module array(s,a,b);

output[15:0] s;

input [7:0]a,b;

wire [63:1]w;

wire [42:1]u;

wire [55:1]c;

//s0

and g8(s[0],a[0],b[0]);

//s1

and g9(w[1],a[1],b[0]);

and g10(w[2],a[0],b[1]);

half h1(s[1],c[1],w[1],w[2]);

//s2

and g11(w[3],a[2],b[0]);

and g12(w[4],a[1],b[1]);

and g13(w[5],a[0],b[2]);

full h2(u[1],c[2],w[3],w[4],w[5]);

half h3(s[2],c[3],c[1],u[1]);

//s3

and g14(w[6],a[3],b[0]);

and g15(w[7],a[2],b[1]);

and g16(w[8],a[1],b[2]);

and g17(w[9],a[0],b[3]);

full h4(u[2],c[4],w[6],w[7],w[8]);

full h5(u[3],c[5],w[9],c[2],c[3]);

half h6(s[3],c[6],u[2],u[3]);

//s4

and g18(w[10],a[4],b[0]);

and g19(w[11],a[3],b[1]);

and g20(w[12],a[2],b[2]);

and g21(w[13],a[1],b[3]);

and g22(w[14],a[0],b[4]);

full h7(u[4],c[7],c[4],c[5],c[6]);

full h8(u[5],c[8],w[10],c[11],c[12]);

half h9(u[6],c[9],w[13],w[14]);

full h10(s[4],c[10],u[4],u[5],u[6]);

//s5

and g23(w[15],a[5],b[0]);

and g24(w[16],a[4],b[1]);

and g25(w[17],a[3],b[2]);

and g26(w[18],a[2],b[3]);

and g27(w[19],a[1],b[4]);

and g28(w[20],a[0],b[5]);

full h11(u[7],c[11],c[7],c[8],c[9]);

full h12(u[8],c[12],w[15],w[16],c[17]);

full h13(u[9],c[13],w[18],w[19],c[20]);

full h14(u[10],c[14],u[8],u[9],u[7]);

half h15(s[5],c[15],u[10],c[10]);

//s6

and g29(w[21],a[6],b[0]);

and g30(w[22],a[5],b[1]);

and g31(w[23],a[4],b[2]);

and g32(w[24],a[3],b[3]);

and g33(w[25],a[2],b[4]);

and g34(w[26],a[1],b[5]);

and g35(w[27],a[0],b[6]);

full h16(u[11],c[16],w[21],w[22],w[23]);

full h17(u[12],c[17],w[24],w[25],w[26]);

full h18(u[13],c[18],w[27],c[11],c[12]);

full h19(u[14],c[19],c[13],c[14],c[15]);

full h20(u[15],c[20],u[11],u[12],u[13]);

half h21(s[6],c[21],u[15],u[14]);

//s7

and g36(w[28],a[7],b[0]);

and g37(w[29],a[6],b[1]);

and g38(w[30],a[5],b[2]);

and g39(w[31],a[4],b[3]);

and g40(w[32],a[3],b[4]);

and g41(w[33],a[2],b[5]);

and g42(w[34],a[1],b[6]);

and g43(w[35],a[0],b[7]);

full h22(u[16],c[22],w[28],w[29],w[30]);

full h23(u[17],c[23],w[31],w[32],w[33]);

full h24(u[18],c[24],w[34],w[35],c[16]);

full h25(u[19],c[25],c[17],c[18],c[19]);

full h26(u[20],c[26],c[20],c[21],u[16]);

full h27(u[21],c[27],u[19],u[17],u[18]);

half h28(s[7],c[28],u[20],u[21]);

//s8

and g44(w[36],a[7],b[1]);

and g45(w[37],a[6],b[2]);

and g46(w[38],a[5],b[3]);

and g47(w[39],a[4],b[4]);

and g48(w[40],a[3],b[5]);

and g49(w[41],a[2],b[6]);

and g50(w[42],a[1],b[7]);

full h29(u[22],c[29],c[22],c[23],c[24]);

full h30(u[23],c[30],c[27],c[26],c[25]);

full h31(u[24],c[31],c[28],w[36], w[37]);

full h32(u[25],c[32],w[40],w[38],w[39]);

full h33(u[26],c[33],u[22],w[41],w[42]);

full h34(u[27],c[34],u[25],u[23],u[24]);

half h35(s[8],c[35],u[27],u[26]);

//s9

and g51(w[43],a[7],b[2]);

and g52(w[44],a[6],b[3]);

and g53(w[45],a[5],b[4]);

and g54(w[46],a[4],b[5]);

and g55(w[47],a[3],b[6]);

and g56(w[48],a[2],b[7]);

full h36(u[28],c[36],w[43],w[44],w[45]);

full h37(u[29],c[37],w[46],w[47],w[48]);

full h38(u[30],c[38],c[29],c[30],c[31]);

full h39(u[31],c[39],c[32],u[33],u[34]);

full h40(u[32],c[40],c[35],u[28],u[29]);

full h41(s[9],c[41],u[31],u[30],u[32]);

//s10

and g57(w[49],a[7],b[3]);

and g58(w[50],a[6],b[4]);

and g59(w[51],a[5],b[5]);

and g60(w[52],a[4],b[6]);

and g61(w[53],a[3],b[7]);

full h42(u[33],c[42],c[38],c[37],c[36]);

full h43(u[34],c[43],c[39],c[40],c[41]);

full h44(u[35],c[44],w[51],w[50],w[49]);

full h45(u[36],c[45],w[52],w[53],u[33]);

full h46(s[10],c[46],u[36],u[35],u[34]);

//s11

and g62(w[54],a[7],b[4]);

and g63(w[55],a[6],b[5]);

and g64(w[56],a[5],b[6]);

and g65(w[57],a[4],b[7]);

full h47(u[37],c[47],w[54],w[55],w[56]);

full h48(u[38],c[48],c[42],c[43],w[57]);

full h49(u[39],c[49],c[44],c[45],c[46]);

full h50(s[11],c[50],u[37],u[38],u[39]);

//s12

and g66(w[58],a[7],b[5]);

and g67(w[59],a[6],b[6]);

and g68(w[60],a[5],b[7]);

full h51(u[40],c[51],w[58],w[59],w[60]);

full h52(u[41],c[52],c[47],c[48],w[49]);

full h53(s[12],c[53],u[40],u[41],c[50]);

//s13

and g69(w[61],a[7],b[6]);

and g70(w[62],a[6],b[7]);

full h54(u[42],c[54],w[61],w[62],c[51]);

full h55(s[13],c[55],c[52],c[53],u[42]);

//s14 and s[15]

and g71(w[63],a[7],b[7]);

full h56(s[14], s[15],c[54],c[55],w[63]);

endmodule

module half(s,c,a,b);

output s,c;

input a,b;

and g1(c,a,b);

xor g2(s,a,b);

endmodule

module full(s,c,a,b,d);

output s,c;

input a,b,d;

wire z1,z2,z3;

xor g3(s,a,b,d);

and g4(z1,a,b);

and g5(z2,a,d);

and g6(z3,d,b);

or g7(c,z1,z2,z3);

endmodule

module atb;

wire [15:0] s;

reg [7:0]a,b;

reg clk;

array uut (s,a,b);

initial

begin

#5 clk=1'b0;

#5 clk=1'b1;

end

initial

begin

$dumpfile("dump.vcd");

$dumpvars(1);

end

always @(posedge clk)

begin

#10 a[7:0]=8'b11111111;b[7:0]=8'b11111111;

#10 $stop;

end

endmodule