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

);

module FA // (a, b, cin, sout, cout)

(

input a, b, cin,  
 output sout, cout

);  
 assign sout = (a ^ b ^ cin);  
 assign cout = ((a & b) | (a & cin) | (b & cin));  
endmodule

module BE // (inp, o) booth encoding

(

input inp[7:0],

output o[7:0]

);

genvar i;  
generate

for (i = 0; i < 8; i++) begin :

if (inp[i] == 0b’0) begin

If (inp[i-1] == 0b’0) assign o[i] = 0b’0;

else if (inp[i-1] == 0b’1) assign o[i] = 0b’1;

end

If (inp == 0b’1) begin

if (inp[i-1] == 0b’0) assign o[i] = 0b’0 - 0b’1;

else if (inp[i-1] == 0b’1) assign 0[i] = 0b’0;

end

end

endmodule

module Arraymultiplier // (inp1, inp2, product)

(

input [7:0] in1,  
 input [7:0] in2,  
 output [15:0] product

);  
 assign product[0]=(in1[0] & in2[0]);

logic [7:0] in3;

genvar i;  
 generate

for (i = 0; i < 8; i++) begin :

BE BE0(in2,in3);

end

logic [15:0] temp1;

logic [15:0] temp2;

logic over1;

logic over2;

assign over = 4b’0;

assign temp1 = 16b’0;

assign temp2 = 16b’0;

FA FA0(in1[1]&in3[0],in1[0]&in3[1],over1,temp2[1],over2);

FA FA1(in1[2]&in3[0],in1[1]&in3[1],over2,temp1[2],over1);

FA FA2(temp1[2],in1[0]&in3[2],over1,temp2[2],over2);

FA FA3(over2,in1[3]&in3[0],over2,temp1[4],over1);

FA FA4(temp1[4],in1[2]&in3[1],over1,temp2[4],over2);

FA FA5(temp2[4],in1[1]&in3[2],over2,temp1[4],over1);

FA FA6(temp1[4],in1[0]&in3[3],over1,temp2[4],over2);

FA FA7(over2,in1[4]&in[0],over2,temp1[5],over1);

FA FA8(temp1[5],in1[3]&in3[1],over1,temp2[5],over2);

FA FA9(temp2[5],in1[2]&in3[2],over2,temp1[5],over1);

FA FA10(temp1[5],in1[1]&in3[3],over1,temp2[5],over2);

FA FA11(temp1[5],in1[0]&in3[4],over2,temp2[5],over1);

FA FA12(over1,in1[5]&in3[0],over2,temp2[6],over1);

FA FA13(temp2[6],in1[4]&in3[1],over2,temp1[6],over1);

FA FA14(temp1[6],in1[3]&in3[2],over1,temp2[6],over2);

FA FA15(temp2[6],in1[2]&in3[3],over2,temp1[6],over1);

FA FA16(temp1[6],in1[1]&in3[4],over1,temp2[6],over2);

FA FA17(temp2[6],in1[0]&in3[5],over2,temp1[4],over1);

FA FA18(temp2[5],in1[2]&in3[3],over1,temp1[5],over2);

FA FA19(temp2[6],in1[3]&in3[3],over2,temp1[6],over1);

FA FA20(temp2[7],in1[4]&in3[3],over1,temp1[7],over2);

FA FA21(temp2[8],in1[5]&in3[3],over2,temp1[8],over1);

FA FA22(temp2[9],in1[6]&in3[3],over1,temp1[9],over2);

FA FA23(temp2[10],in1[7]&in3[3],over2,temp1[10],over1);

FA FA24(temp1[4],in1[0]&in3[4],over1,temp2[4],over2);

FA FA25(temp1[5],in1[1]&in3[4],over2,temp2[5],over1);

FA FA26(temp1[6],in1[2]&in3[4],over1,temp2[6],over2);

FA FA27(temp1[7],in1[3]&in3[4],over2,temp2[7],over1);

FA FA28(temp1[8],in1[4]&in3[4],over1,temp2[8],over2);

FA FA29(temp1[9],in1[5]&in3[4],over2,temp2[9],over1);

FA FA30(temp1[10],in1[6]&in3[4],over1,temp2[10],over2);

FA FA31(temp1[11],in1[7]&in3[4],over2,temp2[11],over1);

FA FA32(temp1[12],in1[8]&in3[4],over1,temp2[12],over2);

FA FA33(temp2[5],in1[0]&in3[5],over2,temp1[5],over1);

FA FA34(temp2[6],in1[1]&in3[5],over1,temp1[6],over2);

FA FA35(temp2[7],in1[2]&in3[5],over2,temp1[7],over1);

FA FA36(temp2[8],in1[3]&in3[5],over1,temp1[8],over2);

FA FA37(temp2[9],in1[4]&in3[5],over2,temp1[9],over1);

FA FA38(temp2[10],in1[5]&in3[5],over1,temp1[10],over2);

FA FA39(temp2[11],in1[6]&in3[5],over2,temp1[11],over1);

FA FA40(temp2[12],in1[7]&in3[5],over1,temp1[12],over2);

FA FA41(temp1[6],in1[0]&in3[6],over2,temp2[6],over1);

FA FA42(temp1[7],in1[1]&in3[6],over1,temp2[7],over2);

FA FA43(temp1[8],in1[2]&in3[6],over2,temp2[8],over1);

FA FA44(temp1[9],in1[3]&in3[6],over1,temp2[9],over2);

FA FA45(temp1[10],in1[4]&in3[6],over2,temp2[10],over1);

FA FA46(temp1[11],in1[5]&in3[6],over1,temp2[11],over2);

FA FA47(temp1[12],in1[6]&in3[6],over2,temp2[12],over1);

FA FA48(temp1[13],in1[7]&in3[6],over1,temp2[13],over2);

FA FA49(temp1[7],in1[0]&in3[7],over2,temp2[7],over1);

FA FA50(temp1[8],in1[1]&in3[7],over1,temp2[8],over2);

FA FA51(temp1[9],in1[2]&in3[7],over2,temp2[9],over1);

FA FA52(temp1[10],in1[3]&in3[7],over1,temp2[10],over2);

FA FA53(temp1[11],in1[4]&in3[7],over2,temp2[11],over1);

FA FA54(temp1[12],in1[5]&in3[7],over1,temp2[12],over2);

FA FA55(temp1[13],in1[6]&in3[7],over2,temp2[13],over1);

FA FA56(temp1[14],in1[7]&in3[7],over1,temp2[14],over2);

assign temp2[15] = over2;

assign product[15:0] = temp2[15:0];  
  
endmodule

endmodule