Foundations of Computer Systems Design Lab (CS2310)
Lab 7: Floating Point Adder and Multiplier

October 14, 2024

In this lab, we will learn how to implement IEEE 754 floating point number addition and multiplication

Note: You are free to use any style of coding (not just structural) for this assignment

1 floating point number addittion

Design a circuit which take two 32 bit IEEE 754 floating point number and return their sum in the same format

Instructions: The exception flag should be set in special case of IEEE 754 representations The module signature should be:

```
module FloatingPointAddition (
    input [31:0] a_operand, // Input in IEEE-754 Representation
    input [31:0] b_operand, // Input in IEEE-754 Representation
    output Exception,       // Exception output flag
    output [31:0] result    // Result in IEEE-754 Representation
);
```

2 floating point number multiplication

Design a circuit which take two 32 bit IEEE 754 floating point number and return their product in the same format

Instructions: The exception flag should be set in special case of IEEE 754 representations,overflow and underflow flag should be set based on the conditions.

The module signature should be:

```
module floating_point_mult (
    input [31:0] a,       // First IEEE-754 floating point number
    input [31:0] b,       // Second IEEE-754 floating point number
    output reg [31:0] product, // Product of a and b in IEEE-754 format
    output reg exception,  // Flag for exception (overflow/underflow/Special case)
    output reg overflow,   // Flag for overflow
    output reg underflow    //Flag for underflow
);
```

3 Bonus floating point number addittion and subtraction

Design a circuit which take two 32 bit IEEE 754 floating point number and return their sum or difference in the same format based on the input flag set

```
module Addition_Subtraction(
    input [31:0] a_operand,b_operand, //Inputs
    input AddBar_Sub, //If AddBar_Sub is low then Addition else Subtraction.
    output Exception,
    output [31:0] result //Outputs
    );
```