Laboratory Report # 2

Name: **Paul Emmanuel G. Corsino** **Date Completed:** **9/12/2023**

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 2C:

For the Laboratory Report, prepare and take screen shots of the following items for Exercise 2C:

- **Verilog HDL Design Entry** of the following:
  - Exercise 2B's Full Adder *(since this is used in Exercise 2C)*
  - Exercise 2C's 4-bit Adder
  - Show your ***name*** and ***course group with schedule*** in the comments
- **Compilation Report for the Flow Summary**
  - Indicate how many logic elements and pins were used.
- **Schematic Diagram of Synthesized Circuit** using RTL Viewer
- **Verilog Testbench File**, showing your ***name*** and ***course group with schedule*** in the comments
- **Testbench Waveform with your annotations**, such as in Figure 9

Figure 1. Flow Summary

```
1  /************************************************
2  *
3  *FILE:          FullAdder.v
4  *AUTHOR:        Paul Emmanuel G. Corsino
5  *CLASS:         CpE 3101L
6  *SCHEDULE:      Group 2 FRI 11:00 AM - 2:00 PM
7  *DESCRIPTION:   Simulates a Full Adder
8  *
9  ************************************************/
10
11 module FullAdder (A, B, C_in, S, C_out);
12
13    input    A, B, C_in;
14    output   S, C_out;
15    wire     w1, w2, w3;
16
17    xor xor1 (w1,A,B);
18    xor xor2 (S,w1,C_in);
19
20    and and1 (w2,A,B);
21    and and2 (w3,C_in,w1);
22
23    or  or1  (C_out,w3,w2);
24
25 endmodule
```

Figure 2. Exercise 2B's Full Adder

```
1  /*************************************************
2  *
3  *FILE:          Adder_4bit
4  *AUTHOR:        Paul Emmanuel G. Corsino
5  *CLASS:         CpE 3101L
6  *SCHEDULE:      Group 2 FRI 11:00 AM - 2:00 PM
7  *DESCRIPTION:   Simulates a 4 bit adder
8  *
9  *************************************************/
10
11 module Adder_4bit (A, B, C_in, S, C_out);
12
13    input    [3:0] A, B;
14    input          C_in;
15    output   [3:0] S;
16    output         C_out;
17    wire           w1, w2, w3;
18
19    FullAdder fa1 (A[0], B[0], C_in, S[0], w1);
20    FullAdder fa2 (A[1], B[1], C_in, S[1], w2);
21    FullAdder fa3 (A[2], B[2], C_in, S[2], w3);
22    FullAdder fa4 (A[3], B[3], C_in, S[3], C_out);
23
24 endmodule
25
```

Figure 3. Exercise 2C's 4-bit Adder

Logic Elements:

Full Adder = 5 (xor1,xor2,and1,and2,or1)

4-bit Adder = 4 (fa1,fa2,fa3,fa4)

Pins:

Full Adder = 5 (A,B,C_in,C_out,S)

4-bit Adder = 14 (A[0],A[1],A[2],A[3],B[0],B[1],B[2],B[3],C_in,C_out,S[0],S[1],S[2],S[3])

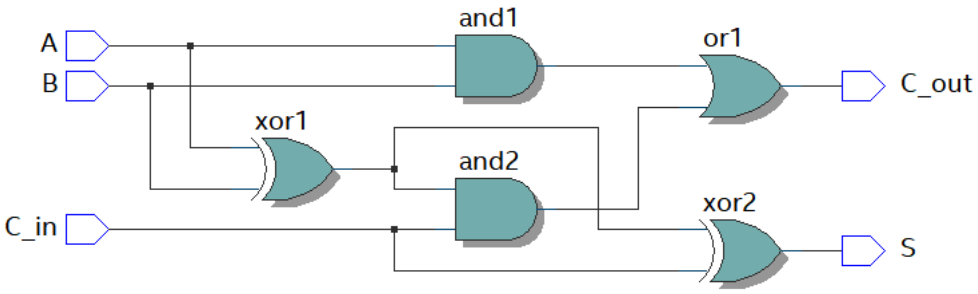

Figure 4. Full Adder Schematic Diagram

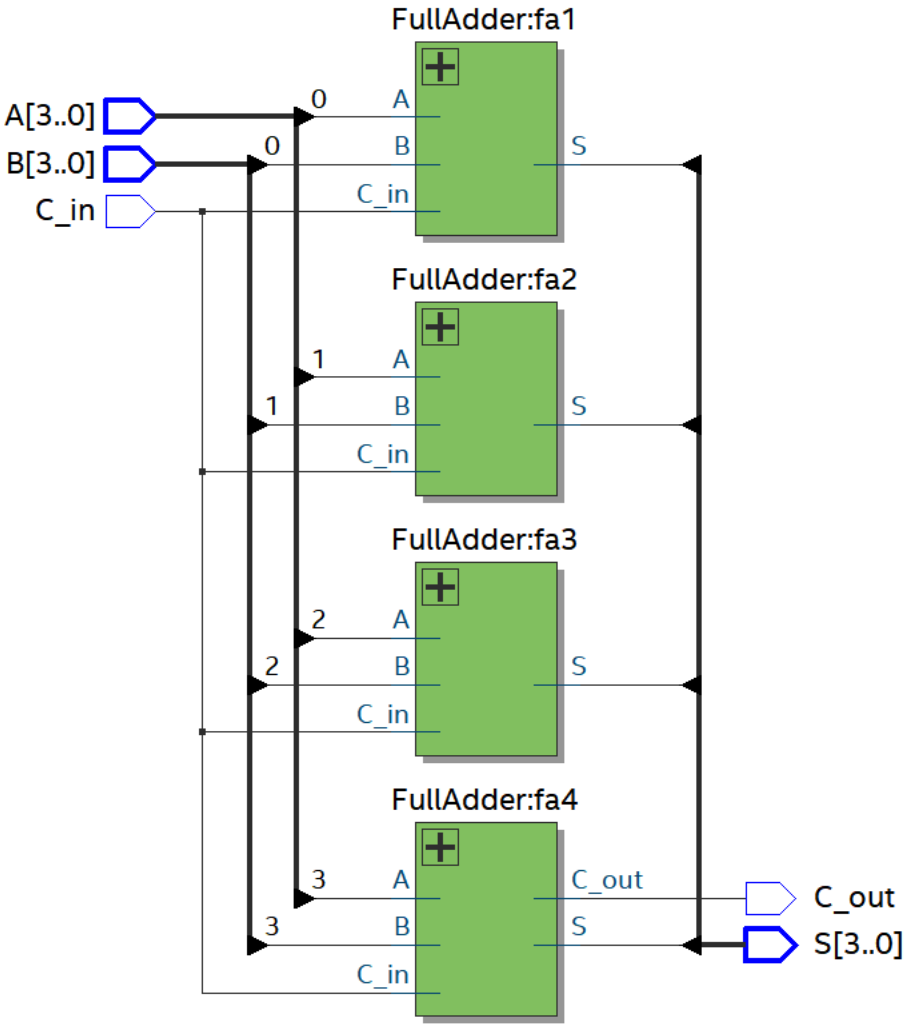

Figure 5. 4-bit Adder Schematic Diagram

```
1  /*************************************************
2  *
3  *FILE:          tb_Adder_4bit.v
4  *AUTHOR:        Paul Emmanuel G. Corsino
5  *CLASS:         CpE 3101L
6  *SCHEDULE:      Group 2 FRI 11:00 AM - 2:00 PM
7  *DESCRIPTION:   Testbench file for Adder_4bit.v
8  *
9  *************************************************/
10
11 `timescale 1 ns / 1 ps
12 module tb_Adder_4bit();
13
14    reg   [3:0]    A, B;
15    reg            C_in;
16
17    wire  [3:0]    S;
18    wire           C_out;
19
20    Adder_4bit UUT (A, B, C_in, S, C_out);
21
22
23    initial
24    begin
25
26       A = 4'd0;   B = 4'd0;   C_in = 0;   #10
27       A = 4'd3;   B = 4'd8;   C_in = 1;   #10
28       A = 4'd11;  B = 4'd3;   C_in = 0;   #10
29       A = 4'd12;  B = 4'd6;   C_in = 0;   #10
30       A = 4'd5;   B = 4'd4;   C_in = 1;   #10
31       A = 4'd1;   B = 4'd9;   C_in = 0;   #10
32       A = 4'd15;  B = 4'd15;  C_in = 0;   #10
33       A = 4'd15;  B = 4'd15;  C_in = 1;   #10
34
35       $stop;
36    end
37
38 endmodule
```

Figure 6. Verilog Testbench File

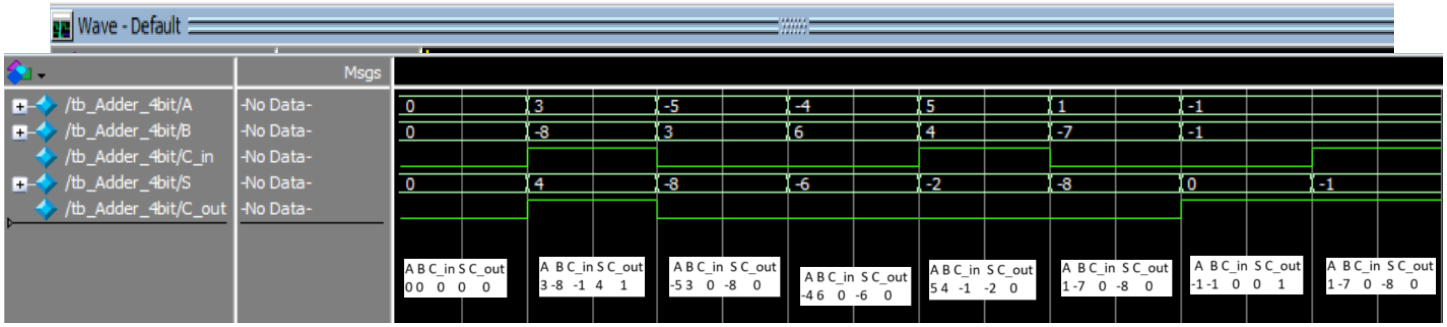

Figure 7. Testbench Waveform w/ annotations

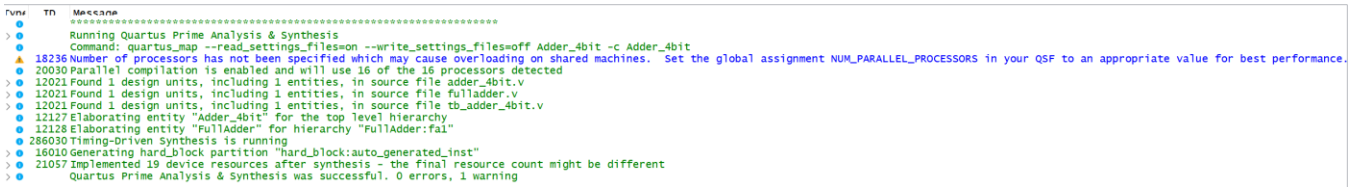

Figure 8. Message window