Birla Institute of Technology and Science Pilani
CS F342 Computer Architecture
Second Semester 2021-22
Evaluative Assignment (22nd Feb. 2022)

Date of Submission: 25th March 2022.
Weightage: 20 Marks
Submission: Online
Group Size: Maximum of Two members per team.

Floating Point Multiplier and Adder/Subtraction

This assignment is about building a floating point ALU which is capable of implementing the floating point multiplication and addition/subtraction operations. The floating point representation used here is the half-precision (16-bit) standard mentioned in IEEE 754. The half-precision notation is exactly similar to that of the single precision standard discussed in the class. The format is shown below. The BIAS is 15. Ignore ths sub-normal numbers in this implementation. The representation of NaN, zero and ±Infinity are similar to that of the IEEE-754 single precision standard discussed in the class.

Sign	**Biased Exponent**	**Mantissa**
1-bit	**5-bits**	**10-bits**

1. Implement a half precision floating point adder unit. Design this as a single combinational circuit. You are free to choose the design styles based on your convenience. Strictly adhere to the following conditions. There is a 2-bit exception flag: 00: Valid output, 01: overflow, 10: underflow, 11: Others
   a. The shifting of the mantissa to align the exponents has to be done using a barrel shifter. Implement the *barrel_shifter* as an individual module.
   b. Include Guard bit and Round Bit, use the Round Up rounding scheme.
   c. Develop a suitable test bench to test the above module. At-least 20 test cases which cover various aspects of the design should be included.

Name of the module: hp_adder
Inputs: [15:0] hp_inA, hp_inB;
Output: [15:0] hp_sum;
Output: [1:0] Exceptions

2. Implement a half precision floating point multiplier unit. Design this as a single combinational circuit. You are free to choose the design styles based on your convenience. Strictly adhere to the following conditions. There is a 2-bit exception flag: 00: Valid output, 01: overflow, 10: underflow, 11: Others

a. The mantissa multiplication has to be implemented using the Radix-4 Booths Algorithm. Implement the radix4_*booth_multiplier* as an individual module.

b. Develop a suitable test bench to test the above module. At-least 20 test cases which cover various aspects of the design should be included.

Name of the module: hp_multiplier
Inputs: [15:0] hp_inA, hp_inB;
Output: [15:0] hp_product;
Output: [1:0] Exceptions

****** The End ******

For clarifications if any, please contact the undersigned
Dr. K. Babu Ravi Teja
Assistant Professor
EEE Department

Appendix-A

Half Precision		Object Represented
Exponent	**Fraction**	
0	0	0 (zero)
0	Non-zero	±Denormalized number
1-30	Anything	±Floating-point number
31	0	± Infinity
31	Non-zero	NaN (Not a Number)