ECE4703 B Term 2010 Laboratory Assignment 3

Project Code and Report Due by 3:00pm 18-Nov-2010

The goals of this laboratory assignment are:

- to familiarize you with the fixed-point digital filter design tools in Matlab and
- to familiarize you with fixed-point real-time FIR and IIR filtering on the TMS320C6713.

This assignment extends the work that you did in Lab 2 to redesign your filters to use only fixed-point calculations. Fixed-point processing is used in many applications requiring high speed, low power, and/or low cost. Although your C6713 processor has a built-in floating point ALU, each part of this assignment requires you to develop code that uses no floating point data types or mathematical operations.

Filter Specifications

The filter specifications for this assignment are the same as Lab 2. Prior to the generation of the header files, your coefficients should be quantized to a coefficient word length of 8 bits (including the sign bit), i.e. the data type for all filter coefficients should be signed character. Double check the header files that Matlab generates to be sure that all of the integer filter coefficients fall in the range -128 to +127. You should also confirm that Matlab is giving the best Q-representation that will allow you to represent your filter coefficients with minimum quantization error while also avoiding overflow. Note that the optimum number of fractional bits might be a negative number. This is ok — it just means that you are shifting the decimal point to the right rather than to the left. Be sure to note the number of fractional bits implicit in your fixed-point filter coefficients and comment your code with this information.

Use `fdatool` to plot the impulse response, magnitude response, and phase response of your filter with quantized coefficients. Compare these results to the unquantized coefficients and discuss any differences in your report.

In the remainder of this assignment, you will be implementing your filters in real-time on the TMS320C6713 DSK by creating C6713 projects and writing code to compute the filter output according to the filter type (FIR or IIR) and the filter realization structure with fixed-point processing. In each part of the assignment, you should profile your filters and confirm that they are running in real time. You should also compare your profiling results to Lab 2.

Part 1: Fixed-Point DF-I FIR filtering

Repeat Part 1 of Lab 2 with your 8-bit fixed-point FIR coefficients and 32-bit signed integer intermediate results. You may want to do this in two steps:

1. Use the quantized filter coefficients, but keep all intermediate results in floats. Like Lab 2, include some code to keep track of the largest positive and largest negative valued intermediate results (including the final output) in the filter. Test your code with full-scale sinusoids in the passband and stopband, as well as with white noise. You shouldn't have any overflow when using floating point intermediate results (except perhaps at the output), but this step will help to determine if you will need to perform any scaling to avoid overflow when the intermediate results are fixed-point.

2. Make the code 100% fixed-point by now using signed 32-bit integers for your intermediate results. Remove all floating point data types from your code. How many fractional bits should your 32-bit intermediate results have? Use your results from the previous part to perform any scaling as needed to avoid overflow in all intermediate results and at the output.

The project that you submit for grading should not have any floating point data types. All intermediate results in your final code should be stored as *32-bit signed integers*. Cast the final 32-bit result as a 16-bit signed integer (short datatype) only prior to output by the AIC23 DAC. Use the same procedures as Lab 2 to generate convincing results that your filter is working correctly. The magnitude response of your real-time FIR filter should be almost indistinguishable from the `fdatool` prediction with quantized coefficients. Profile your code to determine the number of cycles needed to complete the ISR. Does your code run in real-time? How does it compare to the floating point code you wrote in Lab 2?

Part 2: Fixed-Point DF-II Single-Section IIR filtering

Repeat Part 2 of Lab 2 with your 8-bit fixed-point DF-II single-section IIR filter coefficients. If your filter is not stable, you can use 16-bit fixed point IIR filter coefficients (make sure you discuss this in your report). In either case, you will need to think about how your fixed-point processing works with feedback and make sure that you are appropriately shifting the results of products to be consistent prior to storing the result. Like Part 1, you may want to get this working first with floating point intermediate results, and then convert to 100% fixed-point after you are sure your IIR filter is working correctly.

Part 3: Fixed-Point DF-II Second-Order-Sections IIR filtering

Repeat Part 3 of Lab 2 with your 8-bit fixed-point DF-II second-order-sections IIR filter coefficients. Your DF-II SOS filter should be stable with 8-bit coefficients.

In Lab

You will be working in the same teams as in Lab 1.

Specific Items to Discuss in Your Report

For each part of the assignment, please provide he following plots:

1. Theoretical magnitude of floating-point filter and fixed-point filter (on same plot).

2. Theoretical magnitude of fixed-point filter and experimental magnitude results of fixed-point filter (on same plot).

3. Experimental magnitude results of floating-point filter from Lab 2 and experimental magnitude results of fixed-point filter from Lab 3 (on same plot).

Use different colors or line styles to distinguish your results (please try to be consistent, however). Your plots should look professional with axis labels, grid lines, and legends as necessary. Your plots should provide immediately convincing evidence to the grader that your fixed-point filter running on the DSK satisfies the requirements and that that it agrees with the magnitude response predictions from Matlab (with quantized coefficients). Be sure to explain any discrepancies.

There is a lot of opportunity for discussion in this assignment about your methods and results. Discuss any scaling considerations that you used to get your filters working correctly and to avoid overflow in the intermediate results and the output. Did you perform any worst-case analysis to confirm that your filters will not overflow? Discuss the differences between single-section and SOS IIR filtering with fixed-point coefficients. Profile the execution time of all of your filters and discuss any differences and/or similarities in your results with the results that you obtained in Lab 1. Discuss any difficulties you had in getting your filters to work correctly, as well as anything else that you found challenging/interesting while completing this assignment.