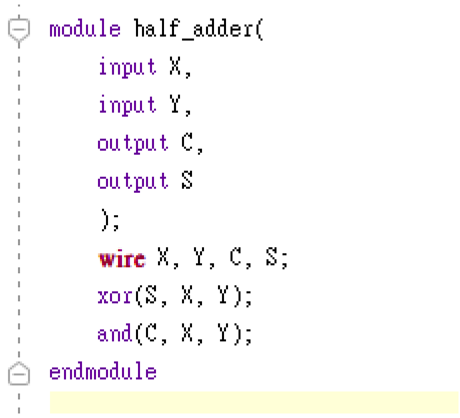
Lab 1 report

Name: Oscar Stark   Student ID: F11015127

In this lab I implemented a half adder and a full adder.

Half adder

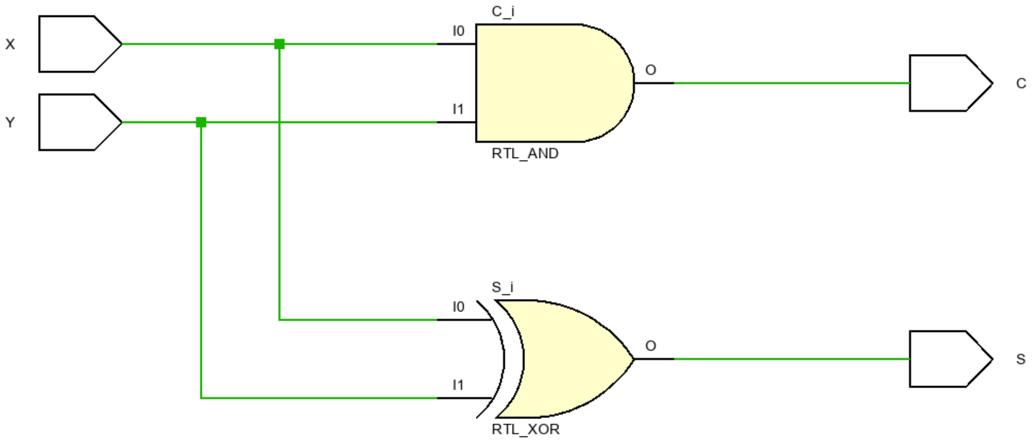
Half adder code



```
module half_adder(
    input X,
    input Y,
    output C,
    output S
    );
    wire X, Y, C, S;
    xor(S, X, Y);
    and(C, X, Y);
endmodule
```

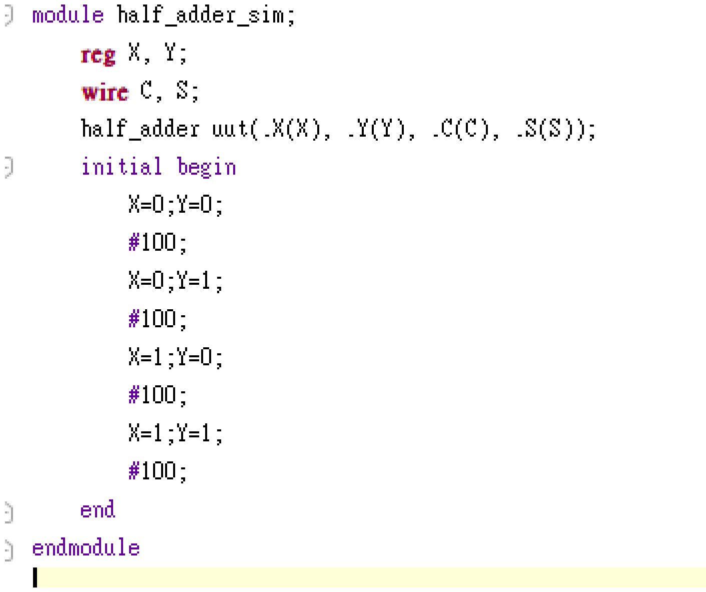
X and Y are the bits being added, C is the carry and S is the sum.

Schematic



The half adder uses 1 AND gate and 1 XOR gate.

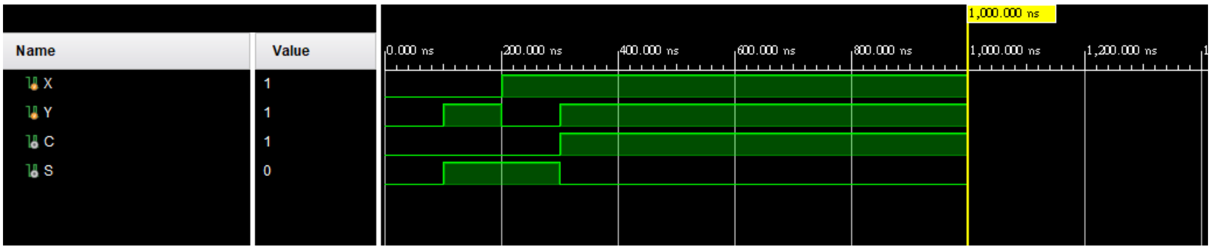
Test bench



```
module half_adder_sim;
    reg X, Y;
    wire C, S;
    half_adder uut(.X(X), .Y(Y), .C(C), .S(S));
    initial begin
        X=0;Y=0;
        #100;
        X=0;Y=1;
        #100;
        X=1;Y=0;
        #100;
        X=1;Y=1;
        #100;
    end
endmodule
```

All possible inputs are simulated.

Simulation results

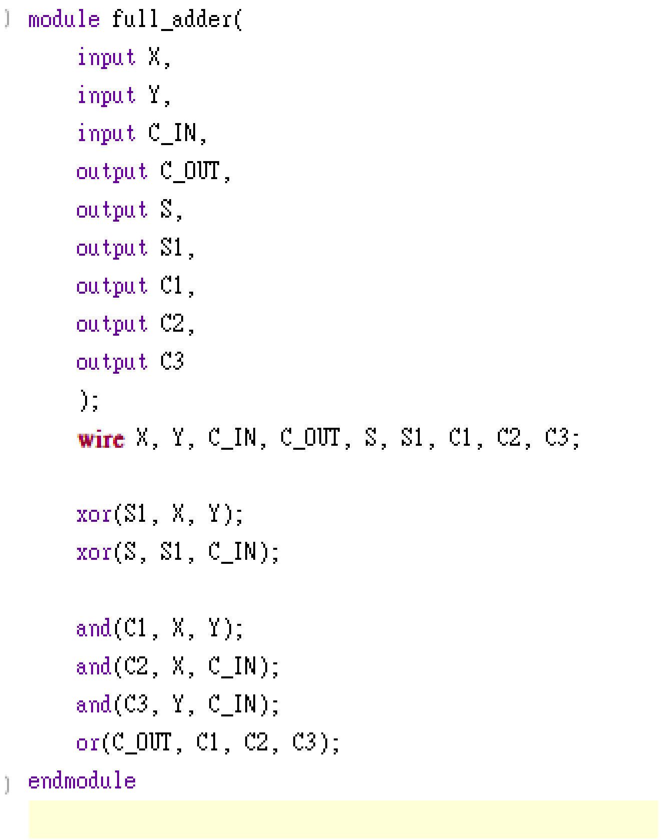


Name	Value
X	1
Y	1
C	1
S	0

X and Y are the bits being added, C is the carry and S is the sum.

Full adder

Full adder code



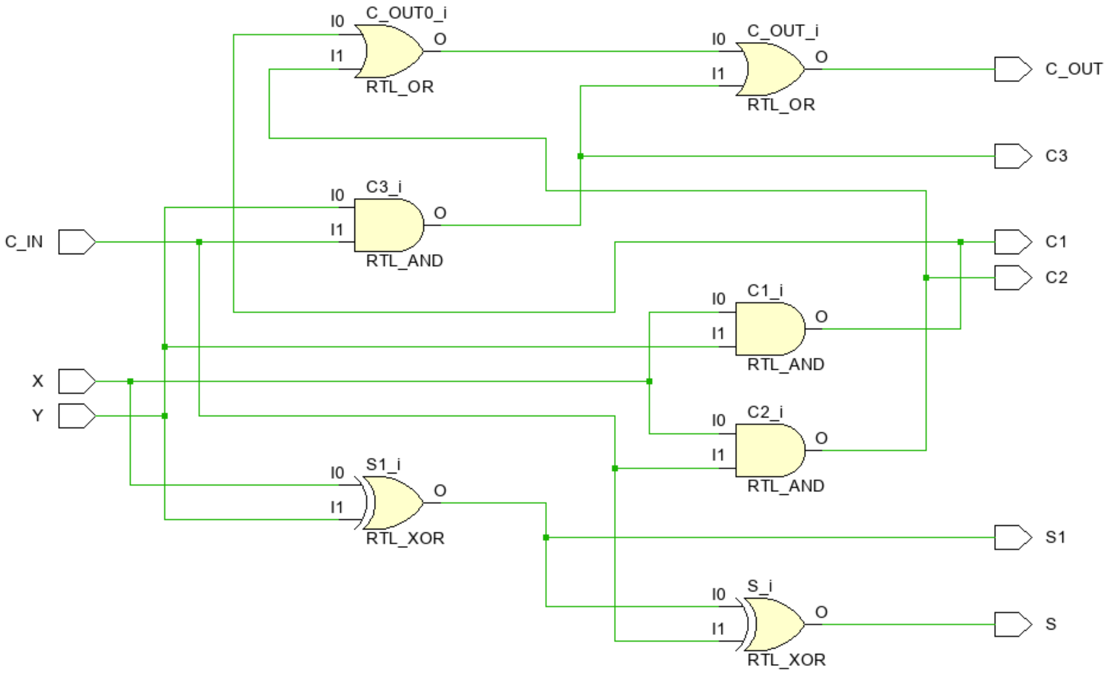
```
module full_adder(
    input X,
    input Y,
    input C_IN,
    output C_OUT,
    output S,
    output S1,
    output C1,
    output C2,
    output C3
    );
    wire X, Y, C_IN, C_OUT, S, S1, C1, C2, C3;

    xor(S1, X, Y);
    xor(S, S1, C_IN);

    and(C1, X, Y);
    and(C2, X, C_IN);
    and(C3, Y, C_IN);
    or(C_OUT, C1, C2, C3);
endmodule
```

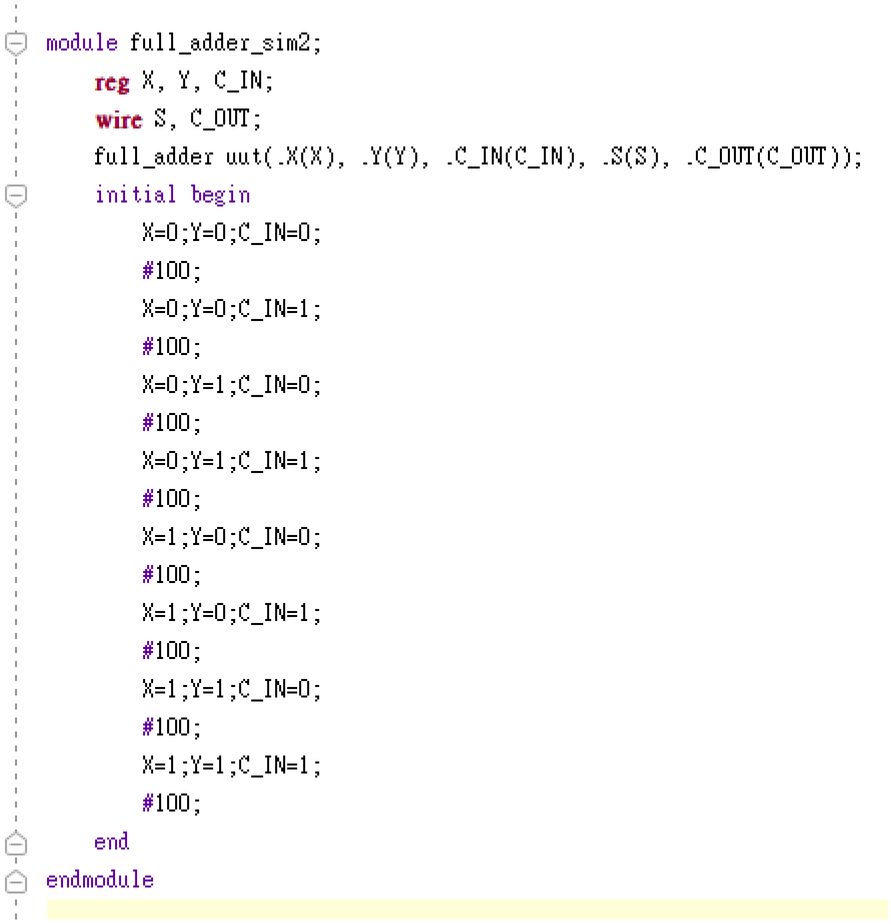
X and Y are the bits being added, C_IN is the carry input, C_OUT is the carry output and S is the sum. S1, C1, C2, C3 are just wires.

Schematic



The circuit has 3 AND gates, 2 XOR gates and 2 OR gates.

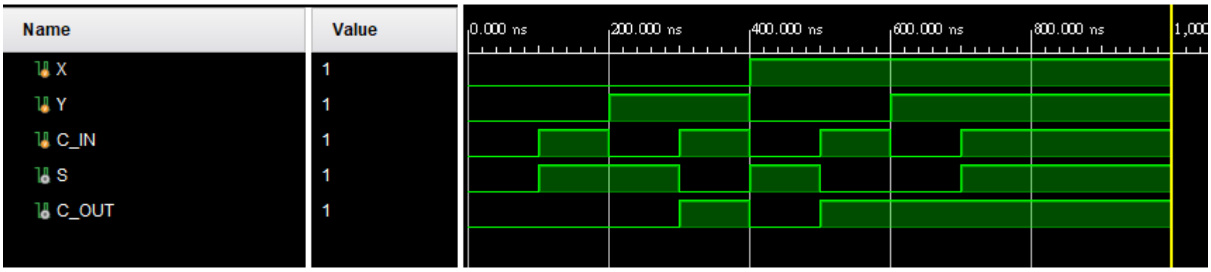
Test bench



```
module full_adder_sim2;
    reg X, Y, C_IN;
    wire S, C_OUT;
    full_adder uut(.X(X), .Y(Y), .C_IN(C_IN), .S(S), .C_OUT(C_OUT));
    initial begin
        X=0;Y=0;C_IN=0;
        #100;
        X=0;Y=0;C_IN=1;
        #100;
        X=0;Y=1;C_IN=0;
        #100;
        X=0;Y=1;C_IN=1;
        #100;
        X=1;Y=0;C_IN=0;
        #100;
        X=1;Y=0;C_IN=1;
        #100;
        X=1;Y=1;C_IN=0;
        #100;
        X=1;Y=1;C_IN=1;
        #100;
    end
endmodule
```

All possible inputs are simulated.

Simulation results



Name	Value
X	1
Y	1
C_IN	1
S	1
C_OUT	1

X, Y and C_IN are the inputs. S and C_OUT are the outputs.