Jefferson Bui

Paul Kyriakos

10/21/19

Yi Yang

CMPEN 275

3 Bit Adder

Lab 5 Report

Objective:

In this lab we will learn how to write a Verilog code for a simple adder circuit using Gate-level modelling, how to use Dataflow and Behavioral modelling to describe the same circuit, and how to test and validate your design on an FPGA.

Codes:

module adder\_gate\_onebit(A,B,C,Cout,S); \*Implements the code and variables to the file

input A,B,C; \* Sets the input variables as A,B and C

output S,Cout; \* Sets the output variables as S and Cout

wire X,Y,Z; \* Sets the wire variables as X, Y and Z

xor a1(X,A,B); \*Uses Xor gate(a1) to connect Variables A,B and X

xor a2(S,X,C); \*Uses Xor gate(a2) to connect Variables S,X and Z

and a3(Y,C,X); \*Uses and gate(a3) to connect Variables Y,C and X

and a4(Z,A,B); \*Uses and gate(a4) to connect Variables Z, A and B

or a5(Cout,Y,Z); \*Uses or gate(a5) to connect Variables Cout Y and Z

Endmodule \* Ends the code



1

2

3

4

5

6

7

8

1- Implements the code and the variables to the file

2- Sets input variables as A0,A1,A2,B0,B1,B2,Cin

3- Sets the output variables as S0,S1,S2,Cout2

4- Sets the wire variables as Cout0 and Cout1

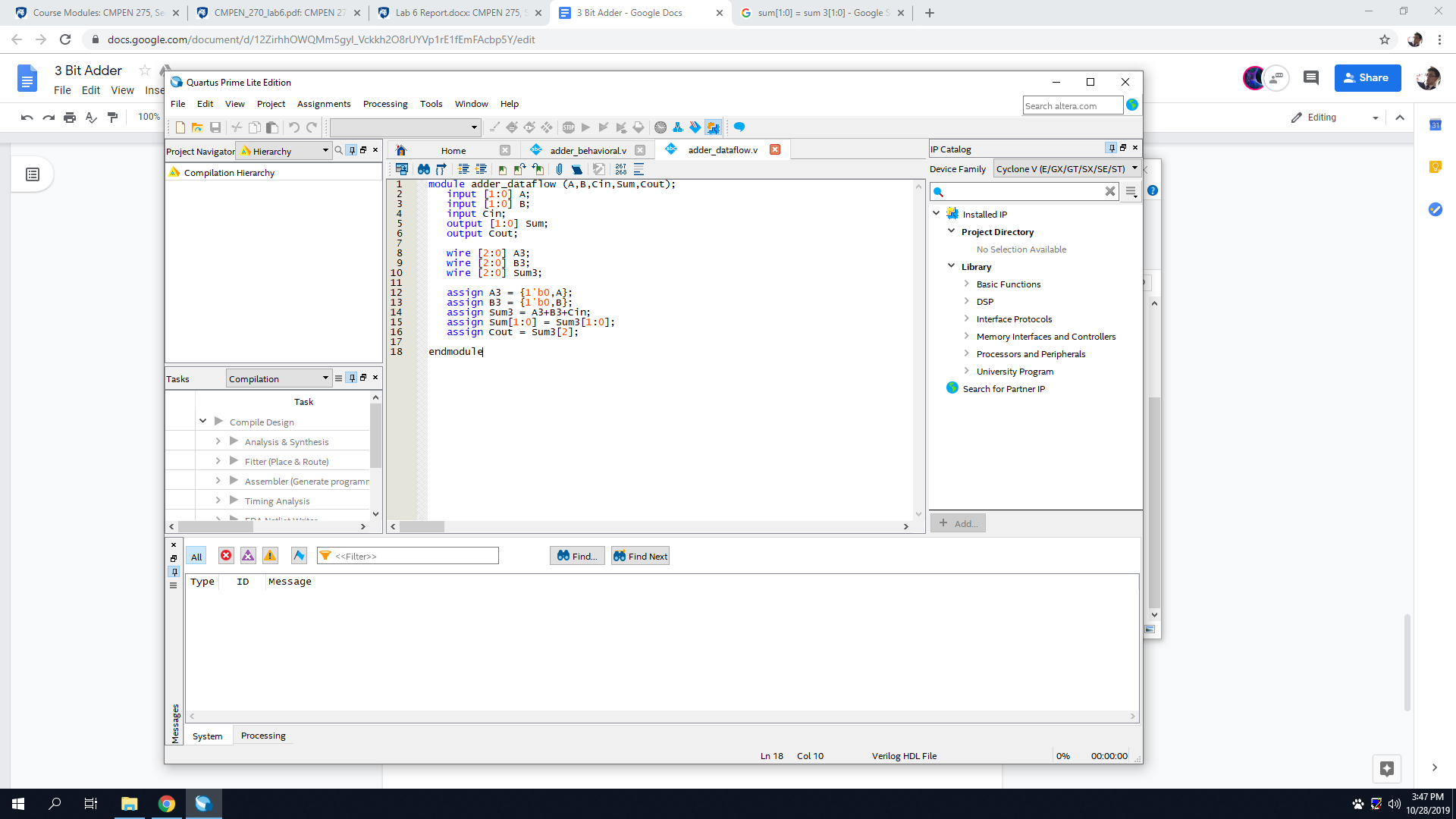
5- Calls back to adder\_gate\_onebit (Adder0) to run for A0, B0, Cin, Cout0, and S0

6- Calls back to adder\_gate\_onebit (Adder1) to run for A1, B1, Cout0, Cout1, and S1

7- Calls back to adder\_gate\_onebit (Adder2) to run for A2, B2, Cout1, Cout2, and S2

8- Ends the code

Dataflow:



1 - Implements the code and variables to the file

2 - Sets input variables as A1,A0,B1,B0, and Cin

3 - Sets output variables as Sum1,Sum0, and Cout

4 - Sets wires as A3[2],A3[1],A3[0], doing the same for B3 and Sum3

5 - Assigns variable A3 as a one-bit binary # with value of 0

6 - Assigns variable B3 as a one-bit binary # with value of 0

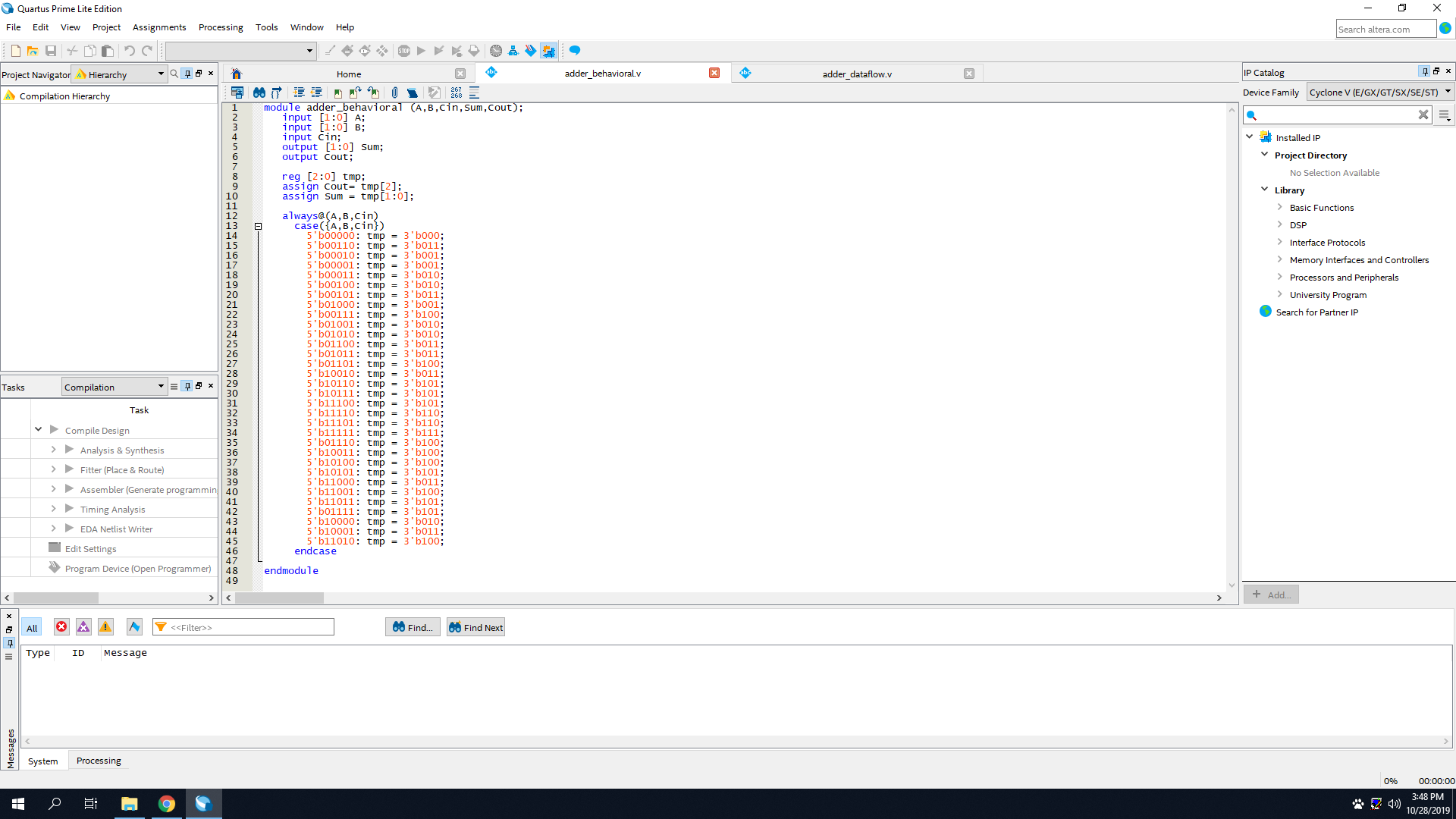
7 - Assigns Sum3 as A3(which is 0)+B3(which is 0)+Cin

8 - Assigns Sum[1] and Sum[0] equal to Sum3[1] and Sum3[0]

9 - Assigns Cout as Sum3[2]

10 - Ends code

Behavioral:



1- Implements the code and the variables to the file

2- Sets input variables as A1, A0, B1, B0, and Cin

3- Sets the output variables as S0,S1,S2,Cout2

4- Sets output variables as Sum1,Sum0, and Cout

5- Sets temporary variables for Cout as 2, and for Sum as 1 and 2

6- Starts the code always at A, B and Cin

7- Sets the variables (A,B,Cin) that affect the procedure

8- Converts 5 bit binary into 3 bit binary for temporarys

9- Ends the case

10- ends the code