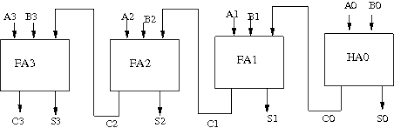
**Ripple Carry Adder:**

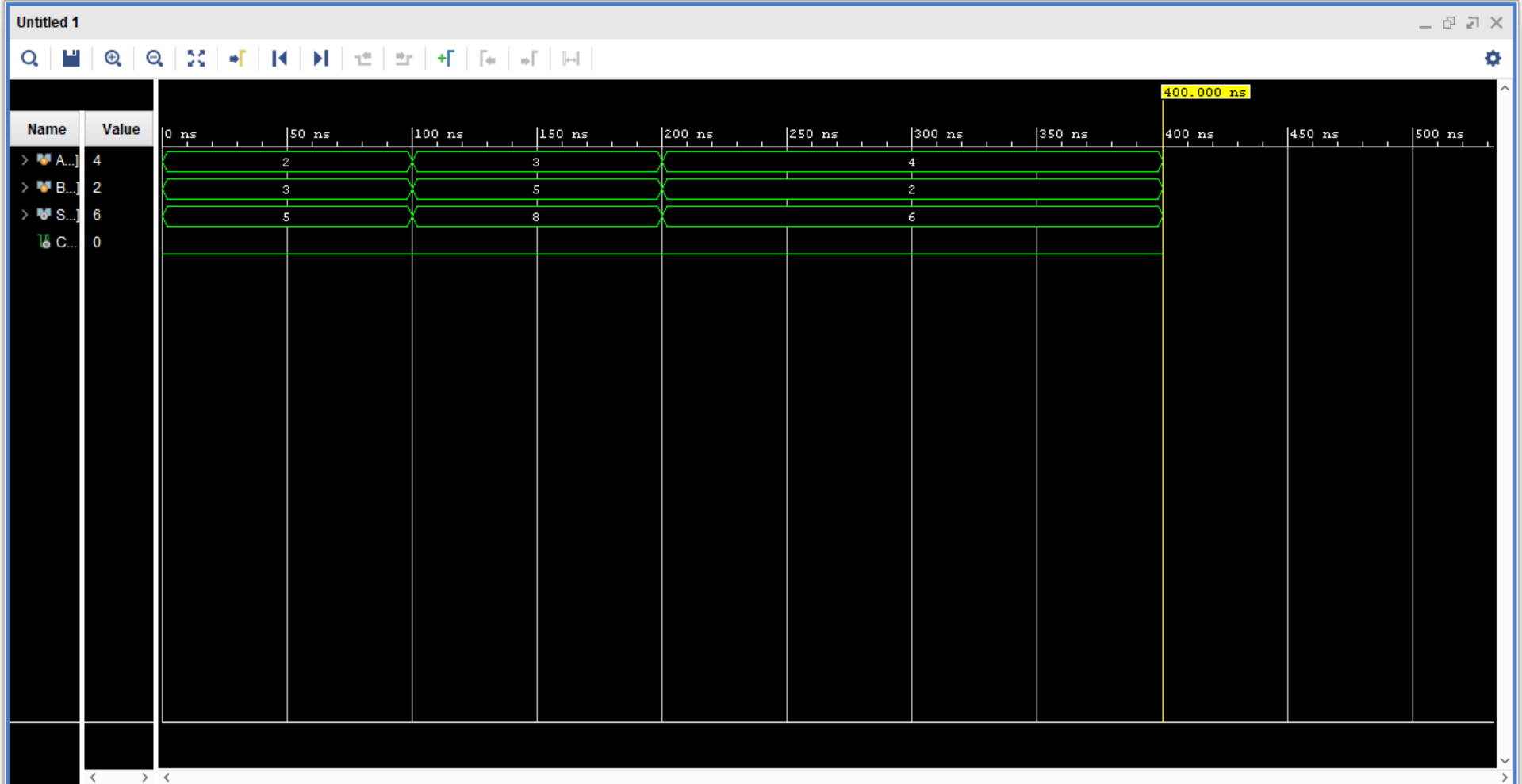
A Ripple-Carry Adder is defined as a method of constructing an N-bit carry propagate adder by chaining together N full adders, where the carry out of one stage acts as the carry in of the next stage. This approach is modular and regular but can be slow for large N due to the carry rippling through the 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:**

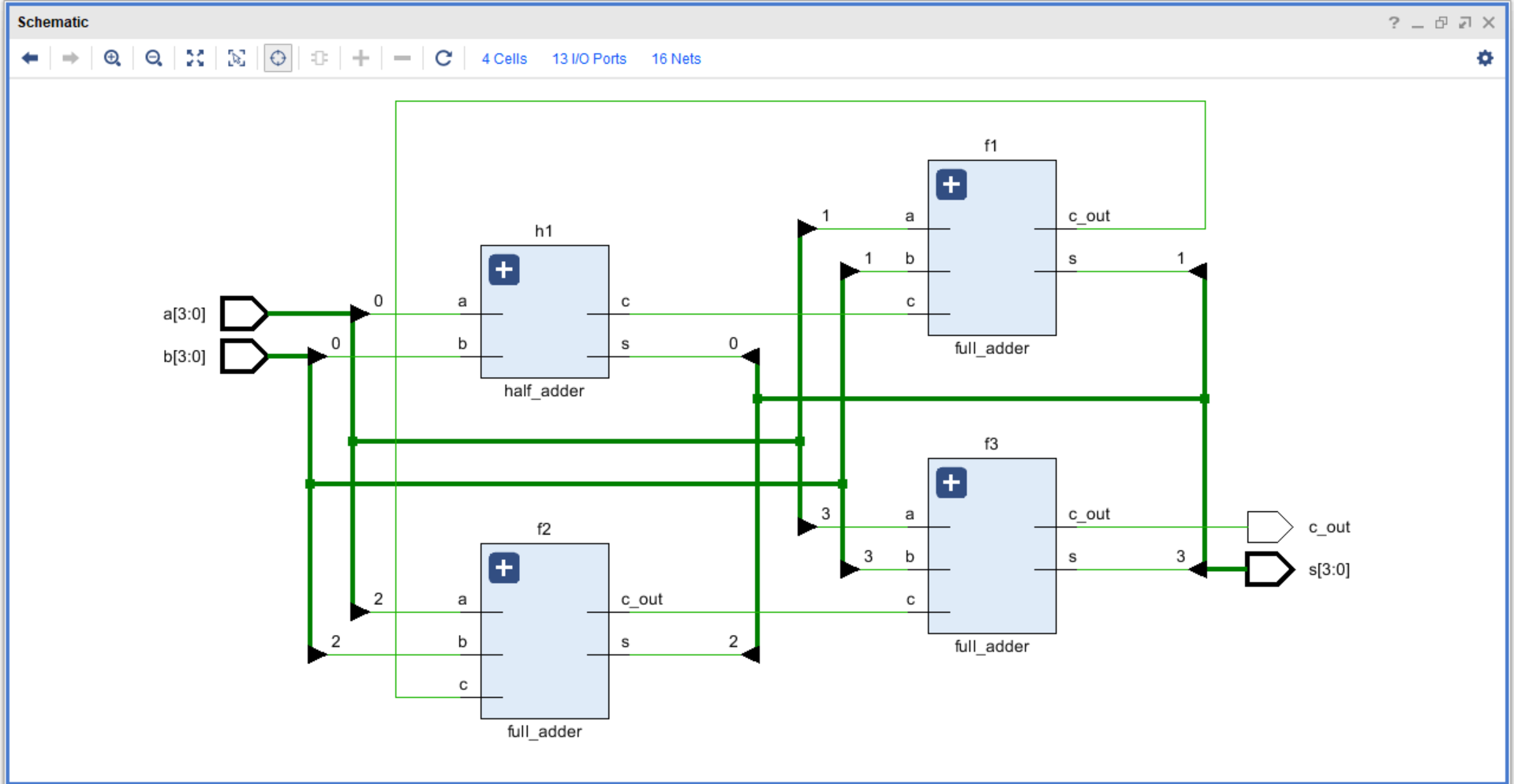
|  |  |
| --- | --- |
| **Verilog Code:**  moduleripple\_carry\_4bit\_adder(a,b,s,c\_out);  input [3:0] a,b;  output [3:0]s;  output c\_out;  wire c1,c2,c3;  half\_adder h1(a[0],b[0],s[0],c1);  full\_adder f1(a[1],b[1],c1,s[1],c2);  full\_adder f2(a[2],b[2],c2,s[2],c3);  full\_adder f3(a[3],b[3],c3,s[3],c\_out);  endmodule | **Testbench:**  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:**  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 | **Verilog Code half adder:**  module half\_adder(a,b,s,c);  input a,b;  output s,c;  xor(s,a,b);  and(c,a,b);  endmodule |

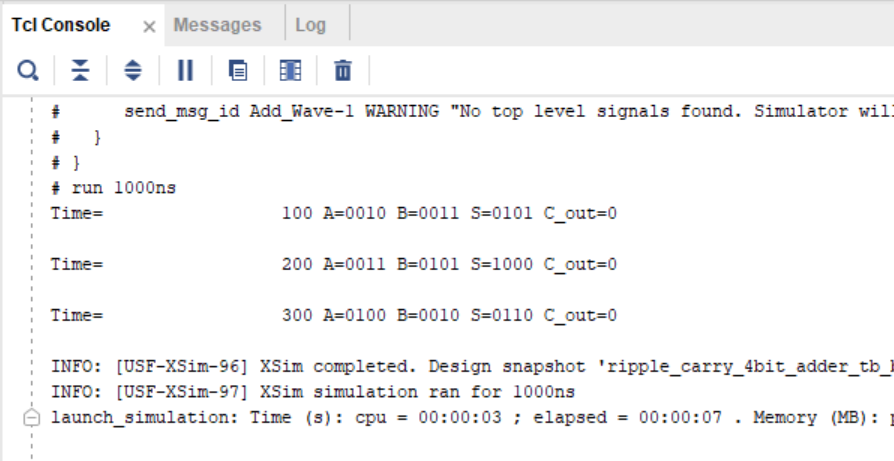
**Simulation:**

****

**RTL design:**

****

**Tcl Console:**

****