Laboratory Report #2

Name: Mileah Zoei Dy **Date Completed: 08/29/25**

Laboratory Exercise Title: Full Adder Circuit

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:

In this exercise, a Verilog HDL description of a full adder circuit was created while incorporating gate primitives and structural modeling. Furthermore, the circuit was simulated using a testbench file to verify the generated waveform.

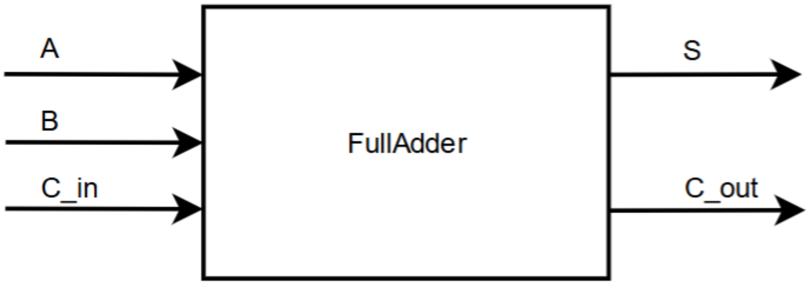


Figure 1. Full Adder Circuit Entity Diagram

The basic function of a full adder circuit is to take the binary sum of two single-bit inputs along with a carry-in to produce a sum and carry output (Figure 1). The inputs and outputs used in this exercise are further shown in Table 1.

Table 1. Full Adder Circuit I/O

I/O	Variable	Description
Inputs	A	First single-bit input
	B	Second single-bit input
	C_in	Carry-In
Outputs	C_out	Carry-Out
	S	Sum

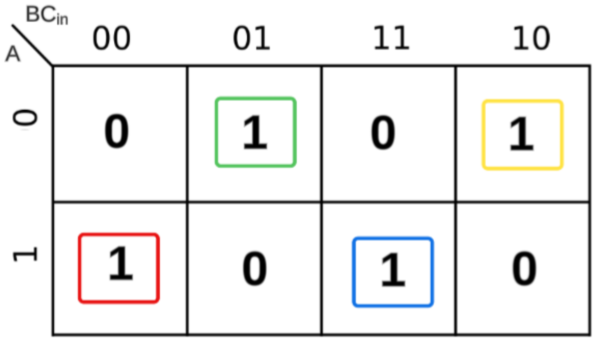
In single-bit binary addition, the sum is 0 if all inputs are 0, the sum is 1 with a 0 carry out if only one input has the value of 1, the sum is 0 with a carry out of 1 if two inputs are 1, and lastly if all inputs are equal to 1, then the sum is 1 with a carry out value of 1. By indicating all possible sum and carry outcomes of the single-bit binary addition of inputs A, B, and C_{in}, the truth table may be formulated likewise:

Table 2. Full Adder Circuit Truth Table

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

To get the boolean functions, a K-map was formulated based on the truth table of the full adder circuit, then the sum of products (SOP) was taken using the grouped terms.

After constructing the K-map of the S output (Figure 2),



A \ BC_{in}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Figure 2. Karnaugh Map of S Output

the SOP expression was further simplified:

$$S = \overline{A}\,\overline{B}C_{in} + ABC_{in} + \overline{A}B\overline{C_{in}} + A\overline{B}\,\overline{C_{in}}$$

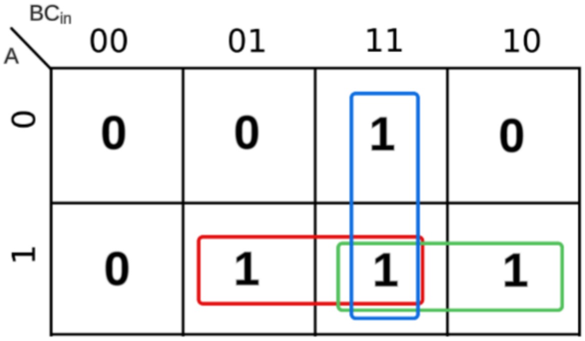
$$S = C_{in}(\overline{A}\,\overline{B} + AB) + \overline{C_{in}}(\overline{A}B + A\overline{B})$$

$$S = C_{in}\overline{(A \oplus B)} + \overline{C}_{in}(A \oplus B)$$

to obtain the boolean function for S:

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

After constructing the K-map of the C_{out} output (Figure 3),



$A \backslash BC_{in}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Figure 3. Karnaugh Map of C_{out} Output

the SOP expression was further manipulated:

$$C_{out} = AB + AC_{in} + BC_{in}$$

$$C_{out} = AB + (A + B)C_{in}$$

$$C_{out} = AB + ((A + B)(A + \overline{A}))C_{in}$$

$$C_{out} = AB + (A + \overline{A}B)C_{in}$$

$$C_{out} = AB + (A(B + \overline{B}) + \overline{A}B)C_{in}$$

$$C_{out} = AB + (AB + A\overline{B} + \overline{A}B)C_{in}$$

$$C_{out} = AB + (AB + A \oplus B)C_{in}$$

$$C_{out} = AB + ABC_{in} + C_{in}(A \oplus B)$$

$$C_{out} = AB(1 + C_{in}) + C_{in}(A \oplus B)$$

to obtain the boolean function for C_{out}:

$$C_{out} = AB + C_{in}(A \oplus B)$$

The logic circuit can then be constructed (Figure 4) according to the derived boolean functions. This shows the circuit to be made up of two half adder circuits (Figure 5) and an additional OR gate. To simplify the HDL description, the half adder module made in the previous exercise was imported to the project and instantiated in the top circuit.

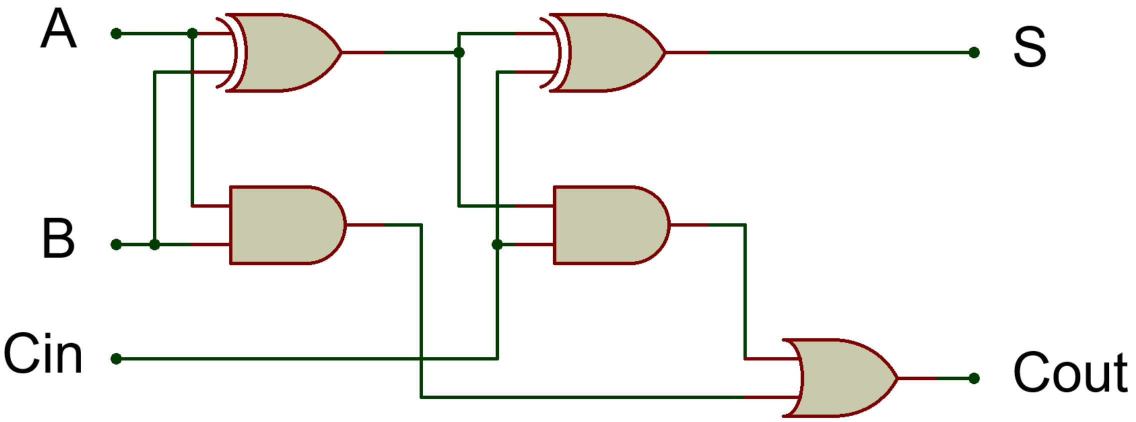


Figure 4. Full Adder Circuit Logic Diagram

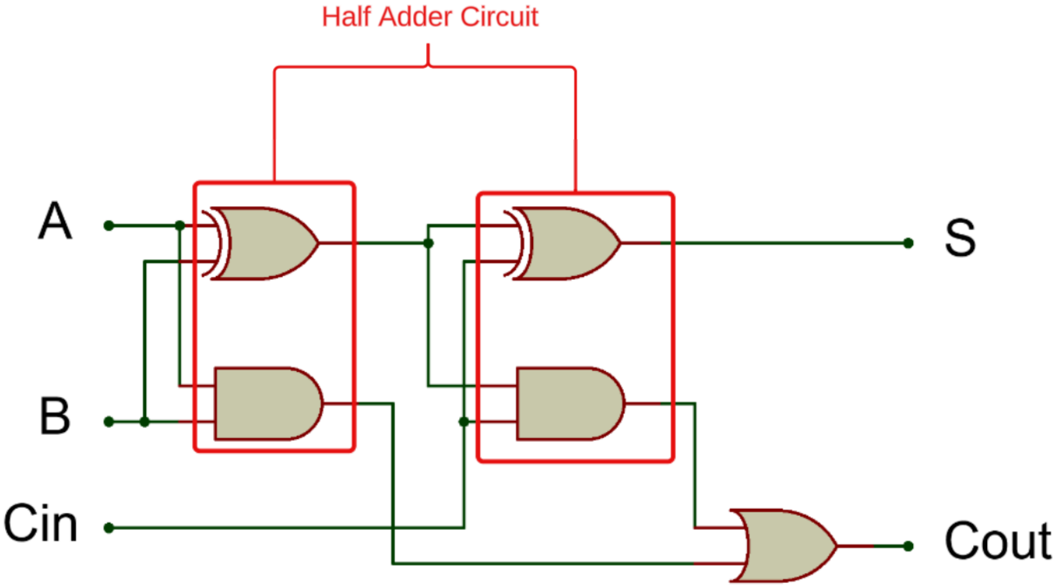
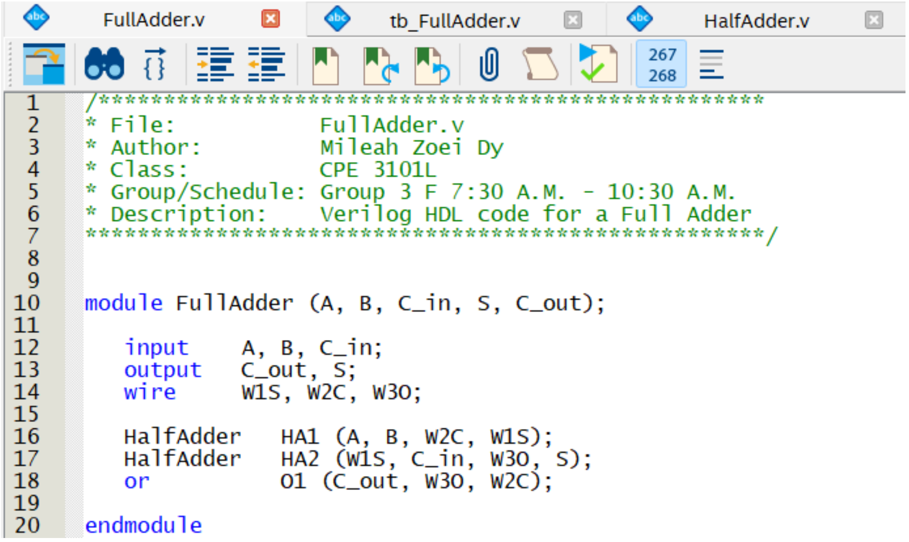


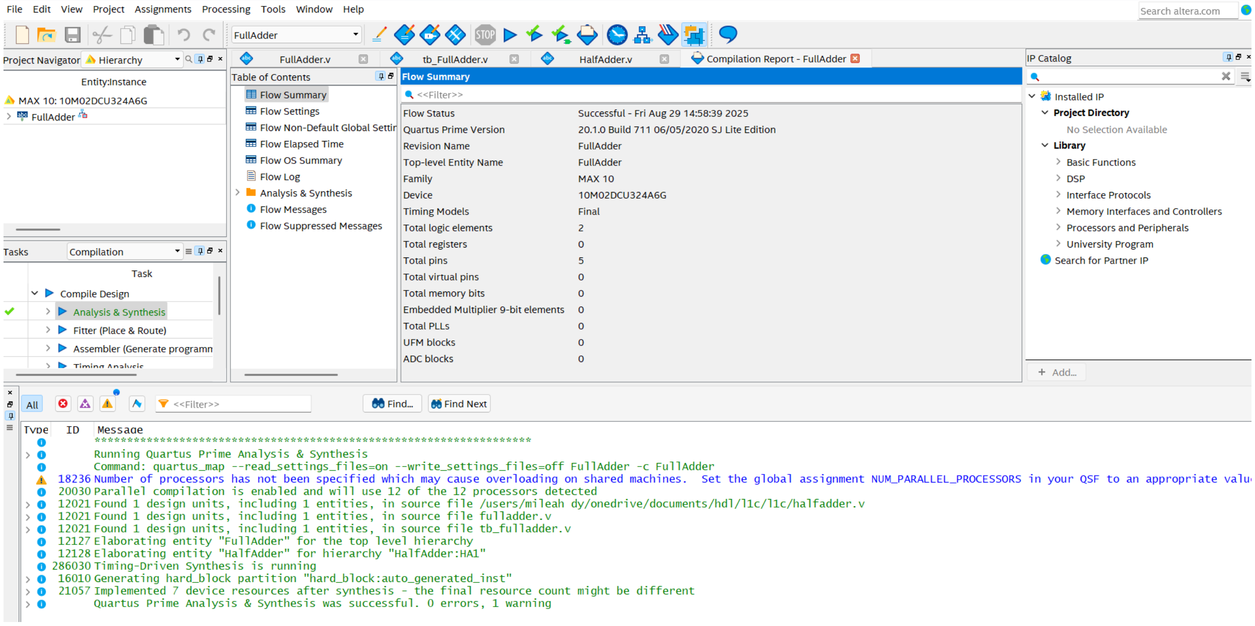
Figure 5. Full Adder Circuit Logic Diagram (with annotations)

Exercise Screenshots:



```
/****************************************************
* File:           FullAdder.v
* Author:         Mileah Zoei Dy
* Class:          CPE 3101L
* Group/Schedule: Group 3 F 7:30 A.M. - 10:30 A.M.
* Description:    Verilog HDL code for a Full Adder
*****************************************************/


module FullAdder (A, B, C_in, S, C_out);

   input    A, B, C_in;
   output   C_out, S;
   wire     W1S, W2C, W3O;

   HalfAdder   HA1 (A, B, W2C, W1S);
   HalfAdder   HA2 (W1S, C_in, W3O, S);
   or          O1 (C_out, W3O, W2C);

endmodule
```

Figure 6. Verilog HDL Design Entry



Flow Summary

Flow Status	Successful - Fri Aug 29 14:58:39 2025
Quartus Prime Version	20.1.0 Build 711 06/05/2020 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

```
Type   ID   Message
            ********************************************************************
            Running Quartus Prime Analysis & Synthesis
            Command: quartus_map --read_settings_files=on --write_settings_files=off FullAdder -c FullAdder
      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 valu
      20030 Parallel compilation is enabled and will use 12 of the 12 processors detected
      12021 Found 1 design units, including 1 entities, in source file /users/mileah dy/onedrive/documents/hdl/l1c/l1c/halfadder.v
      12021 Found 1 design units, including 1 entities, in source file fulladder.v
      12021 Found 1 design units, including 1 entities, in source file tb_fulladder.v
      12127 Elaborating entity "FullAdder" for the top level hierarchy
      12128 Elaborating entity "HalfAdder" for hierarchy "HalfAdder:HA1"
     286030 Timing-Driven Synthesis is running
      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
```

Figure 7. Compilation Report Flow Summary

- Total logic elements used: 2
- Total pins used: 5

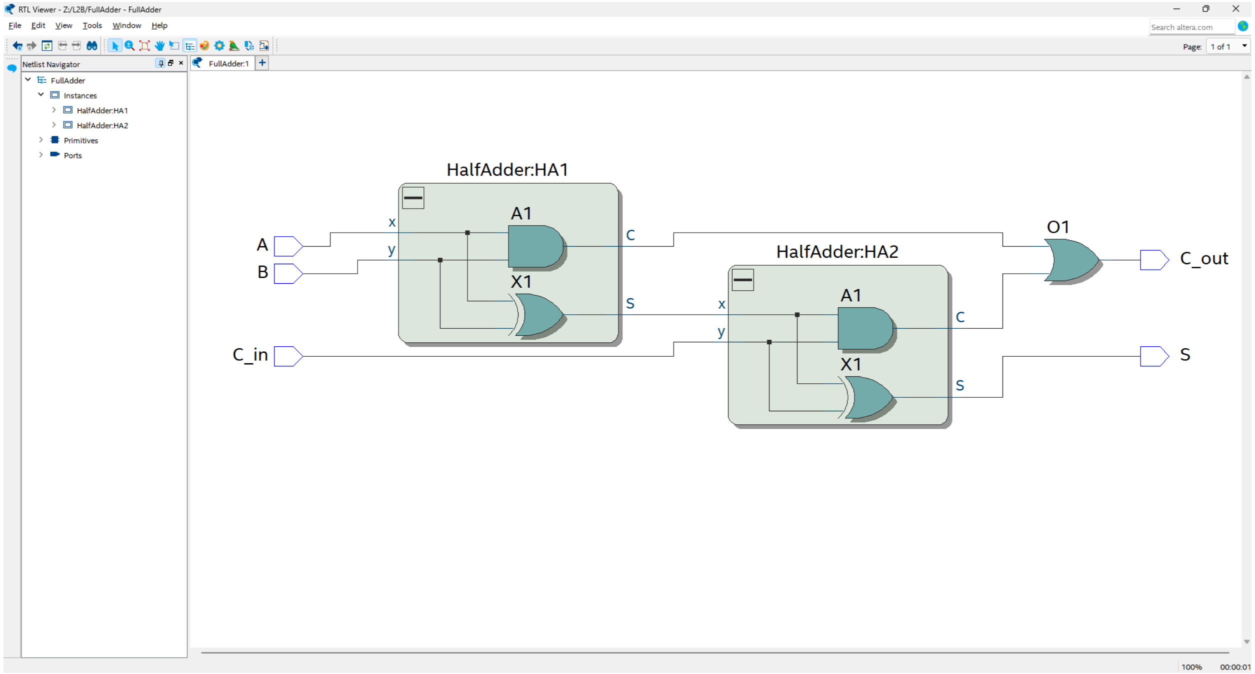
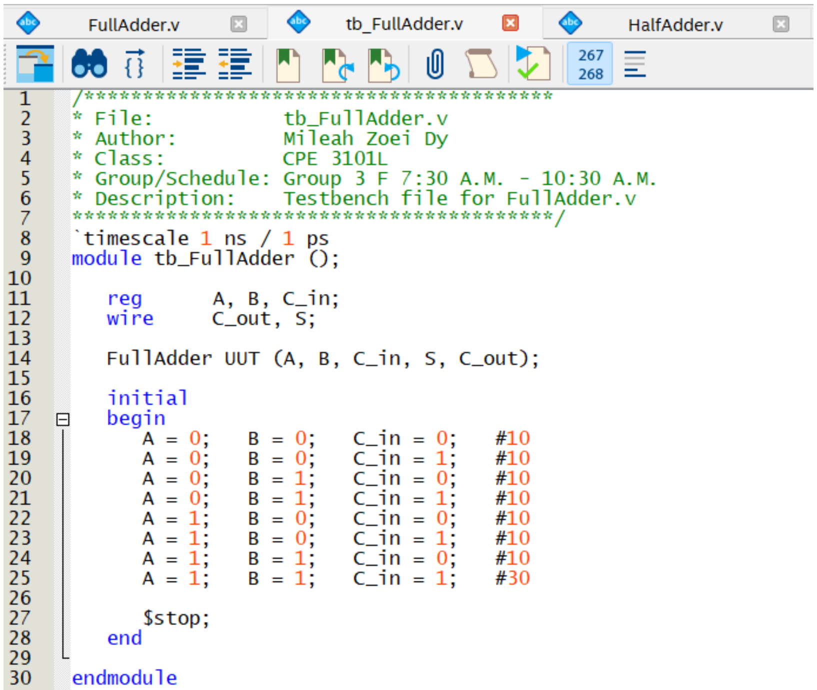


Figure 8. Schematic Diagram of Synthesized Circuit using RTL Viewer



```
/****************************************
* File:           tb_FullAdder.v
* Author:         Mileah Zoei Dy
* Class:          CPE 3101L
* Group/Schedule: Group 3 F 7:30 A.M. - 10:30 A.M.
* Description:    Testbench file for FullAdder.v
****************************************/
`timescale 1 ns / 1 ps
module tb_FullAdder ();

   reg         A, B, C_in;
   wire        C_out, S;

   FullAdder UUT (A, B, C_in, S, C_out);

   initial
   begin
      A = 0;   B = 0;   C_in = 0;   #10
      A = 0;   B = 0;   C_in = 1;   #10
      A = 0;   B = 1;   C_in = 0;   #10
      A = 0;   B = 1;   C_in = 1;   #10
      A = 1;   B = 0;   C_in = 0;   #10
      A = 1;   B = 0;   C_in = 1;   #10
      A = 1;   B = 1;   C_in = 0;   #10
      A = 1;   B = 1;   C_in = 1;   #30

      $stop;
   end

endmodule
```

Figure 9. Verilog Testbench File

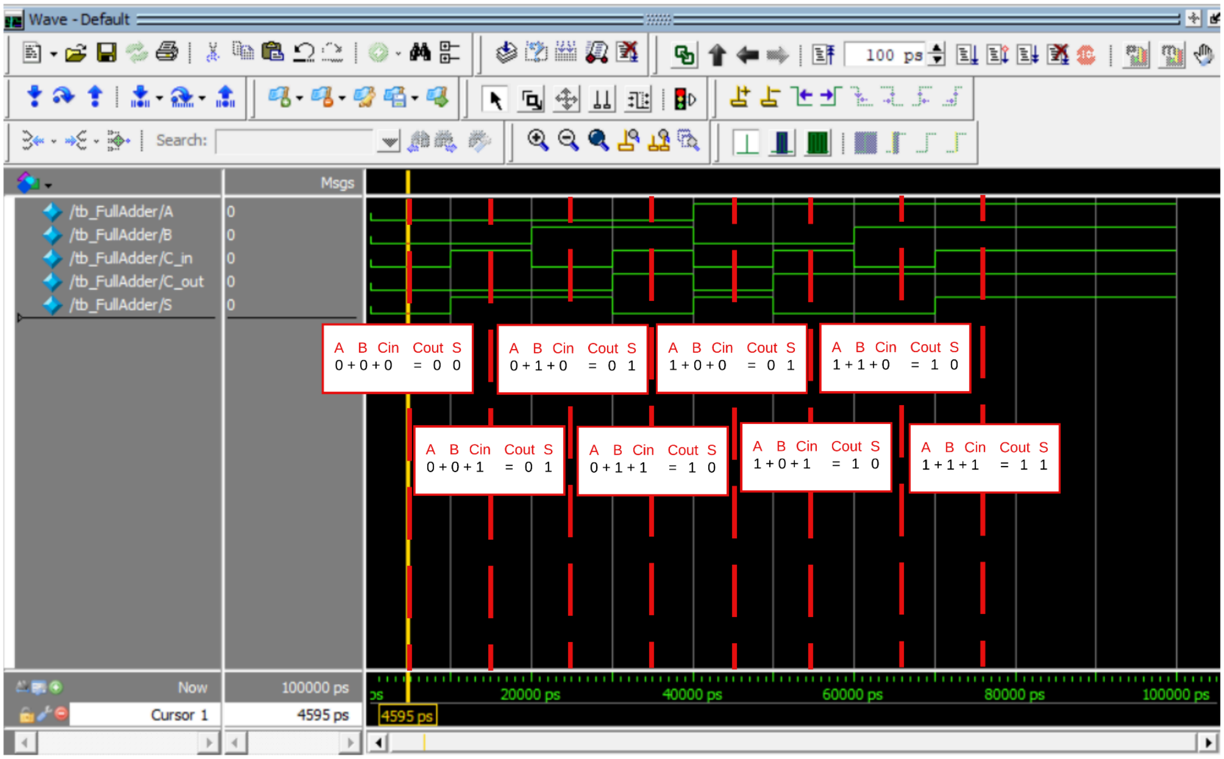


Figure 10. Testbench waveform for FullAdder.v (with annotations)