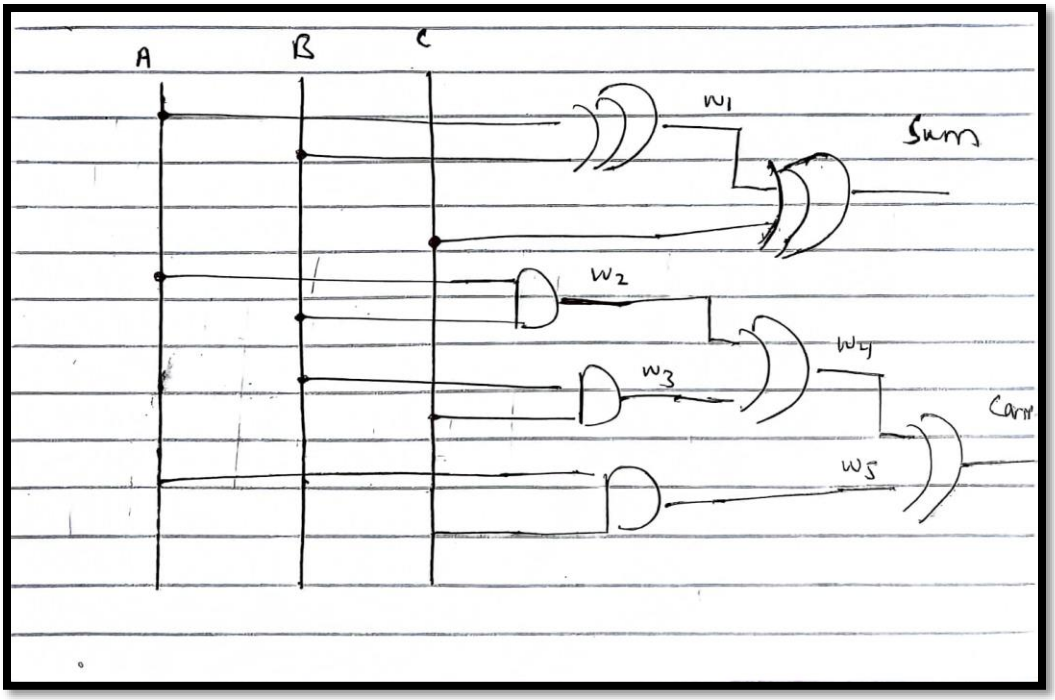
Full adder:

Full Adder is the adder that adds three inputs and produces two outputs. The first two inputs are A and B, and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. The C-OUT is also known as the majority 1's detector, whose output goes high when more than one input is high. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another. we use a full adder because when a carry-in bit is available, another 1-bit adder must be used since a 1-bit half-adder does not take a carry-in bit. A 1-bit full adder adds three operands and generates 2-bit results.

Logic Circuit:



Truth Table:

A	B	C	Sum	Carry
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

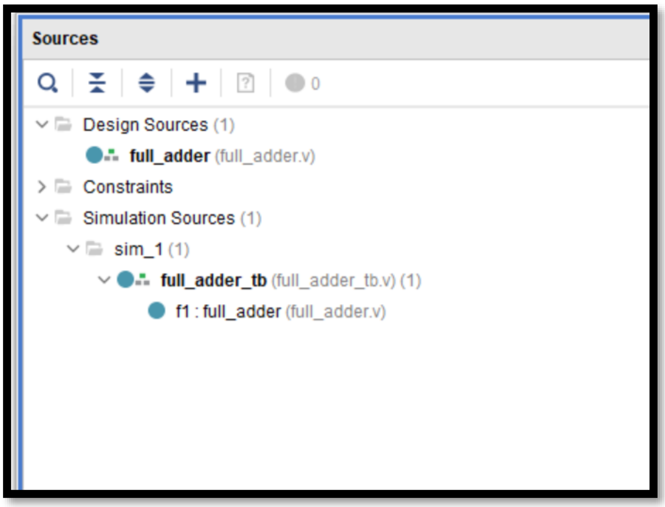
Verilog Code:

```
module full_adder(a,b,c,s,c_out);
input a,b,c;
output s,c_out;
wire w1,w2,w3,w4;
// for sum
assign w1=a^b;
assign s=w1^c;
// for carry
assign w2=a&b;
assign w3=b&c;
assign w4=a&c;
assign c_out=w2|w3|w4;
endmodule
```

Testbench:

```
module full_adder_tb();
reg x,y,z;
wire s,c_out;
full_adder f1(x,y,z,s,c_out);
initial
begin
x=0;y=0;z=0;
#100
x=0;y=0;z=1;
#100
x=0;y=1;z=0;
#100
x=0;y=1;z=1;
#100
x=1;y=0;z=0;
#100
x=1;y=0;z=1;
#100
x=1;y=1;z=0;
#100
x=1;y=1;z=1;
#100 $finish;
end
initial
begin
$display("\t\t\t\ttime\tx\ty\tz\ts\tc_out");
$monitor($time,"\t",x,"\t",y,"\t",z,"\t",s,"\t",c_out);
end
endmodule
```

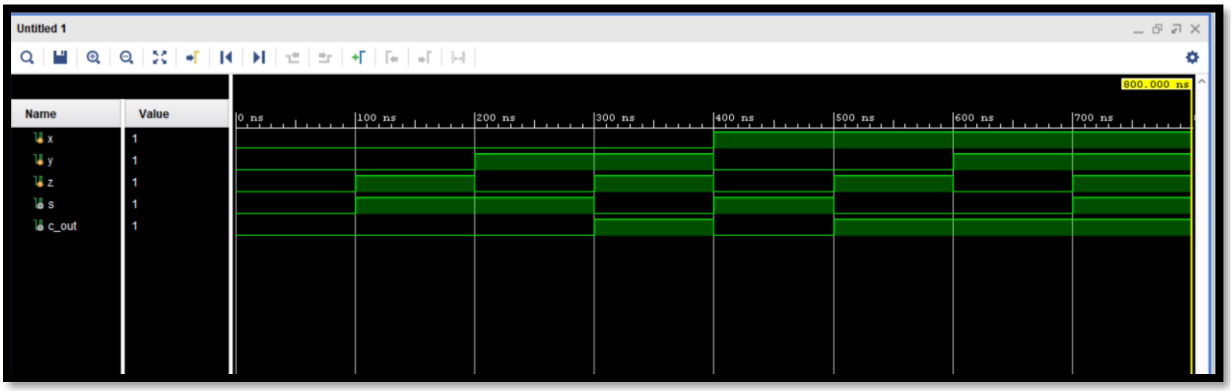
Sources:



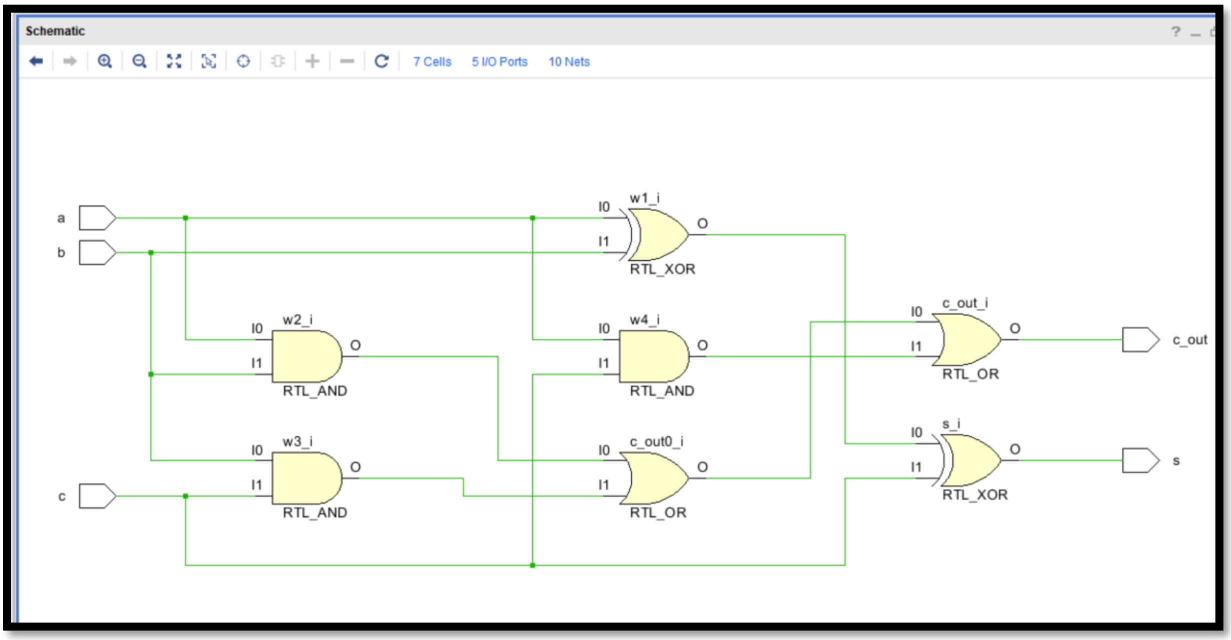
Sources

Design Sources (1)
full_adder (full_adder.v)
Constraints
Simulation Sources (1)
sim_1 (1)
full_adder_tb (full_adder_tb.v) (1)
f1 : full_adder (full_adder.v)

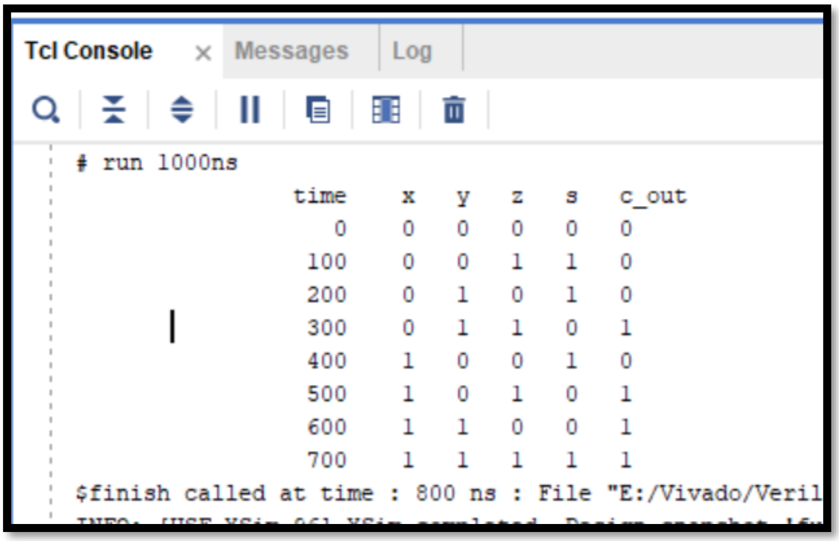
Simulation:



RTL Design:



Tcl Console:



```
# run 1000ns
                    time    x    y    z    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
$finish called at time : 800 ns : File "E:/Vivado/Veril
```

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.