1 Project Description

1.1 Program 1 - FIR

Write a program to execute a FIR filter. Place the FIR filter in a C file named myfir.c. Filter the provided test signal with an input buffer size of 1 sample. The input test signal includes a sinusoid at frequency 1000 Hz and a second at 4000 Hz at a 48 kHz sample rate for a total time duration of 0.25 seconds.

- Use Matlab to design a FIR low-pass filter to remove the 4000 Hz sinusoid. The filter passband ripple should be less than 0.2 dB and the minimum stop-band attenuation should be 60 dB. Comment on your filter design process, the filter order and include the magnitude response in your report.

- On the DSP, filter the input test signal with your FIR filter and save the result to a .dat file.

- Create a Matlab script to perform an equivalent FIR filter process on the test signal. Import the filter output from the DSP into Matlab for comparison. How closely does the DSP FIR filter output match the Matlab result? (Hint: You can use Mean Squared Error(MSE) as your metric and modify your DSP filter implementation to minimize it e.g. 0.13). Your goal should be a minimal MSE number, I will grade on how good your FIR implementation is. Include a graph of the Matlab filtered result and the DSP filtered result together (**limit the number of samples displayed so the graph looks good**). Also include a graph of the difference signal between Matlab theoretical output and the DSP's output.

- What is the delay of your filter? Does the filter delay match expectations? Explain your answer. How did you measure the delay?

- Count the number of cycles required to filter 1 input sample. Could you use this filter to process blocks of input samples in real time? Explain your answer.

- Record the number of cycles required to filter 1 input sample with C optimization turned on for myfir.c. Report the number of cycles for each level (0, 1, 2, 3). With best optimization could you use this filter to process input samples in real time? Explain your answer.

- Reduce the number of cycles as much as possible. Make use of intrinsics, if you haven't already done so. Place the critical pieces of memory in DARAM. What was the lowest cycle count you could achieve? Can your filter be used in real time? What % of the CPU are you consuming to run your filter?

- Characterize the relationship between number of filter coefficients and number of cycles. What is the max number of filter coefficients where your FIR can be run in real time?

- For your demo, filter real-time audio from Stereo-In and write it to Stereo-Out. You can use your smart phone or computer with head phones to hear the result.

```
void myfir(int16_t*       input,
           const int16_t* filterCoeffs,
           int16_t*       output,
           int16_t*       delayLine,
           uint16_t       numberOfInputSamples,
           uint16_t       numberOfFilterCoeffs);
```

Listing 1: Recommended FIR Prototype

2 Report

Your report and all source code is due at the beginning of class the day the project is due. If you work in groups hand in one report per group. Reports must be written in Latex. Comments in your code must be accurate and meaningful.