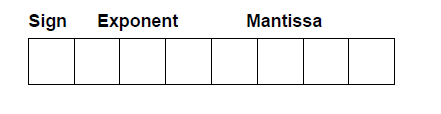
**COEN 313 – Digital System Design II**  **Jafar Abbas  
LAB 4 – A combinational integer to floating-point converter** 26346650  
 Winter 2017 – April 03, 2017

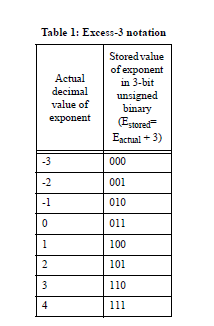
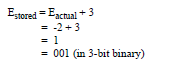
**Objectives**

The purpose of this lab is to become acquainted with using VHDL combinational processes to design combinational logic. Also to become acquainted with some useful command line options to the vcom command and to become better acquainted with the subtle differences between signals and variables.



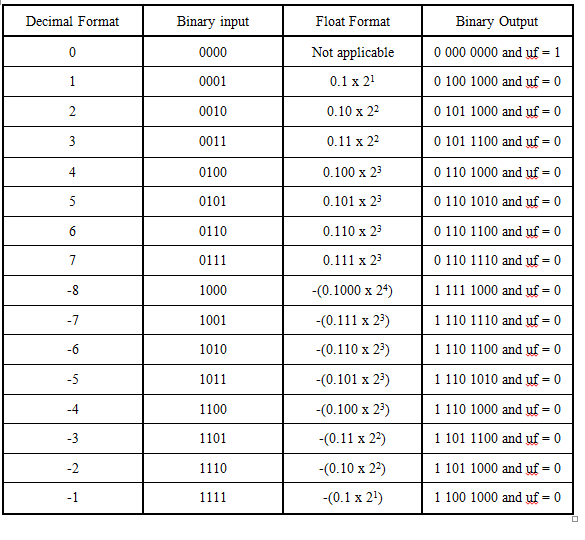
**Procedure**

This lab follows the procedure outlined in Parts I, II, III, and IV of the tutorial “Digital Logic Simulation and Synthesis Using Modelsim, Precision RTL, and Xilinx ISE”. The end-goal is to simulate, synthesize and download to the Xilinx FPGA board the integer to floating-point representation converter (using excess-3 notation) described in the LAB 4 manual.



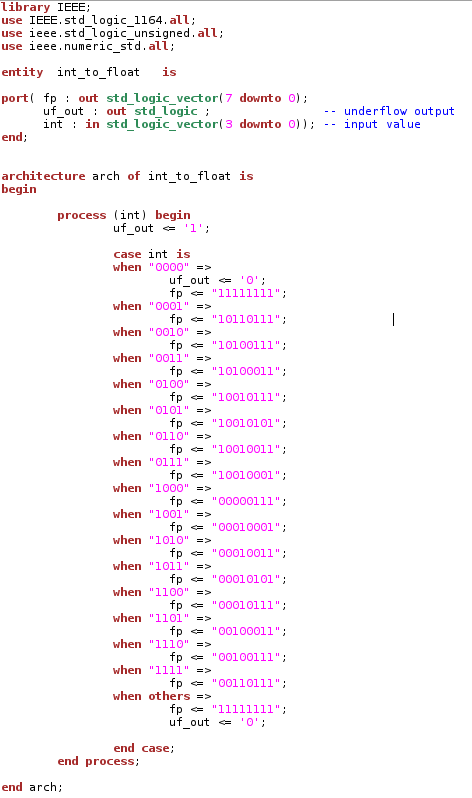
**Requirements**

1. Implement the converter using the brute-force method.
2. Use excess-3 notation for exponent
3. Include precision log file
4. Satisfy the following truth table

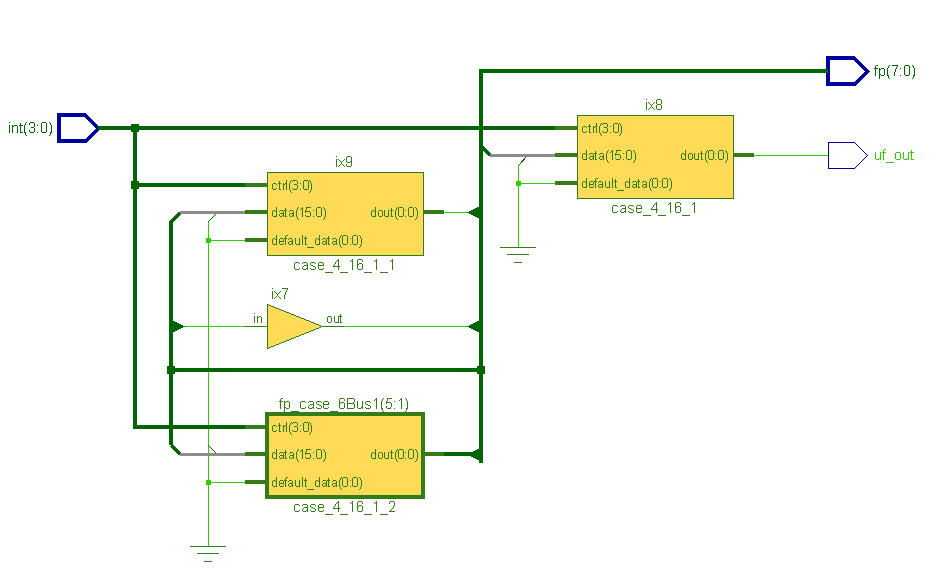
****

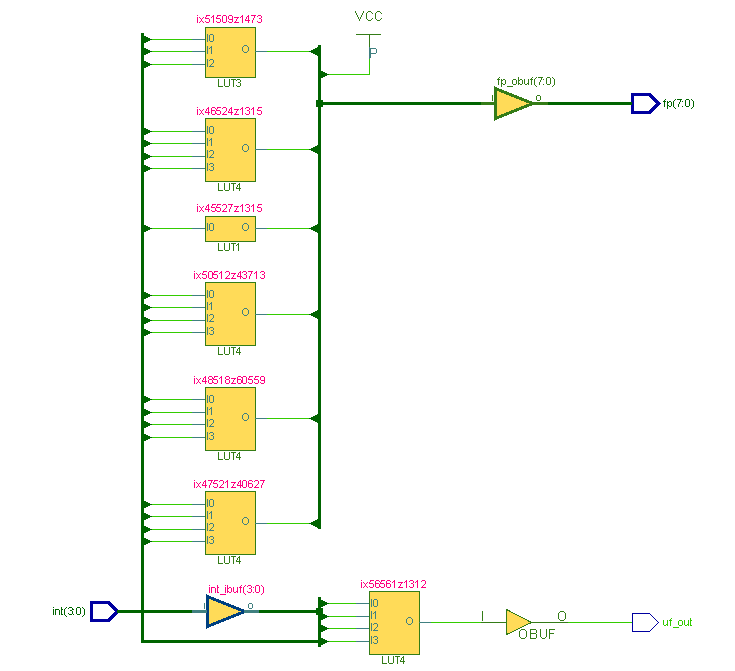
**Results**

*VHDL Code for int\_to\_float*:

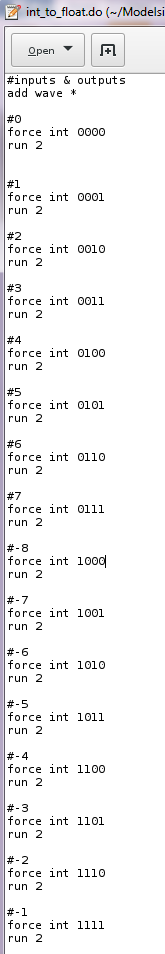
**

*Synthesized circuit:*

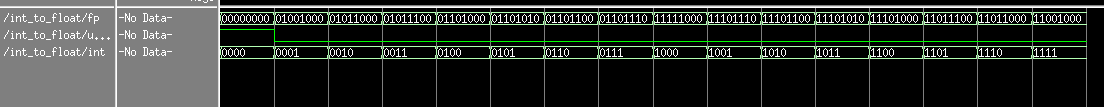


*Technology mapped circuit:*

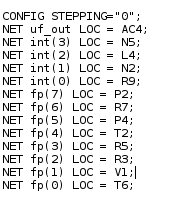
*Do file to simulate my circuit:*

**

*Modelsim Simulation:*

****

*Configuration file (I/O mapping):*

****

*Precision.log file:*

****

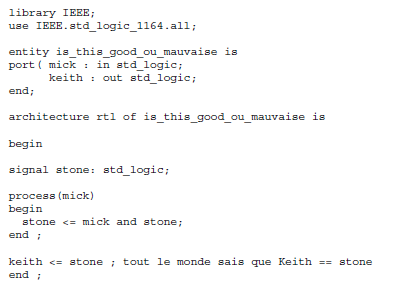
I didn’t get any errors or warnings when I compiled and synthesized my code, probably because it was so simple…

**Questions**

1. Rewrite your VHDL code using a process which counts the number of leading zeros present in the input and based on this number, determines the values of the mantissa and exponent fields of the floating point format accordingly.

We were told not to do this. If we did, we should use a variable to keep track of the number of leading zeros however, because signals only get their values at the end of a process.

1. What will result ( during synthesis) if a signal appears on both sides of the signal assignment operator (<=) within a combinational VHDL process such as:



It is syntactically correct for a signal to be on both sides of a concurrent signal assignment but could cause a closed feedback loop (or latch) which is why it is suggested to avoid this type of syntax even if it would compile. Latches are avoided in sequential circuit design because of undefined behavior.

**Conclusion**

In conclusion, we compiled a VHDL code implementing an in to float converter using combinational logic. It was then compiled and simulated using Modelsim which allowed to confirm its functionality. Having done that an RTL schematic was generated to obtain a physical representation of the entity and using the precision log we were able to identify possible mistakes and errors. Having explored VHDL using combinational logic a suitable follow-up would be synchronous or time-dependant logic circuits.