WES 237C - Project 1: FIR filter

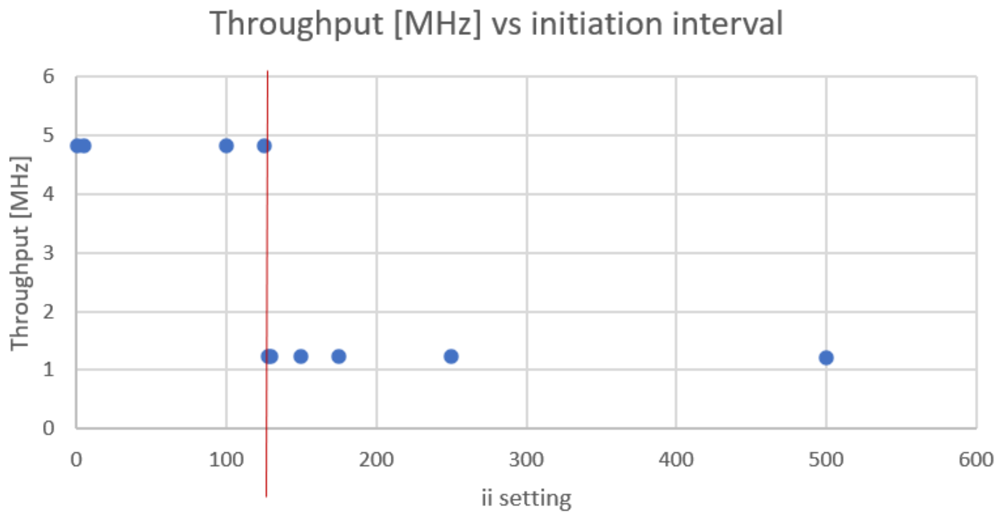
FIR 128

For the 128 FIR filter, our baseline throughput is 1.07 MHz. Let's see if we can improve it with optimization.

Pipelining

By using the #pragma HLS pipeline ii = X setting, we can direct vitis to optimize the IP for initiation interval. For the 128 tap filter, we have very high throughput when we take advantage of the filter's known number of MACs (128 multiply accumulates). When we tell vitis to start the next computation before the last one is finished, we can save a lot of time. It makes sense that once we set the ii to 128 or greater, our throughput takes a big hit because initiating at 129 cycles is basically the same as not optimizing at all in this case. See below on the chart, at the redline for ii = 128, throughput goes from almost 5 MHz down to 1 MHz.

Results:
Throughput w/pipeline @ <128 = 4.81 MHz
Throughput w/pipeline @ >128 = 1.23 MHz or less



Code Hoisting

Since we know that eventually we will get to the 0th entry in our filter, we can remove the if statement asking if i = 0. We put the initial mult-acc after the for loop and thereby hoist the if statement out of the program.

Funnily enough, this only improved latency cycle count from 135 baseline to 134 optimized. Once we add back in pipelining, it goes to 30, and there’s no difference really.

I will keep the code this way though because it is cleaner and more simple.

Results:
Throughput = 1.08 MHz
Throughput w/pipeline = 4.81 MHz