



## 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

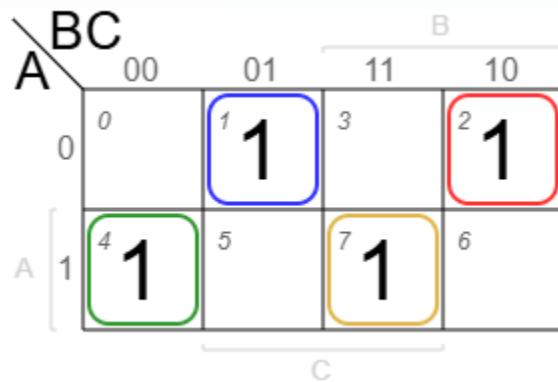


Figure 1. S Kmap

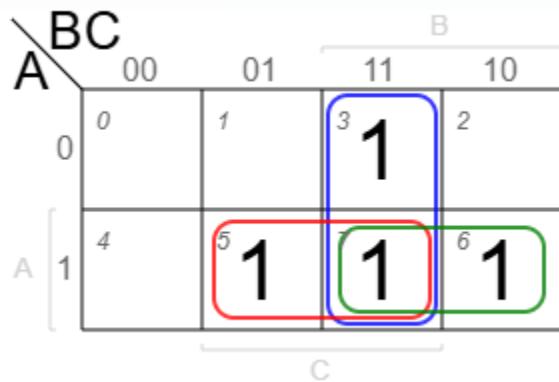


Figure 2. Cout Kmap

#### Boolean Function for the Full Adder Circuit

$$\begin{aligned} \text{Sum} &= A \oplus B \oplus C_{in} \\ \text{Count} &= AB + AC_{in} + BC_{in} \end{aligned}$$



The screenshot shows a dual-pane code editor. The left pane displays the Verilog source code for a Full Adder, while the right pane shows the testbench code. Both panes have tabs at the top labeled 'abc' and show line numbers from 1 to 38.

```
1  //*****  
2  * Name: Christian Jay Gallardo  
3  * Date: 29/08/2025  
4  * Schedule: GRP. 4 CPE 3101L 10:30AM - 1:30PM  
5  * Description: Verilog HDL code for Half Adder and Full Adder  
*****  
6  
7 module HalfAdder (a, b, c, s);  
8  
9     input a, b; // inputs  
10    output c, s; // outputs: carry, sum  
11  
12    // Half Adder Logic  
13    xor X1 (s, a, b); // Sum = a XOR b  
14    and A1 (c, a, b); // Carry = a AND b  
15  
16 endmodule  
17  
18  
19 module FullAdder (a, b, cin, s, cout);  
20  
21    input a, b, cin; // inputs  
22    output s, cout; // outputs  
23  
24    wire c1, s1, c2; // internal connections  
25  
26    // First Half Adder  
27    HalfAdder HA1 (a, b, c1, s1);  
28  
29    // Second Half Adder  
30    HalfAdder HA2 (s1, cin, c2, s);  
31  
32    // OR gate for final carry  
33    or O1 (cout, c1, c2);  
34  
35 endmodule  
36  
37  
38
```

Figure 3. Verilog HDL Design Entry

The screenshot shows the Quartus Prime software interface with multiple windows open. The central window displays the 'Flow Summary' report, which provides detailed statistics about the compilation process. Other windows visible include 'FullAdder.v', 'tb\_FullAdder.v', and 'Compilation Report - FullAdder'. The left sidebar shows the project navigation and tasks, while the bottom pane shows the message log.

**Flow Summary**

Parameter	Value
Flow Status	Successful - Sat Aug 30 20:59:00 2025
Quartus Prime Version	20.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	FullAdder
Top-level Entity Name	FullAdder
Family	MAX 10
Device	10M02DCU324A6G
Timing Models	Final
Total logic elements	2
Total registers	0
Total pins	5
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

**Message Log (Bottom)**

```
Type ID Message  
> 16010 Generating hard_block partition "hardBlock:auto_generated_inst"  
> 21057 Implemented 7 device resources after synthesis - the final resource count might be different  
> 16010 Quartus Prime Analysis & Synthesis was successful. 0 errors, 1 warning  
*****  
> 16010 Running Quartus Prime Netlist Viewers Preprocess  
Command: quartus_npp FullAdder --cc FullAdder --netlist_type=sgate  
> 18236 Number of processors has not been specified which may cause overloading on shared machines. set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value for best performance.  
> 16010 Quartus Prime Netlist Viewers Preprocess was successful. 0 errors, 1 warning
```

Figure 4. Compilation Report for the Flow Summary

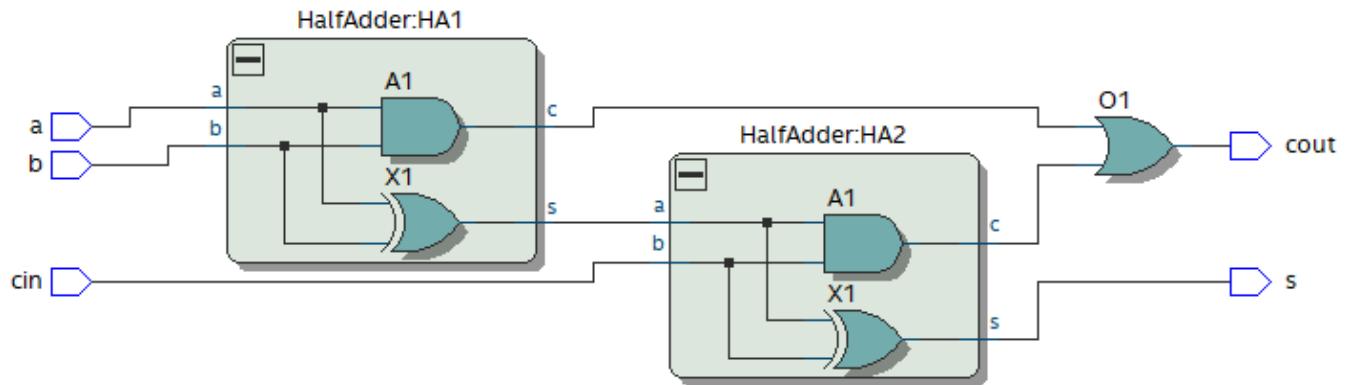


Figure 5. Schematic Diagram of Synthesized Circuit

```
1  /*************************************************************************/
2  * Name: Christian Jay Gallardo
3  * Date: 29/08/2025
4  * Schedule: GRP. 4 CPE 3101L 10:30AM - 1:30PM
5  * Description: Testbench file for Full Adder
6  /*************************************************************************/
7
8 `timescale 1 ns / 1 ps
9
10 module tb_FullAdder();
11
12   reg a, b, cin;           // inputs
13   wire s, cout;          // outputs
14
15   // Instantiate the Unit Under Test (UUT)
16   FullAdder UUT (
17     .a(a),
18     .b(b),
19     .cin(cin),
20     .s(s),
21     .cout(cout)
22   );
23
24   // Apply stimulus
25   initial begin
26     a=0; b=0; cin=0; #10;
27     a=0; b=0; cin=1; #10;
28     a=0; b=1; cin=0; #10;
29     a=0; b=1; cin=1; #10;
30     a=1; b=0; cin=0; #10;
31     a=1; b=0; cin=1; #10;
32     a=1; b=1; cin=0; #10;
33     a=1; b=1; cin=1; #30;
34
35   $stop; // end simulation
36
37
38 endmodule
39
```

Figure 6. Verilog Testbench File

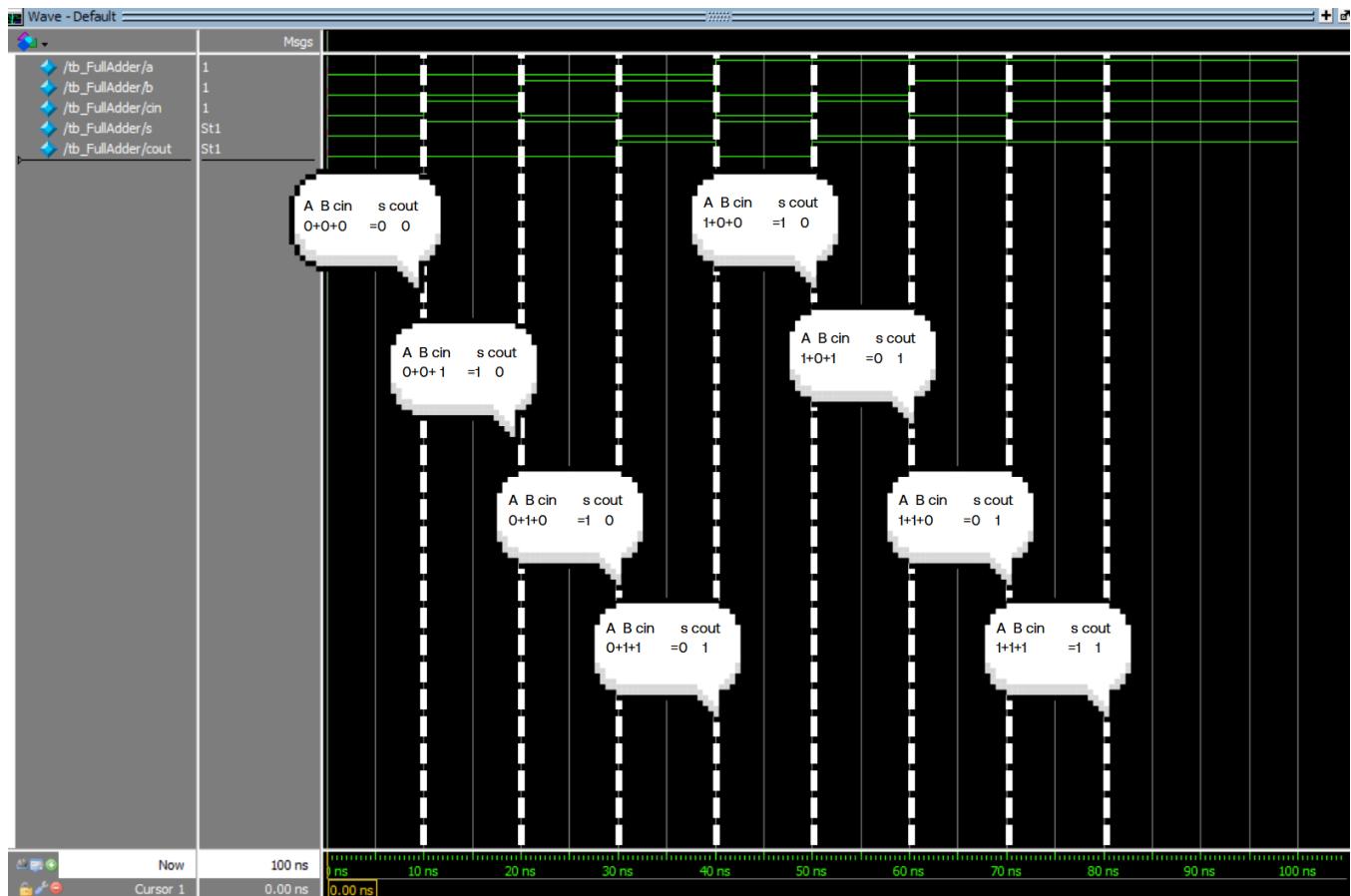
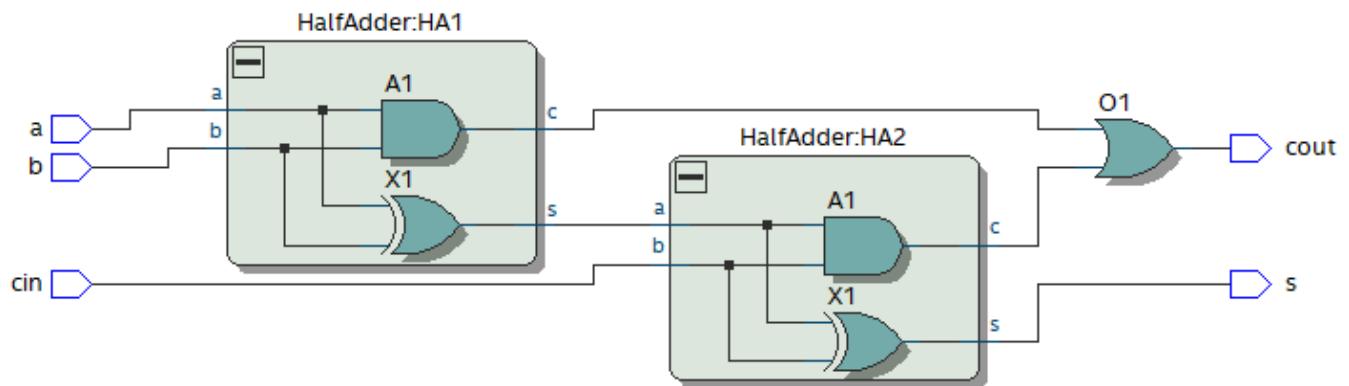


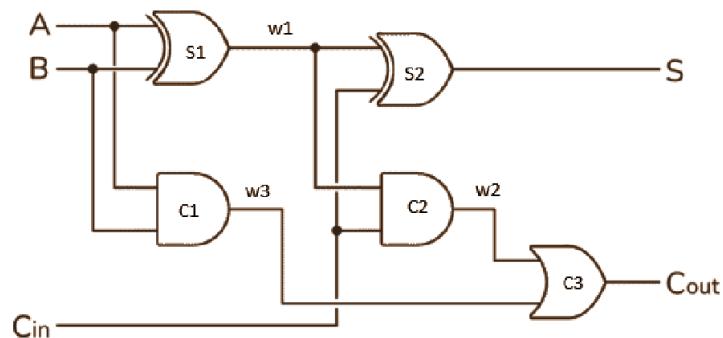
Figure 7. Testbench Waveform with your annotations





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

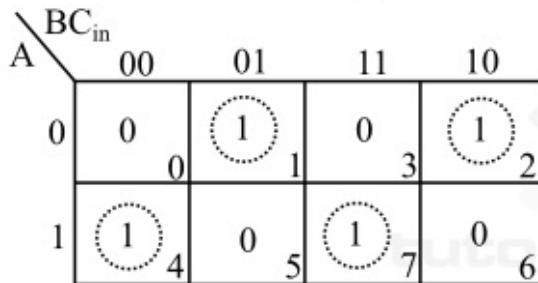


**Figure 2.2** Circuit Diagram of a Full Adder Circuit

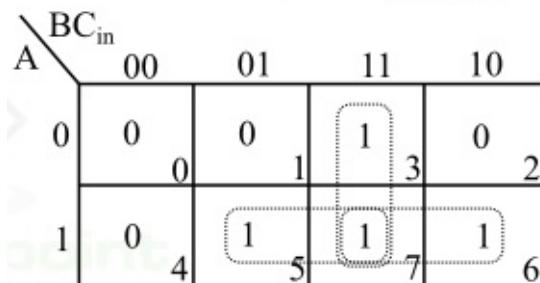
**Karnaugh Map for the Full Adder Circuit**



Karnaugh Map for Sum function



Karnaugh Map for Count function



### Boolean Function for the Full Adder Circuit

$$\begin{aligned} \text{Sum}(S) &= A \oplus B \oplus C_{\text{in}} \\ \text{Count}(C_{\text{out}}) &= AB + AC_{\text{in}} + BC_{\text{in}} \end{aligned}$$

The screenshot shows the Quartus Prime Lite Edition interface with the following details:

- File Menu:** File, Edit, View, Project, Assignments, Processing, Tools, Window, Help.
- Project Navigator:** MAX 10: 10M50DAF484C7G, FullAdder.
- Editor:** FullAdder.v (Verilog code), tb\_FullAdder.v (Testbench), Compilation Report - FullAdder.
- Code Editor Content:**

```
/*  
 * Filename: FullAdder.v  
 * Author: Lorenz Anthony L. Soriano  
 * Class: CPE3101L  
 * Group/Schedule: Group 4 Friday 10:30AM - 1:30PM  
 * Description: Verilog HDL file for full adder  
 */  
  
module FullAdder (A, B, C_in, C_out, S);  
    input A, B, C_in;  
    output C_out, S;  
    wire w1, w2, w3;  
  
    xor S1(w1, A, B);  
    and C1(w1, A, B);  
    xor S2(w2, w1, C_in);  
    and C2(w2, w1, C_in);  
    or C3(w3, w2, w3);  
    assign C_out = w3;  
    assign S = w1;  
endmodule
```
- Task List:** Task, Compile Design (checked), Analysis & Synthesis, Fitter (Place & Route), Assembler (Generate program), Timing Analysis, EDA Nellist Writer, Edit Settings.
- Messages Panel:** Type: All, ID: Message, Message content: 22036 Successfully launched NativeLink simulation (quartus\_sh -t "c:/intelfpga\_lite/20.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl\_sim "FullAdder" "FullAdder")  
22036 For messages from NativeLink execution see the NativeLink log file c:/users/lorenz/documents/cpe3101l/lab2/2b/fulladder\_nativelink\_simulation.rpt

Figure 2.3 Verilog HDL Design Entry

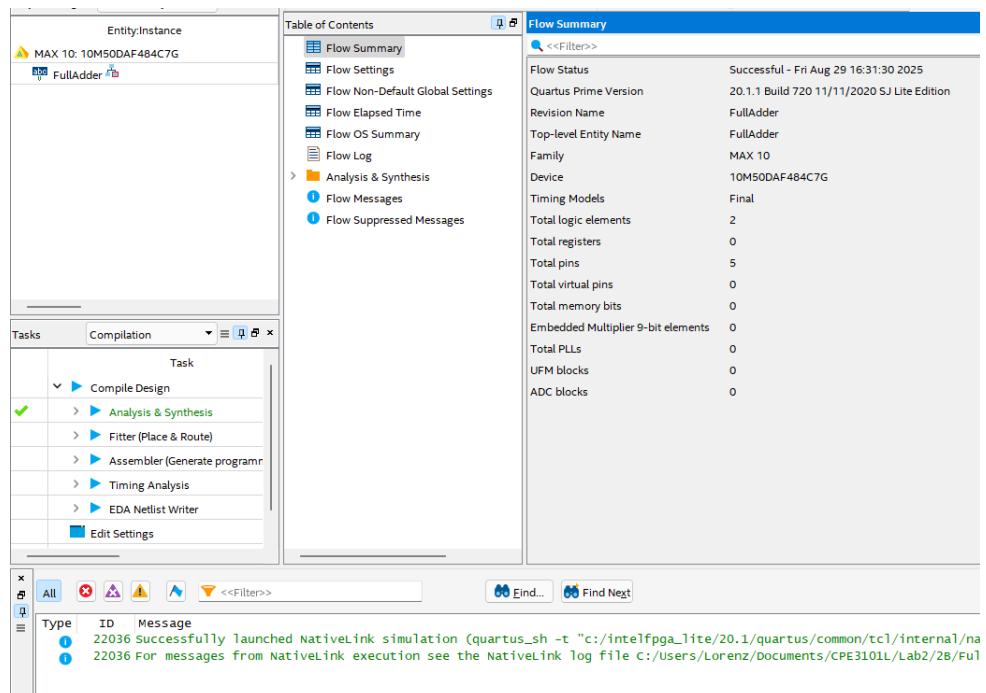


Figure 2.4 Compilation Report for the Flow Summary

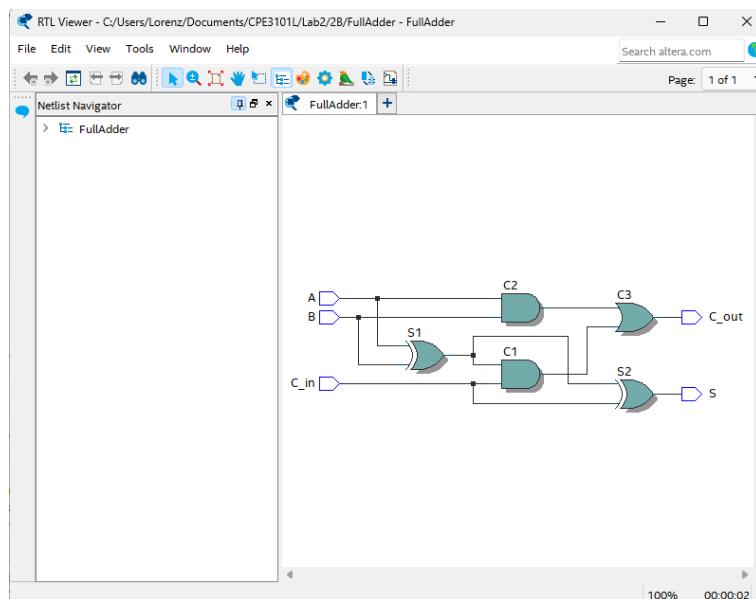
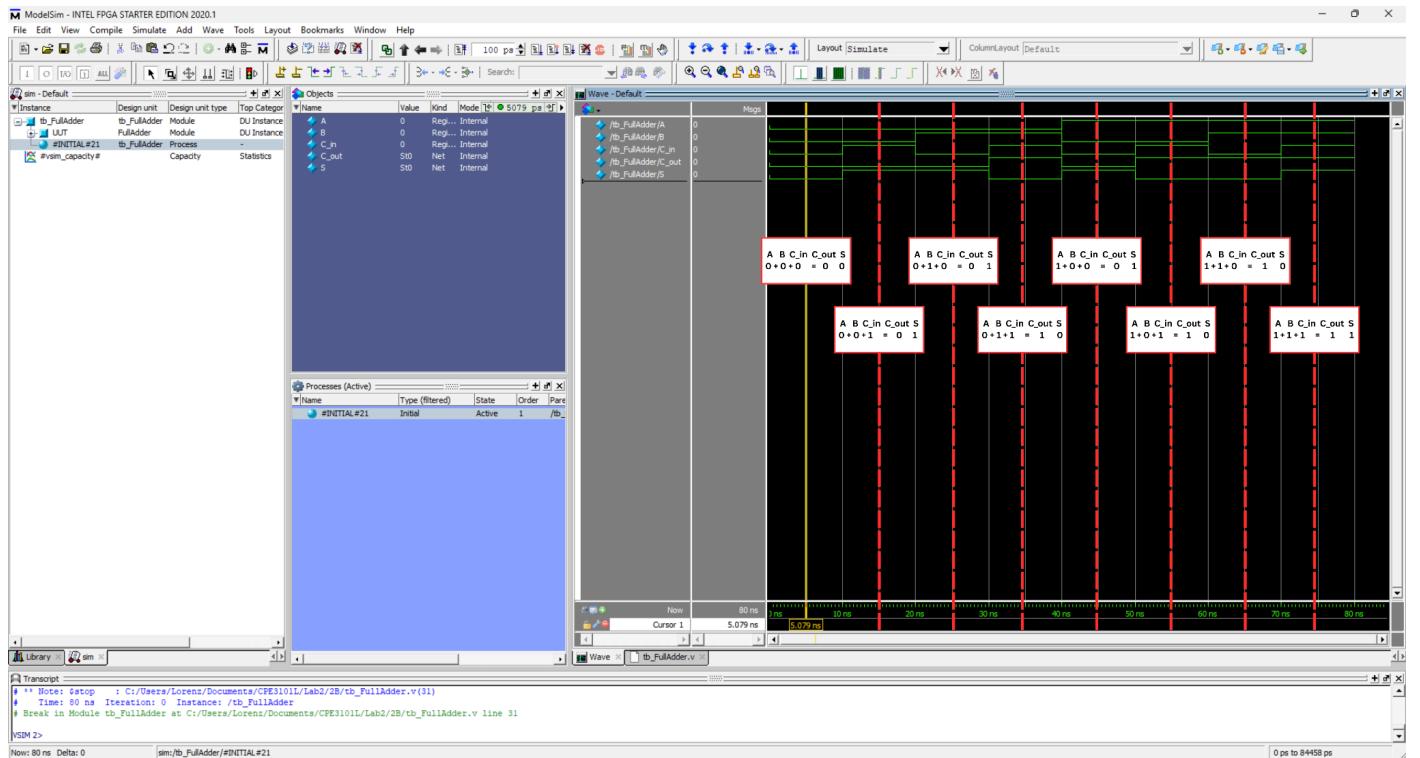


Figure 2.5 Schematic Diagram of Synthesized Circuit using RTL viewer

The screenshot shows the Quartus Prime Lite Edition software interface. The top menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, Help, and a FullAdder icon. The main window displays a VHDL code editor for a file named tb\_FullAdder.v. The code defines a testbench for a full adder, including stimulus generation and a simulation task. To the right of the code editor is an IP Catalog pane. On the left, there's a Project Navigator with a hierarchy tree and a Task list containing a View Report item under Compilation. At the bottom, there are tabs for System (2) and Processing (17), along with a Messages pane showing synthesis and analysis logs.

## **Figure 2.6 Verilog Testbench File**



**Figure 2.7** Testbench Waveform with annotations