**MEMORANDUM**

**Date:** April 21, 2023

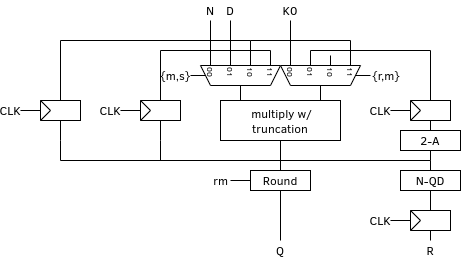
**To**: Dr. James Stine

**From**: Marcus Mellor

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

This week I added hardware to determine the sign of the remainder for rounding. I also implemented a round to nearest even module and a round towards zero module, and modified my testbench to input the rounding mode to the device under test.

The remainder calculation centers around the equation. Last week I used the following architecture to calculate the remainder during the final iteration of Goldschmidt division:



This week, I have adjusted the architecture to a) use a comparator instead of performing a subtraction, and b) perform the comparison after the register instead of every iteration. This means that my Goldschmidt divider no longer outputs the remainder, but only its sign.

The primary motivation for this change is to make sure I fully understand what the synthesizer is doing with my HLS statements. Previously I used a subtraction, but I was not confident that the synthesizer was performing one’s compliment subtraction instead of two’s compliment. This change also reduces the amount of information coming out of my divider, reducing the complexity that I need to consider. Later, I can come back and optimize this design once the overall architecture is functional.

To test my newly implemented rounding modules I used the test vectors provided in the fptests folder. For RNE, my design is correct for 25,659 of the 30,977 test cases. The failure cases all appear to be incorrect by a single ULP, which suggests an uncovered case in my rounding modules. The results for RZ appear to have an identical issue, with a similar failure rate of about 1 in 6.

Overall, the progress I’ve made this week is encouraging, as I’m now very close to a completed, working divider. Next week I plan to identify and fix my rounding issue, then begin implementing square root logic.