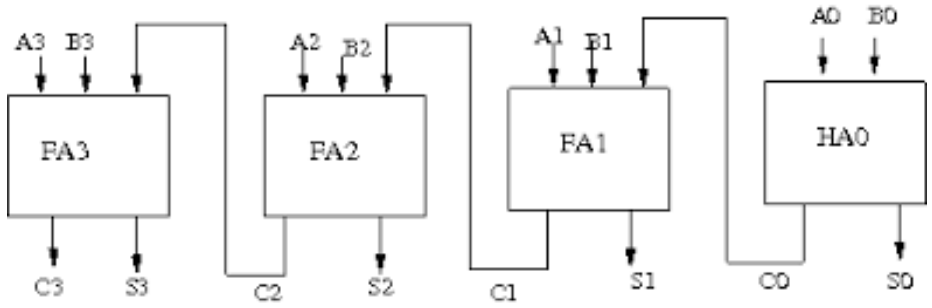
Ripple Carry Adder:

A Ripple-Carry Adder is defined as a method of constructing an N-bit carry propagate adder by chaining together N full adders, where the carry out of one stage acts as the carry in of the next stage. This approach is modular and regular but can be slow for large N due to the carry rippling through the chain.But in our case, we want to design 4-bit ripple carry adder so we did not take C_in and use only one half adder and three full adder.

Circuit Diagram:



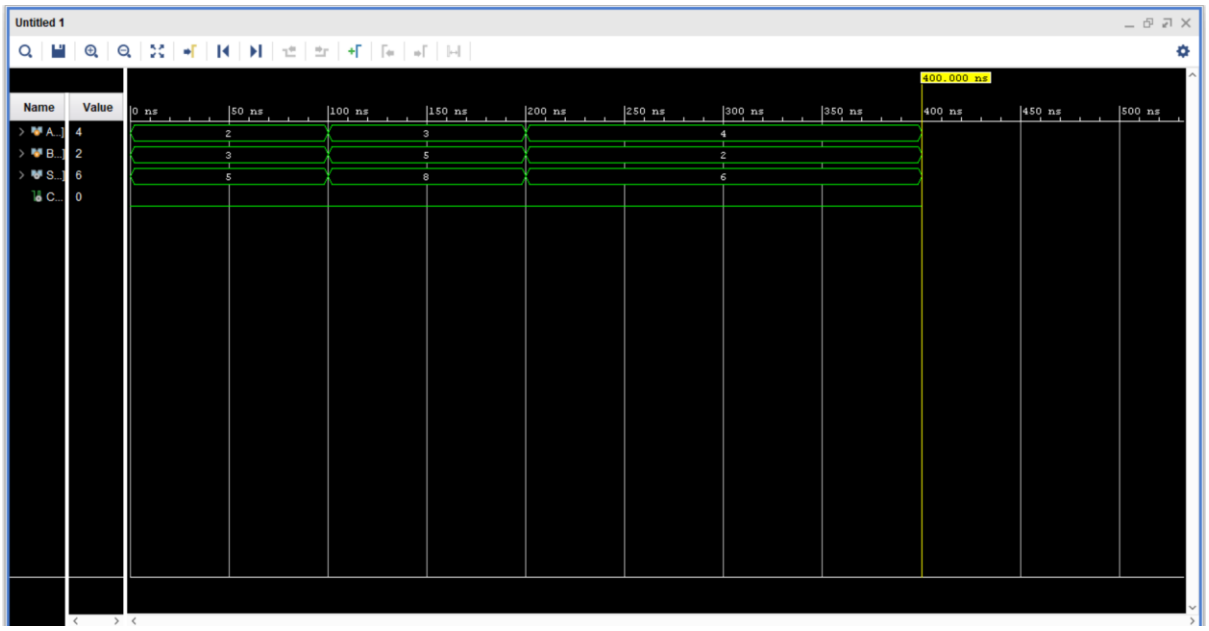
Verilog Code:	**Testbench:**

moduleripple_carry_4bit_adder(a,b,s,c_out);
input [3:0] a,b;
output [3:0]s;
output c_out;
wire c1,c2,c3;
half_adder h1(a[0],b[0],s[0],c1);
full_adder f1(a[1],b[1],c1,s[1],c2);
full_adder f2(a[2],b[2],c2,s[2],c3);
full_adder f3(a[3],b[3],c3,s[3],c_out);
endmodule
	module
	ripple_carry_4bit_adder_tb();
	reg [3:0] A,B;
	wire [3:0] S;
	wire C_out;
	ripple_carry_4bit_adder
	A1(A,B,S,C_out);
	initial
	begin
	A=4'd2;B=4'd3;
	#100
	$display("Time=%d A=%b B=%b
	S=%b
	C_out=%b\n",$time,A,B,S,C_out);
	A=4'd9;B=4'd9;
	#100
	$display("Time=%d A=%b B=%b
	S=%b
	C_out=%b\n",$time,A,B,S,C_out);
	A=4'd5;B=4'd8;
	#100
	$display("Time=%d A=%b B=%b
	S=%b
	C_out=%b\n",$time,A,B,S,C_out);
	#100 $stop;
	end
	endmodule

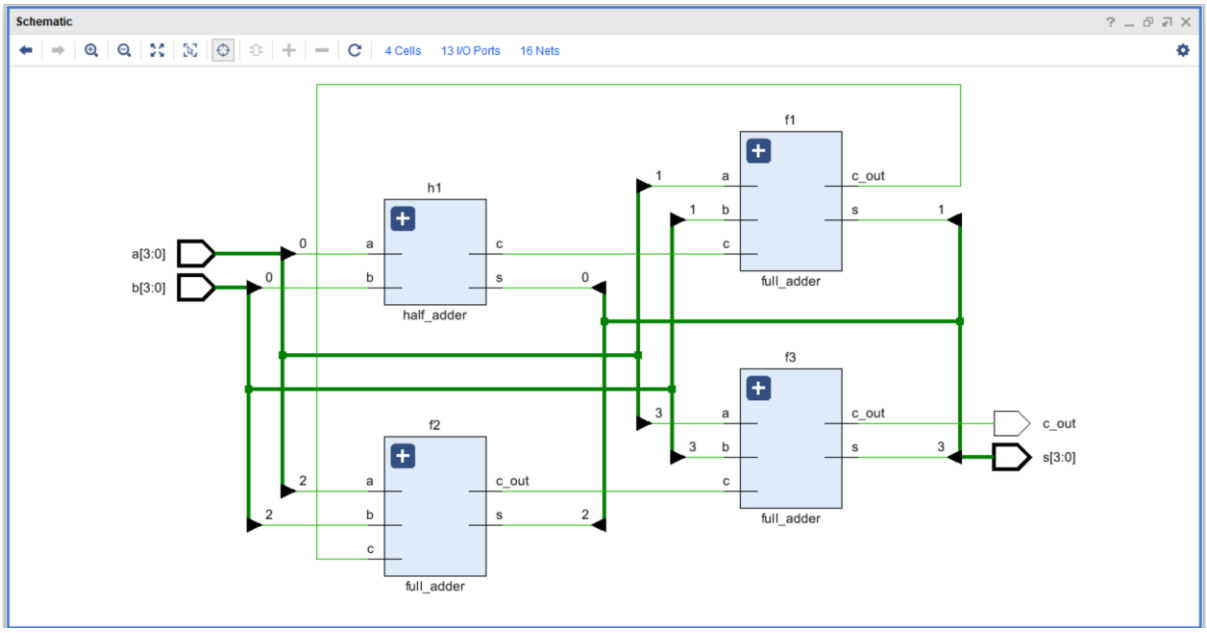
Verilog Code for full adder:	**Verilog Code half adder:**

module full_adder(a,b,c,s,c_out);
input a,b,c;
output s,c_out;
wire w1,w2,w3;
xor(w1,a,b);
and(w2,a,b);
and(w3,c,w1);
xor(s,w1,c);
or(c_out,w2,w3);
endmodule
	module half_adder(a,b,s,c);
	input a,b;
	output s,c;
	xor(s,a,b);
	and(c,a,b);
	endmodule

Simulation:



RTL design:



Tcl Console:

Tcl Console | Messages | Log

```
#      send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator wil
#   }
# }
# run 1000ns
Time=                  100 A=0010 B=0011 S=0101 C_out=0

Time=                  200 A=0011 B=0101 S=1000 C_out=0

Time=                  300 A=0100 B=0010 S=0110 C_out=0

INFO: [USF-XSim-96] XSim completed. Design snapshot 'ripple_carry_4bit_adder_tb_
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:03 ; elapsed = 00:00:07 . Memory (MB):
```