Digital Filter Design How-To (Digital Filter Design Toolkit)

June 2008, 372541A-01

This book contains step-by-step instructions and other information that might be useful as you use the LabVIEW Digital Filter Design Toolkit.

- To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW Help* highlights this topic in the **Contents** tab so you can navigate the related topics.
- © 2008 National Instruments Corporation. All rights reserved.

Designing a Floating-Point Filter (Digital Filter Design Toolkit)

This tutorial introduces you to the basics of <u>designing a floating-point filter</u> by using the LabVIEW Digital Filter Design Toolkit. In this tutorial, you will design a lowpass Butterworth filter. This filter passes frequencies below 100 Hz with a maximum ripple of 0.1 dB and attenuates frequencies above 200 Hz with a minimum attenuation of 60 dB. This filter is suitable for filtering electrocardiograph (ECG) signals.

This tutorial consists of the following three parts:

- <u>Part 1</u>: Entering Floating-Point Filter Specifications
- Part 2: Analyzing a Digital Filter
- Part 3: Using a Digital Filter



Note Refer to the labview\examples\Digital Filter Design\Getting Started\Tutorials\ directory for completed versions of the digital filter you create in this tutorial.

Next: Entering Filter Specifications

Designing a Floating-Point Filter Part 1: Entering Filter Specifications (Digital Filter Design Toolkit)

In Part 1 of this tutorial, you create and <u>specify</u> a floating-point filter design by using the <u>Classical Filter Design</u> Express VI.

Complete the following steps to design a lowpass infinite impulse response (IIR) filter by using this Express VI.

- 1. Launch LabVIEW.
- 2. In the **Getting Started** window, click the **Blank VI** link in the **New** section to create a new, blank VI.
- 3. Display the block diagram by selecting **Window**»Show Block Diagram.
 - **Tip** You can press the <Ctrl-E> keys to switch between the block diagram window and the front panel window of the VI.
- 4. If you do not see the <u>Functions</u> palette, select View»Functions Palette to display this palette.
- 5. On this palette, select **Signal Processing»Digital Filter Design»Filter Design** to display the <u>Filter Design</u> VIs.
- 6. Click the <u>Classical Filter Design</u> Express VI and place it on the block diagram.

■ Place ■ Find

The **Configure Classical Filter Design** dialog box appears.

7. Select **Butterworth** from the **Design method** pull-down menu.

Ensure the filter specifications match the following figure.

Filter type	Lowpass			~
Filter Specifications				
Sampling frequency	1k Hz	۲		
Passband edge frequency	100 Hz	*	400 Hz	A V
Passband ripple	0.1 dB	۲		
Stopband edge frequency	200 Hz	۲	300 Hz	*
Stopband attenuation	60 dB	۲		
Design method	Butterworth			~

- Note The configuration in this example specifies a lowpass Butterworth filter. This filter passes frequencies below 100 Hz with a maximum ripple of 0.1 dB and attenuates frequencies above 200 Hz with a minimum attenuation of 60 dB. This filter is suitable for filtering electrocardiograph (ECG) signals.
- 8. Click the **OK** button to apply the current configuration and close the dialog box.
- 9. Select **File**»**Save As** and save this VI as Designing a Floating-Point Filter.vi in an easily accessible location.

You can experiment with the Classical Filter Design Express VI by choosing other **Design method** options and modifying one or more of the parameters of the filter design. You can enter the exact values for filter parameters by using the controls on the left side of the configuration dialog box or you can use the interactive graphical interface on the upper right side of the dialog box.

To use the graphical interface, click the **Magnitude** graph and drag the passband and stopband lines to modify the location of the passband and stopband frequencies. The corresponding numeric controls and graphs update to show the revised filter design.



Note This tutorial shows one Express VI and one design method. The LabVIEW Digital Filter Design Toolkit includes many other filter design VIs and design methods you can use.

After you enter the filter specifications for a digital filter, you can analyze the characteristics of the filter in <u>Part 2</u> of this tutorial.

 $\overline{\mathbb{N}}$

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Floating-Point Filter\Designing a Floating-Point Filter Part 1.vi for a completed version of the digital filter from this part of the tutorial.

Previous: Designing a Floating-Point Next: Analyzing a Digital Filter Filter Design

Designing a Floating-Point Filter Part 2: Analyzing a Digital Filter (Digital Filter Design Toolkit)

In <u>Part 1</u> of this tutorial, you learned how to create and <u>specify</u> a floatingpoint filter design by using the <u>Classical Filter Design</u> Express VI. In Part 2 of this tutorial, you <u>analyze</u> the characteristics of the filter by evaluating the magnitude response of the filter.

 $\overline{\mathbb{N}}$

Note If you did not complete Part 1 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Floating-Point Filter\Designing a Floating-Point Filter Part 1.vi for a completed version of the digital filter from that part.

Complete the following steps to analyze a filter design by using the <u>Filter</u> <u>Analysis</u> Express VI.

1. Place the Filter Analysis Express VI on the block diagram.

■ Place ■ Find

The **Configure Filter Analysis** dialog box appears with **Sample Data** labels on the graphs. These labels remain on the graph until you run the VI.

- 2. Leave the default settings and click the **OK** button to close the dialog box and return to the block diagram.
- 3. Wire the **filter out** output of the Classical Filter Design Express VI to the **filter in** input of the Filter Analysis Express VI.
- 4. Click the Run button to run the VI.
- 5. Double-click the Filter Analysis Express VI to open the configuration dialog box and view the graphical analysis of the filter.
- 6. Click the **OK** button to close the dialog box and return to the block diagram.
- 7. Right-click the **magnitude** output of the Filter Analysis Express VI and select **Create»Graph Indicator** from the shortcut menu.

A new indicator appears to the right of the Filter Analysis Express VI. This indicator represents a waveform graph in the front panel window. This graph displays data from the **magnitude** output of the Filter Analysis Express VI.

The block diagram now resembles the following figure.



- 8. Double-click the **magnitude** indicator to display the front panel window of the VI and see the graph.
- 9. Click the **Run** button to run the VI. The following figure displays the magnitude graph of the filter design.



Note Graph indicators you create from the Filter Analysis Express VI automatically show two lines on the plot legend to facilitate comparison between floating-point and fixedpoint filters. In this tutorial, only the graph for the floatingpoint filter appears.

In the previous figure, notice the filter passes frequencies below 100 Hz with a 0.1 dB passband ripple and attenuates frequencies

above 200 Hz with a minimum attenuation of 60 dB. This behavior is in accordance with the filter specifications you specified in Part 1.

- (Optional) Create graph indicators for other outputs of the Filter Analysis Express VI, such as the phase delay, group delay, Z Plane, impulse, and step outputs. Then, display the front panel window and run the VI again.
- 11. Select File»Save to save the VI.

Designing a filter is a trial-and-error process. If the filter does not meet the filtering requirements, you can <u>modify the filter specifications</u> and analyze the filter again. After you design an appropriate filter, you can use the filter to process an input signal in <u>Part 3</u> of this tutorial.



Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Floating-Point Filter\Designing a Floating-Point Filter Part 2.vi for a completed version of the digital filter from this part of the tutorial.

Previous: Entering Filter Specifications

Next: Using a Digital Filter

Designing a Floating-Point Filter Part 3: Using a Digital Filter (Digital Filter Design Toolkit)

In <u>Part 2</u> of this tutorial, you learned how to <u>analyze</u> the filter you designed in <u>Part 1</u>. In Part 3 of this tutorial, you use the filter to <u>process</u> an input signal.



Note If you did not complete Part 2 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Floating-Point Filter\Designing a Floating-Point Filter Part 2.vi for a completed version of the digital filter from that part.

Adding and Configuring an Input Signal

Complete the following steps to add and configure an input signal.

- 1. If you do not see the <u>Functions</u> palette, select View»Functions Palette to display this palette.
- 2. On this palette, select **Express**»Input to display the Input Express VIs.
- 3. Click the <u>Simulate Signal</u> Express VI and place it on the block diagram.

The **Configure Simulate Signal** dialog box appears. Make sure **Sine** is selected from the **Signal type** pull-down menu.

- 4. Enter 100 in the **Frequency (Hz)** text box.
- 5. Place a checkmark in the **Add noise** checkbox. Make sure **Uniform White Noise** is selected from the **Noise type** pull-down menu.
- 6. Click the **OK** button to close the dialog box and return to the block diagram.
- 7. On the Functions palette, select **Express**»**Signal Manipulation** to display the <u>Signal Manipulation</u> Express VIs.
- 8. Click the <u>Convert from Dynamic Data</u> Express VI and place it on the block diagram. You use this Express VI to <u>convert</u> the input signal into the 1D array data type.

The **Configure Convert from Dynamic Data** dialog box appears. Make sure **1D array of scalars - automatic** is selected from the **Resulting data type** list.

- 9. Click the **OK** button to close the dialog box and return to the block diagram.
- 10. Wire the **Sine with Uniform Noise** output of the Simulate Signal Express VI to the **Dynamic Data Type** input of the Convert from Dynamic Data Express VI.

Processing an Input Signal

Complete the following steps to process an input signal with the floating-point filter by using the \underline{DFD} Filtering VI.

1. Place the DFD Filtering VI on the block diagram.

Place Find

- 2. Wire the **Array** output of the Convert from Dynamic Data Express VI to the **signal in** input of the DFD Filtering VI.
- 3. Wire the **filter out** output of the <u>Classical Filter Design</u> Express VI to the **filter in** input of the DFD Filtering VI.

The block diagram now resembles the following figure.



- 4. On the Functions palette, select **Programming**»**Array** to display the <u>Array</u> functions.
- 5. Click the <u>Build Array</u> function and place it on the block diagram. The Build Array function concatenates the original input signal and the filtered signal so that you can analyze the effectiveness of the floating-point filter.
- 6. Move the cursor over the Build Array function. When a doubleheaded arrow appears, click and drag the border of the Build

Array function to add one more **element** terminal.

- 7. Wire the **Array** output of the Convert from Dynamic Data Express VI to the first **element** input of the Build Array function.
- 8. Wire the **signal out** output of the DFD Filtering VI to the second **element** input of the Build Array function.
- 9. Display the front panel window. If you do not see the <u>Controls</u> palette, select **View**»**Controls Palette** to display this palette.
- 10. On this palette, select **Modern»Graph** to display the Graph indicators.
- 11. Click the <u>Waveform Graph</u> indicator and add it to the front panel. A Waveform Graph indicator also appears on the block diagram.
- 12. Display the block diagram and move the Waveform Graph indicator to the right of the Build Array function.
- 13. Wire the **appended array** output of the Build Array function to the input of the waveform graph.



The block diagram now resembles the following figure.

14. Display the front panel window and move the cursor over the plot legend on the waveform graph. When a double-headed arrow appears, click and drag the border of the plot legend to add one item to the legend. When you release the mouse button, the second plot name appears.

15. Click the **Run** button to run the VI. The following figure displays the waveform graph of the original input signal and the filtered signal.



Note You can customize the graph by clicking the plot legend and configuring the plot labels, display, and colors.

16. Select File»Save to save the VI.

The infinite impulse response (IIR) floating-point filter you designed in this tutorial is a lowpass Butterworth filter. This filter passes frequencies below 100 Hz with a maximum ripple of 0.1 dB and attenuates frequencies above 200 Hz with a minimum attenuation of 60 dB. This filter is suitable for filtering electrocardiograph (ECG) signals.



Note Refer to labview/examples/Digital Filter Design/Getting Started/Tutorials/Designing a Floating-Point Filter/Designing a Floating-Point Filter Part 3.vi for a completed version of the digital filter from this part of the tutorial.

Previous: Analyzing the Digital Filter Design

Designing a Fixed-Point Filter (Digital Filter Design Toolkit)

This tutorial introduces you to the basics of <u>designing a fixed-point filter</u> by using the LabVIEW Digital Filter Design Toolkit. This tutorial consists of the following six parts:

- Part 1: Selecting a Filter Structure
- <u>Part 2</u>: Scaling Filter Coefficients
- Part 3: Quantizing a Floating-Point Filter
- <u>Part 4</u>: Analyzing a Fixed-Point Filter
- Part 5: Creating a Fixed-Point Filter Model
- <u>Part 6</u>: Simulating a Fixed-Point Filter Model

To design a fixed-point filter, you first must design a floating-point filter, also known as a reference filter. Complete <u>Part 1</u> of the <u>Designing a</u> <u>Floating-Point Filter</u> tutorial to design a floating-point filter.



Note Refer to the labview\examples\Digital Filter Design\Getting Started\Tutorials\ directory for completed versions of the digital filter you create in this tutorial.

Next: Selecting a Filter Structure

Designing a Fixed-Point Filter Part 1: Selecting a Filter Structure (Digital Filter Design Toolkit)

In fixed-point implementations, different <u>filter structures</u> have different memory and multiplier requirements. These requirements can cause different <u>finite word length effects</u>. In this situation, converting the default filter structure of a filter design to a different structure might provide better finite word length performance.

In Part 1 of this tutorial, you select a filter structure for the reference floating-point filter. Later in the design process, you can change the implementation structure if the filter design does not meet the filtering requirements.

Complete the following steps to select a filter structure for the reference floating-point filter by using the <u>DFD Convert Structure</u> VI.

- Design a reference <u>floating-point filter</u>, if you have not already done so. This tutorial uses the lowpass Butterworth filter from <u>Part 1</u> of the <u>Designing a Floating-Point Filter</u> tutorial.
 - **Note** Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Floating-Point Filter\Designing a Floating-Point Filter Part 1.vi for the reference floating-point filter you use in this tutorial.
- 2. Place the DFD Convert Structure VI on the block diagram.

Place Find

- 3. Wire the **filter out** and **error out** outputs of the <u>Classical Filter</u> <u>Design</u> Express VI to the **filter in** and **error in** inputs of the DFD Convert Structure VI.
- 4. Right-click the **target structure** input of the DFD Convert Structure VI and select **Create»Constant** from the shortcut menu.

A <u>block diagram constant</u> appears to the left of the DFD Convert Structure VI. This constant represents the target structure that this VI uses to convert the filter structure.

5. Click the **target structure** constant to select the target structure

this VI uses to convert the filter structure.

Note This tutorial uses the IIR Cascaded Second-Order Sections Form II Transposed structure, which is the default structure for Butterworth filters.

The block diagram now resembles the following figure.



6. Select **File**»**Save As** and save this VI as Designing a Fixed-Point Filter.vi in an easily accessible location.

After you select a filter structure for the reference floating-point filter, you can scale the filter coefficients in <u>Part 2</u> of this tutorial.

N

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 1.vi for a completed version of the digital filter from this tutorial.

Previous: Designing a Fixed-Point Filter Next: Scaling Filter Coefficients

Designing a Fixed-Point Filter Part 2: Scaling Filter Coefficients (Digital Filter Design Toolkit)

In <u>Part 1</u> of this tutorial, you <u>selected a filter structure</u> for the reference floating-point filter. In Part 2 of this tutorial, you <u>scale the coefficients</u> of the filter.

Every filter structure contains accumulators, which might use different data ranges. In this situation, scaling the coefficients of the reference floating-point filter modifies the accumulators to use the same data range and can help you obtain a better filtering result.



Note If you did not complete Part 1 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 1.vi for a completed version of the digital filter from that part.

Complete the following steps to scale the coefficients of the reference floating-point filter by using the <u>DFD Scale Filter</u> VI.

1. Place the DFD Scale Filter VI on the block diagram.

■ Place ■ Find

- 2. Wire the **filter out** and **error out** outputs of the DFD Convert Structure VI to the **filter in** and **error in** inputs of the DFD Scale Filter VI.
- 3. Right-click the **scale type** input of the DFD Scale Filter VI and select **Create**»**Constant** from the shortcut menu.

A block diagram constant appears above the DFD Scale Filter VI. This constant represents the scale type that this VI uses to scale the filter coefficients.

4. Click the **scale type** constant to select the scale type this VI uses to scale the filter coefficients.



Note This tutorial uses **Time Domain 1-Norm**, the default scale type.



The block diagram now resembles the following figure.

5. Select File»Save to save the VI.

After you scale the filter coefficients, you can quantize the coefficients of the reference floating-point filter in <u>Part 3</u> of this tutorial.

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 2.vi for a completed version of the digital filter from this part of the tutorial.

Previous: Selecting a Filter Structure Next: Quantizing a Floating-Point Filter

Designing a Fixed-Point Filter Part 3: Quantizing a Floating-Point Filter (Digital Filter Design Toolkit)

In <u>Part 2</u> of this tutorial, you <u>scaled the coefficients</u> of the reference floating-point filter. In Part 3 of this tutorial, you <u>quantize the coefficients</u> of the filter.

Quantization is the process of approximating a fixed-point value for each reference floating-point value. You then can use the fixed-point values in fixed-point mathematical computation or a hardware implementation. By quantizing the coefficients of the reference floating-point filter, you convert a floating-point filter to a fixed-point filter.



Note If you did not complete Part 2 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 2.vi for a completed version of the digital filter from that part.

Complete the following steps to quantize a floating-point filter by using the $\underline{\text{DFD FXP Quantize Coef}}$ VI.

1. Place the DFD FXP Quantize Coef VI on the block diagram.

■ Place ■ Find

- 2. Wire the **filter out** and **error out** outputs of the DFD Scale Filter VI to the **filter in** and **error in** inputs of the DFD FXP Quantize Coef VI.
- 3. Right-click the **gain settings** input of the DFD FXP Quantize Coef VI and select **Create»Constant** from the shortcut menu.

A block diagram constant appears below the DFD FXP Quantize Coef VI. This constant represents the filter gain settings.

Click the gain settings constant to specify whether this VI processes the filter gain on a host machine or a fixed-point target. If you set gain settings to On Target, you can specify the word length, in number of bits, that represents the filter gain.

Note This tutorial uses On Target and 16 as the gain settings.



The block diagram now resembles the following figure.

5. (Optional) Right-click the **coefficients b/v word length** input of the DFD FXP Quantize Coef VI and select **Create»Constant** from the shortcut menu.

A block diagram constant appears above the DFD FXP Quantize Coef VI. This constant represents the <u>coefficients word length</u>, in number of bits, that this VI uses to quantize the coefficients b/v. The default value for the word length is 16. Later in the design process, you can change the value for the word length if the filter design does not meet the filtering requirements.

6. Select File»Save to save the VI.

After you quantize the floating-point filter, you can analyze the fixed-point filter in $\frac{Part 4}{2}$ of this tutorial.

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 3.vi for a completed version of the digital filter from this part of the tutorial.

Previous: Scaling Filter Coefficients Next: Analyzing a Fixed-Point Filter

Designing a Fixed-Point Filter Part 4: Analyzing a Fixed-Point Filter (Digital Filter Design Toolkit)

In <u>Part 3</u> of this tutorial, you <u>quantized the coefficients</u> of the reference floating-point filter. In Part 4 of this tutorial, you <u>analyze the fixed-point</u> <u>filter</u> to determine how the characteristics of the quantized fixed-point filter deviate from the characteristics of the reference floating-point filter.

Note If you did not complete Part 3 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 3.vi for a completed version of the digital filter from that part.

Complete the following steps to analyze a fixed-point filter by using the <u>Filter Analysis</u> Express VI.

1. Place the Filter Analysis Express VI on the block diagram.

■ Place ■ Find

The **Configure Filter Analysis** dialog box appears with **Sample Data** labels on the graphs. These labels remain on the graph until you run the VI.

- 2. Leave the default settings and click the **OK** button to close the dialog box and return to the block diagram.
- 3. Wire the **filter out** and **error out** outputs of the DFD FXP Quantize Coef VI to the **filter in** and **error in** inputs of the Filter Analysis Express VI.
 - Note If you do not see the error in input to the Filter Analysis Express VI, move the cursor over the down arrows at the bottom of the Filter Analysis Express VI. When a double-headed arrow appears, click and drag the border of the Express VI to add two terminals. When you release the border, the error in input appears.
- 4. Right-click the **magnitude** output of the Filter Analysis Express VI and select **Create»Graph Indicator** from the shortcut menu.

A new indicator appears to the right of the Filter Analysis Express

VI. This indicator represents a waveform graph in the front panel window. This graph displays data from the **magnitude** output of the Filter Analysis Express VI.



The block diagram now resembles the following figure.

- 5. Double-click the **magnitude** indicator to display the front panel window of the VI and see the graph.
- 6. Click the **Run** button to run the VI. The following figure displays the magnitude graph of the filter design.



Note You can customize the graph by clicking the plot legend and configuring the plot labels, display, and colors.

In the previous figure, notice the magnitude response of the quantized fixed-point filter matches the magnitude response of the reference floating-point filter.

- 7. (Optional) Create graph indicators for other outputs of the Filter Analysis Express VI, such as the phase delay, group delay, Z **Plane**, **impulse**, and **step** outputs. Then, display the front panel window and run the VI again. Make sure the filter analysis graphs of the fixed-point filter match the filter analysis graphs of the reference floating-point filter.
- 8. Select File»Save to save the VI.

After you analyze and confirm that the fixed-point filter meets the filtering requirements, you can create a fixed-point filter model in Part 5 of this tutorial. If the filter does not meet the filtering requirements, try one or more of the following steps:

- Redefine the <u>filter specifications</u> of the reference floating-point filter.
- Change the implementation structure of the reference floatingpoint filter.
- Modify the <u>quantization settings</u> of the filter coefficients.

Note Refer to labview/examples/Digital Filter Design/Getting N Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 4.vi for a completed version of the digital filter from this part of the tutorial.

Point Filter

Previous: Quantizing a Floating- Next: Creating a Fixed-Point Filter Model

Designing a Fixed-Point Filter Part 5: Creating a Fixed-Point Filter Model (Digital Filter Design Toolkit)

In Part 3 of this tutorial, you <u>quantized the coefficients</u> of the reference floating-point filter. In Part 4 of this tutorial, you <u>analyzed the fixed-point</u> filter to determine if the filter design meets the filtering requirements. In Part 5 of this tutorial, you <u>create a fixed-point filter model</u> by configuring all other <u>quantizers</u>, such as the input, output, and multiplicand quantizers.

You must create a fixed-point filter model before you simulate the filtering process or generate code from the filter. To use the DFD FXP Modeling VI to create a fixed-point filter model, you must specify the <u>word lengths</u> for the input and output signals and set the <u>rounding mode</u> for the output signal.

Note If you did not complete Part 4 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 4.vi for a completed version of the digital filter from that part.

Complete the following steps to create a fixed-point filter model by using the DFD FXP Modeling VI.

- 1. Remove the Filter Analysis Express VI and the **magnitude** indicator from the block diagram by selecting the icons and pressing the <Delete> key.
- 2. Remove all broken wires by selecting **Edit**»Remove Broken Wires or by pressing the <Ctrl-B> keys.
- 3. Place the DFD FXP Modeling VI on the block diagram.

■ Place ■ Find

4. Wire the **filter out** and **error out** outputs of the DFD FXP Quantize Coef VI to the **filter in** and **error in** inputs of the DFD FXP Modeling VI.



The block diagram now resembles the following figure.

- 5. (Optional) The default word lengths for the input and output signals are 16. To specify different word lengths, right-click the input word length and the output word length inputs of the DFD FXP Modeling VI and select Create»Constant or Create»Control from the shortcut menu. Then, use the constants or controls to specify the value(s) to use for the word lengths.
- 6. (Optional) The default <u>rounding mode</u> for the output signal is **Truncation**. To specify a different output rounding mode option, right-click the **output rounding mode** input of the DFD FXP Modeling VI and select **Create*Constant** or **Create*Control** from the shortcut menu. Then, use the constant or control to specify the rounding option to use.
- 7. Select File»Save to save the VI.

After you create the fixed-point filter model, you can simulate the fixedpoint filter model in <u>Part 6</u> of this tutorial. Later in the design process, you can change the value for the word lengths and rounding mode if the filter design does not meet the filtering requirements.

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 5.vi for a completed version of the digital filter from this part of the tutorial.

Previous: Analyzing a Fixed-Point Next: Simulating a Fixed-Point Filter Filter Model

Designing a Fixed-Point Filter Part 6: Simulating a Fixed-Point Filter Model (Digital Filter Design Toolkit)

In <u>Part 5</u> of this tutorial, you <u>created a fixed-point filter model</u>. In Part 6 of this tutorial, you <u>simulate the fixed-point filter model</u> to verify that this model works as you expect.



Note If you did not complete Part 5 of this tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 5.vi for a completed version of the digital filter from that part.

Adding and Configuring an Input Signal

Complete the following steps to add and configure an input signal.

- 1. If you do not see the <u>Functions</u> palette, select View»Functions Palette to display this palette.
- 2. On this palette, select **Express**»Input to display the Input Express VIs.
- 3. Click the <u>Simulate Signal</u> Express VI and place it on the block diagram.

The **Configure Simulate Signal** dialog box appears. Make sure **Sine** is selected from the **Signal type** pull-down menu.

- 4. Enter 100 in the **Frequency (Hz)** text box.
- 5. Place a checkmark in the **Add noise** checkbox. Make sure **Uniform White Noise** is selected from the **Noise type** pull-down menu.
- 6. Click the **OK** button to close the dialog box and return to the block diagram.
- 7. On the Functions palette, select **Express**»**Signal Manipulation** to display the <u>Signal Manipulation</u> Express VIs.
- 8. Click the <u>Convert from Dynamic Data</u> Express VI and place it on the block diagram. You use this Express VI to <u>convert</u> the input signal into the 1D array data type.

The **Configure Convert from Dynamic Data** dialog box appears. Make sure **1D array of scalars - automatic** is selected from the **Resulting data type** list.

- 9. Click the **OK** button to close the dialog box and return to the block diagram.
- 10. Wire the **Sine with Uniform Noise** output of the Simulate Signal Express VI to the **Dynamic Data Type** input of the Convert from Dynamic Data Express VI.

Simulating a Fixed-Point Filter

Complete the following steps to simulate a fixed-point filter model by using the DFD FXP Simulation VI.

1. Place the DFD FXP Simulation VI on the block diagram.

■ Place ■ Find

- 2. Wire the **filter out** and **error out** outputs of the DFD FXP Modeling VI to the **filter in** and **error in** inputs of the DFD FXP Simulation VI.
- 3. Place the <u>DFD Filtering</u> VI on the block diagram. You use this VI to process the input signal with the reference floating-point filter.

■ Place ■ Find

- 4. Wire the **Array** output of the Convert from Dynamic Data Express VI to the **signal in** input of the DFD Filtering VI.
- 5. Wire the **filter out** output of the <u>Classical Filter Design</u> Express VI to the **filter in** input of the DFD Filtering VI.
- 6. Wire the **Array** output of the Convert from Dynamic Data Express VI to the **signal in** input of the DFD FXP Simulation VI.



The block diagram now resembles the following figure.

7. On the Functions palette, select Programming»Array to display

the Array functions.

- 8. Click the <u>Build Array</u> function and place it on the block diagram. The Build Array function concatenates the original input signal, the reference floating-point filtered signal, and the fixed-point filtered signal so that you can compare the characteristics of the floating-point and the fixed-point filters.
- 9. Move the cursor over the Build Array function. When a doubleheaded arrow appears, click and drag the border of the Build Array function to add two more **element** terminals.
- 10. Wire the **Array** output of the Convert from Dynamic Data Express VI to the first **element** input of the Build Array function. This input is the Sine with Uniform Noise input signal.
- 11. Wire the **signal out** output of the DFD Filtering VI to the second **element** input of the Build Array function. This input is the reference floating-point filtered signal.
- 12. Wire the **DBL signal out** output of the DFD Simulation VI to the third **element** input of the Build Array function. This input is the fixed-point filtered signal.
- 13. Display the front panel window. If you do not see the <u>Controls</u> palette, select **View**»**Controls Palette** to display this palette.
- 14. On this palette, select **Modern»Graph** to display the Graph indicators.
- 15. Click the <u>Waveform Graph</u> indicator and add it to the front panel. A Waveform Graph indicator also appears on the block diagram.
- 16. Display the block diagram and move the Waveform Graph indicator to the right of the Build Array function.
- 17. Wire the **appended array** output of the **Build Array** function to the input of the waveform graph.

The block diagram now resembles the following figure.



- 18. Display the front panel window and move the cursor over the top of the plot legend on the waveform graph. When a doubleheaded arrow appears, click and drag the border of the plot legend to add two items to the legend. When you release the mouse button, three plot names appear.
- 19. Click the **Run** button to run the VI. The following figure displays the waveform graph of the original input signal, the reference floating-point filtered signal, and the fixed-point filtered signal.



Note You can customize the graph by clicking the plot legend and configuring the plot labels, display, and colors.

In this graph, the white line represents the original input signal, which is the Sine with Uniform Noise input signal. The red line represents the reference floating-point filtered signal. The green dots represent the fixed-point filtered signal. Notice the characteristics of the fixed-point filtered signal match the characteristics of the reference floating-point filtered signal. The fixed-point filter works well and is ready to be deployed to a fixed-point hardware target, such as a National Instruments field-programmable gate array (FPGA) target.

20. Select File»Save to save the VI.

Simulation is a trial-and-error process. If the fixed-point filtered signal does not match the reference floating-point filtered signal, try one of more of the following steps:

- Redefine the <u>filter specifications</u> of the reference floating-point filter.
- Change the <u>implementation structure</u> of the reference floatingpoint filter.
- Modify the <u>quantization settings</u> of the filter coefficients.
- Modify the <u>fixed-point filter model</u>.

After you verify that the characteristics of the fixed-point filter model match the characteristics of the reference floating-point filter, you can <u>implement the fixed-point filter</u> on a <u>supported execution target</u>.

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 6.vi for a completed version of the digital filter from this part of the tutorial.

Generating Code from a Fixed-Point Filter (Digital Filter Design Toolkit)

After you <u>design a fixed-point filter</u>, you can implement the fixed-point filter on a <u>supported execution target</u>. You can use the LabVIEW Digital Filter Design Toolkit to generate the following types of code:

- 1. Fixed-point C code
- 2. Integer LabVIEW code
- 3. LabVIEW field-programmable gate array (FPGA) code
- Note Refer to the labview\examples\Digital Filter Design\Getting Started\Tutorials\ directory for completed versions of the digital filter you create in this tutorial.

Fixed-Point C Code (Digital Filter Design Toolkit)

In the <u>Designing a Fixed-Point Filter</u> tutorial, you learned the basics of designing a fixed-point filter using the LabVIEW Digital Filter Design Toolkit. After you design a fixed-point filter, you can generate <u>fixed-point</u> <u>C code</u> and implement the fixed-point filter on a <u>supported execution</u> target, such as a 32-bit microprocessor.



Note If you did not complete the Designing a Fixed-Point Filter tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 6.vi for a completed version of the digital filter from that tutorial.

Generating Fixed-Point C Code

Complete the following steps to generate fixed-point C code from a fixed-point filter by using the <u>DFD FXP Code Generator</u> VI.

1. Place the DFD FXP Code Generator VI on the block diagram.

■ Place ■ Find

- 2. Wire the **filter out** and **error out** outputs of the DFD FXP Modeling VI to the **filter in** and **error in** inputs of the DFD FXP Code Generator VI.
- 3. Click the <u>polymorphic VI selector</u> of the DFD FXP Code Generator VI and select **C Code** from the pull-down menu.
- 4. Right-click the **destination folder** input of this VI and select **Create»Control** from the shortcut menu.

A front panel control appears to the left of this VI. This control represents the destination folder in which this VI generates the code.

5. Right-click the **filter name** input of this VI and select **Create»Control** from the shortcut menu.

A front panel control appears to the left of this VI. This control represents the filter name that this VI uses for the code generation.

The block diagram now resembles the following figure.



- 6. Display the front panel window and click the **Browse** button on the **destination folder** control to select the folder in which this VI generates the code.
- 7. Click the **filter name** control and enter the name for the generated code.
- 8. Click the Run button to run the VI.
- 9. Select **File**»**Save As** and save this VI as Generating C Code.vi in an easily accessible location
- $\overline{\mathbb{N}}$

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Generating Code\Generating C Code.vi for a completed version of the digital filter from this part of the tutorial.

Implementing Fixed-Point C Code on 32-bit Microprocessors

After you generate C code from the fixed-point filter model, you can compile the C code to run on <u>32-bit microprocessors</u>.

The following figure shows example code the DFD FXP Code Generator VI generates.

```
/* ----Demo Program---- */
/* void main()
*
       //Declare and create State
*
     filtername_State state;
 *
     state = filtername_CreateState();
 *
       . . .
*
     //Initialize State
 *
     filtername_InitState (state);
 *
     //Processing
 *
      while ()
 *
       {
 *
         . . .
 *
          y = filtername_Filtering(x, state);
 *
          . . . .
 *
     };
       . . .
    //Dispose State
 *
       filtername DisposeState (state);
 *
       - - -
 *
     return;
* }
*/
```

You can include this code in any C development platform. You also can include this code in the following LabVIEW add-ons that generate C code.

- LabVIEW Embedded Module for Blackfin Processors
- LabVIEW Embedded Module for ARM Microcontrollers
- LabVIEW Microprocessor SDK
- LabVIEW Mobile Module

• LabVIEW Touch Panel Module



Note Refer to the <u>National Instruments Web site</u> for information about these products.

Integer LabVIEW Code (Digital Filter Design Toolkit)

In the <u>Designing a Fixed-Point Filter</u> tutorial, you learned the basics of designing a fixed-point filter using the LabVIEW Digital Filter Design Toolkit. After you design a fixed-point filter, you can generate <u>integer</u> <u>LabVIEW code</u> and implement the fixed-point filter on any platform that LabVIEW supports.



Note If you did not complete the Designing a Fixed-Point Filter tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 6.vi for a completed version of the digital filter from that tutorial.

Generating Integer LabVIEW Code

Complete the following steps to generate integer LabVIEW code from a fixed-point filter by using the <u>DFD FXP Code Generator</u> VI.

1. Place the DFD FXP Code Generator VI on the block diagram.

■ Place ■ Find

- 2. Wire the **filter out** and **error out** outputs of the DFD FXP Modeling VI to the **filter in** and **error in** inputs of the DFD FXP Code Generator VI.
- 3. Click the polymorphic VI selector of the DFD FXP Code Generator VI and select **Integer LabVIEW Code** from the pulldown menu.
- 4. Right-click the **destination folder** input of this VI and select **Create»Control** from the shortcut menu.

A front panel control appears to the left of this VI. This control represents the destination folder in which this VI generates the code.

5. Right-click the **filter name** input of this VI and select **Create»Control** from the shortcut menu.

A front panel control appears to the left of this VI. This control represents the filter name this VI uses for the code generation.

6. Right-click the **Ivproj path** output of this VI and select **Create»Indicator** from the shortcut menu.

A front panel indicator appears to the right of this VI. This indicator represents the path to the generated project file.

The block diagram now resembles the following figure.



- 7. Display the front panel window and click the **Browse** button on the **destination folder** control to select the folder in which this VI generates the code.
- 8. Click the **filter name** control and enter the name for the generated code.
- 9. Click the Run button to run the VI.
- 10. Select **File**»**Save As** and save this VI as Generating Integer LabVIEW Code.vi in an easily accessible location
- $\overline{\mathbb{N}}$

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Generating Code\Generating Integer LabVIEW Code.vi for a completed version of the digital filter from this part of the tutorial.

Implementing Integer LabVIEW Code

After you generate integer LabVIEW code, you can implement the code on any platform or target on which you can run LabVIEW VIs.

The Digital Filter Design Toolkit uses <u>LabVIEW projects</u> to manage the resulting integer LabVIEW code. The following figure displays an example project file that contains integer LabVIEW code.



LabVIEW FPGA Code (Digital Filter Design Toolkit)

In the <u>Designing a Fixed-Point Filter</u> tutorial, you learned the basics of designing a fixed-point filter using the LabVIEW Digital Filter Design Toolkit. After you design a fixed-point filter, you can generate <u>LabVIEW</u> <u>field-programmable gate array (FPGA) code</u> and implement the fixed-point filter on any <u>NI Reconfigurable I/O (RIO) device</u>.



Note To generate LabVIEW FPGA code, you must install the LabVIEW FPGA Module and NI-RIO driver software with R Series support. To execute the FPGA code, you also need an FPGA target on which to run the code.



Note If you did not complete the Designing a Fixed-Point Filter tutorial, refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Designing a Fixed-Point Filter\Designing a Fixed-Point Filter Part 6.vi for a completed version of the digital filter from that tutorial.

Generating LabVIEW FPGA Code

Complete the following steps to generate FPGA code from a fixed-point filter by using the DFD FXP Code Generator VI.

1. Place the DFD FXP Code Generator VI on the block diagram.

■ Place ■ Find

- 2. Wire the **filter out** and **error out** outputs of the DFD FXP Modeling VI to the **filter in** and **error in** inputs of the DFD FXP Code Generator VI.
- 3. Click the polymorphic VI selector of the DFD FXP Code Generator VI and select LabVIEW FPGA Code from the pulldown menu.
- 4. Right-click the **destination folder** input of this VI and select **Create»Control** from the shortcut menu.

A front panel control appears to the left of this VI. This control represents the destination folder in which this VI generates the code.

5. Right-click the **filter name** input of this VI and select **Create»Control** from the shortcut menu.

A front panel control appears to the left of this VI. This control represents the filter name this VI uses for the code generation.

6. Right-click the **Ivproj path** output of this VI and select **Create»Indicator** from the shortcut menu.

A front panel indicator appears to the right of this VI. This indicator represents the path to the generated project file.

The block diagram now resembles the following figure.



- 7. Display the front panel window and click the **Browse** button on the **destination folder** control to select the folder in which you this VI generates the code.
- 8. Click the **filter name** control and enter the name for the generated code.
- 9. Click the Run button to run the VI.
- 10. Select **File**»**Save As** and save this VI as Generating FPGA LabVIEW Code.vi in an easily accessible location
- $\overline{\mathbb{Z}}$

Note Refer to labview\examples\Digital Filter Design\Getting Started\Tutorials\Generating Code\Generating FPGA Code.vi for a completed version of the digital filter from this part of the tutorial.

Implementing LabVIEW FPGA Code

After you generate <u>LabVIEW FPGA code</u>, you can implement the code on any <u>National Instruments FPGA</u> target, such as an NI-RIO device.

The Digital Filter Design Toolkit uses <u>LabVIEW projects</u> to manage the resulting LabVIEW FPGA code. The following figure displays an example project file that contains LabVIEW FPGA code.

