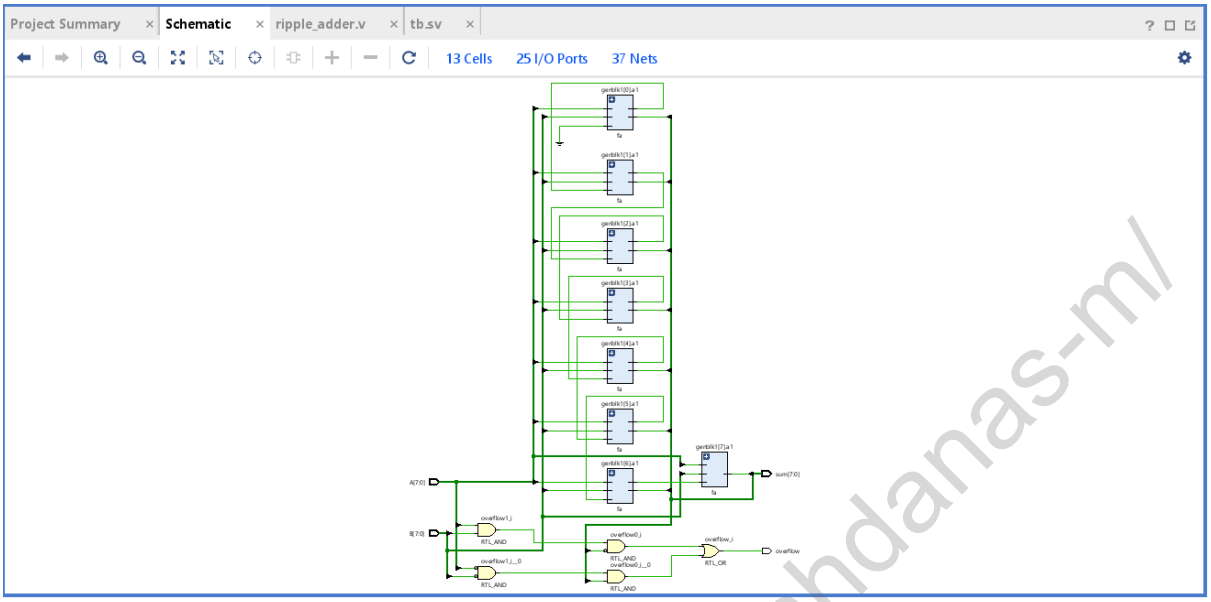
Problem

Assume that you have two 8-bit 2's complement numbers, a[7:O] and b[7:O]. These numbers are added to produce s[7:O]. Also compute whether a (signed) overflow has occurred.

Design

```
module fa( input a,b,c,
output sum,carr
 );
// assign {carr,sum}=a+b+c;
wire w1,w2,w3;
xor a1(sum,a,b,c);
and a2(w1,a,b),
a4 (w2,a,c),
a3 (w3,b,c);
or a5(carr,w1,w2,w3);


endmodule

module ripple_adder#(parameter Bitwidth=8)( input [Bitwidth-1:0] A,B,
                output[Bitwidth-1:0] sum,
                output overflow );
    wire [Bitwidth:0]c;
assign c[0]=0;

    generate
        genvar i;
        for (i = 0; i < Bitwidth; i = i + 1) begin
         fa a1(A[i],B[i],c[i],sum[i],c[i+1]);
        end
    endgenerate

 assign overflow =(A[7]&B[7]&~sum[7])|(~A[7]&~B[7]&sum[7]) ;

endmodule
```

Circuit



Testbench

```
module day15_tb;
reg  [7:0] a;
reg [7:0] b;
wire [7:0] s;
wire overflow;
  ripple_adder uut (a,b,s,overflow); //instantiation

initial begin
    a = 8'd51; b = 8'd65;    //both positive but sum less than 127
     #10                     //over flow not occured
    a = 8'd124;  b = 8'd12;  //both positive but sum greater than 127
      #10                   //overflow occured

    a = -8'd46;  b = -8'd99;  //both negative but sum below -127
      #10                    //overflow occured

    a = -8'd60;  b = 8'd40;   //one positive and one negative but sum
within -127 to +127
     #10                     //overflow not occured
    $finish;
end
endmodule
```

Waveform