FIR Filtering*

Swaroop Appadwedula
Douglas L. Jones
Daniel Sachs

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 1.0†

Abstract

This module introduces the design and implementation of FIR filters on the 56002.

1 FIR filter implementation

Review the **crucial 4 instructions** that actually implement an FIR filter. Explain, in your own words, what each of these instructions does, and how they together implement an FIR filter. Explain also any necessary initializations for this segment of code.

- `clr a x0,x:(r0)+ y:(r4)+,y0`
- `rep #ntaps-1`
- `mac x0,y0,a x:(r0)+,x0 y:(r4)+,y0`
- `macr x0,y0,a (r0)-`

Implement an FIR filter by modifying the program you will be given. The routine should take in samples from the A/D converter at 16kHz, filter them using an FIR filter, and play the result back through the D/A. Your modifications to this file will include the following three additions:

1. Assembler directives that reserve and initialize memory space for filter coefficients and filter states. For this implementation, use the filter coefficients found in the documentation you have received.
2. The crucial 4 instructions that actually implement the FIR filter.
3. Code to properly initialize the address registers before filtering begins.

Assemble this program, load it into the EVM, and execute the program. By changing the frequency of the sinewave input to the EVM, you can measure the magnitude frequency response of the FIR filter implemented. Do this, making sure that you have a sufficient number of measurements around the regions of interest, such as the band edges. As part of your report for this module, compute the frequency response of this filter (from the coefficients) using Matlab and compare with the measured frequency response. What happens to the frequency response if you change the sampling frequency of the program?

Attempt to induce overflow by rescaling the coefficients of your filter. Can you observe evidence of overflow on the oscilloscope?

*Version 2.9: Feb 25, 2004 12:26 pm -0600
†http://creativecommons.org/licenses/by/1.0

For an FIR filter of N taps, how many instructions does it take to filter one sample? The 56002EVM operates with a clock rate of 24.576MHz, and most instructions take two clock cycles to execute. At a sampling rate of 16kHz, how many instructions can be performed per sample? Can you suggest a better way to organize the program so that unused processor time can be applied to other tasks?

2 FIR filter design

Some speech has been recorded in the presence of a loud, annoying, high frequency hum. Measurements have indicated that the noise energy is isolated to frequencies in the range of 2850Hz to 2950Hz. Your job is to remove as much of this unwanted noise as possible without affecting the speech any more than necessary.

Use Matlab to design an FIR notch filter to remove these unwanted components. Design the filter for a sampling rate of 16kHz. Implement this filter on the 56002 and verify its operation using the oscilloscope. You will have to change the sampling rate of the DSP to 16kHz inside another given file. Once you are confident in its operation, demonstrate for the instructor its operation on some of the recorded speech.