module de1soc\_top

(

// These are the board inputs/outputs required for all the ECE342 labs.

// Each lab can use the subset it needs -- unused pins will be ignored.

// Clock pins

input CLOCK\_50,

// Seven Segment Displays

output [6:0] HEX0,

output [6:0] HEX1,

output [6:0] HEX2,

output [6:0] HEX3,

output [6:0] HEX4,

output [6:0] HEX5,

// Pushbuttons

input [3:0] KEY,

// LEDs

output [9:0] LEDR,

// Slider Switches

input [9:0] SW,

// VGA

output [7:0] VGA\_B,

output VGA\_BLANK\_N,

output VGA\_CLK,

output [7:0] VGA\_G,

output VGA\_HS,

output [7:0] VGA\_R,

output VGA\_SYNC\_N,

output VGA\_VS

);

// VGA adapter and signals

logic [8:0] vga\_x;

logic [7:0] vga\_y;

logic [2:0] vga\_color;

logic vga\_plot;

vga\_adapter #

(

.BITS\_PER\_CHANNEL(1)

)

vga\_inst

(

.CLOCK\_50(CLOCK\_50),

.VGA\_R(VGA\_R),

.VGA\_G(VGA\_G),

.VGA\_B(VGA\_B),

.VGA\_HS(VGA\_HS),

.VGA\_VS(VGA\_VS),

.VGA\_SYNC\_N(VGA\_SYNC\_N),

.VGA\_BLANK\_N(VGA\_BLANK\_N),

.VGA\_CLK(VGA\_CLK),

.x(vga\_x),

.y(vga\_y),

.color(vga\_color),

.plot(vga\_plot)

);

// This generates a one-time active-high asynchronous reset

// signal on powerup. You can use it if you need it.

// All the KEY inputs are occupied, so we can't use one as a reset.

logic reset;

logic [1:0] reset\_reg;

always\_ff @ (posedge CLOCK\_50) begin

reset <= ~reset\_reg[0];

reset\_reg <= {1'b1, reset\_reg[1]};

end

//

// PUT YOUR UI AND LDA MODULE INSTANTIATIONS HERE

logic [8:0] x, x0, x1;

logic [7:0] y, y0, y1;

logic color;

logic plot;

logic done, start;

datapath UI\_datapath

(

.SetX(x),

.SetY(y),

.SetCol(color),

.GO(start),

.done(done),

.x0(x\_0),

.x1(x\_1),

.y0(y\_0),

.y1(y\_1),

.color(color\_out),

.start(\_start)

);

datapath LDA\_datapath

(

.x0(x\_0),

.x1(x\_1),

.y0(y\_0),

.y1(y\_1),

.color(color\_out),

.start(\_start)

.x(x),

.y(y),

.plot(start\_),

.color\_out(color),

.done(done)

);

control \_control

(

.SetX(x),  
 .SetY(y),  
 .SetCol(color),  
 .GO(start),  
 .done(done),  
 .o\_x,  
 .o\_y,  
 .plot(start\_),  
 .color(color\_out)

);

endmodule