```
module Top_Module_Main(
    input btnC,
    input btnU,
    input btnD,
    input btnR,
    input btnL,
    input [15:0] sw,
    input clkin,
    output Hsync,
    output Vsync,
    output [3:0] vgaRed,
    output [3:0] vgaGreen,
    output [3:0] vgaBlue,
    output [15:0] led,
    output [6:0] seg,
    output dp,
    output [3:0]an
    );
lab7_clks not_so_slow (.clkin(clkin), .greset(btnR), .clk(clk), .digsel(digsel));

wire [11:0] H_Count_Value;
wire [11:0] V_Count_Value, frog_position, treecol1, treecol2, treecol3, treerow1,
treerow2, treerow3;
wire [15:0] bit16out;
wire rowFinish, EDrowFinish, frame;
wire moveup, movedown, moving, start_timer, frog_blink, delay_signal;
//VGA CONNECTIONS
//need a module to count where we are horizontally (pass in H_Count_Value)
Edge_Detector finishRow (.clk(clk), .btn(rowFinish), .out(EDrowFinish));
HSync_Tracker hsynch_track (.clk(clk), .RowFinish(rowFinish),
.Position(H_Count_Value));
//need a module to count where we are vertically   (pass in H_Count_Value)
VSync_Tracker vsynch_track (.clk(clk), .NextCol(EDrowFinish), .Frame(frame),
.Position(V_Count_Value));

//3 pixels / frame clk
wire [3:0] count;
wire ThreePixFrameClk;
countUD4L faster_clk (.Up(count < 3), .Dw(1'b0), .LD(frame), .Q(4'b0), .clk(clk),
.Reset(1'b0), .Qout(count));
assign ThreePixFrameClk = count < 3;

// VGA sync outputs
assign Hsync = (H_Count_Value >= 656 & H_Count_Value <= 751) ? 1'b0 : 1'b1;
assign Vsync = (V_Count_Value >= 489 & V_Count_Value <= 490) ? 1'b0 : 1'b1;
```

```
//LOGIC
//add state machine
wire edge_L, rungame, blink, reset_game, initial_state, plant_hit;
StateMachine statemachine (.Up(btnU), .Down(btnD), .Center(btnC),
.SEC2(delay_signal), .Moving(moving),
                           .HIT(plant_hit&!sw[0] | sw[2]), .clk(clk),
.TimerStart2Sec(start_timer), .Frog_Up(moveup), .INITSTATE(initial_state),
//switches used to simulate HIT
                           .Frog_Down(movedown), .Rungame(rungame),
.Frog_Blink(frog_blink), .Reset(reset_game));

//Logic for making the segments and the frog blink
Blink blinker (.InputSignal(frog_blink), .Framerate(frame), .clk(clk),
.OutputSignal(blink));

//Logic for the switches and LEDs
assign led = sw;
//add instances of 3 trees
wire resetPlantPosition;        //while in the END_DELAY state, btnC is pressed and
we can start over
Edge_Detector rstPlants (.clk(clk), .btn(btnC&!rungame), .out(resetPlantPosition));
wire goAheadPlant2, goAheadPlant3;
Plants plant1 (.Rungame(!sw[3]&rungame), .INIT(initial_state | start_timer),
.clk(clk), .FirstPlant(1'b1), .Row(treerow1), .Col(treecol1), .INITPosition(12'd300),
               .ResetPlant(resetPlantPosition), .Framerate(frame),
.ThreePixFrameClk(ThreePixFrameClk));

//FDRE #(.INIT(1'b0)) readyplant2 (.C(clk), .R(resetPlantPosition), .CE(treecol1 ==
470), .D(1'b1), .Q(goAheadPlant2));  //when plant1 is in the middle of the screen
Plants plant2 (.Rungame(!sw[3]&rungame), .INIT(initial_state | start_timer),
.clk(clk), .FirstPlant(1'b0), .Row(treerow2), .Col(treecol2), .INITPosition(12'd530),
               .ResetPlant(resetPlantPosition), .Framerate(frame),
.ThreePixFrameClk(ThreePixFrameClk));

//FDRE #(.INIT(1'b0)) readyplant3 (.C(clk), .R(resetPlantPosition), .CE(treecol1 ==
267), .D(1'b1), .Q(goAheadPlant3)); //when plant1 is in the first third of the screen
Plants plant3 (.Rungame(!sw[3]&rungame), .INIT( | start_timer), .clk(clk),
.FirstPlant(1'b0), .Row(treerow3), .Col(treecol3), .INITPosition(12'd745),
               .ResetPlant(resetPlantPosition), .Framerate(frame),
.ThreePixFrameClk(ThreePixFrameClk));

//add 2 sec timer
Delay2sec timer (.Start(start_timer), .FrameClk(frame), .clk(clk),
.Signal(delay_signal));

//add frog
```

```
Frog frog (.MoveUp(moveup), .MoveDown(movedown), .Reset_btnC(btnC),
.Framerate(frame), .In_End_Delay(frog_blink), .clk(clk),
           .ThreePixPerFrame(ThreePixFrameClk), .INIT(initial_state),
.Moving(moving), .Rungame(rungame), .FrogBlink(blink), .Position(frog_position));
//count number of times frog passed a plant
wire utc1, utc2, utc3, increment;
wire [15:0]switch;
countUD4L trackpoint1 (.Up(rungame&increment),                .Dw(1'b0),
.LD(btnC&!rungame), .Reset(1'b0), .Q(4'b0), .clk(frame), .UTC(utc1),
.Qout(switch[3:0]));
countUD4L trackpoint2 (.Up(rungame&increment&utc1),           .Dw(1'b0),
.LD(btnC&!rungame), .Reset(1'b0), .Q(4'b0), .clk(frame), .UTC(utc2),
.Qout(switch[7:4]));
countUD4L trackpoint3 (.Up(rungame&increment&utc1&utc2),      .Dw(1'b0),
.LD(btnC&!rungame), .Reset(1'b0), .Q(4'b0), .clk(frame), .UTC(utc3),
.Qout(switch[11:8]));
countUD4L trackpoint4 (.Up(rungame&increment&utc1&utc2&utc3), .Dw(1'b0),
.LD(btnC&!rungame), .Reset(1'b0), .Q(4'b0), .clk(frame),
.Qout(switch[15:12]));
assign bit16out[3] = switch[0], bit16out[2] = switch[1], bit16out[1] = switch[2],
bit16out[0] = switch[3];
assign bit16out[7] = switch[4], bit16out[6] = switch[5], bit16out[5] = switch[6],
bit16out[4] = switch[7];
assign bit16out[11] = switch[8], bit16out[10] = switch[9], bit16out[9] = switch[10],
bit16out[8] = switch[11];
assign bit16out[15] = switch[12], bit16out[14] = switch[13], bit16out[13] =
switch[14], bit16out[12] = switch[15];

//Logic for the VGA colors
VGA_Control vga (.Frog_Position(frog_position), .Tree1Row(treerow1),
.Tree2Row(treerow2), .Tree3Row(treerow3), .Tree1Col(treecol1), .Tree2Col(treecol2),
.AddOne(increment),
                 .Tree3Col(treecol3), .currentHCount(H_Count_Value),
.currentVCount(V_Count_Value), .Red(vgaRed), .Green(vgaGreen), .Blue(vgaBlue),
.HIT(plant_hit));
wire [3:0]Qring;
RingCounter ring_cntr (.digsel(digsel), .clk(clk), .Q(Qring));
assign an[0] = !(Qring[0]) | blink;
assign an[1] = !(Qring[1]) | blink;
assign an[2] = !(Qring[2]) | blink;
assign an[3] = !(Qring[3]) | blink;
assign dp = 1;
wire [3:0]sel;
Selector select(.sel(Qring), .N(bit16out), .H(sel));
hex7seg segment_disp (.n(sel), .seg(seg));
endmodule
```