, syncC}));

wire [2:0] testscore;
water watta ( .Vpixel(Vpixel), .water(wet));

frog frogga (.Vpixel(Vpixel), .Hpixel(Hpixel), .clk(clk), .new_frame(new_frame),
.up(up), .down(down), .reset(res), .position(frogsition), .frog(frog));

cactus cacttal (.Vpixel(Vpixel), .Hpixel(Hpixel), .clk(clk), .pause(pause),
.reset(res), .defaultWhore(16'd297), .rng(rand[3:0]), .new_frame(new_frame),
.cactus(cactus), .scored(testscore[0]), .xpos(xpos1),
.ypos(ypos1));
cactus cactta2 (.Vpixel(Vpixel), .Hpixel(Hpixel), .clk(clk), .pause(pause),
.reset(res), .defaultWhore(16'd537), .rng(rand[3:0]), .new_frame(new_frame),
.cactus(cactus2), .scored(testscore[1]), .xpos(xpos2),
.ypos(ypos2));
cactus cactta3 (.Vpixel(Vpixel), .Hpixel(Hpixel), .clk(clk), .pause(pause),
.reset(res), .defaultWhore(16'd777), .rng(rand[3:0]), .new_frame(new_frame),
.cactus(cactus3), .scored(testscore[2]), .xpos(xpos3),
.ypos(ypos3));

wire twoSec, colision, frog232, frog136, frog328; // inputs for sm

col colla (.frogpos(frogsition), .x1(xpos1), .x2(xpos2), .x3(xpos3) ,
.y1(ypos1), .y2(ypos2), .y3(ypos3), .colision(colision));
//frogsition mania

```

```

//assign colision = 1'b0;
assign frog232 = (frogsition == 16'd232) ? 1'b1 :1'b0;
assign frog136 = (frogsition <= 16'd136) ? 1'b1 :1'b0;
assign frog328 = (frogsition >= 16'd328) ? 1'b1 :1'b0;

SM statemachine (.clk(clk) , .btnC(syncC), .btnD(syncD), .btnU(syncU),
.twoSec(twoSec), .colision(colision & ~sw[1]), .frog232(frog232), .frog136(frog136),
.frog328(frog328),
.up(up), .down(down), .reset(res), .pause(pause),
.blink(blink), .resetTime(resetTime), .blinkscore(blinkscore));

wire [15:0] seconds;
wire [15:0] blinkTime;
wire [3:0] qsecTimesTWO;

countUD16L timer (.clk(clk), .Up(new_frame), .Dw(1'b0), .LW(resetTime),
.d(16'b0), .q(seconds));

countUD16L blinkTimer (.clk(clk), .Up(new_frame), .Dw(1'b0), .LW((blinkTime >=
16'd180) | ~blink), .d(16'b0), .q(blinkTime));

assign twoSec = (seconds >= 15'd360) ? 1'b1: 1'b0;
assign qsecTimesTWO = (blinkTime >= 90) ? 4'b0 : 4'b1111;

RNG rng (.clk(clk),.Q(rand));

//pass rand into cactus to determin the center position of the cactus

wire [3:0] greenPriority;
// frog (qsecTimesTWO & blink & 4'b0000)
assign greenPriority = ((cactus == 4'hf) | (cactus2 == 4'hf) | (cactus3 == 4'hf))
? (4'b0000): 4'hf;

assign vgaRed = red & (frog & qsecTimesTWO );

assign vgaBlue = blue & ((frog & qsecTimesTWO) | (wet & greenPriority)); // or
wet with frog and then and to blue

assign vgaGreen = green & ((frog & qsecTimesTWO) | (cactus | cactus2 | cactus3));

assign score = testscore[0] | testscore[1] | testscore[2];

```

```

//score counter
wire [15:0] scoreOut;

countUD16L scoreCount (.clk(clk), .Up(score), .Dw(1'b0), .LW(res), .d(16'b0),
.q(scoreOut));

// assign scoreOut = 16'b1111001100000001;
wire [3:0] Rout;
Ring_Counter Rcounter (.clk(clk), .Adv(digsel), .out(Rout));
//seven seg
wire [3:0] sevin;
Selector sevInseg (.sel(Rout), .N(scoreOut[15:2]), .H(sevin));

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

//just making switches turn on LEDs no reson for it
assign led = sw;

assign dp =1'b1;

assign an = ~Rout | ( {blinkscore, blinkscore, blinkscore, blinkscore} &
qsecTimesTWO);

endmodule

```