

## Problem

Create a 100-bit binary adder. The adder adds two 100-bit numbers and a carry-in to produce a 100-bit sum and carry out.

## 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

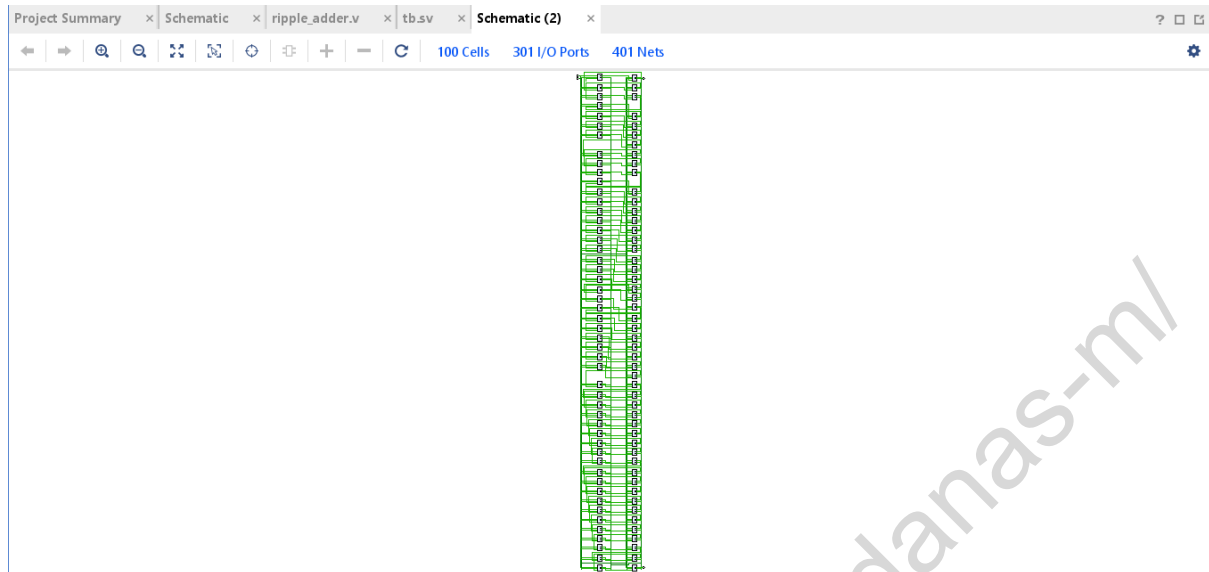
module ripple_adder#(parameter Bitwidth=100)( input [Bitwidth-1:0] A,B,
output[Bitwidth-1:0] sum,
output carry );
    wire [Bitwidth:0]c;
    assign c[0]=0;

    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 carry = c[100] ;

endmodule
```

## Circuit



## Testbench

```
module day16_tb;
reg [99:0] a;
reg [99:0] b;
wire [99:0] s;
wire carry;
    ripple_adder uut (a,b,s,carry); //instantiation

initial begin
repeat(10) begin
a=$random;b=$random;
#10;
end
    $finish;
end
endmodule
```

## Waveform

