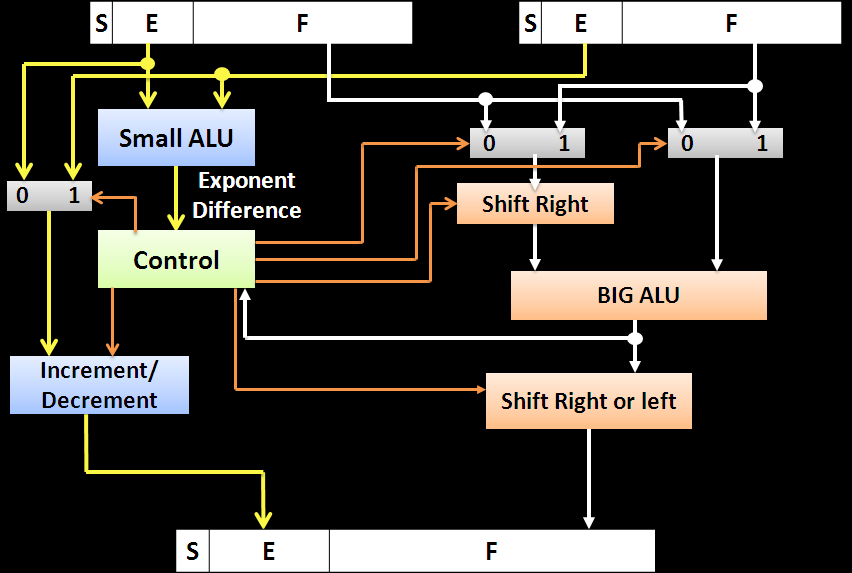
**Experiment No 9: Implementation of Floating point Adder**

The aim of this experiment is to implement a floating point Adder circuit and synthesize to check LUT usage and delay of the circuit.



Sel1

Sel0

Intermediate Mantissa

ShiftMantisssa

B

A

IncDecE

Carryout

SelE

ShiftAmt

MUX C

MUX0

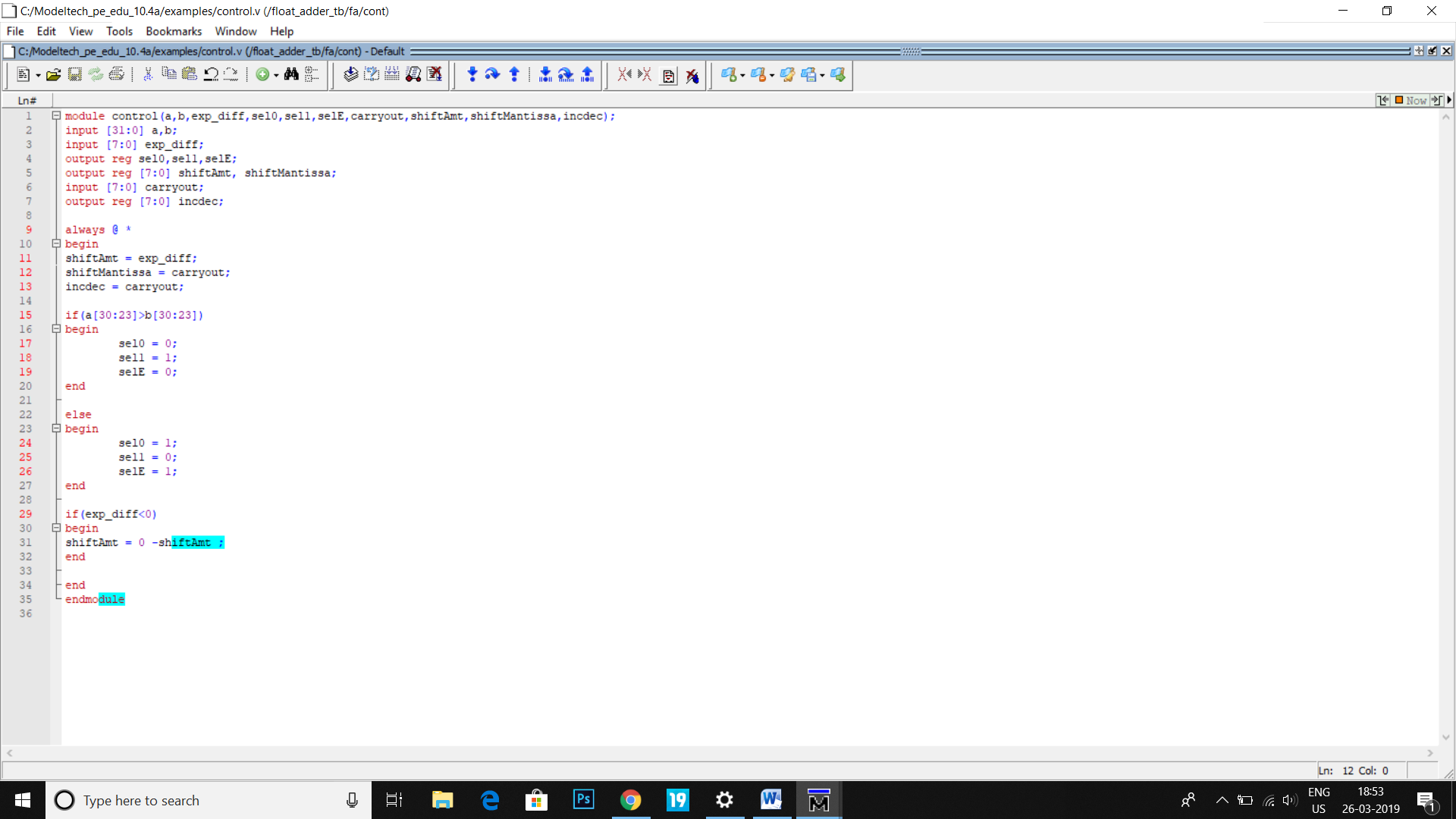
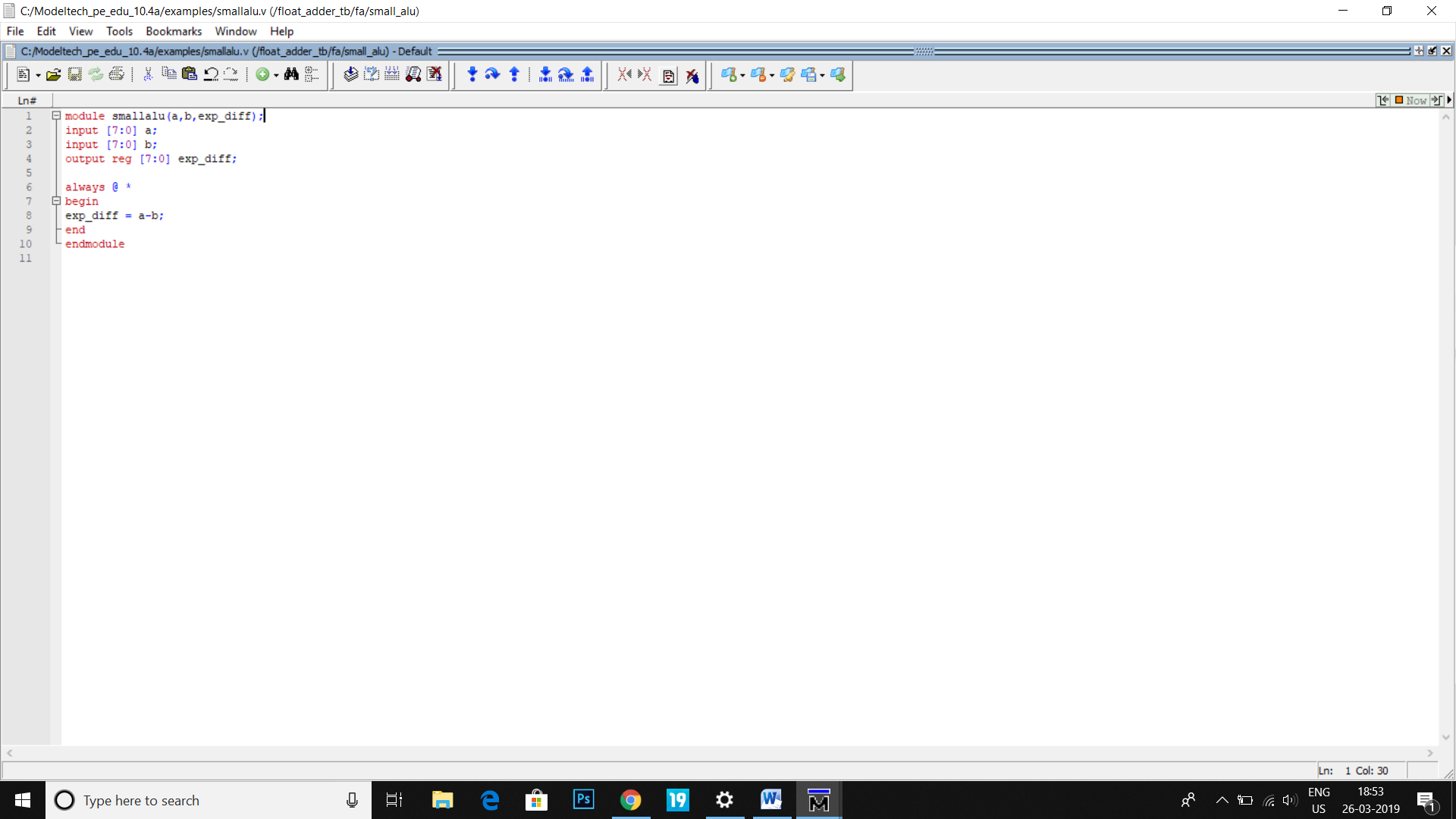
MUX1

**Exercise 9.1 Implement a floating point adder. Assume the floating point numbers are in IEEE 754 single precision format. Assume both the inputs will have same sign.**

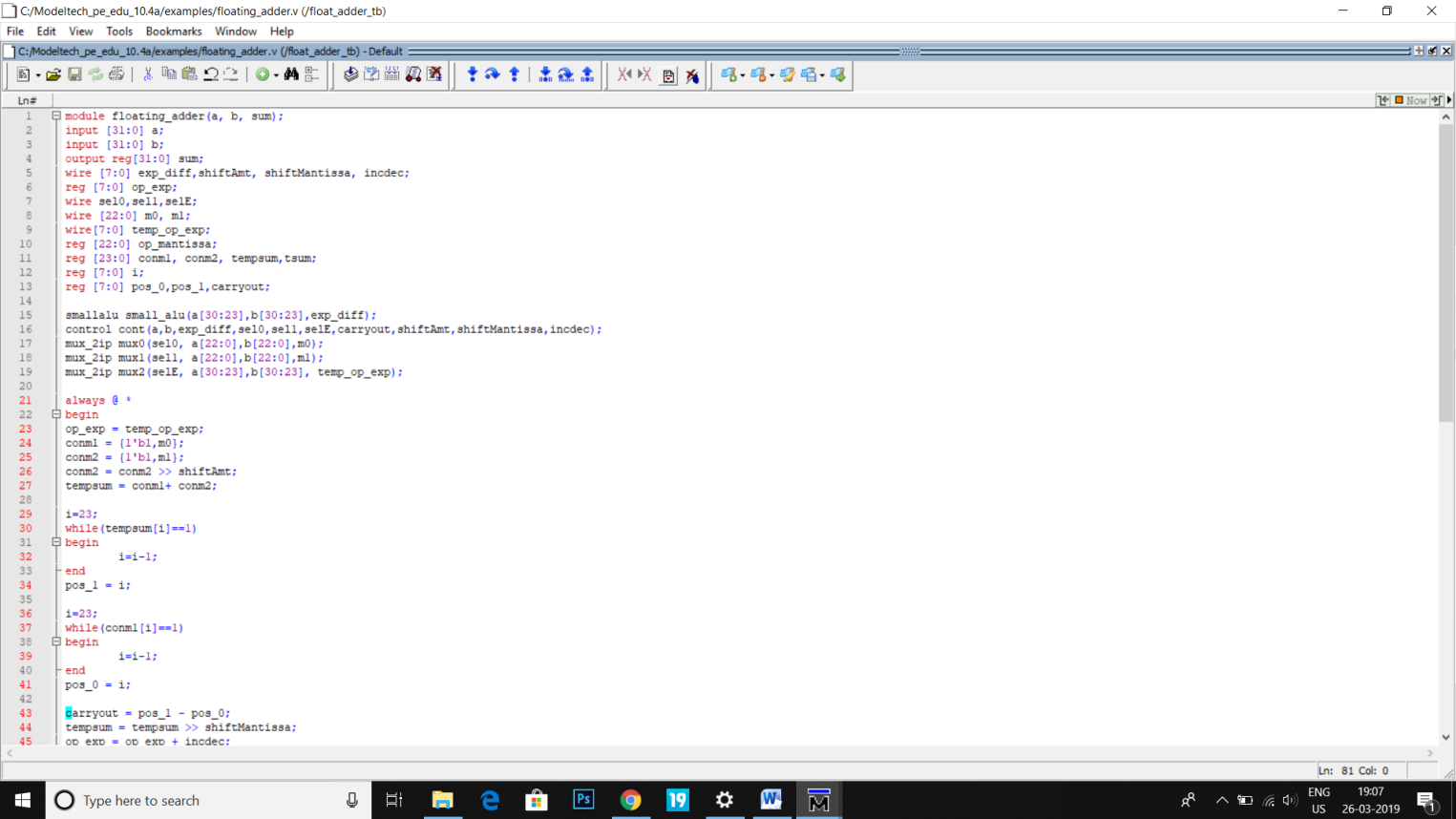
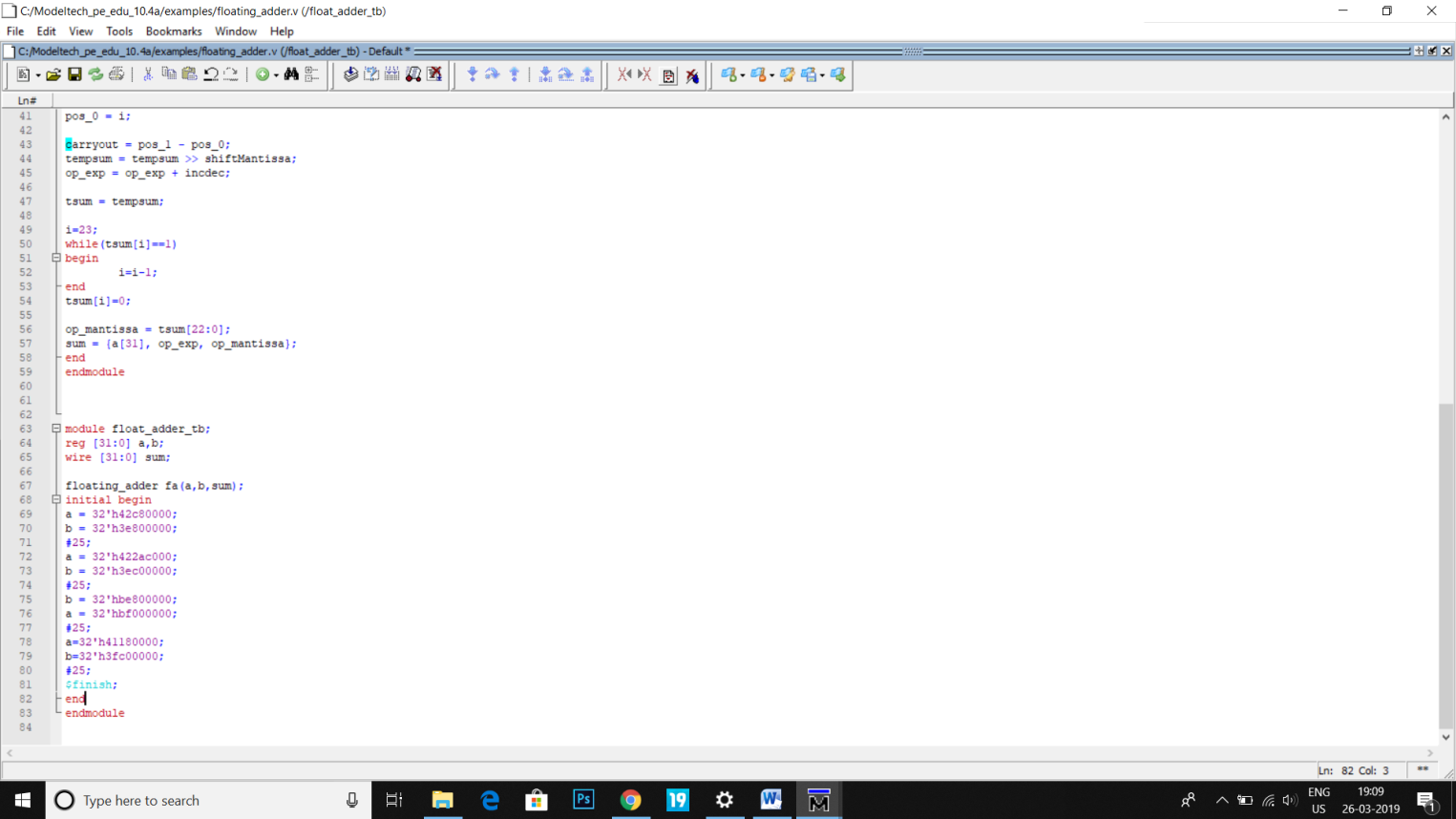
Any other assumptions made should be clearly specified here

Assumptions: Sign of both the numbers is same i.e. both are positive or both are negative.   
There is no overflow or underflow.

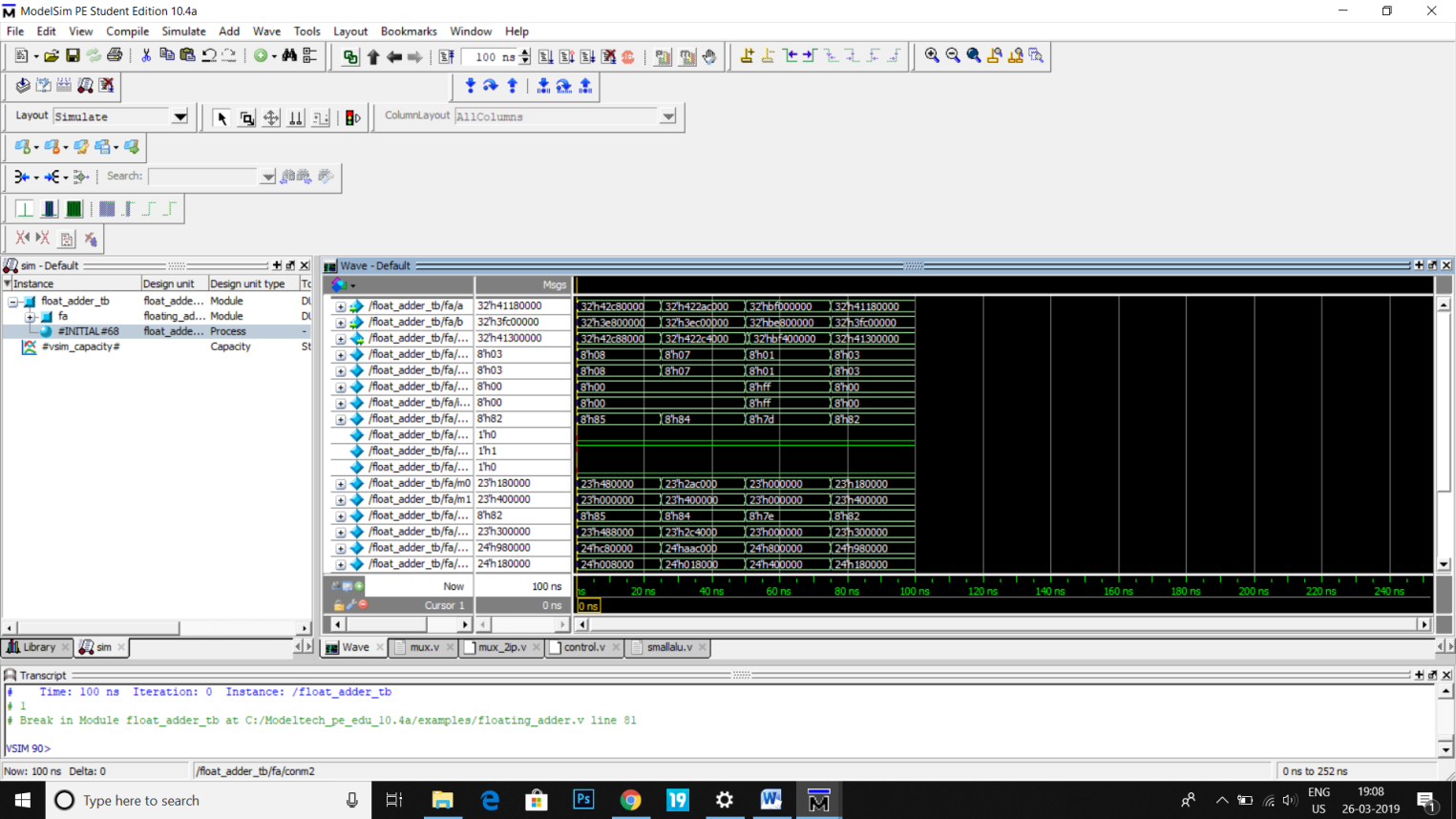
1. **Implement the submodules which are necessary for implementation of floating point addition and copy the images of those Verilog codes here.**

Answer: 

1. **Implement complete floating point adder in Verilog (by instantiating all the submodules). Copy the image of Verilog code of the floating point adder here.**

Answer: 

1. **Copy the image of waveform window that is generated for your Testbench?**

Answer: 

1. **From the synthesis report find the Delay and Resource information your floating point adder on Spartan 3E XC3S1600E.**

Answer:

1. **Once design, test and verification are complete call one of the instructors and get your design/output verified.**
2. **List the concepts you learnt from this experiment (Conclusions/Observations)**

Answer: We learnt the implementation of a floating point adder circuit and the methodology of adding and storing floating point numbers  
  
Sumbitted by   
Neil Thanawala 2015A8PS0517G