

Lab 5 – Digital FIR Filters

# Preparation

* [Read the lecture notes from UCCS on FIR filters](http://www.eas.uccs.edu/~mwickert/ece2610/lecture_notes/ece2610_chap5.pdf)
* Go through lectures again
* Download, load into CCS, and run the “EGCP\_450\_Sine\_Table.zip” example
* Make a copy of the “EGCP\_450\_Sine\_Table” example and rename it Lab 5
* [Watch the related video(s) here](https://www.youtube.com/playlist?list=PLxYT_0WQMg35FiE5EDnCwW-1p7-E3gNtJ)

# Purpose

The purpose of this lab is to learn how to design a digital Finite Impulse Response (FIR) filter in Matlab and then to implement it on the MSP432 LaunchPad. Although this is **not** a real-time example, it will give you valuable experience in coding a digital filter and analyzing the output.

# System Requirements

In this lab, you will first design a digital FIR low-pass filter in Matlab. A low-pass filter is a filter that passes signals with a frequency lower than a certain cutoff frequency (*fc*) and attenuates (i.e., reduces the amplitude of) signals with frequencies higher than the cutoff frequency (see Figure 1). The amount of attenuation for each frequency depends on the filter design. The coefficients, which will come from Matlab, for the filter will be used in your embedded program on the board to filter a signal. The signal will be generated programmatically in the embedded program and will be the sum of a 375 kHz sinewave and a 1 MHz sinewave where the 1 MHz sinewave has amplitude twice that of the 375 kHz sinewave (Figure 2). The filter should be designed such that the 1 MHz sinewave is removed from the signal (see Figure 3).

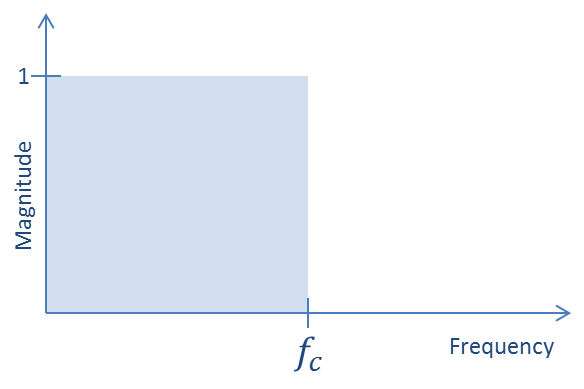


Figure 1: Ideal magnitude response of a digital low-pass filter where *fc* is the cutoff frequency.



Figure 2: A time-domain plot of a 375 kHz sinewave (top), a 1 MHz sinewave (middle), and the summation (bottom).

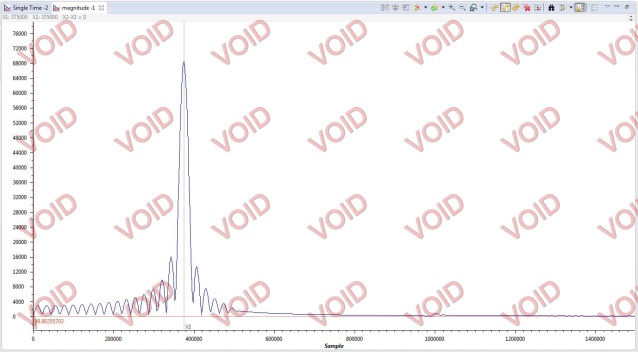
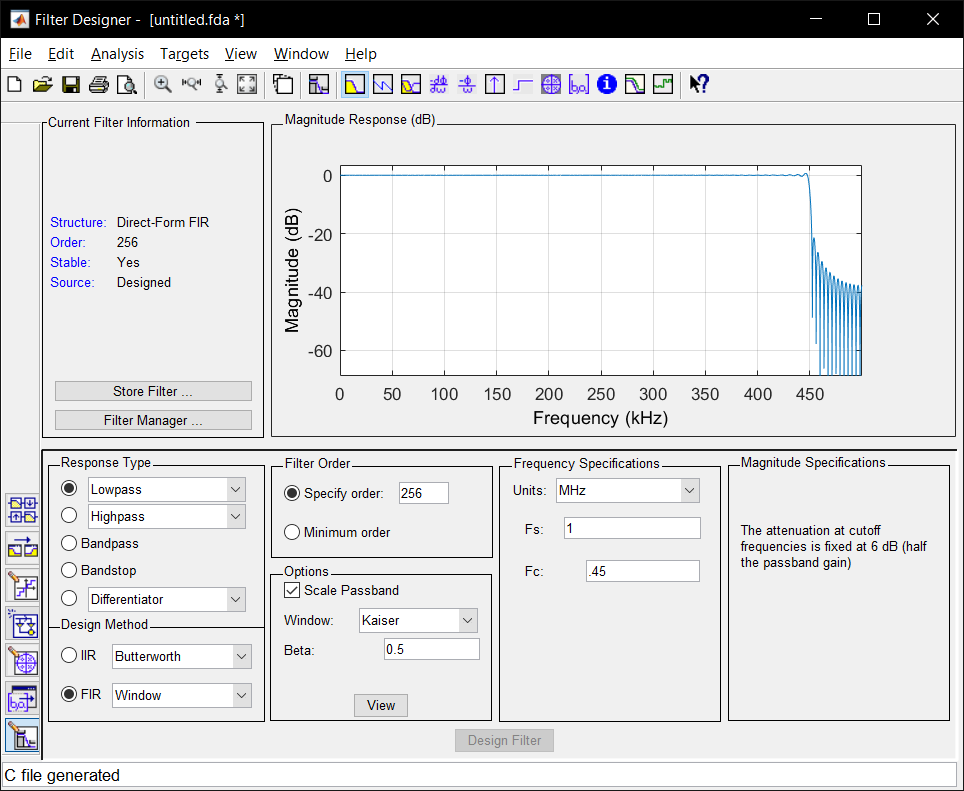
 

Figure 3: Magnitude plots of signal pre-filter (left) and post-filter (right) filtering.



# Procedure

You will do two seemingly disparate activities to design and implement the digital FIR filter on the MSP432 LaunchPad: 1) Matlab activities, and 2) Code Composer Studio (CCS) activities. Matlab will be used to: generate the sine tables containing the samples of the sinusoids, design the filter, and to generate the filter coefficients. The coefficients will then be exported to the proper format and used in the embedded program to filter a signal.

## Part 1: Matlab Activities

1. In Matlab, design a FIR low-pass filter using the window method. Select an appropriate cutoff frequency (*fc*) to attenuate the 1 MHz signal. Put a screenshot of FDATool with all the settings in the “Questions/Deliverables” section, Part 3.a of your report.
2. Generate the samples for the 375 kHz and 1 MHz sinusoids. Export them to a comma Separated Value (CSV) file. I might ask you to change the frequencies of the sinewaves during the demo. So, write a Matlab script where the parameters can be changed easily.

## Part 2: Code Composer Studio (CCS) Activities

1. Export the coefficients from Part 1.a above and put them in the “fir.cof” file of your embedded program.
2. Copy the samples for the CSV from Part 1.b above for each sinusoid into their own table (e.g., sine\_table\_375kHz and sine\_table\_1MHz) in the “main.c” file.
3. Add a FIR filter function (fir\_filt) to “main.c” to do the FIR filtering. There must only be **one** FIR filter function that takes a pointer to a buffer (buffer\_ptr), and the number of samples in the buffer (buffer\_n). **DO NOT** create multiple FIR filter functions. **DO NOT** pass the buffer to the function by value. **DO NOT** return a value back from the function. The buffer must be updated via reference. The declaration of the function **must** be:

void fir\_filt(short\* buffer\_ptr, short buffer\_n);

1. In the main function, first create a local buffer of 200 samples and fill it with the sum of the 375 kHz and 1 MHz sinusoids. However, before summing, add a gain to the 1 MHz sinewave such that it has twice the amplitude of the 375 kHz sinewave.
2. Next, filter the buffer with the fir\_filt function.
3. Lastly, loop forever in the main loop doing nothing.
4. Enter the debugger and place breakpoints on the call to the fir\_filt function and the main loop. Execute your code, and once it stops on the first breakpoint, plot the time and magnitude plots of the buffer. This will give you the pre-filter contents of the buffer. Put screenshots of the plots in the “Questions/Deliverables” section, Part 3.b of your report. Your magnitude plot should look something like Figure 3, left and the time domain plot should look something like Figure 4, left.
5. Now, click “step over” to continue executing to the next breakpoint. Again, once it stops on this breakpoint, plot the time and magnitude plots of the buffer. This will give you the post-filter contents of the buffer. Put screenshots of the plots in the “Questions/Deliverables” section, Part 3.c of your report. Your magnitude plot should look something like Figure 3, right and the time domain plot should look something like Figure 4, right.

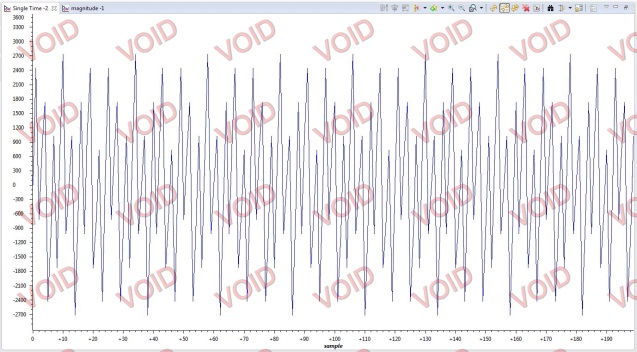
 

Figure 4: Time-domain plots of signal pre-filter (left) and post-filter (right) filtering.

# Demonstration

You will show the instructor your program operation on the MSP432 LaunchPad. The instructor may look at your magnitude and/or time domain plots, and expect you to understand what is being shown. Be prepared to explain what parameters you used to design your filter and why (e.g., cutoff frequency, sampling rate, window used, etc.). Why is the filtered signal delayed? How do you know that you generated the correct samples for a 375 kHz and/or a 1 MHz sinewave in Matlab? I might ask you to change the frequencies of the sinewaves. So, write a Matlab script where the parameters can be changed easily.

# Lab Report

The lab report is how I will grade you on your labs. Usually, the report is due 1 week following the completion of the lab (see TITANium for due dates). However, do to unforeseen circumstances, due dates may change. I will try my best to keep everyone informed of any changes. With this said, it is your responsibility to turn the report in during the scheduled due date. **There is a 10% penalty for late reports**. **Your report must be in MS Word Doc format. Submitting the report in another format will result in a 10% penalty.** Your lab report must include the cover sheet from the lab report template available on TITANium. **Not including the cover sheet will result in a 10% penalty.** The template contains instructions for the report and the rubric used for grading the labs. Please, read it thoroughly. **Don’t forget to include pertinent information such as code, flowcharts, waveform output, etc. in your report**. Please, no “spaghetti” code, keep your code clean and use comments. Remember, your code will affect your lab grade. If I can’t understand it, then I will assume it’s incorrect.