Laboratory Report #2

Name: Christian Jay Y. Gallardo **Date Completed:** August 29, 2025

Laboratory Exercise Title: Basic Constructs in Verilog HDL

Target Course Outcomes:

CO1: Create descriptions of digital hardware components such as in combinational and sequential circuits using synthesizable Verilog HDL constructs.

CO2: Verify the functionality of HDL-based components through design verification tools.

Exercise 2B:

This exercise focused on simulating Half Adder and Full Adder circuits using Verilog HDL in ModelSim*-Intel® FPGA Starter Edition. Exercise 2A introduced testbenches and waveform generation through the Half Adder, while Exercise 2B applied the same fundamentals to a Full Adder in a separate project.

Table 1. Truth Table of a Full Adder Circuit

A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

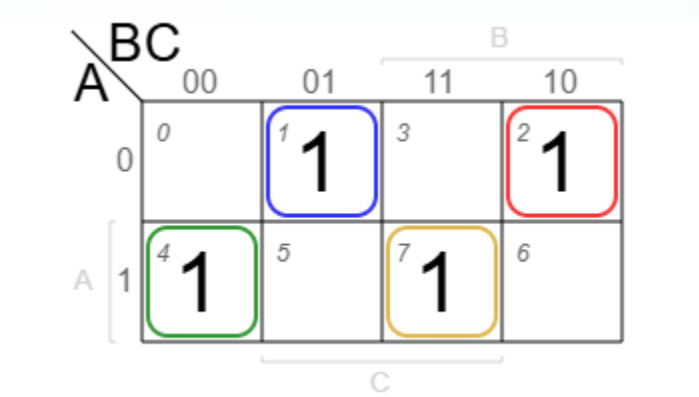


Figure 1. S Kmap

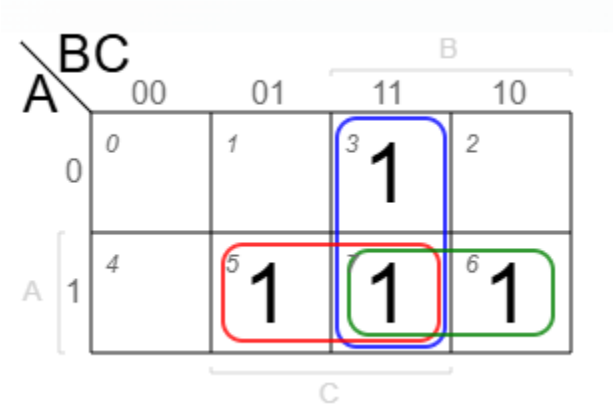


Figure 2. Cout Kmap

Boolean Function for the Full Adder Circuit

Sum = $A \oplus B \oplus C_{in}$

Count = $AB + AC_{in} + BC_{in}$

FullAdder.v tb_FullAdder.v

```
/*************************************************************************
* Name:         Christian Jay Gallardo
* Date:         29/08/2025
* Schedule:     GRP. 4 CPE 3101L 10:30AM - 1:30PM
* Description: Verilog HDL code for Half Adder and Full Adder
*************************************************************************/

module HalfAdder (a, b, c, s);

    input  a, b;    // inputs
    output c, s;    // outputs: carry, sum

    // Half Adder Logic
    xor X1 (s, a, b);   // Sum = a XOR b
    and A1 (c, a, b);   // Carry = a AND b

endmodule


module FullAdder (a, b, cin, s, cout);

    input  a, b, cin;   // inputs
    output s, cout;     // outputs

    wire c1, s1, c2;    // internal connections

    // First Half Adder
    HalfAdder HA1 (a, b, c1, s1);

    // Second Half Adder
    HalfAdder HA2 (s1, cin, c2, s);

    // OR gate for final carry
    or O1 (cout, c1, c2);

endmodule
```

Figure 3. Verilog HDL Design Entry

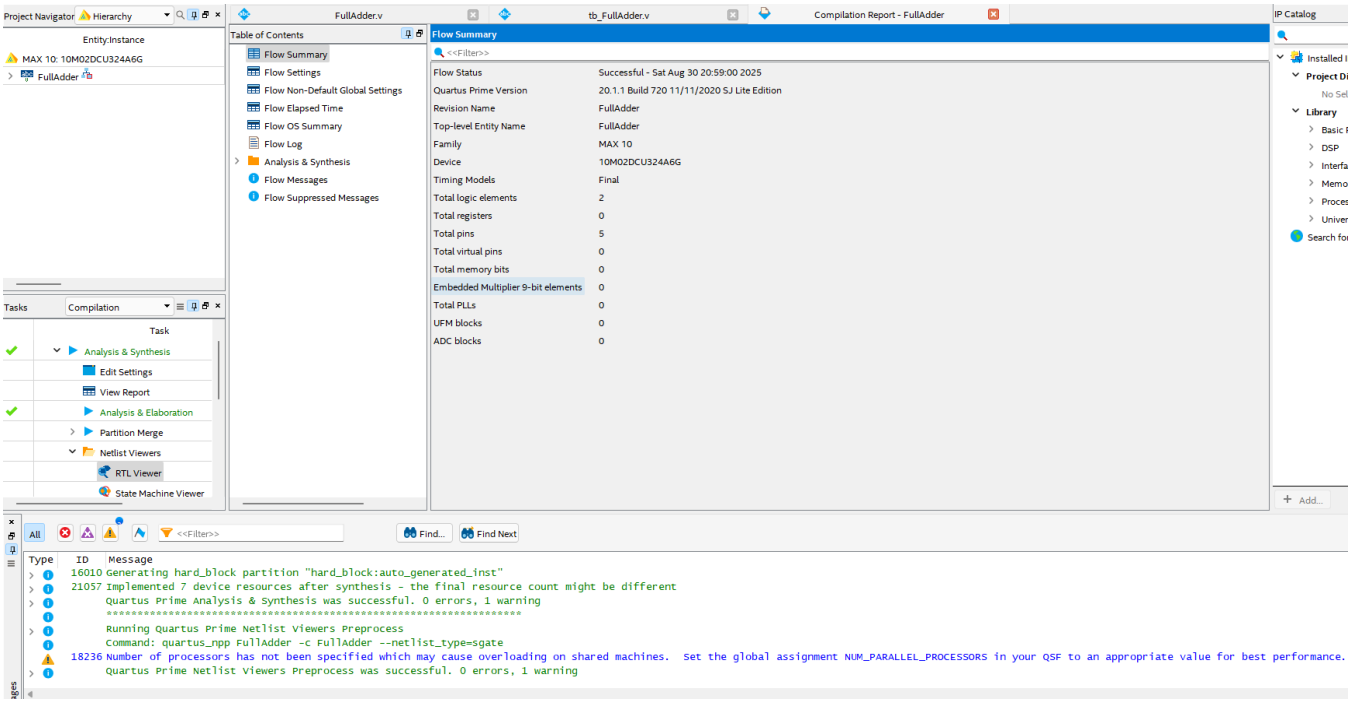


Figure 4. Compilation Report for the Flow Summary

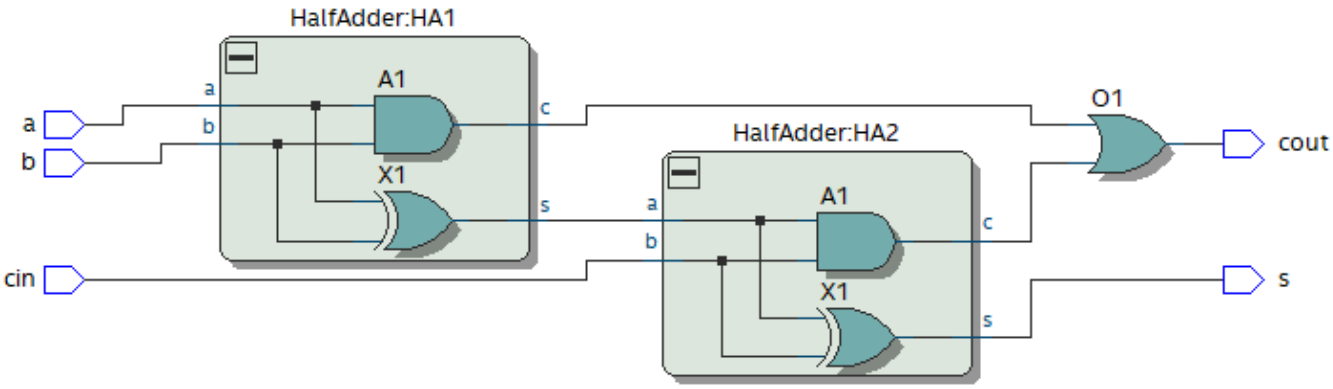


Figure 5. Schematic Diagram of Synthesized Circuit

```
/*************************************************************************
* Name:         Christian Jay Gallardo
* Date:         29/08/2025
* Schedule:     GRP. 4 CPE 3101L 10:30AM - 1:30PM
* Description: Testbench file for Full Adder
*************************************************************************/

`timescale 1 ns / 1 ps

module tb_FullAdder();

    reg a, b, cin;        // inputs
    wire s, cout;         // outputs

    // Instantiate the Unit Under Test (UUT)
    FullAdder UUT (
        .a(a),
        .b(b),
        .cin(cin),
        .s(s),
        .cout(cout)
    );

    // Apply stimulus
    initial begin
        a=0; b=0; cin=0; #10;
        a=0; b=0; cin=1; #10;
        a=0; b=1; cin=0; #10;
        a=0; b=1; cin=1; #10;
        a=1; b=0; cin=0; #10;
        a=1; b=0; cin=1; #10;
        a=1; b=1; cin=0; #10;
        a=1; b=1; cin=1; #30;

        $stop;    // end simulation
    end

endmodule
```

Figure 6. Verilog Testbench File

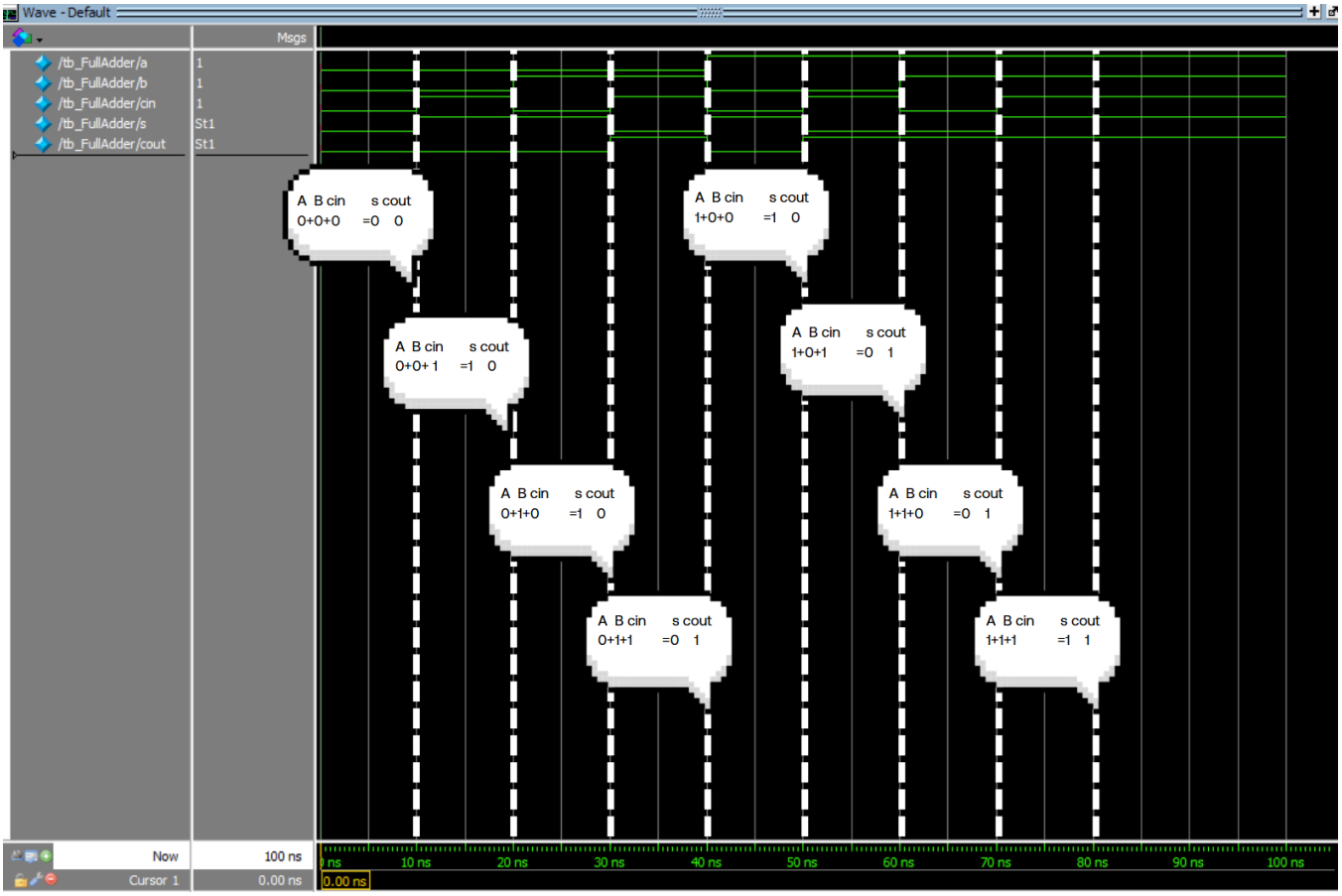
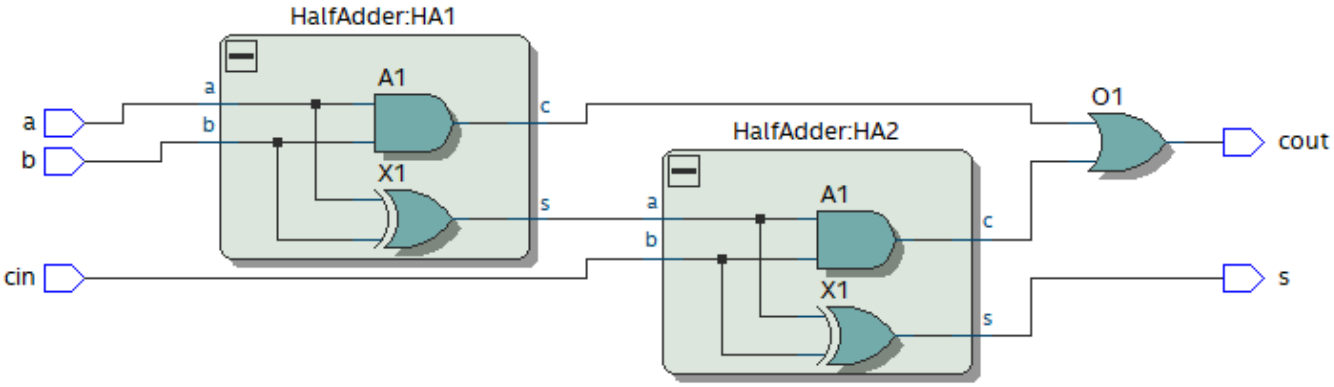


Figure 7. Testbench Waveform with your annotations


HalfAdder:HA1
A1
X1
a
b
c
s
HalfAdder:HA2
A1
X1
a
b
c
s
a
b
cin
O1
cout
s

A	B	C_in	C_out	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 2.1 Truth Table of a Full Adder Circuit

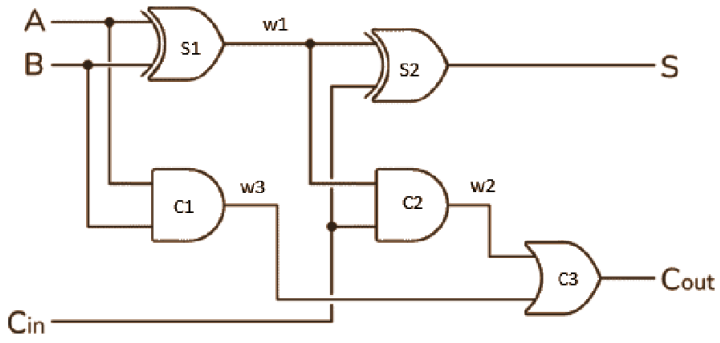
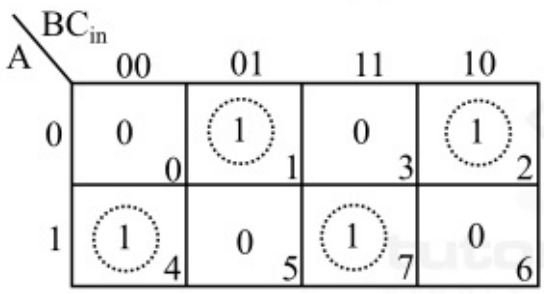


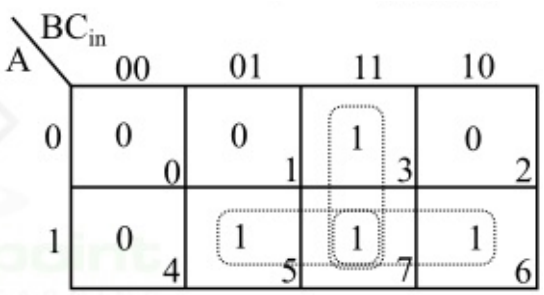
Figure 2.2 Circuit Diagram of a Full Adder Circuit

Karnaugh Map for the Full Adder Circuit

Karnaugh Map for Sum function



Karnaugh Map for Count function



Boolean Function for the Full Adder Circuit

Sum(S) = $A \oplus B \oplus C_{in}$

Count(C_{out}) = $AB + AC_{in} + BC_{in}$

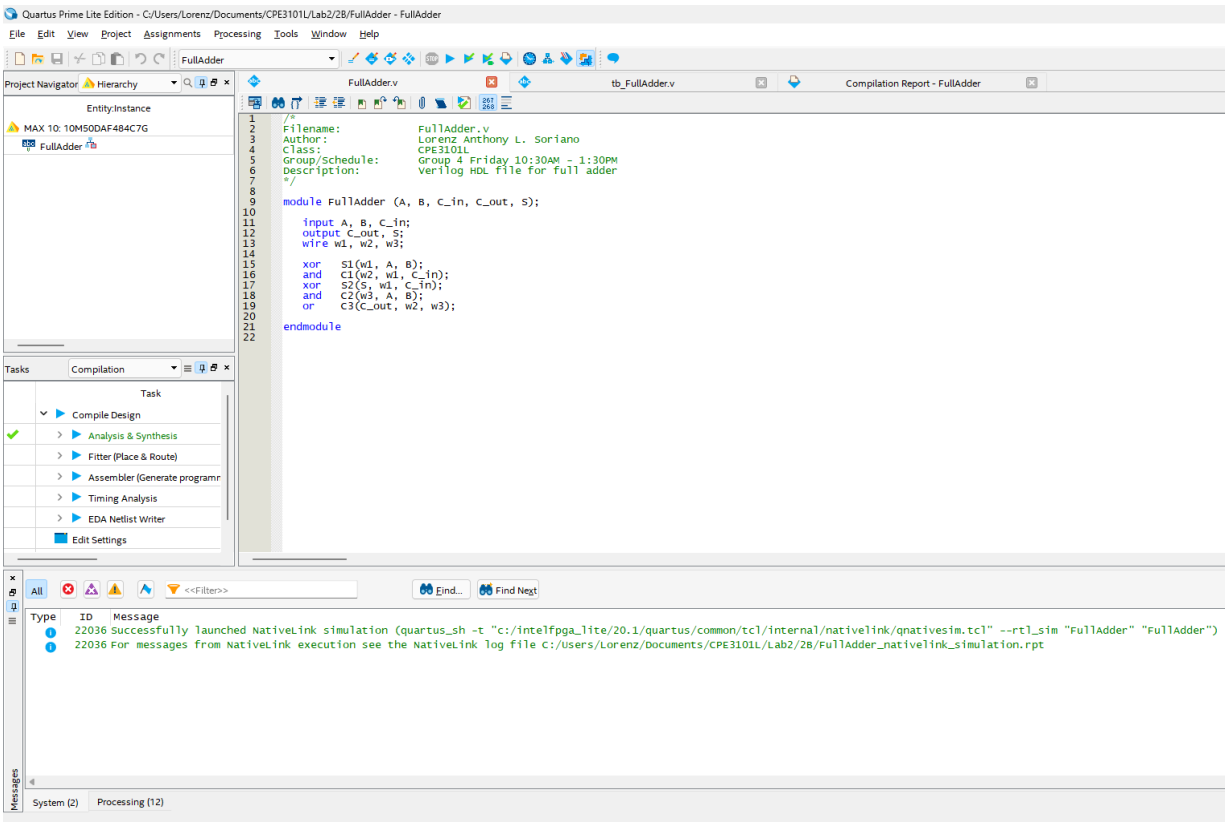


Figure 2.3 Verilog HDL Design Entry

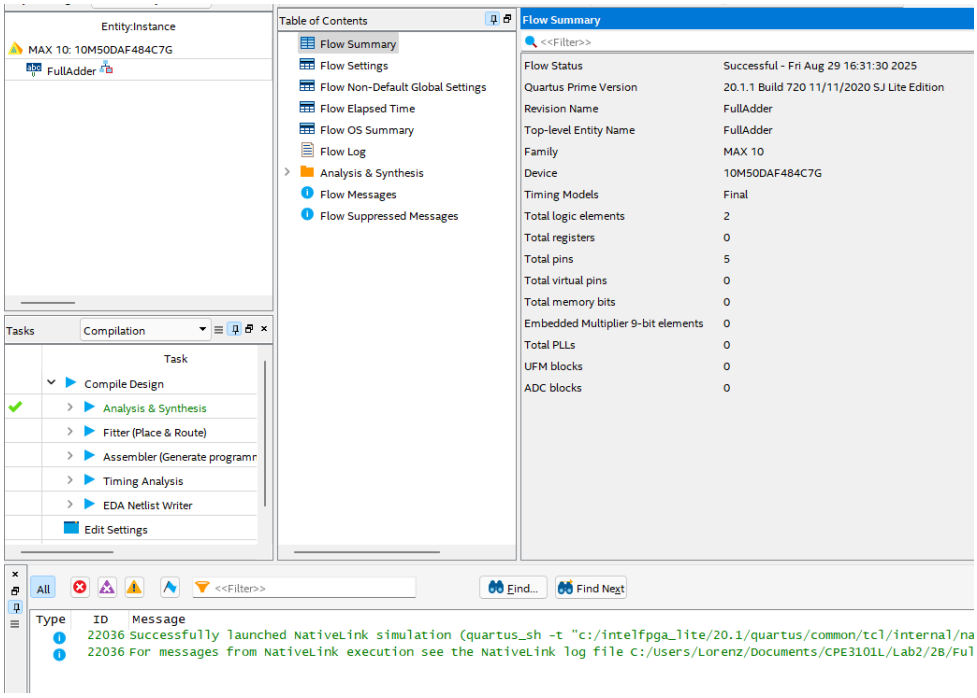


Figure 2.4 Compilation Report for the Flow Summary

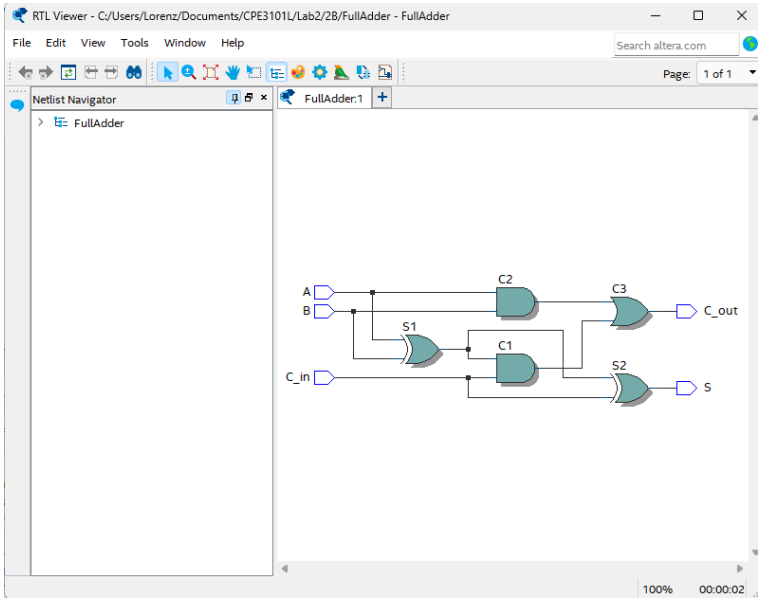


Figure 2.5 Schematic Diagram of Synthesized Circuit using RTL viewer

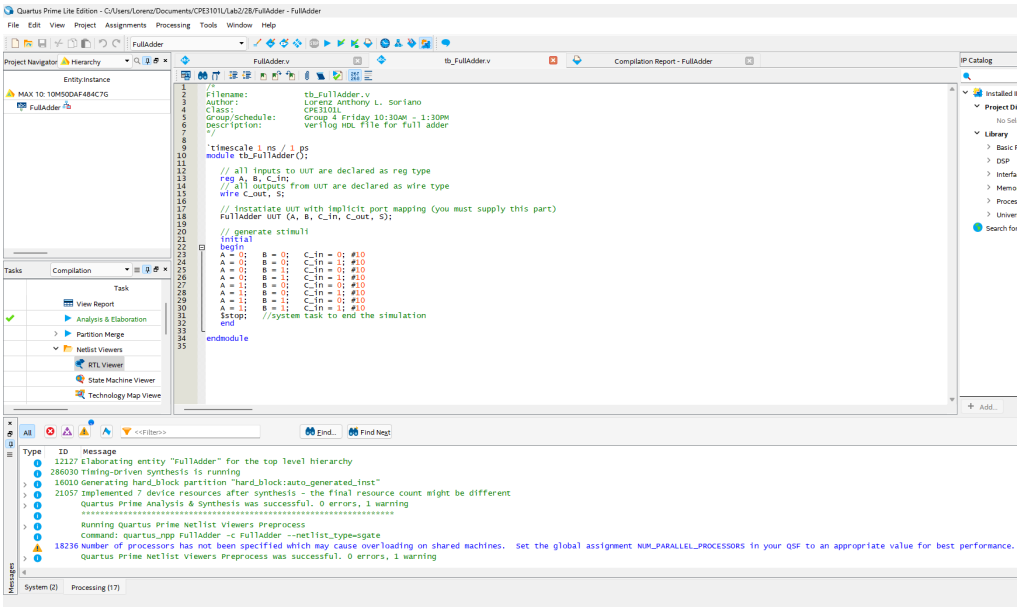


Figure 2.6 Verilog Testbench File

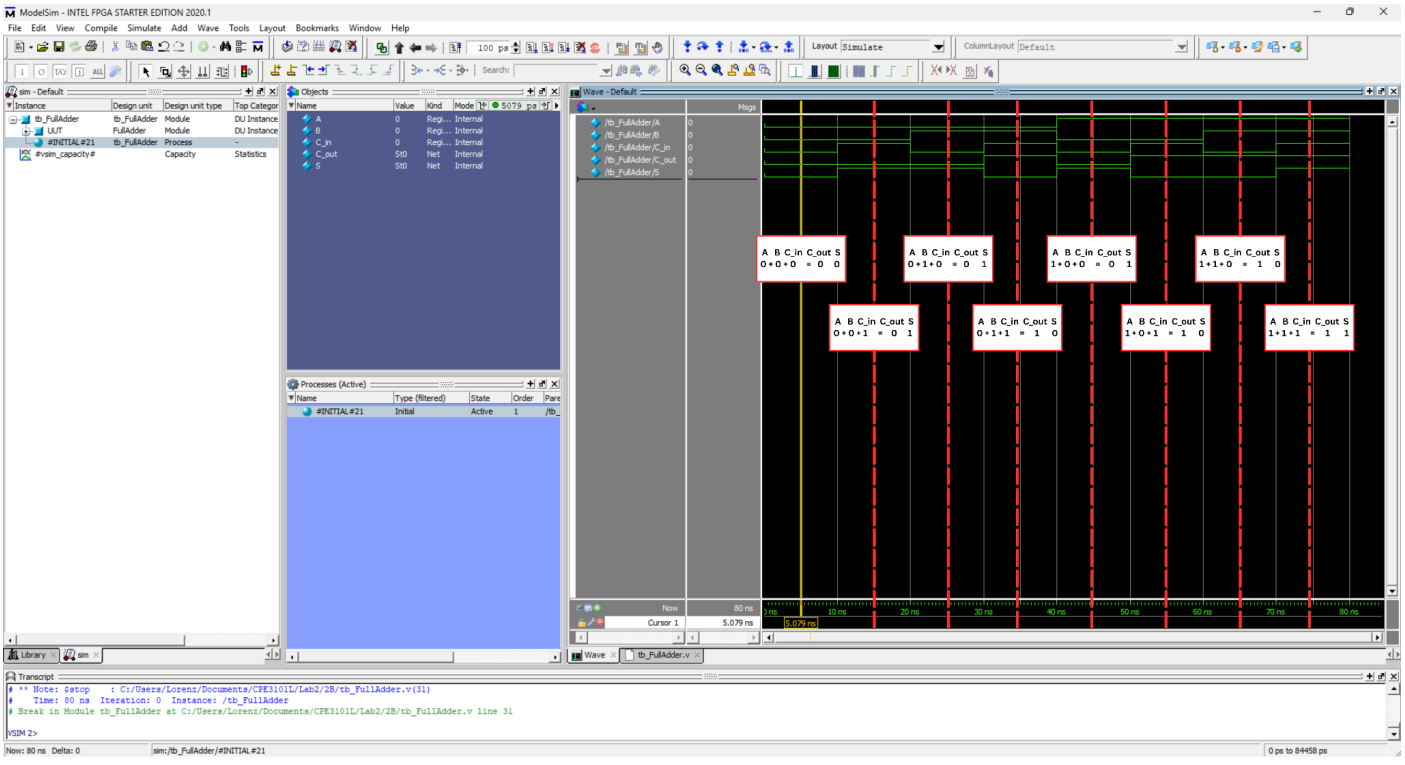


Figure 2.7 Testbench Waveform with annotations