

# **FIRDesign**

## **Finite Impulse Response Digital Filter Design Utility User's Manual**

Author : Felix Meier  
Version : 1.0  
Created : February 11, 2007 3:00 PM  
Modified : February 11, 2007 2:16 PM  
File Name : C:\users\meier\VC8\FIRDesign\doc\FIRDesign.doc

## Table of Contents

<b>1.</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Purpose .....	3
<b>2.</b>	<b>Program Use .....</b>	<b>4</b>
2.1	Menu Items.....	4
2.1.1	File	4
2.1.2	Edit	4
2.1.3	View	4
2.1.4	Extras	5
2.1.5	Help	5
2.2	Parameter Input.....	5
2.2.1	Filter Type	5
2.2.2	# of Taps	6
2.2.3	Design Method	6
2.2.4	Window Type	7
2.2.5	Impulse Response Scaling	7
2.2.6	Sampling Rate	7
2.2.7	First Corner	7
2.2.8	Second Corner	7
2.2.9	sin(x)/x Compensation	8
2.2.10	Impulse Response Scale	8
2.2.11	Target Filter	8
2.3	Buttons .....	8
2.3.1	Design	8
2.3.2	Read File	8
2.3.3	Make File	8
2.3.4	Close	8
2.4	Windows.....	9
2.4.1	Magnitude	9
2.4.2	Phase	9
2.4.3	Impulse Response	9
2.4.4	Step Response	9
2.4.5	Parameter Entry Dialogs	9
<b>3.</b>	<b>Number Format Ranges.....</b>	<b>13</b>

## Abbreviations

xxx a placeholder

## References

- [1] Oppenheim, A. R. and Schafer, R. W., Digital Signal Processing, Prentice Hall, 1975
- [2] Rabiner, L. R. and Gold, B., Theory and Application of Digital Signal Processing, Prentice Hall, 1975
- [3] Oppenheim, A. R. and Schafer, R. W., Discrete Time Signal Processing, Prentice Hall, 1989, 1999
- [4] Rorabaugh, C. B., DSP Primer, McGraw-Hill 1998
- [5] Schlichthärle, D., Digital Filters: Basics and Design, Springer 2000
- [6] Lyons, R. G., Understanding Digital Signal Processing, Prentice Hall 1997
- [7] Harris, F., On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proc. IEEE, Vol. 66, No. 1, January 1978, pp. 51 – 83
- [8] Rabiner, L. R., McClellan, J. H. and Parks, T., FIR Digital Filter Design Techniques Using Weighted Chebyshev Approximation, Selected Papers in Digital Signal Processing, II, IEEE Press 1976, reprinted from Proc. IEEE, Vol. 63, April 1975, pp. 595 – 610
- [9] McClellan, J. H., Parks, T. W. and Rabiner, L. R., A Computer Program for Designing Optimum FIR Linear Phase Digital Filters, Selected Papers in Digital Signal Processing, II, IEEE Press 1976, reprinted from IEEE Trans. Audio Electroacoust., Vol. AU21, Dec. 1973, pp. 506 – 526
- [10] McClellan, J. H., Parks, T. W. and Rabiner, L. R., FIR Linear Phase Filter Design Program, Programs for Digital Signal Processing, IEEE Press 1979
- [11] Rabiner, L. R. and McGonegal, C. A., FIR Windowed Filter Design Program – WINDOW, Programs for Digital Signal Processing, IEEE Press 1979
- [12] Williams, C. S., Designing Digital Filters, Prentice-Hall 1986
- [13] Hamming, R. W., Digital Filters, 3<sup>rd</sup> edition, Dover 1998
- [14] Hamming, R. W., Numerical Methods for Scientists and Engineers, 2<sup>nd</sup> edition, Dover 1989
- [15] Gold, B. and Rader, C. M., Digital Processing of Signals, McGraw-Hill 1969
- [16] Lynch, P., The Dolph-Chebyshev Window: A Simple Optimal Filter, Monthly Weather Review, Vol. 125, April 1997, pp. 655 – 660

## Modifications

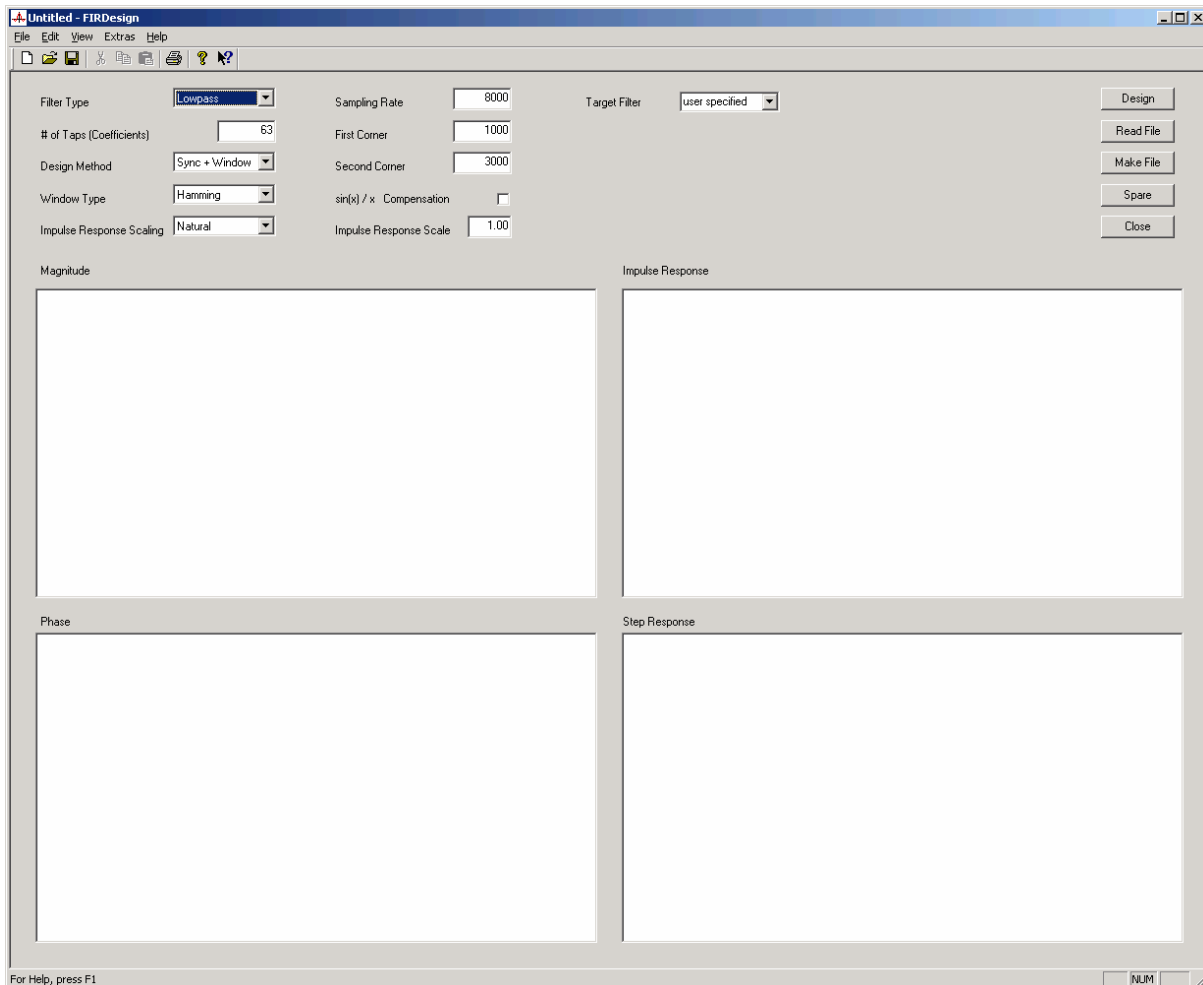
1.0 Feb. 12, 2007 first issue

## 1. Introduction

### 1.1 Purpose

FIRDesign is a tool for the design of Finite Impulse Response filters. It offers three design methods, namely the sinc plus window design, the frequency sampling design and the Remez exchange (Parks – McClellan) design. The program may be used to create source code in C or C++ for the filter coefficient tables to be imported into your DSP application software.

## 2. Program Use



### 2.1 Menu Items

#### 2.1.1 File

The File menu offers the standard commands for any Windows<sup>®</sup> application.

#### 2.1.2 Edit

No items are active on the Edit menu.

#### 2.1.3 View

The View menu offers some standard commands for any Windows<sup>®</sup> application.

**2.1.4 Extras**

The Window command creates a plot of the characteristics the selected window type with the selected number of taps. The impulse response shows the window in the time domain. The step response shows the integral over the impulse response. The magnitude and phase are the results of applying the Fourier transform to the time domain impulse response.

**2.1.5 Help**

Help is currently not implemented.

**2.2 Parameter Input**

Before you can trigger a filter design, you must provide some parameters.

**2.2.1 Filter Type**

The filter type can be any of the following:

	applicable design method			
	window	frequency sampling	Remez exchange	direct
lowpass	x	x	x	o
highpass	x	x	x	o
bandpass	x	x	x	o
bandstop	x	x	x	o
Hilbert transformer	x	o	x	o
differentiator	x	o	x	o
Gaussian edge detector	o	o	o	x
allpass	o	o	o	x
moving average	o	o	o	x

For more information about design methods, see later in this document. Do not confuse filter type (this item) with filter case (1 to 4) as cited in the literature:

taps	impulse response symmetry	case	types
odd	even	1	lowpass highpass bandpass bandstop allpass moving average
even	even	2	lowpass highpass bandpass bandstop allpass
odd	odd	3	moving average Hilbert transformer differentiator
even	even	4	Gaussian edge Hilbert transformer differentiator Gaussian edge

### 2.2.2 # of Taps

This specifies the length of the filter and thus the amount of computation required in an implementation. Note that there are some restrictions on the number of taps depending on the filter case and type:

Highpass or bandstop filters should not be defined with an even number of taps, since the frequency response of case 2 filters is forced to 0 at  $\omega = \pi = f_s/2$ .

If a case 3 filter is chosen for a differentiator or a Hilbert transformer, the upper band limit should be at or below  $\omega = 0.9 * \pi = 0.9 * f_s/2$  for the same reason.

For a Hilbert transformer, the lower band limit should be at or above  $\omega = 0.1 * \pi = 0.1 * f_s/2$  since both case 3 and case 4 filters have a zero frequency response at  $\omega = 0$ .

For more details, see [2], sections 3.36 and 3.37.

### 2.2.3 Design Method

There are four design methods available. For the restrictions on the use of these design methods for the different types of filters, see above.

#### 2.2.3.1 Sync plus Window

The rectangular version of the desired frequency response is used to calculate the impulse response coefficients in the time domain analytically. The frequency response is always band limited to  $f_s/2$  maximum. Thus, the time domain representation is unlimited in time, which, in turn, would produce aliasing when performing the forward Fourier transform, and which would also imply a filter of infinite length. Therefore, the impulse response is multiplied by a window function which limits its extent in time and tapers off smoothly at both ends. A rectangular frequency response transforms into a sinc impulse response in the time domain. The effective frequency response is the Fourier transform of the windowed impulse response. Depending on the type of filter, the time domain impulse response is calculated by means of different formulas. For more details, see the chapter on theory, section on filter types.

#### 2.2.3.2 Frequency Sampling

The frequency response is specified by an array of N samples covering the frequencies from 0 to  $f_s/2$  (inclusive). The program first completes the frequency response to cover the range from 0 to  $f_s$  by mirroring the specification about the sample at  $f_s/2$ . This results in a spectrum of  $(2N - 1)$  samples. Then, the impulse response is calculated by means of the inverse DFT. The DFT size is  $(2N - 2)$ , since the last sample in the spectrum is synonymous with the first one (circular property of the DFT !). Next, the specified window is applied. Finally, the impulse response is mapped to the frequency domain again in order to obtain the resulting frequency and phase responses. Note that the number of taps of the filter is defined by the number of values used for specifying the frequency response in equidistant samples from 0 to  $f_s/2$ . Specifying a different number of taps would require a full blown sampling rate conversion, which is beyond the scope of this program. Because of this handling (doubling), the number of taps in a filter will always be odd.

You may specify an arbitrary shape of the frequency response, such as multiple pass bands etc.

#### 2.2.3.3 Remez Exchange (or Parks – McClellan)

This method yields a filter which is optimum in the minmax sense, i.e. the maximum error relative to the specification is minimized. For the theory of this design, see [8], [9], [10] and section 3.25 of [1].

#### 2.2.3.4 Direct

For these filters, the impulse response can be calculated directly in the time domain.

### 2.2.4 Window Type

This parameter applies to the Sync plus Window and Frequency design methods only. The available window types are:

Rectangular	The simplest but also the most crude window. Provides very poor stopband attenuation.
Hanning	The first sidelobe in the stopband is substantial, but higher sidelobes diminish quickly.
Hamming	A lower first sidelobe in the stopband, but slower decay for further sidelobes.
Blackman	A very low first sidelobe in the stopband and fast decay, at the cost of a slope that is less steep than for the other windows.
Dolph – Chebyshev	Steep slope and almost constant sidelobe level in the stopband. Pretty good filters.
Kaiser	A window that can be parameterized to emulate some of the other filters.

For details about the different windows, see [7] or the chapter on theory.

### 2.2.5 Impulse Response Scaling

The following options are available for the scaling of the impulse response:

#### 2.2.5.1 Natural

The scaling of the impulse response is left as is, i.e. by scaling the inverse Fourier Transform coefficients by  $1/2\pi$ .

#### 2.2.5.2 Normalized

The impulse response is normalized. All samples are re-scaled to make the maximum absolute value 1.0.

#### 2.2.5.3 Optimized

The impulse response is re-scaled to make the maximum value equal to a power of two. This maximizes the resolution that can be obtained when using a fixed point number format.

#### 2.2.5.4 Custom

The scale to be applied to the impulse response can be specified by the user in a separate edit box. This allows for fine tuning to avoid arithmetic overflow in the accumulator.

### 2.2.6 Sampling Rate

The sampling rate (in Hz) can be chosen freely.

### 2.2.7 First Corner

The first corner frequency (in Hz) defines the passband corner frequency for lowpass and highpass filters and the lower band edge for bandpass and bandstop filters as well as for differentiators and Hilbert transformers.

### 2.2.8 Second Corner

The second corner Frequency (in Hz) defines the upper band edge for bandpass and bandstop filters as well as for differentiators and Hilbert transformers. For lowpass and highpass filters, this value is irrelevant.

### 2.2.9 $\sin(x)/x$ Compensation

The distortion that is introduced if the output of a digital filter is converted to an analog signal by means of a sample and hold circuit (DA – converter) can be pre-compensated by adjusting the filter coefficients.

### 2.2.10 Impulse Response Scale

If custom scaling is specified for the impulse response, the scale to be applied is entered via this field. The range is limited to  $0.1 \leq x \leq 10.0$ .

### 2.2.11 Target Filter

Besides the user defined filter, a palette of reference designs can be chosen as examples. These mostly serve to verify the Remez Exchange designs against examples as given in [9].

## 2.3 Buttons

### 2.3.1 Design

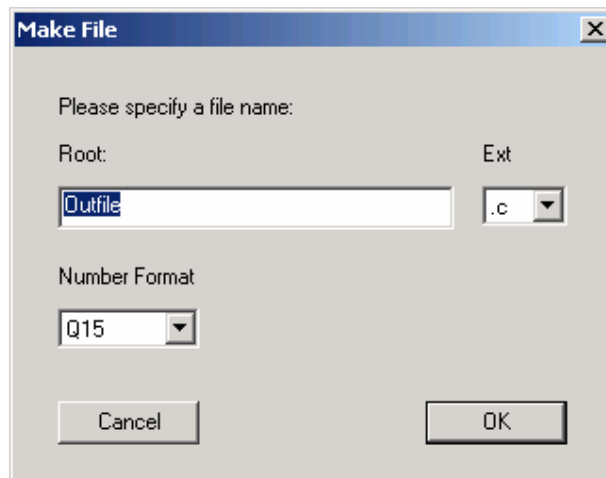
This button triggers the design of the specified filter. Depending on the design method chosen and / or on the window type, some secondary dialogs may appear after starting the design, asking for more input from the user.

### 2.3.2 Read File

This button triggers the reading of a file with frequency response samples for use by the frequency sampling design method. After clicking this button, a file selection dialog appears. The file must be a plain text file with one sample value per line. The number format is fixed point with an arbitrary number of digits to the right of the decimal point, e.g. "0.001".

### 2.3.3 Make File

This button triggers the making of a C or C++ source file with the coefficients  $h(t)$  for the previously designed filter.



When you click this button, a dialog opens, asking for the file name root and extension and for the number format to be used. Once you filled in the data and click OK, the files will be created in the local directory.

### 2.3.4 Close

The purpose of this button is obvious.

## 2.4 Windows

All four windows have a crosshair feature for easy readout of function values at particular points on the curve. If you press the left mouse button with the cursor inside a plot, you will get crosshair lines from the tip of the cursor to the horizontal and vertical scales.

### 2.4.1 Magnitude

This window shows the frequency response magnitude from 0 Hz up to half the sampling frequency. For all filters except for differentiators and Hilbert transformers, the scaling is logarithmic. For differentiators and Hilbert transformers, the scaling is linear.

When "Extras->Window" is selected on the menu, this window shows a DFT transformed version of the time domain impulse response.

### 2.4.2 Phase

This window shows the frequency response phase reduced modulo  $2\pi$ .

### 2.4.3 Impulse Response

This window shows the impulse response.

When "Extras->Window" is selected, this window shows the selected window function in the time domain.

### 2.4.4 Step Response

This window shows the step response.

### 2.4.5 Parameter Entry Dialogs

For some designs, additional parameters must be entered. This is done via special dialogs that pop up when the design is started:

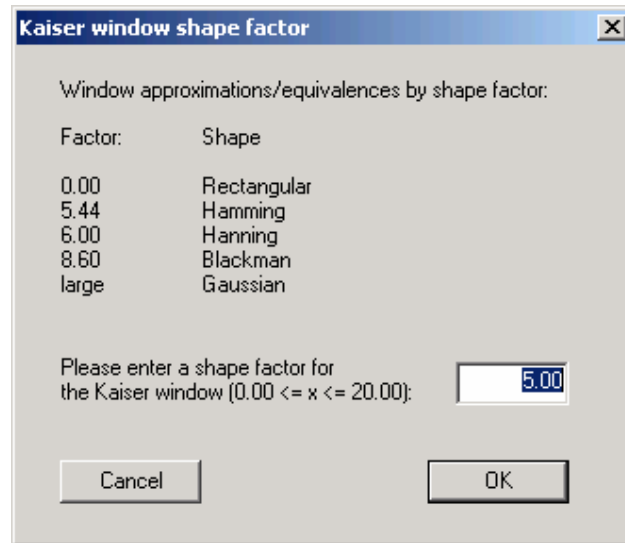
#### 2.4.5.1 Dolph-Chebyshev Stopband Attenuation

For Window designs with the Dolph-Chebyshev window, the stopband attenuation must be specified:



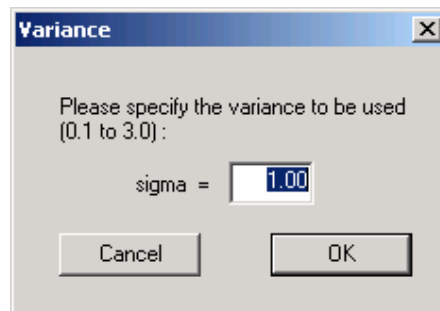
### 2.4.5.2 Kaiser Window Shape Factor

For the Kaiser window, a shape factor must be specified:



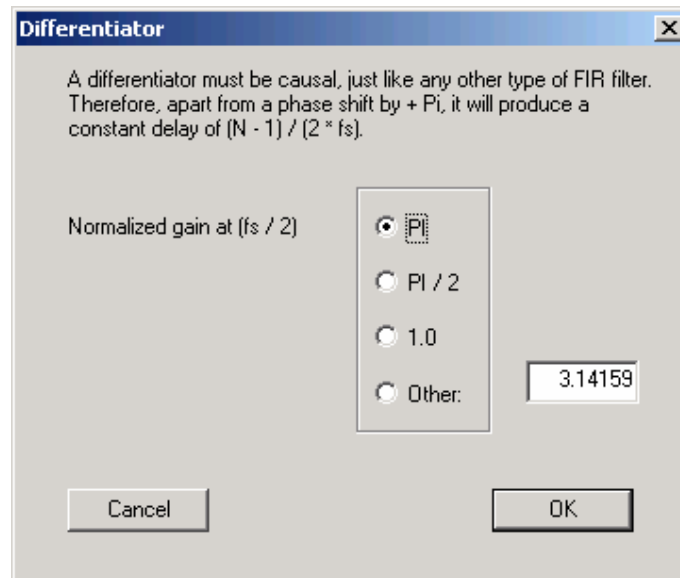
### 2.4.5.3 Gauss Function Variance

For the Gaussian edge detector, the variance of the Gaussian function can be specified:



#### 2.4.5.4 Differentiator Gain

For differentiators, the gain at  $f_s/2$  can be specified. For Window designs, this is done in a special dialog, for Remez exchange designs, this is done in the band specification dialog:



### 2.4.5.5 Remez Design Band Definitions

For Remez Exchange designs, the desired frequency response in at least one band must be specified:

Band	Use	Lower Edge Hz	Upper Edge Hz	Value 0.00 - 1.00	Weight 1.00 - 100.00
1	<input checked="" type="checkbox"/>	0	0	0.00	0.00
2	<input type="checkbox"/>	0	0	0.00	0.00
3	<input type="checkbox"/>	0	0	0.00	0.00
4	<input type="checkbox"/>	0	0	0.00	0.00
5	<input type="checkbox"/>	0	0	0.00	0.00
6	<input type="checkbox"/>	0	0	0.00	0.00
7	<input type="checkbox"/>	0	0	0.00	0.00
8	<input type="checkbox"/>	0	0	0.00	0.00

Grid Density: 16

Buttons: Cancel, OK

Bands need not be continuous, i.e. there may be a gap between the upper edge of the previous band and the lower edge of the next band. This is even advantageous, since it allows for the algorithm to optimize the transition band behavior to the benefit of the performance within the specified bands.

At least one band must be in use. The width of a band must not be zero, and bands must be specified in ascending order of frequency.

The value specifies the desired frequency response magnitude. For differentiators, the value specifies the gain at  $f_s/2$ . In that case, the permitted range is 0.1 to 3.2.

The weight specifies the relative ripple in the band. A 63 tap lowpass filter with a transition band from 1000 Hz to 1200 Hz and a stop band value of 0.0 and weight 1.0 will exhibit a stopband ripple of about  $-34.4$  dB.

### 3. Number Format Ranges

The ranges represented by the various number formats are:

Name	Bytes		Range
Q7	1	signed char	0x80 (-1) to 0x7f (1-1/128)
Q15	2	signed short int	0x8000 (-1) to 0x7fff (1-1/32768)
Q31	4	signed long int	0x80000000L (-1) to 0x7fffffffL (1-1/2'147'483'648)
float	4	none	3.4E +/- 38 (7 digits)
double	8	none	1.7E +/- 308 (15 digits)
long double	10	none	1.2E +/- 4932 (19 digits)