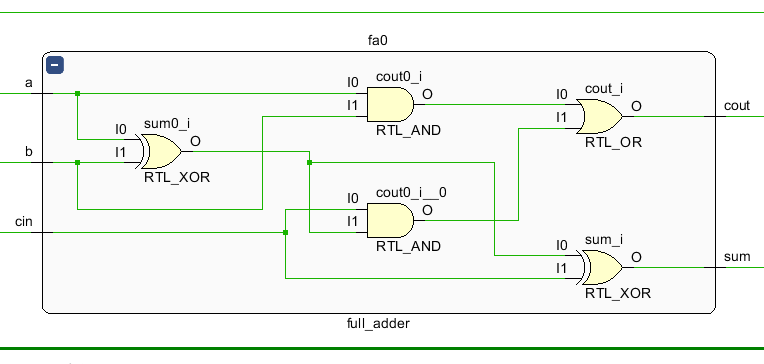
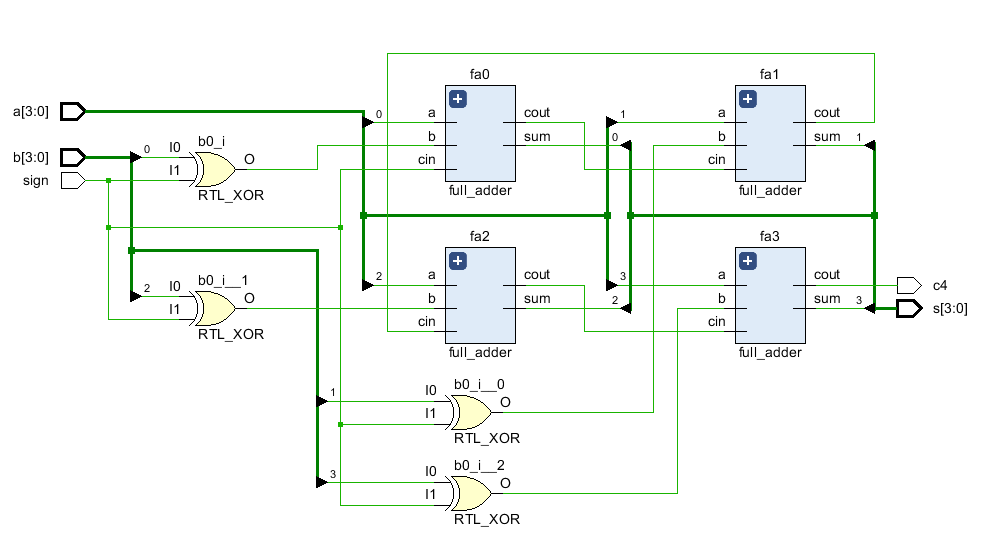
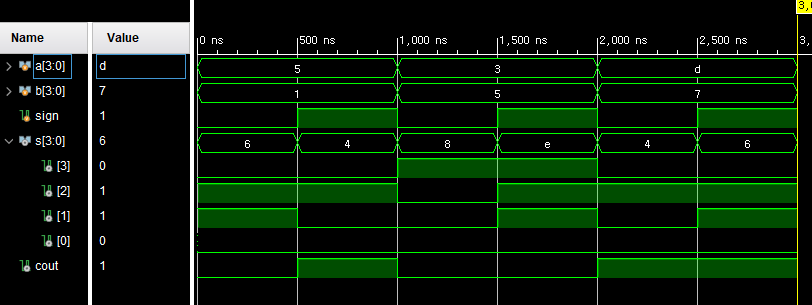
Fa

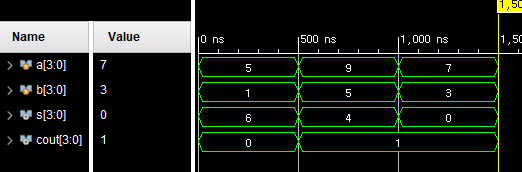


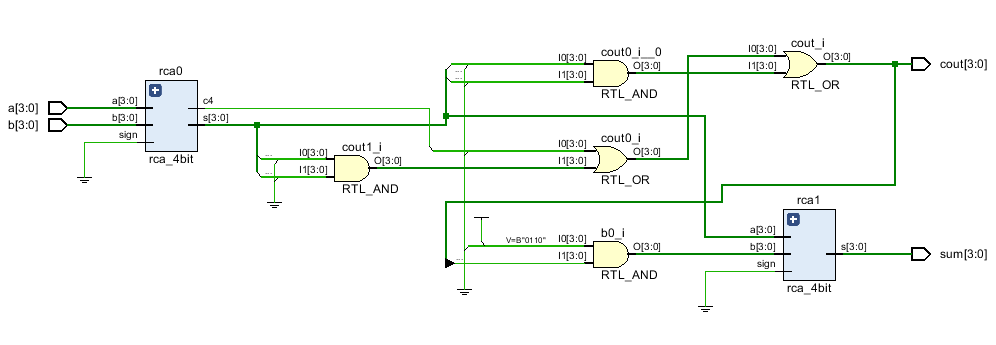
rca





Bcd adder





`timescale 1ns / 1ps

module full\_adder(a, b, cin, sum, cout);

input a, b, cin;

output sum, cout;

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

assign sum = (a^b)^cin;

endmodule

module rca\_4bit(a, b, sign, s, c4);

input [3:0] a, b;

output [3:0] s;

input sign;

output c4;

wire c1, c2, c3;

full\_adder fa0(a[0], b[0]^sign, sign, s[0], c1);

full\_adder fa1(a[1], b[1]^sign, c1, s[1], c2);

full\_adder fa2(a[2], b[2]^sign, c2, s[2], c3);

full\_adder fa3(a[3], b[3]^sign, c3, s[3], c4);

endmodule

module bcd\_adder(a, b, cin, sum, cout);

input [3:0] a, b;

input cin;

output [3:0] sum, cout;

wire carryOut;

wire [3:0] binarySum;

rca\_4bit rca0(a, b, cin, binarySum, carryOut);

assign cout = carryOut|(binarySum[3]&binarySum[2])|(binarySum[3]&binarySum[1]);

rca\_4bit rca1(binarySum, 4'b0110 & {4{cout[0]}}, 0, sum);

endmodule

`timescale 1ns / 1ps

module rcasim;

reg [3:0] a, b;

reg sign;

wire [3:0] s;

wire cout;

initial a = 4'b0101;

initial b = 4'b0001;

initial sign = 1'b0;

rca\_4bit rca4(a, b, sign, s, cout);

initial begin

#500

assign sign = 1'b1;

#500

assign sign = 1'b0;

assign a=4'b0011;

assign b=4'b0101;

#500

assign sign = 1'b1;

#500

assign sign = 1'b0;

assign a=4'b1101;

assign b=4'b0111;

#500

assign sign = 1'b1;

#500

$finish;

end

endmodule

module bcdsim;

reg [3:0] a, b;

wire [3:0] s, cout;

bcd\_adder bcd(a, b, 0, s, cout);

initial a = 4'b0101; //BCD 5

initial b = 4'b0001; //BCD 1

// 5+1 = 6

initial begin

#500

assign a=4'b1001; //BCD 9

assign b=4'b0101; //BCD 5

// 9+5 = 14

#500

assign a=4'b0111; //BCD 7

assign b=4'b0011; //BCD 3

// 7+3 = 10

#500

$finish;

end

endmodule