ECE 153A Lab 3A

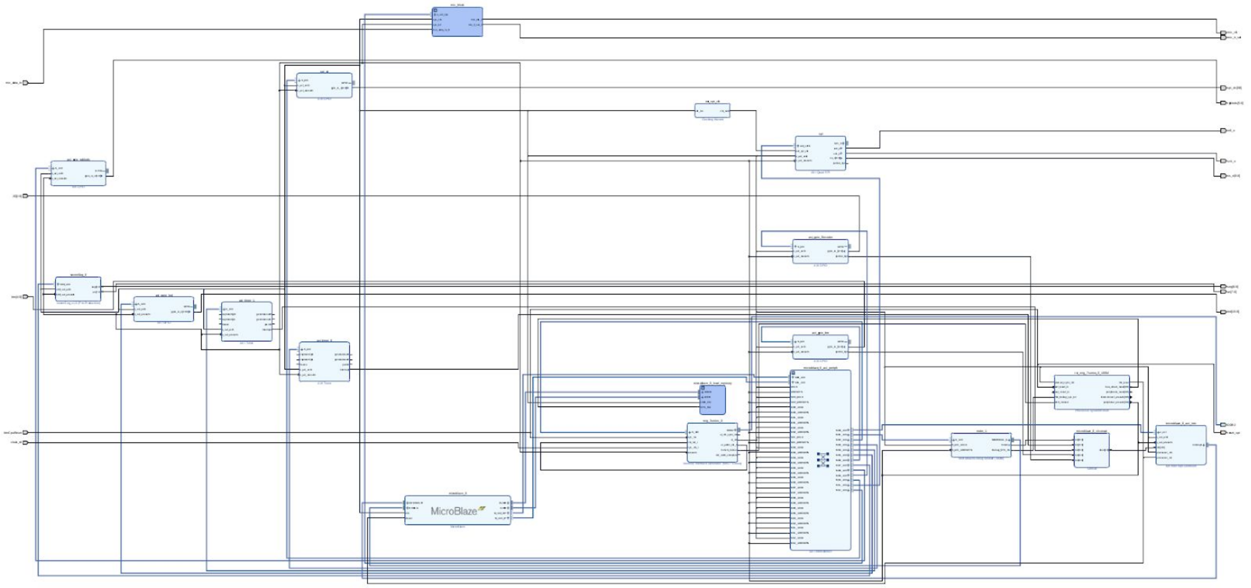
Microphone: Computing FFT

Luke Lewis | Eduardo Olmos

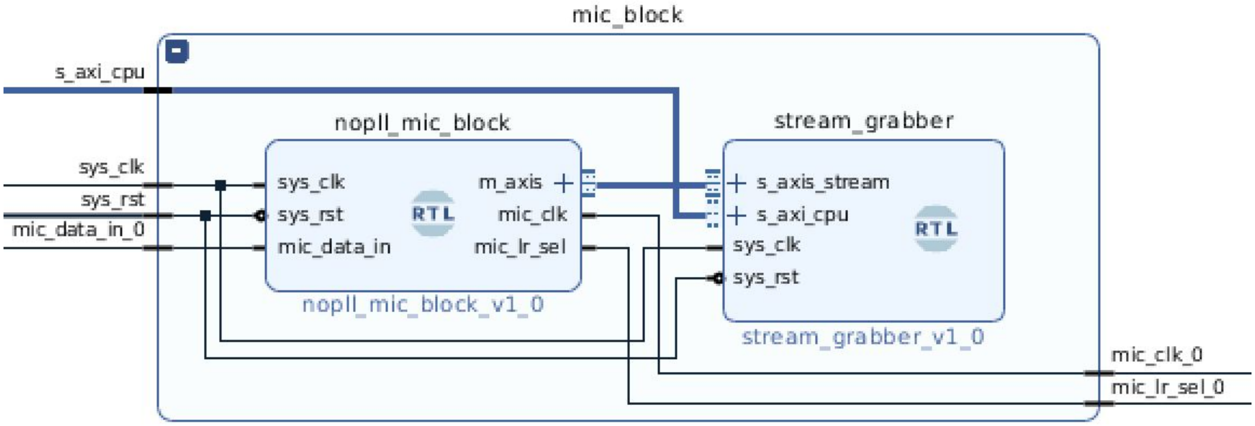
Purpose

In this lab, we were given existing code which utilized the onboard microphone to detect sound, which is then read into a buffer to be analyzed by a written FFT function, outputting the supposed frequency of the input. The purpose of this lab was to increase the resolution of the existing process in order to get an accuracy within 10 Hz, as well as lowering its latency to a goal of < 30 mS by optimizing both the input processing as well as the FFT function itself.

Block Design



Mic_Block



FFT Optimization

The following steps were used to optimize the original FFT code

- Instead of computing the sin and cosine values at run-time using the provided functions, we created a sine and cosine lookup tables at the start of the program and pre-computed these values. When we need a particular sine or cosine value, we simply compute the correct index and access it from our buffer, rather than relying on the computationally-expensive operations of the provided functions.
- At every place where repeated division operations were performed on the same non-changing numbers, we compute it with a single division operation at the beginning to compute the value and we store this result to be used as required in the respective places.
- We also combined two loops that had independent computations.
- We were able to find a good balance between accuracy and timing so we did not have to change our FFT code depending on a high or low frequency.

Timing Statistics

The original run time with the template code was ranging between ~1270-1285 mS. Our code changes improved this latency to 23-27 mS. We profiled the speed of each program execution module with our performance monitor at a resolution of 1ms. As expected, the FFT function took the largest chunk of time, 15-18mS, of our total computation time which ranged from 23-27mS. The rest of the time is mostly going to be allotted to the stream grabbing and buffer filling module.