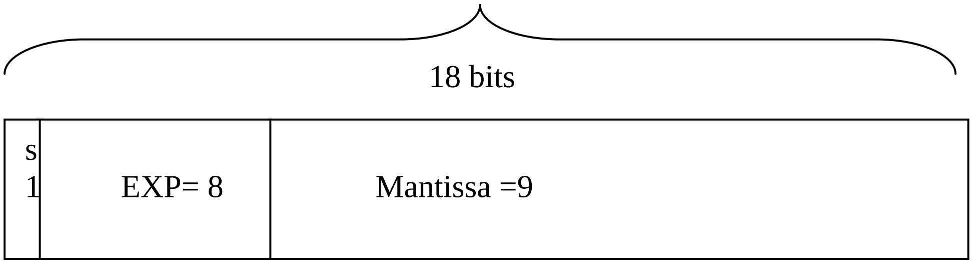
Dept. of Computer Science and Engineering
CSE4210 – Architecture and Hardware for DSP
Lab 2
Floating Point Multipliers

Introduction

This lab will introduce you to floating point numbers. You will be asked to design a floating point multiplier, then you have to reduce (not necessarily minimize) the clock cycle of a circuit that uses the multiplier by using pipelining.

In the lecture, we studies IEEE754 standard for floating point representation. In this lab, we will introduce a simpler format for FP representation that allows you to design a multiplier and reduce the clock cycle time.

In this lab, we use a total of 18 bits to represent a floating point number.



The format of the number is as follows

Sign bit: 1 bit for sign. 0 means positive, 1 means negative

Exponent: 8 bits for exponent, with a bias of 127

Mantissa: 9 bits for the mantissa.

We will be implementing all the features in IEEE754, there is no NaN, infinite, or denom. We assume that the most significant bit of the mantissa is always 1, and the value of the number is represented as

Number = $(-1)^s * 2^{(e-127)} * 0.m$

Before you start in designing your system, please answer these questions.

What is the maximum positive number that can be represented in this format?

What is the minimum positive number that can be represented in this format?

What is the maximum negative number that can be represented in this format?

What is the minimum negative number that can be represented in this format?

Objective:

Design a floating point multiplier, together with an integer to FP converter, and a FP to integer converter.

Int to FP Converter:

This modules converts 16-bit 2’s complement number to the above mentioned FP.

FP to Int Converter

This module converts an FP number to 12-bit 2’s complement format. If the number is too small to fit in the int format, it should be represented as 0 (±). If the number is too big for the int format, it should be represented as the largest number (±) that could be represented.

FP Multiplier

The FP multiplier multiplies 2 FP numbers in the above mentioned format. Use saturation arithmetic to represent numbers that could not fit in the format. The format does not represent infinite or NoN. If the result is greater than the maximum positive number that can be represented, then represent it as the maximum positive number. Similarly for

After completing your design, how long does it takes to complete one FP multiplication?

Use the principle of pipelining to decrease the cycle time of multiplication. How many stages do you suggest? How long is the longest stage? –Note that there is no unique solution for this part, just try to understand pipelining and how to use it to increase throughput