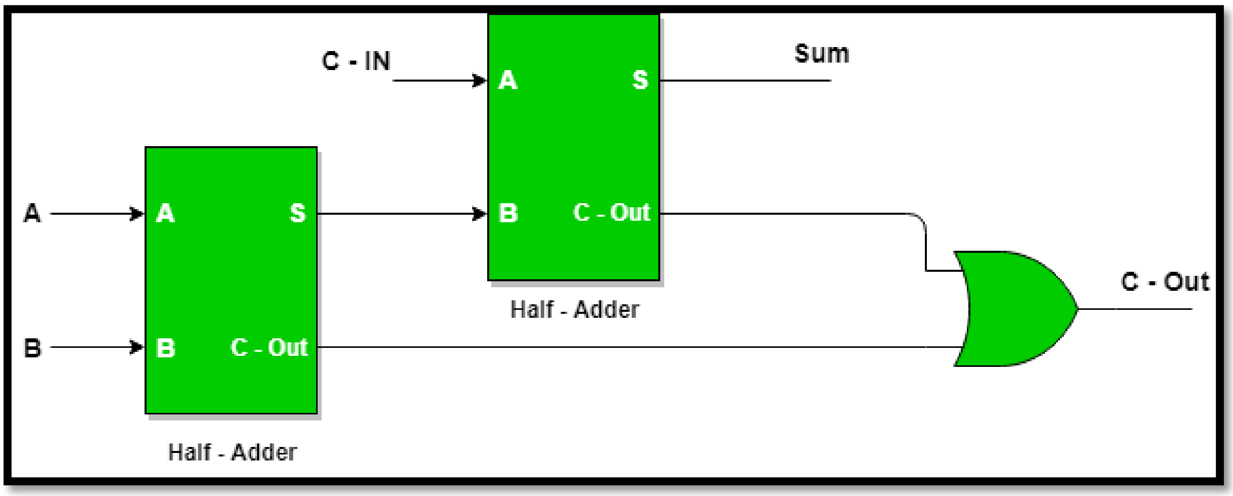
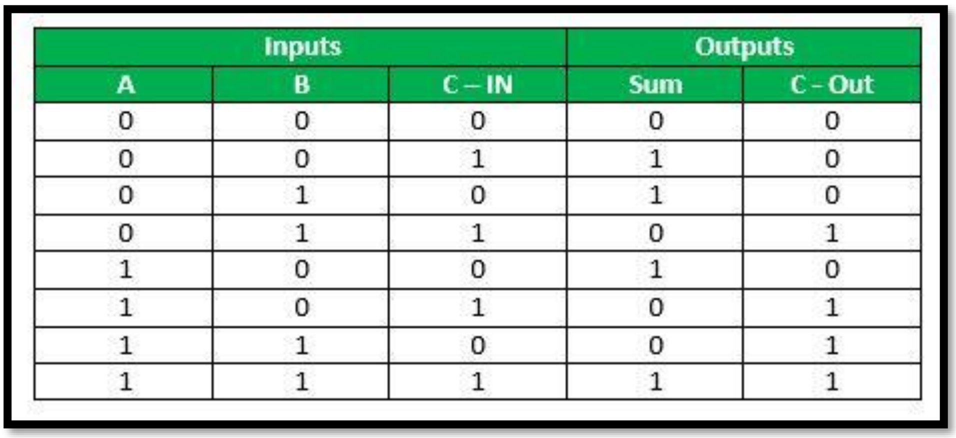
Full adder using two half adders:

In this implementation we need 2 Half Adders, and an OR gate is required to implement a Full Adder. With this logic circuit, two bits can be added together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude.

Logic Symbol:



Truth Table:



Inputs			Outputs	
A	B	C – IN	Sum	C – Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Verilog Code:

```
module full_adder_using_two_half_adder(a,b,c,s,c_out);
input a,b,c;
output s,c_out;
wire s1,c1,c2;

half_adder h1(.a(a),.b(b),.s(s1),.c(c1));

half_adder h2(s1,c,s,c2);
or(c_out,c1,c2);
endmodule
```

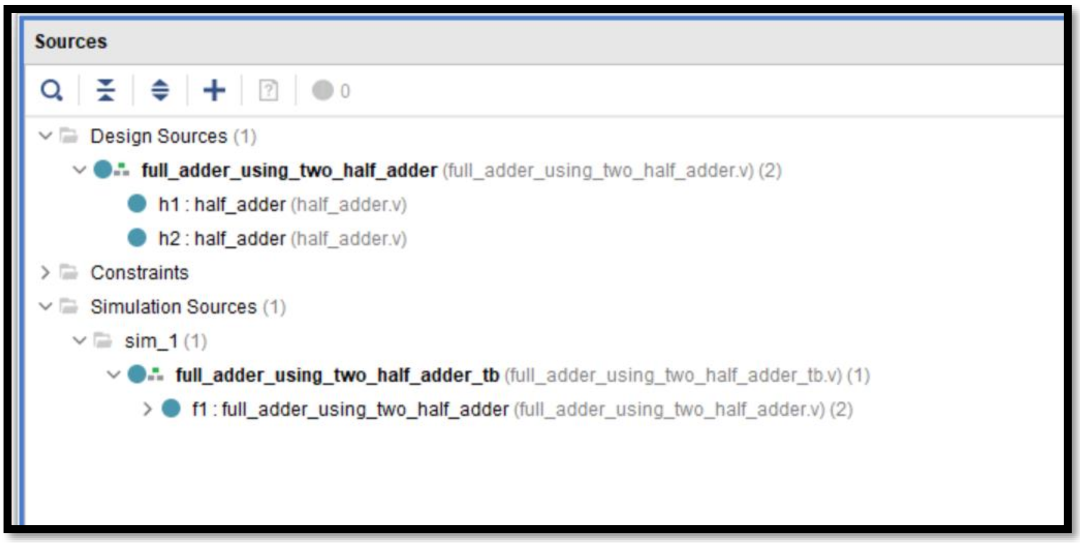
Verilog Code for half adder:

```
module half_adder(a,b,s,c);
input a,b;
output s,c;
xor(s,a,b);
and(c,a,b);
endmodule
```

Testbench:

```
module full_adder_using_two_half_adder_tb();
reg a,b,c;
wire s,c_out;
full_adder_using_two_half_adder f1(.a(a),.b(b),.c(c),.s(s),.c_out(c_out));
initial
begin
a=0;b=0;c=0;
#100
a=0;b=0;c=1;
#100
a=0;b=1;c=0;
#100
a=0;b=1;c=1;
#100
a=1;b=0;c=0;
#100
a=1;b=0;c=1;
#100
a=1;b=1;c=0;
#100
a=1;b=1;c=1;
#100 $finish;
end
initial
begin
$display("\t\t\t\t Time \ta \t b \t c \t s \t c_out");
$monitor($time,"\t",a,"\t",b,"\t",c,"\t",s,"\t", c_out);
end
endmodule
```

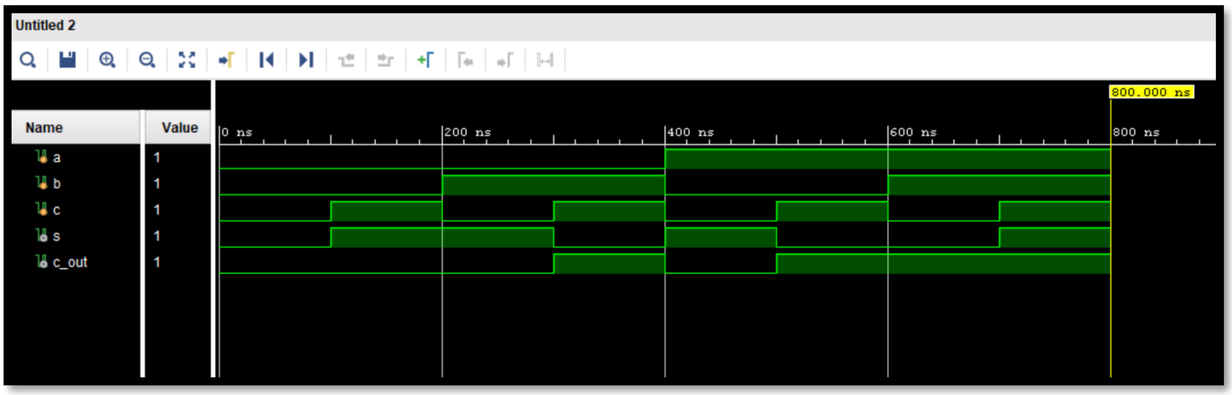
Sources:



Sources

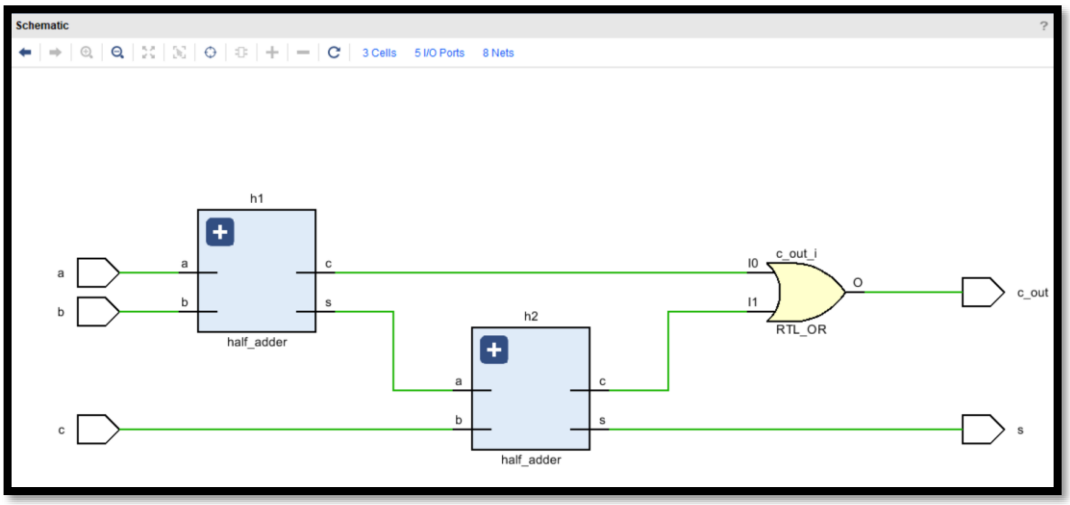
- Design Sources (1)
  - full_adder_using_two_half_adder (full_adder_using_two_half_adder.v) (2)
    - h1 : half_adder (half_adder.v)
    - h2 : half_adder (half_adder.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - full_adder_using_two_half_adder_tb (full_adder_using_two_half_adder_tb.v) (1)
      - f1 : full_adder_using_two_half_adder (full_adder_using_two_half_adder.v) (2)

Simulation:

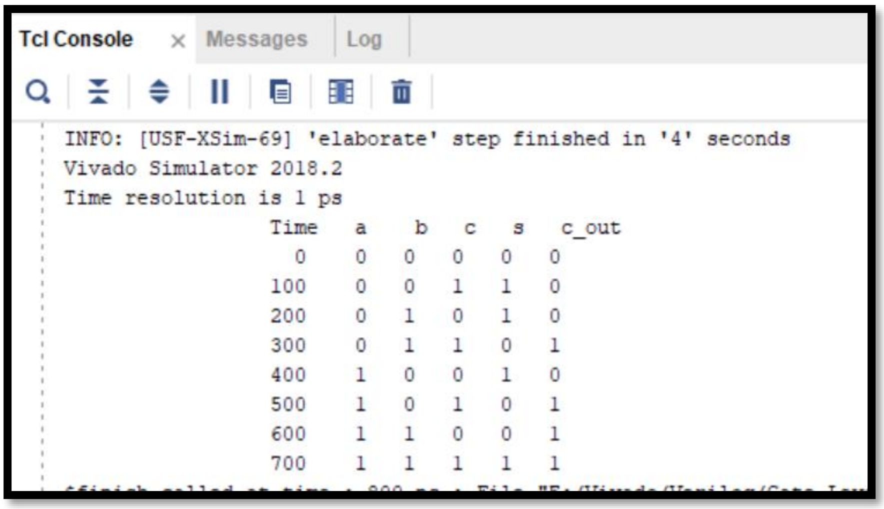


Name	Value
a	1
b	1
c	1
s	1
c_out	1

RTL Design:



Tcl Console:



```
INFO: [USF-XSim-69] 'elaborate' step finished in '4' seconds
Vivado Simulator 2018.2
Time resolution is 1 ps
                    Time   a    b   c   s   c_out
                      0    0    0   0   0   0
                    100    0    0   1   1   0
                    200    0    1   0   1   0
                    300    0    1   1   0   1
                    400    1    0   0   1   0
                    500    1    0   1   0   1
                    600    1    1   0   0   1
                    700    1    1   1   1   1
```

Application:

1.Arithmetic circuits: Full adders are utilized in math circuits to add twofold numbers. At the point when different full adders are associated in a chain, they can add multi-bit paired numbers.
2.Data handling: Full adders are utilized in information handling applications like advanced signal handling, information encryption, and mistake rectification.
3.Counters: Full adders are utilized in counters to addition or decrement the count by one.
4.Multiplexers and demultiplexers: Full adders are utilized in multiplexers and demultiplexers to choose and course information.
5.Memory tending to: Full adders are utilized in memory addressing circuits to produce the location of a particular memory area.
6.ALUs: Full adders are a fundamental part of Number juggling Rationale Units (ALUs) utilized in chip and computerized signal processors.