

# CSE 102 – Assignment 5

---

## Introduction

This assignment will mostly exercise your ability to author programs in assembly language, but it is also a preview for the kinds of processing that is commonly used to turn noisy signals that originally come from sensors (e.g., the temperature sensor we used earlier in the semester) into useful signals that tell us information about the physical world being sensed.

You may work in pairs of up to two individuals on this assignment.

## Finite Impulse Response (FIR) Filter

When we filtered the temperature signal, we simply averaged  $N$  raw (unfiltered) input values to give an output (filtered) temperature. This was a super simple version of what is more generally known as a finite impulse response (FIR) filter. Each output  $y_i$  of an FIR filter can be described by the equation

$$y_i = \sum_{k=0}^{N-1} a_k x_{i-k}$$

where  $x_j$  is the  $j$ th input value and  $a_k$  is the  $k$ th coefficient of the filter (the coefficients are constant values provided by the filter designer). The value of  $N$  is referred to as the number of “taps” in the filter (i.e., each summation adds together  $N$  products of a coefficient  $a$  and an input value  $x$  to yield an output value  $y$ ).

Each  $y_i$  is a function of  $x_i, x_{i-1}, x_{i-2}, \dots, x_{i-(N-1)}$  and one needs to deal with the potential for negative indices in an appropriate manner. For this assignment, you need not compute  $y_i$  for  $0 \leq i < N - 1$ , but can start at  $y_{N-1}$  as the first valid output value. (If you wish, you can set  $y_i = 0$  for  $0 \leq i < N - 1$ ). In the results files described below, we start at  $y_{N-1}$ .

Because this filter computation is often the limiting factor on the speed of an application, a designer might legitimately consider writing it in assembly language. We will do just that.

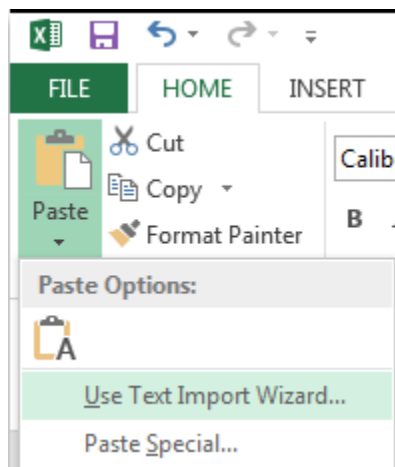
## Assembly Language Implementation

Your task is to author code that computes a 27-tap FIR filter in inline assembly language. The coefficients of the filter are provided in the file [fir1.h](#), two test input sets are provided in the files [input1.h](#) and [input2.h](#), and their respective (filtered) outputs are in the files [Results1.txt](#) and [Results2.txt](#). These output results files are made available for debugging purposes and to enable you to ensure that your code is operating correctly for the two input sets. Feel free to implement the filter in C prior to tackling the problem in assembly language. In your final implementation, only the filter

computation need be in assembly language. The rest of the application (e.g., printing results) can stay in C.

There are a number of challenges that you must surmount to complete this assignment correctly.

1. The input values, coefficients, and output values are all 16-bit integers. That means that the products (in the summation) are 32-bit long integers, and the summation itself needs to be performed using 32-bit arithmetic. One place that describes how to do 16-bit by 16-bit multiply giving a 32-bit result is [here](#). The link doesn't use inline syntax, and you won't be able to use r2 through r5 to store your result, but it should get you started in the right direction.
2. The coefficients  $a$  (in the array `fir1[ ]`), inputs  $x$  (in the arrays `input1[ ]` and `input2[ ]`, which you can rename to `input[ ]` if you wish to make switching between them easier), and outputs  $y$  (you need to define an output array) will need to be accessed appropriately. Use what you learned from [Studio 5](#) to do this.
3. To plot both the input and output signals, you will need to deliver them to the PC and use the plotting package of your choice. If you output ordered pairs (input  $x_i$ , output  $y_i$ ) separated by commas to the PC, you can copy and paste the results into Excel for plotting.



In Excel, you can use the Paste -> Text Import Wizard to select the “,” (comma) as your delimiter.

If a program were to estimate the frequency of the input (raw) and output (filtered) signals by either counting the peaks or counting the zero-crossings, how would the answer be different when transitioning from raw to filtered signal?

## Submitting Your Work

When you have finished your assignment and demonstrated it to the instructor or TA, make sure they record your completion.