**MEMORANDUM**

**Date:** April 4, 2023

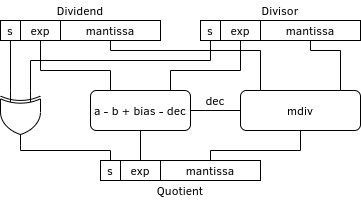
**To**: Dr. James Stine

**From**: Marcus Mellor

**Subject**: ECEN-4233 Weekly Project Update

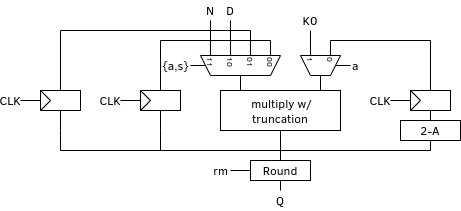
This week I completed an HLS implementation of a floating point divider in SystemVerilog. This module was tested using a very simple testbench which checks a single floating point division operation using binary vectors generated by my Rust code. I also collaborated with Patrick Laverty to develop concepts for implementing a mantissa divider.

By extrapolating from the multiplication diagrams shown in lecture slides, I developed and implemented a floating point divider with the following structure:



Just like radix 10 arithmetic, the exponent is calculated by subtracting the exponents while the magnitude is calculated by dividing. The sign is determined by a simple XOR operation. Unlike radix 10 arithmetic, the exponent has a bias. The magnitude also has a bounding issue, which is solved by checking whether the quotient is less than 1, and if so, multiplying it by 2 (via a left shift) and subtracting 1 from the exponent.

The division operation in my current implementation uses HLS expressions. This will eventually have to be converted to a Goldschmidt divider instead. I collaborated with Patrick to develop a diagram of the Goldshchmidt implementation and some ideas for control. The result is the following diagram:



Patrick and I also discussed a control implementation based around a ring FSM to implement a fixed number of iterations, and the costs and benefits of using two multipliers.