



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

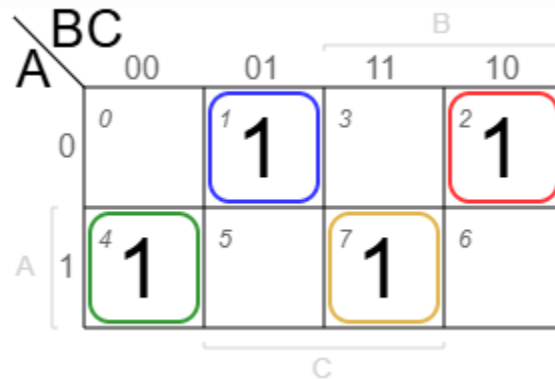


Figure 1. S Kmap

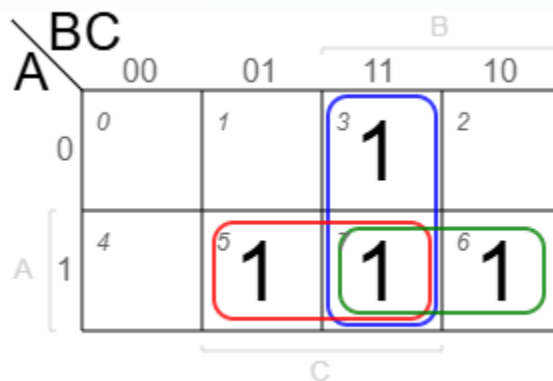


Figure 2. Cout Kmap

Boolean Function for the Full Adder Circuit

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

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



```
FullAdder.v      tb_FullAdder.v
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
8  module HalfAdder (a, b, c, s);
9
10     input a, b;    // inputs
11     output c, s;   // outputs: carry, sum
12
13     // Half Adder Logic
14     xor X1 (s, a, b); // Sum = a XOR b
15     and A1 (c, a, b); // Carry = a AND b
16
17 endmodule
18
19
20 module FullAdder (a, b, cin, s, cout);
21
22     input a, b, cin; // inputs
23     output s, cout;  // outputs
24
25     wire c1, s1, c2; // internal connections
26
27     // First Half Adder
28     HalfAdder HA1 (a, b, c1, s1);
29
30     // Second Half Adder
31     HalfAdder HA2 (s1, cin, c2, s);
32
33     // OR gate for final carry
34     or O1 (cout, c1, c2);
35
36 endmodule
37
38
```

Figure 3. Verilog HDL Design Entry

Project Navigator Entity/Instance: MAX 10: 10M02DCU324A6G, FullAdder

Table of Contents: Flow Summary, Flow Settings, Flow Non-Default Global Settings, Flow Elapsed Time, Flow OS Summary, Flow Log, Analysis & Synthesis, Flow Messages, Flow Suppressed Messages

Tasks: Compilation, Task, Analysis & Synthesis, Edit Settings, View Report, Analysis & Elaboration, Partition Merge, Netlist Viewers, RTL Viewer, State Machine Viewer

Flow Summary

Flow Status	Successful - Sat Aug 30 20:59:00 2025
Quartus Prime Version	20.1.1 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

Messages:

- 16010 Generating hard_block partition "hard_block:auto_generated_inst"
- 21057 Implemented 7 device resources after synthesis - the final resource count might be different
- quartus Prime Analysis & Synthesis was successful. 0 errors, 1 warning
- running quartus Prime Netlist Viewers Preprocess
- Command: quartus_rnp FullAdder -c 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.
- Quartus Prime Netlist viewers Preprocess was successful. 0 errors, 1 warning

Figure 4. Compilation Report for the Flow Summary

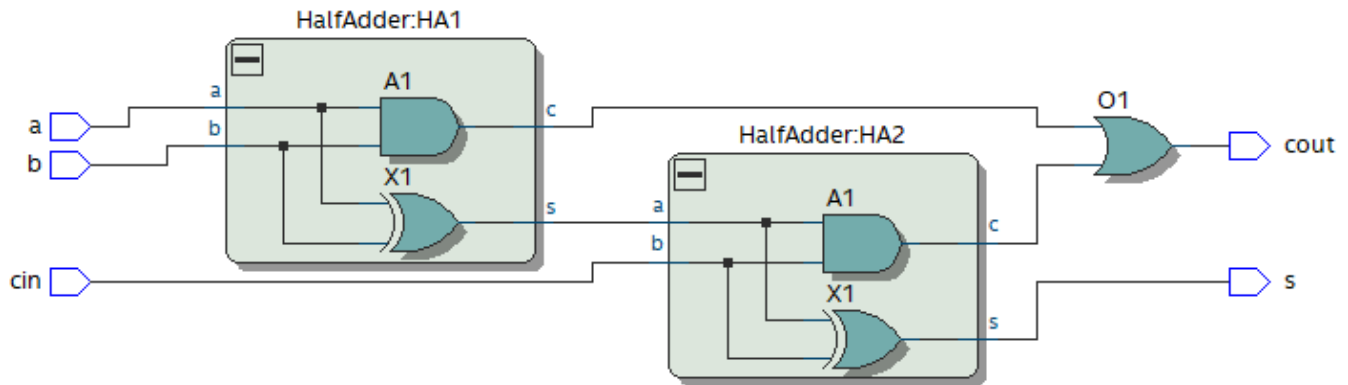


Figure 5. Schematic Diagram of Synthesized Circuit

```

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     end
37 endmodule
38
39

```

Figure 6. Verilog Testbench File

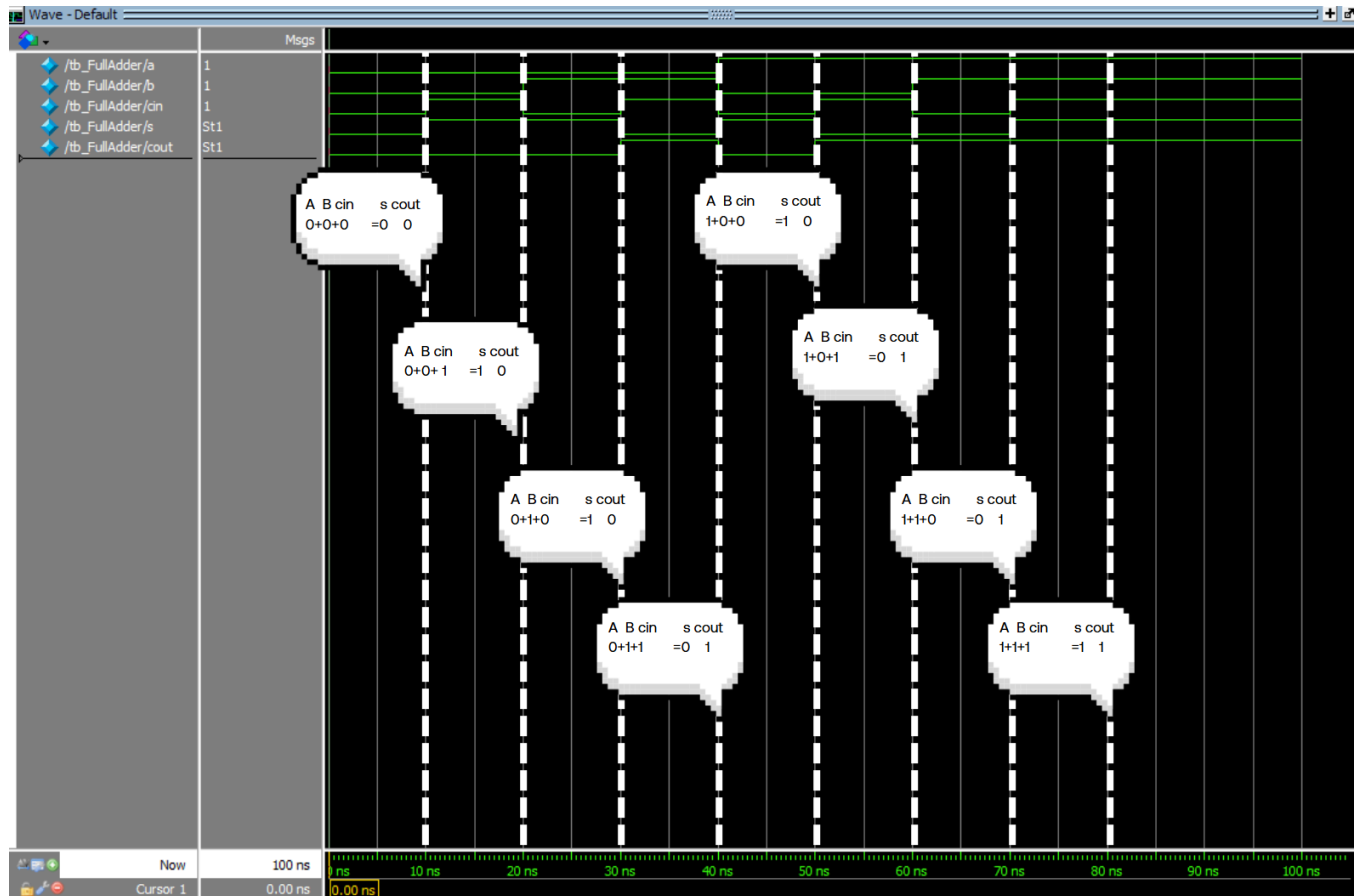
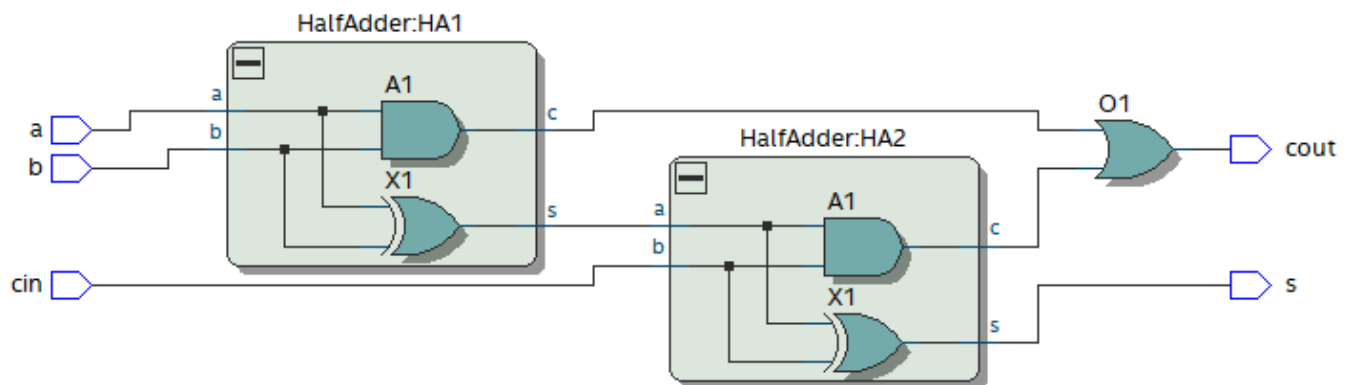


Figure 7. Testbench Waveform with your annotations





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

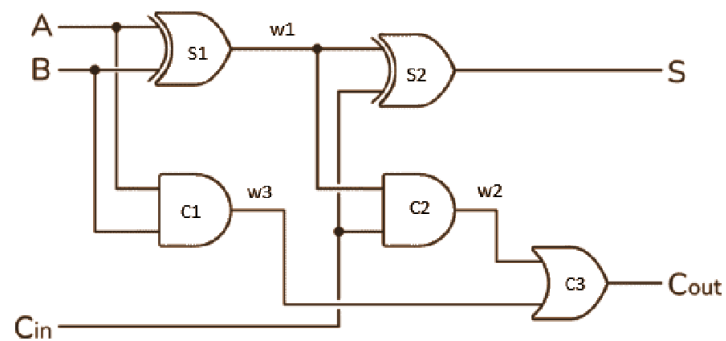


Figure 2.2 Circuit Diagram of a Full Adder Circuit

Karnaugh Map for the Full Adder Circuit



Karnaugh Map for Sum function

A \ BC _{in}				
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Karnaugh Map for Count function

A \ BC _{in}				
	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Boolean Function for the Full Adder Circuit

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

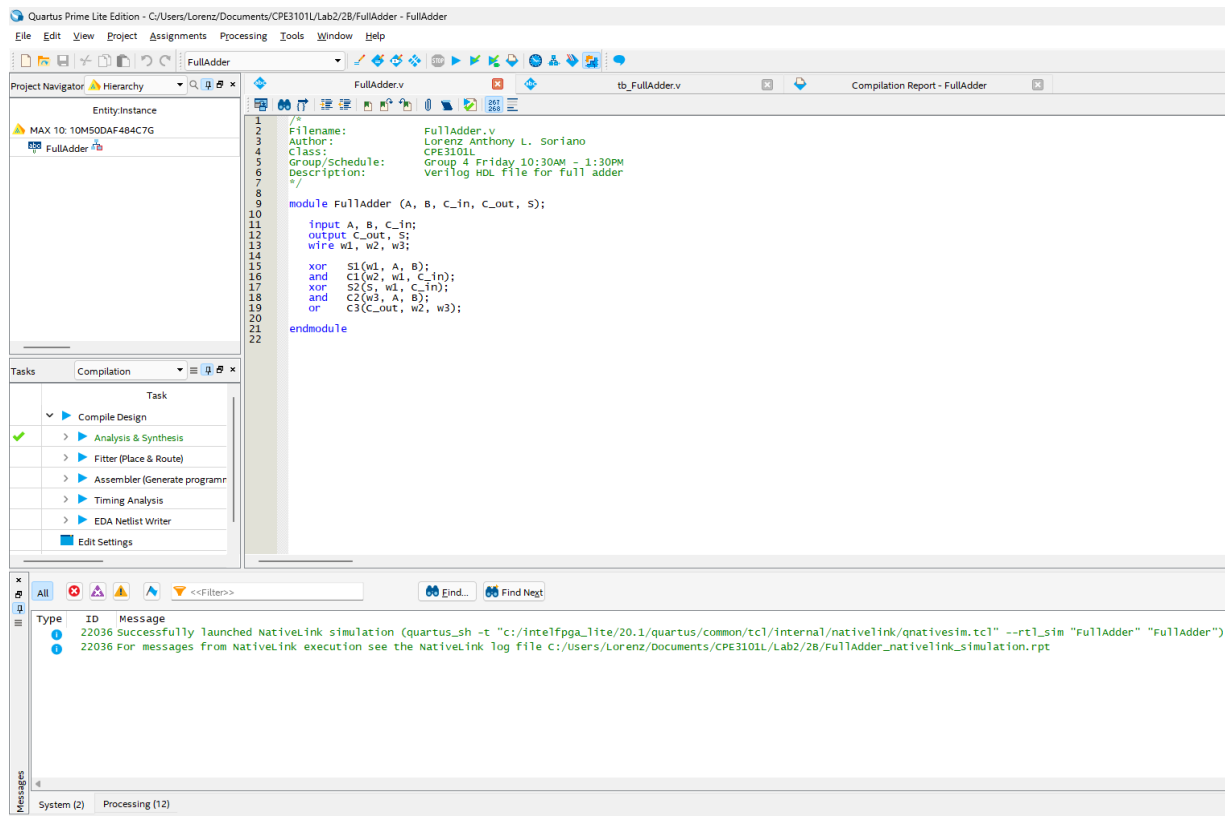


Figure 2.3 Verilog HDL Design Entry

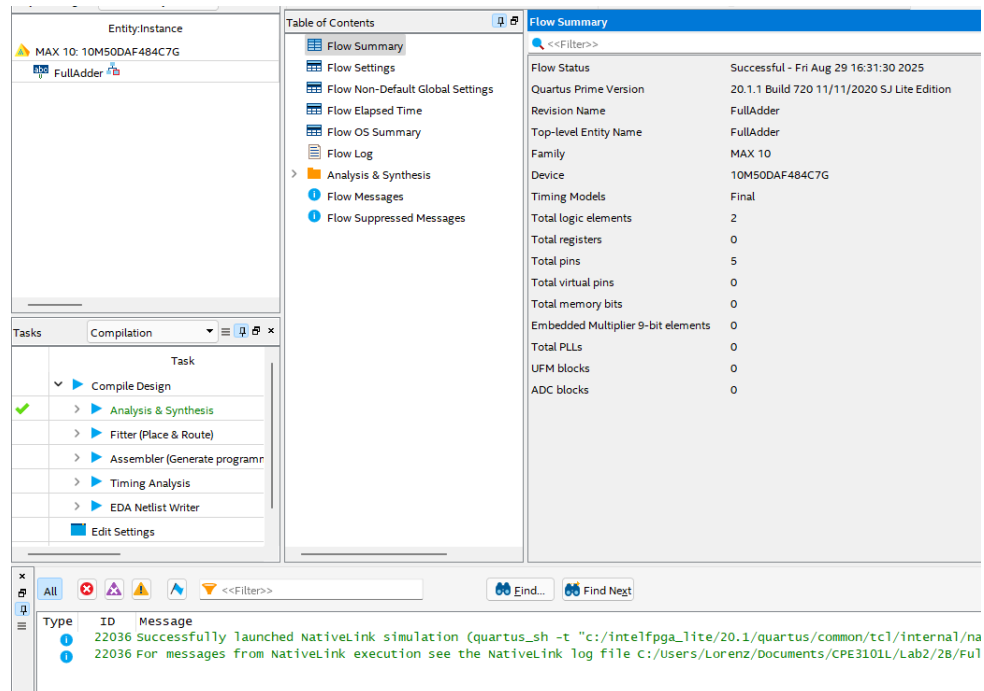


Figure 2.4 Compilation Report for the Flow Summary

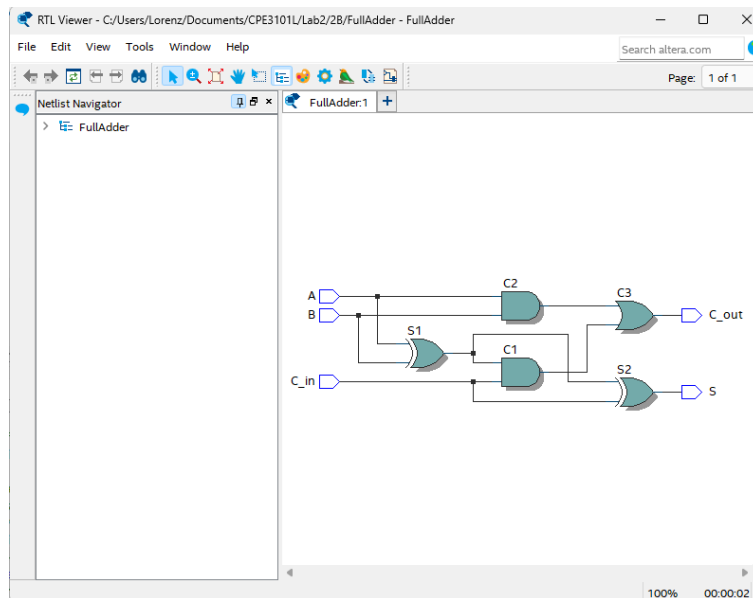


Figure 2.5 Schematic Diagram of Synthesized Circuit using RTL viewer

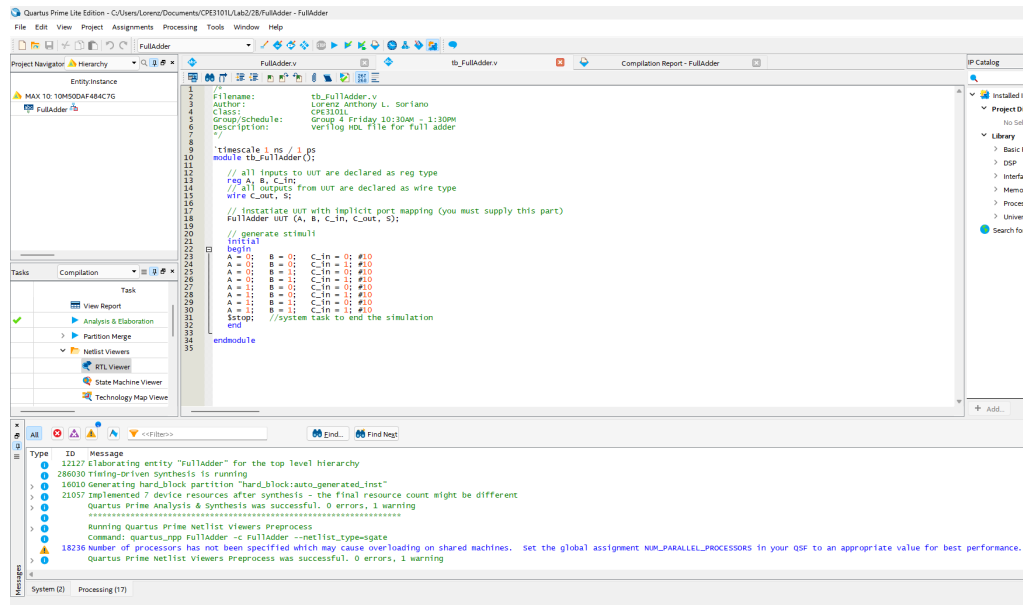


Figure 2.6 Verilog Testbench File

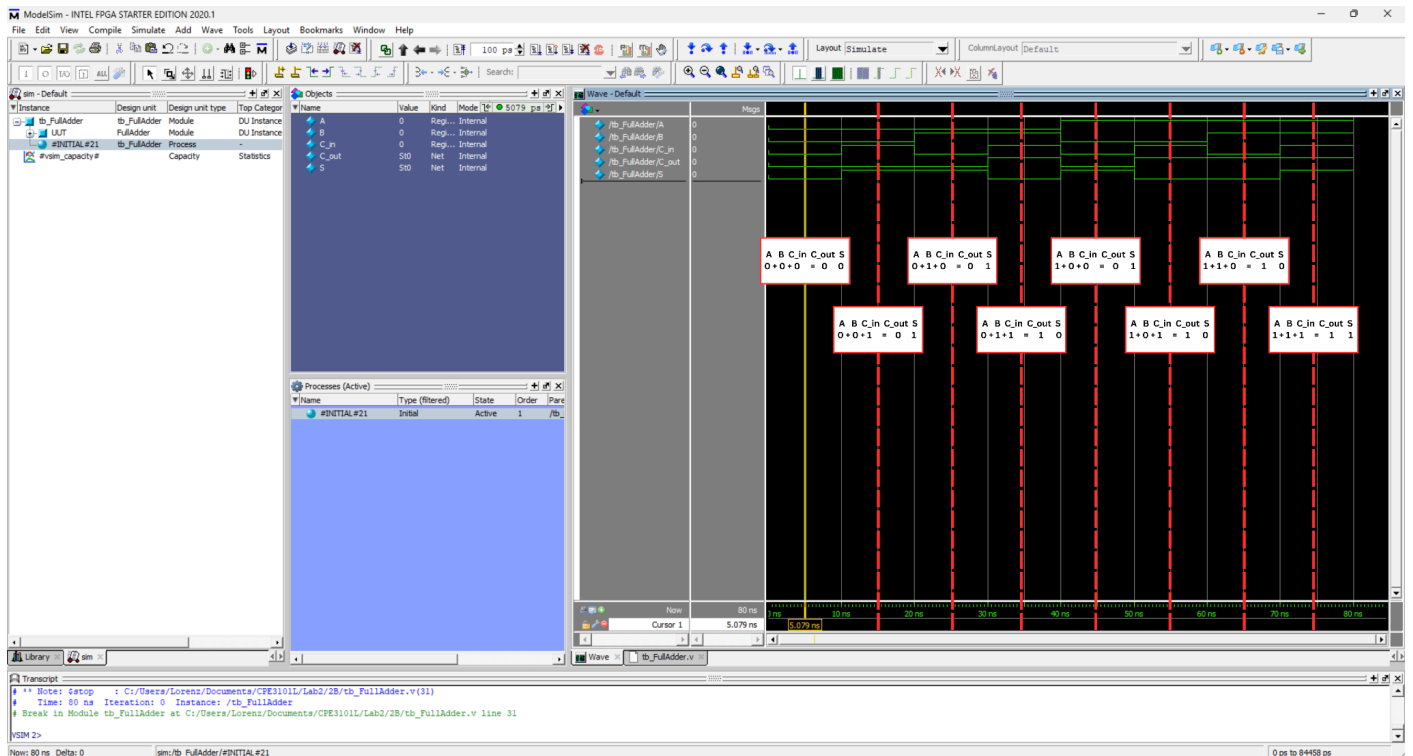


Figure 2.7 Testbench Waveform with annotations