

Introduction

In this short application note we present a simple, automated way to export filter coefficients using Filter Design Toolbox®, the popular toolbox extension to Matlab®.

The application note begins with the specification of a simple low-pass FIR Equiripple filter and proceeds to demonstrate how the resulting filter coefficients may be exported to the Matlab workspace. The application note then shows how the coefficients may be written to a text file in a format suitable for use in the Zipcores VHDL filter design package.

Filter Specification

In this example, a simple FIR Equiripple low-pass filter is specified with a normalized frequency. To invoke Filter Design Toolbox, simply type 'fdatool' at the Matlab command prompt. Then enter the parameters in the filter design window as shown in Figure 1 below:

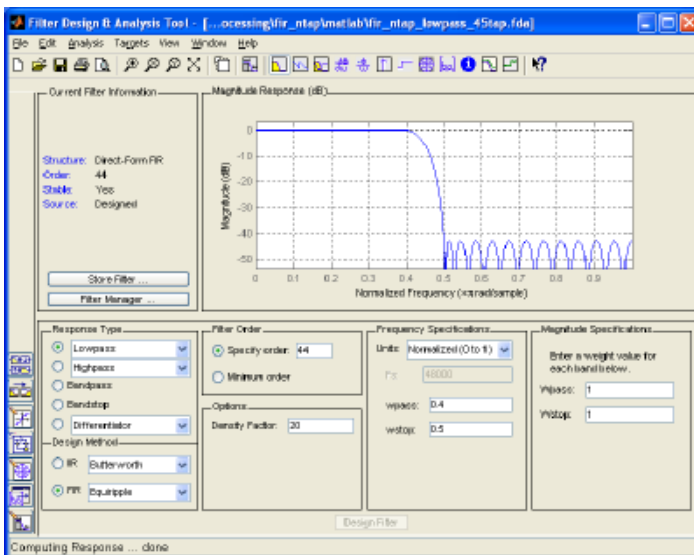


Figure 1: Fdatool Filter design window

The FIR has an order of 44 (or 45 taps) with normalized passband and stopband frequencies set to 0.4 and 0.5 respectively. Once the user is satisfied with the filter specification, the coefficients may then be exported to the Matlab workspace.

Exporting Coefficients

From the top menu bar in Fdatool choose File → Export in order to export the filter coefficients to the Matlab workspace. As the filter in this example is an FIR filter then the transfer function only has Numerator coefficients. By default, the variable name is normally 'Num' for this particular style of filter. Figure 2 shows the export window in more detail.

After export, the next step is to quantize the filter coefficients to match the coefficient width and fraction width of our chosen implementation. In the VHDL package file (e.g. fir_ntap_pack.vhd) the total width of the coefficients is defined as the constant *cw* and the number of fraction bits as the constant *fw*.



Figure 2: Fdatool export window

This example uses values of *cw* = 10 and *fw* = 9 representing coefficients in the range [-1, 0.99805]. However, in the VHDL package file, the coefficients are defined as integers so we need to convert the coefficients to integers rounded to the nearest whole number. This is achieved by multiplying each coefficient by 2^{fw} . A short Matlab script of the following format does just that:

```
% Coefficient width
cw = 10;

% Fraction width
fw = 9;

% Quantize the coefficients
coeffs = round(Num * 2^fw);

% Calculate the DC gain
gain = sum(coeffs)/2^fw;
```

After processing this script in the command window, the variable *coeffs* contains all 45 quantized integer coefficients. Note that it's sometimes useful to work out the gain of the filter as a sanity check to see if the coefficients are correct. For instance, a symmetrical FIR filter will have a gain of unity (or close to unity taking into account quantization error). Filters with inverse symmetry will have a gain of zero.

Writing the Coefficients to a File

Finally, the coefficients may be written to a text file using a script of the following form:

```
% Write coefficients to a file
fid = fopen('coeffs.txt', 'w');

for I = 1:length(Num)
    count = fprintf(fid, '%4d\n', coeffs(I));
end

fclose(fid);
```

It's then a simple matter of cutting and pasting the coefficient values into the VHDL filter design package.

Conclusion

In this application note, a simple method for exporting filter coefficients using Filter Design Toolbox has been presented. The application note explains how to export the coefficients, quantize them and write them out to a text file in a format suitable for the Zipcores VHDL design package.

The application note uses a simple low-pass FIR filter as an example, however, the procedure for generating coefficients is equally relevant for any form of Zipcores digital filter.

An example Matlab script is shipped with all Zipcores filter IP Cores and is included as part of the design package.

Revision History

Revision	Change description	Date
1.0	Initial revision	10/07/2008
1.1	Added note about how to invoke FDAtool from the Matlab command prompt	21/01/2013