Laboratory Report # 3

Name: Lauron, John Enrico D. **Date Completed:** September 5, 2025

Laboratory Exercise Title: Structural Modeling of Combinational Circuits

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 3A: 2x4 Decoder

In this exercise, we designed a 2-to-4 line decoder with an active-high enable (E) using Verilog HDL with structural modeling and gate primitives. The circuit decodes the 2-bit input (A) into four unique outputs (D) based on the enable signal. Furthermore, the design was synthesized and simulated using a testbench file to verify the correct output waveform.

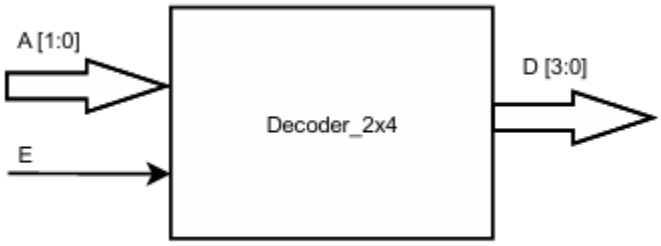


Figure 1.Entity Diagram of 2x4 Decoder

Truth Table:

Inputs		Output
E	A	D
0	xx	0000
1	00	0001
1	01	0010
1	10	0100
1	11	1000

Table 1. Truth Table for 2x4 Decoder

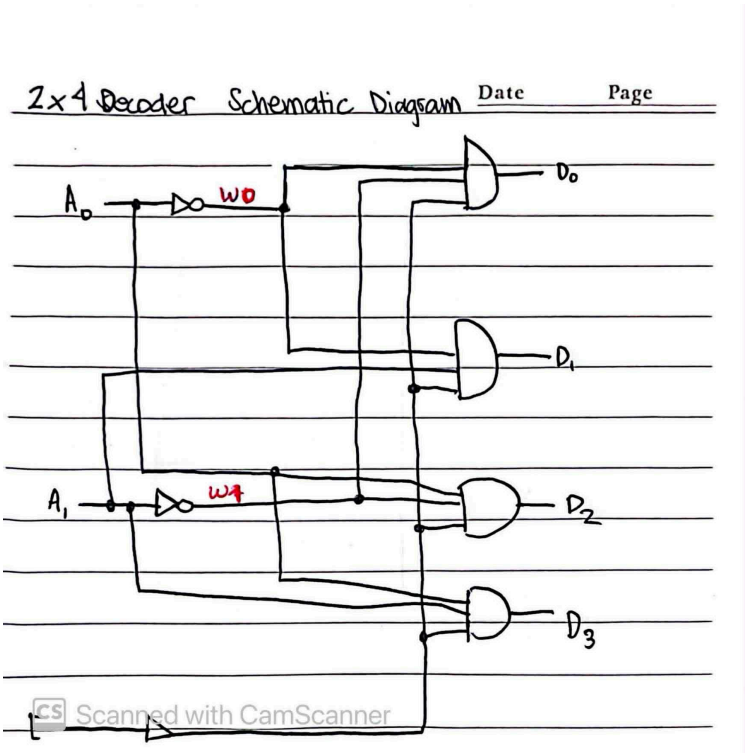


Figure 2. Schematic Diagram of a 2x4 Decoder

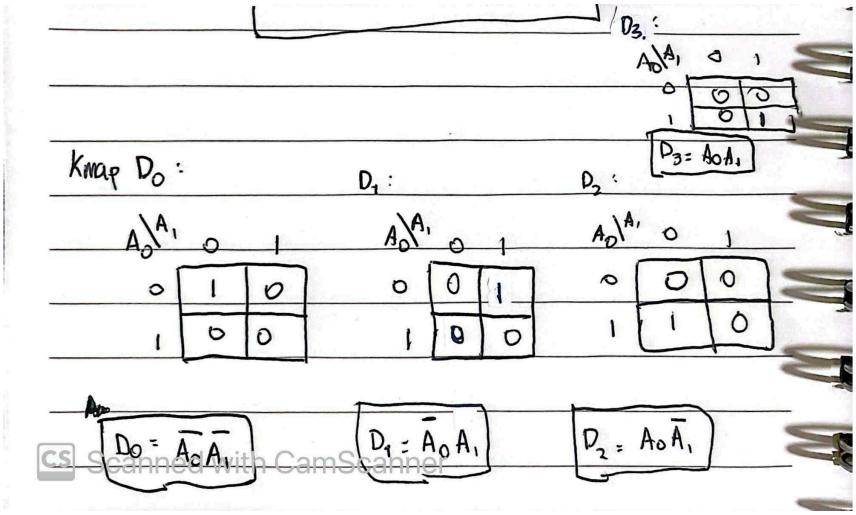


Figure 3. K-Maps for the 2x4 Decoder

Boolean Functions:

$\mathbf{D_0} = A_0' A_1'$ $\mathbf{D_1} = A_0' A_1$ $\mathbf{D_2} = A_0 A_1'$ $\mathbf{D_3} = A_0 A_1$

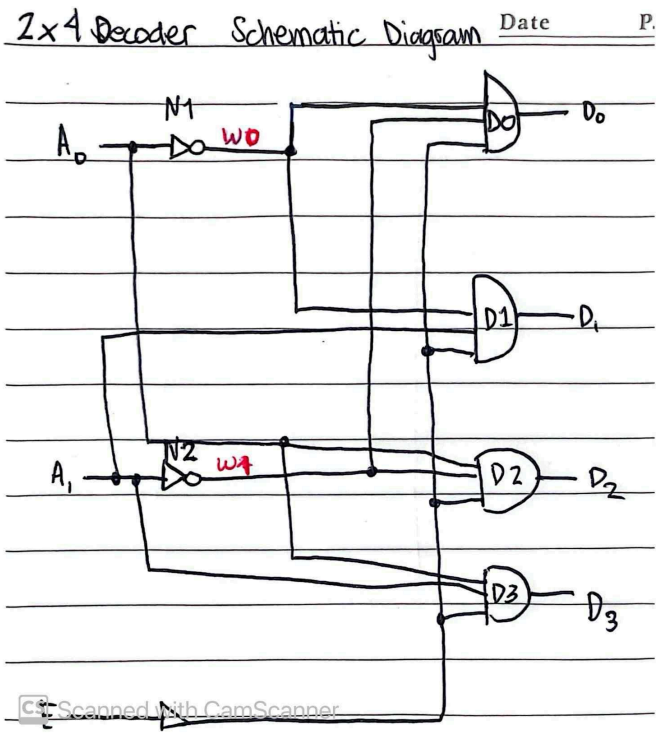


Figure 4. Circuit Diagram for 2x4 Decoder with Verilog HDL Labels

Flow Summary

<<Filter>>

Flow Status	Successful - Fri Sep 05 12:43:36 2025
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	Dec_2x4
Top-level Entity Name	Dec_2x4
Family	MAX 10
Device	10M50DCF484C7G
Timing Models	Final
Total logic elements	4
Total registers	0
Total pins	7
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Figure 5. Flow Summary of 2x4 Decoder

Type	ID	Message
		**********
		Running Quartus Prime Analysis & Synthesis
		Command: quartus_map --read_settings_files=on --write_settings_files=off Dec_2x4 -c Dec_2x4
⚠	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.
	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 dec_2x4.v
	12127	Elaborating entity "Dec_2x4" for the top level hierarchy
	286030	Timing-Driven Synthesis is running
	16010	Generating hard_block partition "hard_block:auto_generated_inst"
	21057	Implemented 11 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_npp Dec_2x4 -c Dec_2x4 --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 6. Synthesis Result of 2x4 Decoder

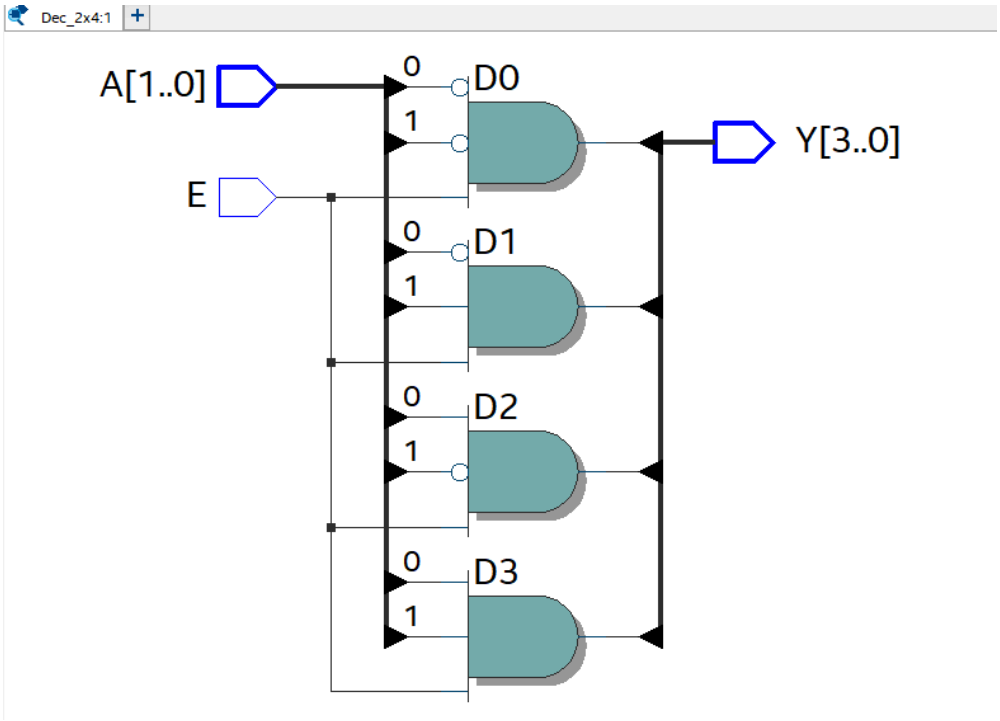


Figure 7. RTL View of 2x4 Decoder

Dec_2x4.v

```
1  /*****************************************
2  * File:                Dec_2x4.v
3  * Author:              John Enrico Desalago Lauron
4  * Class:               CPE 3101L
5  * Group/Schedule:      Group 1 Fri 7:30 - 10:30 AM
6  * Description:         Verilog HDL Code of a 2x4 Decoder
7  *****************************************/
8
9  // Structural Modeling of 2x4 Decoder
10 module Dec_2x4 (
11     input [1:0]     A,
12     input           E,
13     output [3:0]    Y
14 );
15
16
17     // Declare Signals
18     wire [1:0]      w;
19
20     // Implement with gate primitives
21     not N1 (w[0], A[0]);
22     not N2 (w[1], A[1]);
23
24     and D0 (Y[0], w[0], w[1], E);
25     and D1 (Y[1], w[0], A[1], E);
26     and D2 (Y[2], A[0], w[1], E);
27     and D3 (Y[3], A[0], A[1], E);
28
29 endmodule
30
```

Figure 8. Verilog HDL Code of a 2x4 Decoder

Dec_2x4.v | tb_Dec_2x4.v

```
/*************************************
* File:               tb_Dec_2x4.v
* Author:             John Enrico Desalago Lauron
* Class:              CPE 3101L
* Group/Schedule:     Group 1 Fri 7:30 - 10:30 AM
* Description:        Testbench file for Dec_2x4.v
*************************************/

`timescale 1 ns / 1 ps
module tb_Dec_2x4 ();

   // Testbench Signals
      reg [1:0]   A;
      reg         E;
      wire [3:0]  Y;

   // Instantiate the UUT
      Dec_2x4 UUT(A, E, Y);

   // generate stimuli
      initial begin
      $display ("Starting simulation at 0%d ns...", $time);

      E = 1'b0;    A = 2'b00;  #10
                   A = 2'b01;  #10
                   A = 2'b10;  #10
                   A = 2'b11;  #10

      E = 1'b1;    A = 2'b00;  #10
                   A = 2'b01;  #10
                   A = 2'b10;  #10
                   A = 2'b11;  #10
      $display ("Finished simulation at %0dns.", $time);

      $stop;
      end

endmodule

```

Figure 9. Verilog HDL Code of Testbench for a 2x4 Decoder

Figure 10. Testbench Waveform of a 2x4 Decoder (with annotations)

Exercise 3B: 4-bit Adder

In this exercise, we designed a 4-bit adder using Verilog HDL with structural modeling, based on the previously created full adder module. The circuit adds two 4-bit inputs (A and B) along with a carry-in (Cin) to generate a 4-bit sum (S) and a carry-out (Cout). Furthermore, the design was synthesized and simulated using a testbench file to confirm the functionality through waveform verification.

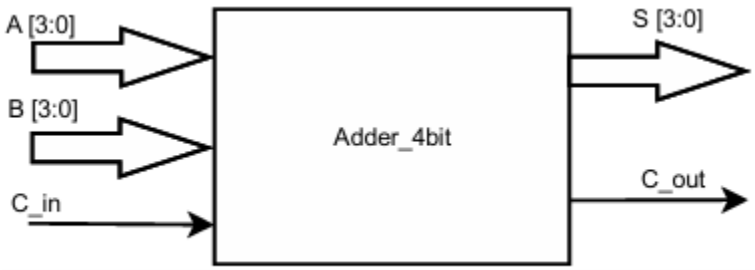


Figure 2. Entity Diagram of a 4-Bit Adder

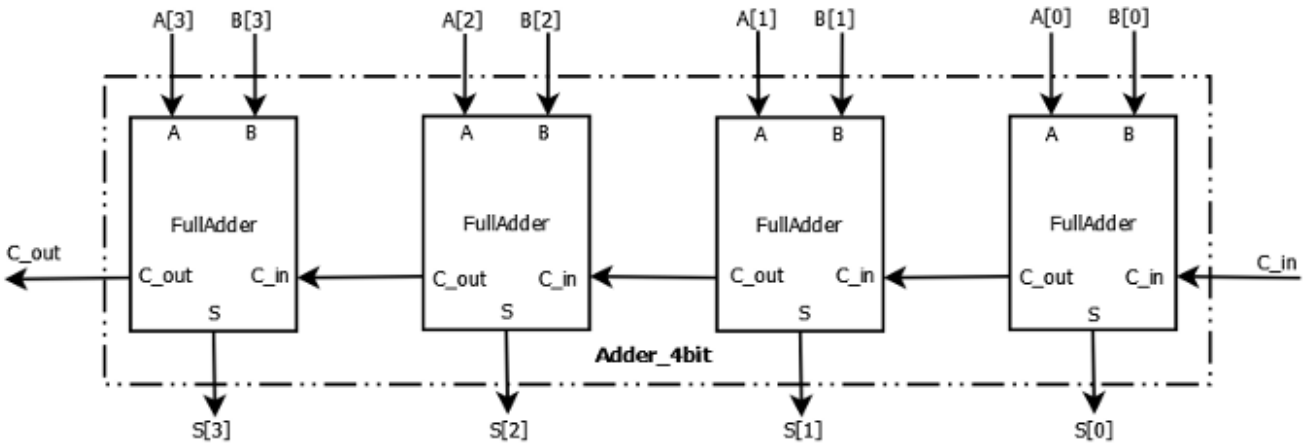


Figure 3. Structural Diagram of a 4-Bit Adder

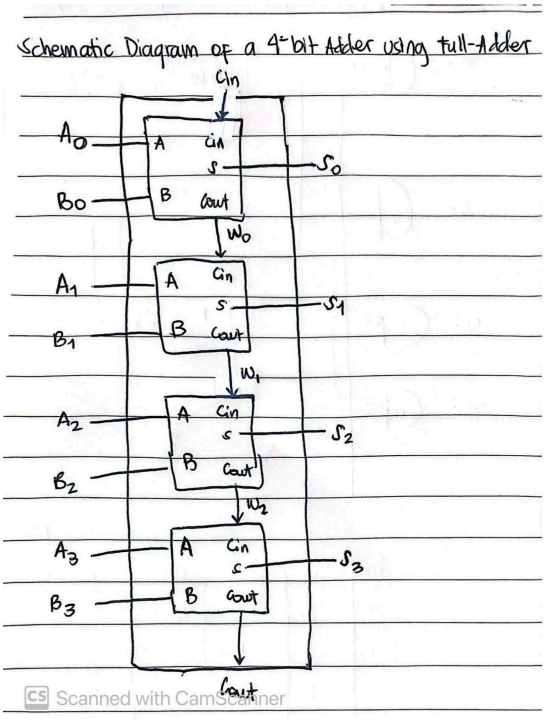
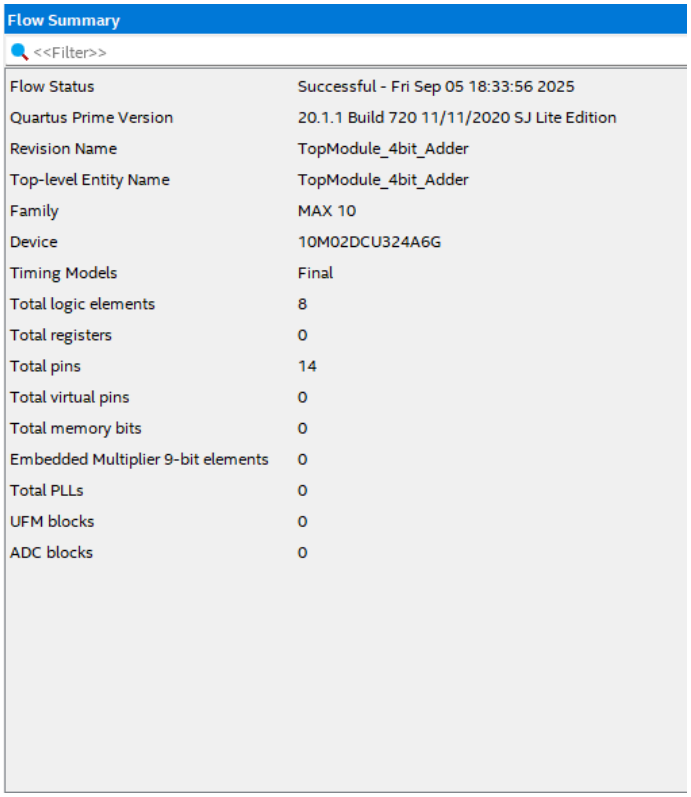


Figure 11. Circuit Diagram of a 4-bit Adder using Full-Adder with Verilog HDL Labels

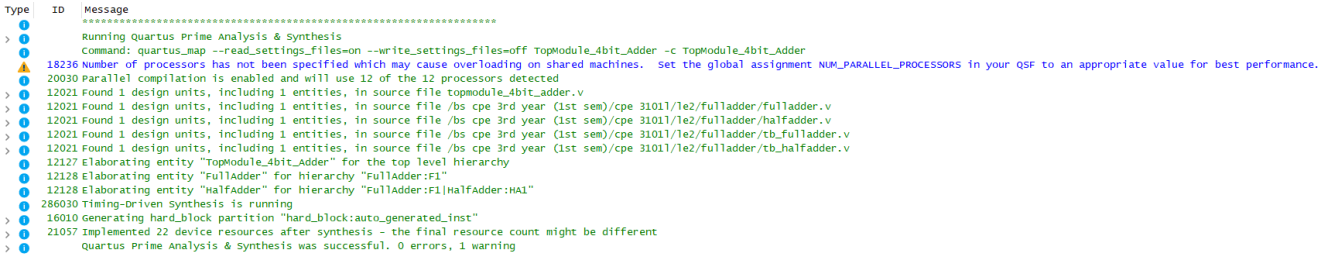


Flow Summary

<<Filter>>

Flow Status	Successful - Fri Sep 05 18:33:56 2025
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	TopModule_4bit_Adder
Top-level Entity Name	TopModule_4bit_Adder
Family	MAX 10
Device	10M02DCU324A6G
Timing Models	Final
Total logic elements	8
Total registers	0
Total pins	14
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Figure 12. Flow Summary of TopModule_4bit_Adder



```
Type   ID     Message
       ********************************************************************
       Running Quartus Prime Analysis & Synthesis
       Command: quartus_map --read_settings_files=on --write_settings_files=off TopModule_4bit_Adder -c TopModule_4bit_Adder
  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.
  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 topmodule_4bit_adder.v
  12021 Found 1 design units, including 1 entities, in source file /bs cpe 3rd year (1st sem)/cpe 3101l/le2/fulladder/fulladder.v
  12021 Found 1 design units, including 1 entities, in source file /bs cpe 3rd year (1st sem)/cpe 3101l/le2/fulladder/halfadder.v
  12021 Found 1 design units, including 1 entities, in source file /bs cpe 3rd year (1st sem)/cpe 3101l/le2/fulladder/tb_fulladder.v
  12021 Found 1 design units, including 1 entities, in source file /bs cpe 3rd year (1st sem)/cpe 3101l/le2/fulladder/tb_halfadder.v
  12127 Elaborating entity "TopModule_4bit_Adder" for the top level hierarchy
  12128 Elaborating entity "FullAdder" for hierarchy "FullAdder:F1"
  12128 Elaborating entity "HalfAdder" for hierarchy "FullAdder:F1|HalfAdder:HA1"
 286030 Timing-Driven Synthesis is running
  16010 Generating hard_block partition "hard_block:auto_generated_inst"
  21057 Implemented 22 device resources after synthesis - the final resource count might be different
       Quartus Prime Analysis & Synthesis was successful. 0 errors, 1 warning
```

Figure 13. Synthesis Result of TopModule_4bit_Adder

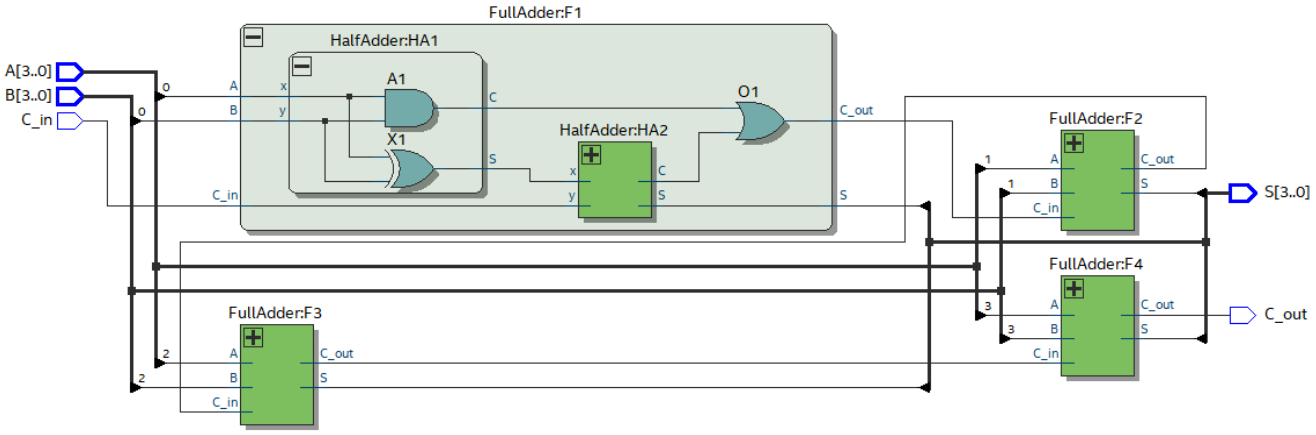


Figure 14. RTL View of TopModule_4bit_Adder

TopModule_4bit_Adder.v | FullAdder.v | HalfAdder.v | Compilation Report - TopModule_4bit_Adder

```
1  /************************************
2  * File:              TopModule_4bit_Adder.v
3  * Author:            John Enrico Desalago Lauron
4  * Class:             CPE 3101L
5  * Group/Schedule:    Group 1 Fri 7:30 - 10:30 AM
6  * Description:       Verilog HDL Code of a 4-bit Adder using Full-Adder
7  ************************************/
8  // Structural Modeling of 4-bit Ripple Carry Adder
9  module TopModule_4bit_Adder (
10     input [3:0]    A,
11     input [3:0]    B,
12     input          C_in,
13     output         C_out,
14     output [3:0]   S
15 );
16     // Declare Signals
17     wire [2:0] w;
18
19     // Instantiate First Full Adder (LSB)
20     FullAdder F1(
21         .A      (A[0]),
22         .B      (B[0]),
23         .C_in   (C_in),
24         .C_out  (w[0]),
25         .S      (S[0])
26     );
27
28     // Instantiate Second Full Adder
29     FullAdder F2(
30         .A      (A[1]),
31         .B      (B[1]),
32         .C_in   (w[0]),
33         .C_out  (w[1]),
34         .S      (S[1])
35     );
36     // Instantiate Third Full Adder
37     FullAdder F3(
38         .A      (A[2]),
39         .B      (B[2]),
40         .C_in   (w[1]),
41         .C_out  (w[2]),
42         .S      (S[2])
43     );
44
45     // Instantiate Fourth Full Adder (MSB)
46     FullAdder F4(
47         .A      (A[3]),
48         .B      (B[3]),
49         .C_in   (w[2]),
50         .C_out  (C_out),
51         .S      (S[3])
52     );
53
54 endmodule
```

Figure 15. Verilog HDL Code of TopModule_4bit_Adder

TopModule_4bit_Adder.v | FullAdder.v | HalfAdder.v | tb_TopModule_4bit_Adder.v* | Compilation Report - TopModule_4bit_Adder

```
 3   * Author:          John Enrico Desalago Lauron
 4   * Class:           CPE 3101L
 5   * Group/Schedule:  Group 1 Fri 7:30 - 10:30 AM
 6   * Description:     Testbench for TopModule_4bit_Adder.v
 7   ************************************/
 8
 9   `timescale 1 ns / 1 ps
10
11   // Testbench Module
12   module tb_TopModule_4bit_Adder;
13
14       // Declare Testbench Signals
15       reg  [3:0] A;
16       reg  [3:0] B;
17       reg        C_in;
18       wire [3:0] S;
19       wire       C_out;
20
21       // Instantiate the UUT
22       TopModule_4bit_Adder UUT (
23           .A     (A),
24           .B     (B),
25           .C_in  (C_in),
26           .S     (S),
27           .C_out (C_out)
28       );
29
30       // generate stimuli
31       initial begin
32           // Display header in simulation log
33           $display("Time\tA\tB\tC_in\t|\tS\tC_out");
34           $display("-------------------------------------------------");
35
36           // Apply another 8 input combinations
37               A = 4'b0100; B = 4'b0010; C_in = 0; #10;
38               A = 4'b0110; B = 4'b0011; C_in = 1; #10;
39               A = 4'b1001; B = 4'b0100; C_in = 0; #10;
40               A = 4'b0111; B = 4'b1000; C_in = 1; #10;
41               A = 4'b1011; B = 4'b0110; C_in = 0; #10;
42               A = 4'b1100; B = 4'b1100; C_in = 1; #10;
43               A = 4'b1000; B = 4'b1000; C_in = 0; #10;
44               A = 4'b0110; B = 4'b0110; C_in = 1; #10;
45
46           // End simulation
47           $stop;
48       end
49
50       // Monitor signal changes
51       initial begin
52           $monitor("%0dns\t%b\t%b\t%b\t|\t%b\t%b",
53                    $time, A, B, C_in, S, C_out);
54       end
55
56   endmodule
```

Figure 16. Testbench Verilog HDL Code of TopModule_4bit_Adder

```
# run -all
# Time  A       B       C_in    |       S       C_out
# -------------------------------------------------
# 0ns   0000    0000    0       |       0000    0
# 10ns  0001    0010    0       |       0011    0
# 20ns  0011    0011    1       |       0111    0
# 30ns  0101    0011    0       |       1000    0
# 40ns  1111    0001    0       |       0000    1
# 50ns  1010    0101    1       |       0000    1
# 60ns  1111    1111    0       |       1110    1
# 70ns  1111    1111    1       |       1111    1
# ** Note: $stop    : D:/BS CpE 3rd Year (1st Sem)/CPE 3101L/LE3/3B/tb_TopModule_4bit_Adder.v(47)
#    Time: 80 ns  Iteration: 0  Instance: /tb_TopModule_4bit_Adder
# Break in Module tb_TopModule_4bit_Adder at D:/BS CpE 3rd Year (1st Sem)/CPE 3101L/LE3/3B/tb_TopModule_4bit_Adder.v line 47
```

Figure 17. Transcript for the Testbench Output

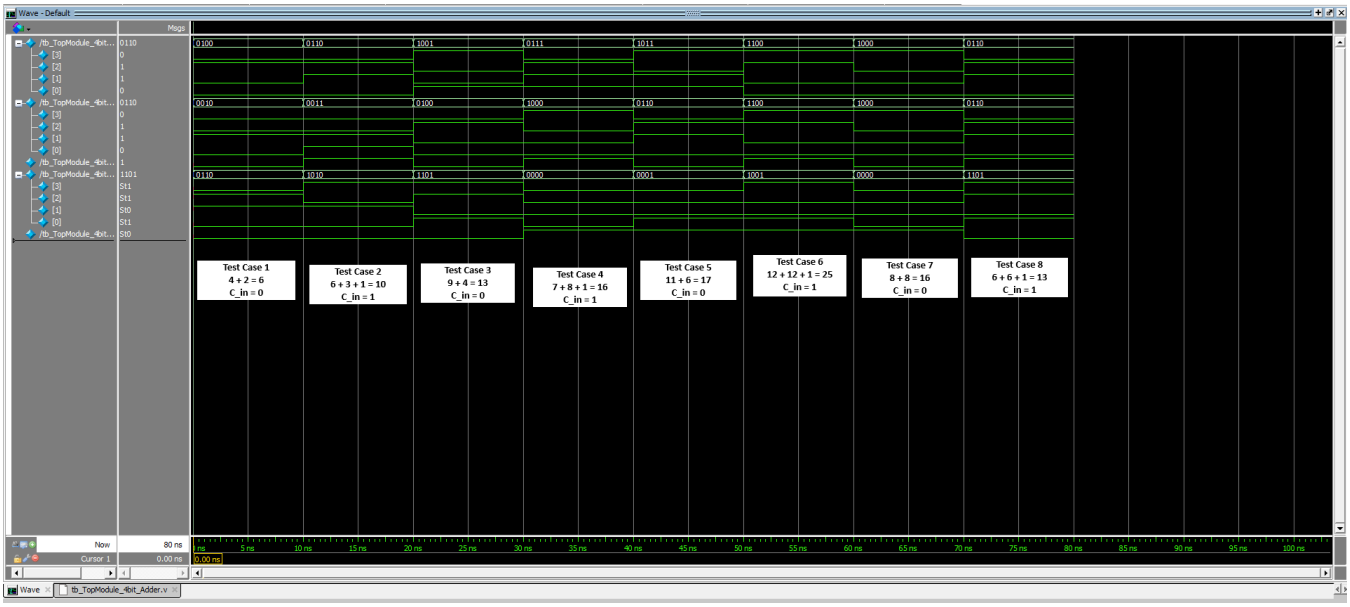


Figure 18. Testbench Waveform of TopModule_4bit_Adder (with annotations)