# ECE4703 B Term 2006 Project 2

Signoff due by 1:50pm on 8-Nov-2006
Report due by 4:00pm on 9-Nov-2006

The goals of this laboratory assignment are:

- to familiarize you with finite impulse response (FIR) filtering on the TMS320C6713,
- to familiarize you with the digital filter design tools in Matlab
- to demonstrate the differences between floating point and fixed point mathematics in FIR filtering, and
- to demonstrate the effects of coefficient quantization in FIR filtering

## 1 Problem Statement

In this assignment, you will write a C program for the TMS320C6713 DSK that implements two digital filters that can be used to separate the "low tones" from the "high tones" in a Dual Tone Multi-Frequency (DTMF) signal. DTMF used by "touch tone" telephones to send control signals to a call switching center, usually for the purpose of dialing a telephone number. For more information about DTMF, see the various resources on the web including wikipedia.com.

The "low tones" in DTMF are at 697Hz, 770Hz, 852Hz, and 941Hz. The "high tones" are at 1209Hz, 1336Hz, 1477Hz, and 1633Hz. Each time a button is pressed on a touch tone phone, one low tone and one high tone are added together and transmitted over the telephone line. The switching equipment at the other end of the line decodes these tones to determine which button you pressed on your phone. Your assignment is to use the Matlab tools to develop and implement a lowpass filter that extracts the low tone from the DTMF signal and sends it to the left output channel and also develop and implement a highpass filter that extracts the high tone from the DTMF signal and send it to the right channel.

Global assignment requirements:

- Your filters must be FIR.
- You must use interrupts for all AIC23 I/O.
- The lowpass filter must provide at least 30dB suppression for all frequencies above 1200Hz and must be within $\pm 0.5$dB for all frequencies below 950Hz.
- The highpass filter must provide at least 30dB suppression for all frequencies below 950Hz and must be within $\pm 0.5$dB for all frequencies above 1200Hz.

## 2 Part 1: Floating Point Design and Implementation

Using the Matlab filter design tools, design lowpass and highpass filters that satisfy the requirements. You can use any Matlab functions you wish but `fdatool` is probably the most convenient. Note that part of the filter design process is choosing a sampling rate. You may want to try out different sampling rates to see if they make any difference in the number of filter coefficients needed to realize your design. Also, recall that a lower sampling rate gives you more time to complete your calculations in real-time. Plot the impulse response magnitude response, and phase response of your filter. Use `fdatool` to generate a C header file with the filter coefficients in single-precision `float` format for use in your TMS320C6713 code.

Once you have your filters designed and have the coefficeint files loaded into your project, you can write a C program to realize your filters in real-time. A suggested outline for your program is given below (you are welcome to any approach that makes sense):

1. Declare variables including a buffer for the DTMF input samples.
2. Initialize the DSK, codec, etc.
3. In the ISR, read in the DTMF tone from the left channel of the AIC23 codec. Because we are reading the channels in stereo, you will also get the right channel too but you don't need to use it.
4. Put the sample into a buffer.
5. Compute the lowpass filter output using the lowpass filter coefficients and the DTMF buffer. See the Kehtarnavaz text for examples if you are sure how to do this.
6. Compute the highpass filter output using the highpass filter coefficients and the DTMF buffer. You can probably save some space and make debugging easier by converting your lowpass filter code to a function and calling it twice, each time with different filter coefficients.
7. Write the lowpass filter output to the codec (left channel) and the highpass filter output to the codec (right channel).
8. Loop back to number 3.

You may also want to buffer the lowpass and highpass filter outputs for troubleshooting and/or visualization. This isn't required, however.

All intermediate results in this part of the assignment should be *single-precision floating point numbers*. Convert the final result to a short only prior to output by the AIC23. Before testing your filter on DTMF tones, carefully plot the magnitude response and compare it to the theoretical predictions from Matlab. Make sure you are getting approximately 0dB gain in the passbands (you may need to fiddle with the AIC23 volume settings to make this work). Profile your code to determine the number of cycles needed to compute the filter outputs. Test your code on DTMF tones (these can be generated in Matlab). You should be able to hear only the low tone in the left channel and only the high tone in the right channel.

# 3 Part 2: 16-bit Fixed Point Design and Implementation

Repeat Part 1 except quantize all filter coefficients to 16-bit signed integers (short) and store all intermediate results in fixed-point format. You are not permitted to use any floating-point math in this part of the assignment. To avoid overflow, the results of all multiplies and additions can be stored in 32-bit signed integers (int) but the final result will need to be converted to a 16-bit signed integer before output by the AIC23. You can use the “set quantization parameters” button in Matlabs `fdatool` to have Matlab generate the filter coefficients for you. Note that the “coefficient word length” setting in `fdatool` should be set to 15 because one bit is required for the sign. Matlab will automatically plot the expected frequency response with quantized filter coefficients. You should not see much difference between this case and Part 1.

Probably the trickiest part of this assignment is making sure you properly scale the signals at each step of the process. Careful scaling is necessary to avoid overflow and/or underflow and to ensure that the AIC23 generates outputs with the correct amplitude. You should discuss how you did this scaling in your report.

As in Part 1, carefully plot the magnitude response of your filter and compare it to the theoretical predictions from Matlab. Profile your code and compare it to the floating point version. Test your code on DTMF tones as well.

# 4 Part 3: 6-bit Fixed Point Design and Implementation

Repeat Part 2 except quantize all filter coefficients to 6-bit signed integers (5 bit coefficient word length) and store all intermediate results in fixed-point format. There is no 6-bit signed datatype on the TMS320C6713 but you can store these quantized coefficients in 16-bit short arrays. The coefficients should all be in the range -32 to +31 (disregarding the decimal point). Matlab’s `fdatool` should show significant changes in the magnitude response of your filter, with respect to Part 1, in this case. Does the filter still meet the specifications?

As in the previous parts, carefully plot the magnitude response of your filter and compare it to the theoretical predictions from Matlab. Test your code on DTMF tones as well.

# 5 Additional Remarks

One problem that has caused some confusion in the past is using input signals that have too much amplitude for the AIC23 codec and end up saturating the ADC. You may want to include a “bypass” option in your code where you check a DIP switch and, if pressed, simply copy the input samples to the output of the AIC23 codec (without any filtering). This will allow you to ensure that your code is at least running and to also see if there is any distortion in the bypassed output waveform, probably resulting from using too large of an input signal. Attenuate the input waveform until the output looks undistorted. Use this setting for all of your testing.

It is important to follow the parts of this assignment in order. The easiest part to get working is the floating point FIR filter (Part 1) since no scaling will be required. Make sure you have this working before moving to the next part of the assignment. You can use most of your code from Part 1 to get the fixed point assignments working but you will need to add some new code to handle the necessary scaling. Pay attention to data types.

If you aren’t sure how to use `fdatool`, type “doc fdatool” at the Matlab command prompt for lots of useful information.

# 6 In Lab

Teams of two are permitted. In the case of an odd number of students in the course, one team of one/three will be formed with permission of the instructor. You will keep these lab partner(s) for all of the laboratory assignments in this course. All lab partners are expected to attend each lab sign-off session and will receive a single grade for the sign-off. You and your lab partner(s) will also submit joint lab reports that receive a single grade.

# 7 Laboratory Report and Grading

Documenting and communicating your work to others is an essential part of the development process and is something that all engineers must master. In ECE4703, the laboratory report is a tool for you to capture the work you did on the assignment and to communicate this work to the Instructor and Teaching Assistant. A good laboratory report will remain useful long after the exercise, even the course, has completed. This section of the assignment lays out the general guidelines for laboratory reports in ECE4703 as well as any specific issues, questions, and topics that should be covered in your report.

## 7.1 General Report Guidelines

All laboratory reports in ECE4703 should follow the following format:

1. One **Cover Page** including the course number, the date, a title, and the report authors.

2. A concise **Introduction** including the problem description as well as goals and specifications. Do not just copy and paste the problem description from the assignment. Use your own words and try to motivate the reader into reading the rest of the report. You may want to even conclude the introduction section with a highlight from your solution and/or results.

3. **Background**. This section should include a concise discussion of any background information (especially theory) that is necessary to understand the methods, solution, and/or results. You can assume that the reader has a junior-level ECE education in the sense that you don't need to explain every little detail, e.g. Fourier analysis. You should, however, highlight any special theory or background material that will help the reader to better understand your work.

4. **Methods**. This section should include information about how you developed and tested your solution to the problem. Any tradeoffs that were considered should be discussed here. Any special techniques that were critical to the solution should be discussed here. This shouldn't be written as a chronological diary of your work but, rather, as a logical justification of how you came to your final solution.

5. **Problem Solution**. This section presents the specifics of your final solution. Your solution should be presented at two levels: first an overview and then the details. The overview is necessary to provide context to the reader. The overview should contain high-level flow charts and/or state diagrams to allow the reader to get an overall understanding of your solution. From this overview, you can then document the details of each specific part of your overall solution. You are encouraged to use diagrams liberally to illustrate details of your design. You don't have to justify your choices here — that was already done in the methods section.

6. **Results**. This is where you present your results as well as answers to any specific questions in the assignment. You are encouraged to try out more than the minimum asked in the lab assignment. Always explain the precise conditions of each test, discuss what the results mean, and provide at least some intuition as to why they make sense. Results without explanation have little value. Where possible, refer to the theory in your background section (or cite a textbook) to justify your results.

7. **Conclusions**. In this section, you should summarize your accomplishments, document any lessons learned and any insight gained, highlight any particular struggles you had in developing the solution, and even suggest directions for future research and/or development. Did you uncover any errors in the problem description? If so what did you do about them? If you were allowed different constraints in the laboratory could you have designed a better, faster, or cheaper system? If so, how?

8. **Code**. Your source code goes here. Good source code listings have numbered lines. Use a smaller font (like 9 or 10 point) for your listings to avoid killing too many trees. Also list all libraries/files that your code links to.

9. **Appendices** (not always required). Appendices are a good place to put things that help to document your work but don't contribute to the overall flow and readability of the report. If you collected a lot of data and it fills a table that takes up 3 pages, it's probably better to just discuss a summary of the results in the Results section and refer to the table in the appendix.

10. **References**. You should document the reference sources you used. That way, if you ever need to find the information again, you'll know where to go. Web references are ok but are less preferable to printed references, e.g. textbook or published papers.

You should use 1.5 line spacing for your report to allow for written comments from the grader. Break large sections into smaller subsections (and even subsubsections) as necessary to improve readability. Your grade is not based on the length of the report; a concise, clearly written report with the key results is much better than a long, wordy, confusing report.

Your report should be of professional quality and, in addition to being a pleasure to read, should look nice. All figures must be numbered, have a descriptive caption, and referenced in the text. All tables must be numbered, have a descriptive caption, and referenced in the text. Good visualizations are important. Sloppy diagrams, plots with missing axis labels, plots with axis dimensions that don't make sense, and plots that fail to show the important features of the results will receive little or no credit.

Don't plagiarize.

## 7.2 Specific Items to Discuss in Your Report

In your report, justify your choice of sampling frequency and provide theoretical and experimental magnitude response results in the same plot (use different colors or line styles). Be sure to explain any discrepancies. Discuss the effects of coefficient quantization on the frequency response of the filter. Discuss any insight gained from the experimental data and any special steps you took to get the fixed-point implementations of your filters working correctly. Discuss any insight obtained from the profiling results.