



# DSP Floating Point to Fixed Point Conversion Tutorial

Lattice Semiconductor Corporation  
5555 NE Moore Court  
Hillsboro, OR 97124  
(503) 268-8000

November 2008

---

---

## Copyright

Copyright © 2008 Lattice Semiconductor Corporation.

This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Lattice Semiconductor Corporation.

## Trademarks

Lattice Semiconductor Corporation, L Lattice Semiconductor Corporation (logo), L (stylized), L (design), Lattice (design), LSC, E2CMOS, Extreme Performance, flexiMAC, flexiPCS, FreedomChip, GAL, GDX, Generic Array Logic, HDL Explorer, IPexpress, ISP, ispATE, ispClock, ispDOWNLOAD, ispGAL, ispGDS, ispGDX, ispGDXV, ispGDX2, ispGENERATOR, ispJTAG, ispLEVER, ispLeverCORE, ispLSI, ispMACH, ispPAC, ispTRACY, ispTURBO, ispVIRTUAL MACHINE, ispVM, ispXP, ispXPGA, ispXPLD, LatticeEC, LatticeECP, LatticeECP-DSP, LatticeECP2, LatticeECP2M, LatticeMico8, LatticeMico32, LatticeSC, LatticeSCM, LatticeXP, MACH, MachXO, MACO, ORCA, PAC, PAC-Designer, PAL, Performance Analyst, PURESPEED, Reveal, Silicon Forest, Speedlocked, Speed Locking, SuperBIG, SuperCOOL, SuperFAST, SuperWIDE, sysCLOCK, sysCONFIG, sysDSP, sysHSI, sysI/O, sysMEM, The Simple Machine for Complex Design, TransFR, UltraMOS, and specific product designations are either registered trademarks or trademarks of Lattice Semiconductor Corporation or its subsidiaries in the United States and/or other countries. ISP, Bringing the Best Together, and More of the Best are service marks of Lattice Semiconductor Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

## Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL LATTICE SEMICONDUCTOR CORPORATION (LSC) OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF LSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

LSC may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. LSC makes no commitment to update this documentation. LSC reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. LSC recommends its customers obtain the

---

latest version of the relevant information to establish, before ordering, that the information being relied upon is current.

## Type Conventions Used in This Document

Convention	Meaning or Use
<b>Bold</b>	Items in the user interface that you select or click. Text that you type into the user interface.
<i>&lt;Italic&gt;</i>	Variables in commands, code syntax, and path names.
<b>Ctrl+L</b>	Press the two keys at the same time.
<i>Courier</i>	Code examples. Messages, reports, and prompts from the software.
...	Omitted material in a line of code.
.	Omitted lines in code and report examples.
[ ]	Optional items in syntax descriptions. In bus specifications, the brackets are required.
( )	Grouped items in syntax descriptions.
{ }	Repeatable items in syntax descriptions.
	A choice between items in syntax descriptions.



# Contents

<b>DSP Floating Point to Fixed Point Conversion Tutorial</b>	<b>1</b>
Learning Objectives	1
Time to Complete This Tutorial	1
System Requirements	2
Accessing Online Help	2
About the Tutorial Design	2
Prerequisites	2
Model Examples	2
Task 1: Create the FIR Filter	3
Task 2: Create a Subsystem	6
Task 3: Complete the Testbench	6
Summary	8

---

# DSP Floating Point to Fixed Point Conversion Tutorial

This tutorial covers the basic steps in taking a floating point design in Simulink and converting it to a fixed-point design using the Lattice ispLeverDSP blockset for MATLAB/Simulink.

Inexperienced users of Lattice ispLeverDSP blockset for MATLAB/Simulink are advised to perform the “System Design Using ispLeverDSP” tutorial before attempting this tutorial.

---

## Learning Objectives

---

When you have completed this tutorial, you should be able to do the following:

- ◆ Create a floating-point design in Simulink that achieves the desired system performance criteria.
- ◆ Create a testbench to compare the two versions of the design.
- ◆ Simulate floating-point and fixed-point versions of the design and compare the results.

---

## Time to Complete This Tutorial

---

The time to complete this tutorial is about 30 minutes.

---

## System Requirements

---

One of the following software configurations is required to complete this tutorial:

- ◆ ispLEVER software latest ispLeverDSP MATLAB/Simulink blockset installed.
- ◆ Active license for The MathWorks MATLAB/Simulink software.

---

## Accessing Online Help

---

You can find online help information on any block used in this tutorial by pressing the Help button in the block mask.

To access the ispLeverDSP Help, you need manually copy help files from ispLEVER directory into MATLAB directory as follows:

1. Browse to the `<ispLEVER_install_path>\ispLeverDSP` directory.
2. Highlight and copy the **help** folder.
3. Inside the MATLAB directory, paste the **help** folder on top of the existing help folder.
4. In the Confirm Folder Replace dialog box, choose **Yes to All**.

---

## About the Tutorial Design

---

It is assumed you are familiar with the steps required to produce a floating-point Simulink model. If help is needed with this step, refer to the appropriate Mathworks documentation. For this example, we use the FDA tool to create a Simulink implementation of a 8th order low-pass FIR filter. This will be the reference for creating our Lattice blockset design.

### Prerequisites

Before beginning the tutorial, you must have the Matlab/Simulink software already installed. To install the software, follow the installation instructions that accompany the MATLAB/Simulink software and the ispLeverDSP software.

You should be familiar with the FPGA design process before beginning the tutorial. To learn about the FPGA design process, see the FPGA and Crossover Design section of the ispLEVER online Help.

### Model Examples

A directory of model examples is provided with ispLeverDSP and is available at the following location:

`<ispLEVER_install_path>\ispLeverDSP\examples`

To view the examples for each of the tasks of this tutorial, open the following files in the examples folder:

- ◆ DSP\_tutorial3.mdl
- ◆ DSP\_tutorial4.mdl
- ◆ DSP\_tutorial5.mdl

#### Note

You can backup the files into another directory, so that other users can use them later.

---

## Task 1: Create the FIR Filter

---

1. Start the MATLAB software, using the instructions in the Mathworks documentation.
2. If you haven't already done so, choose **File > Set Path** to add the following ispLEVER paths to the Set Path dialog box.

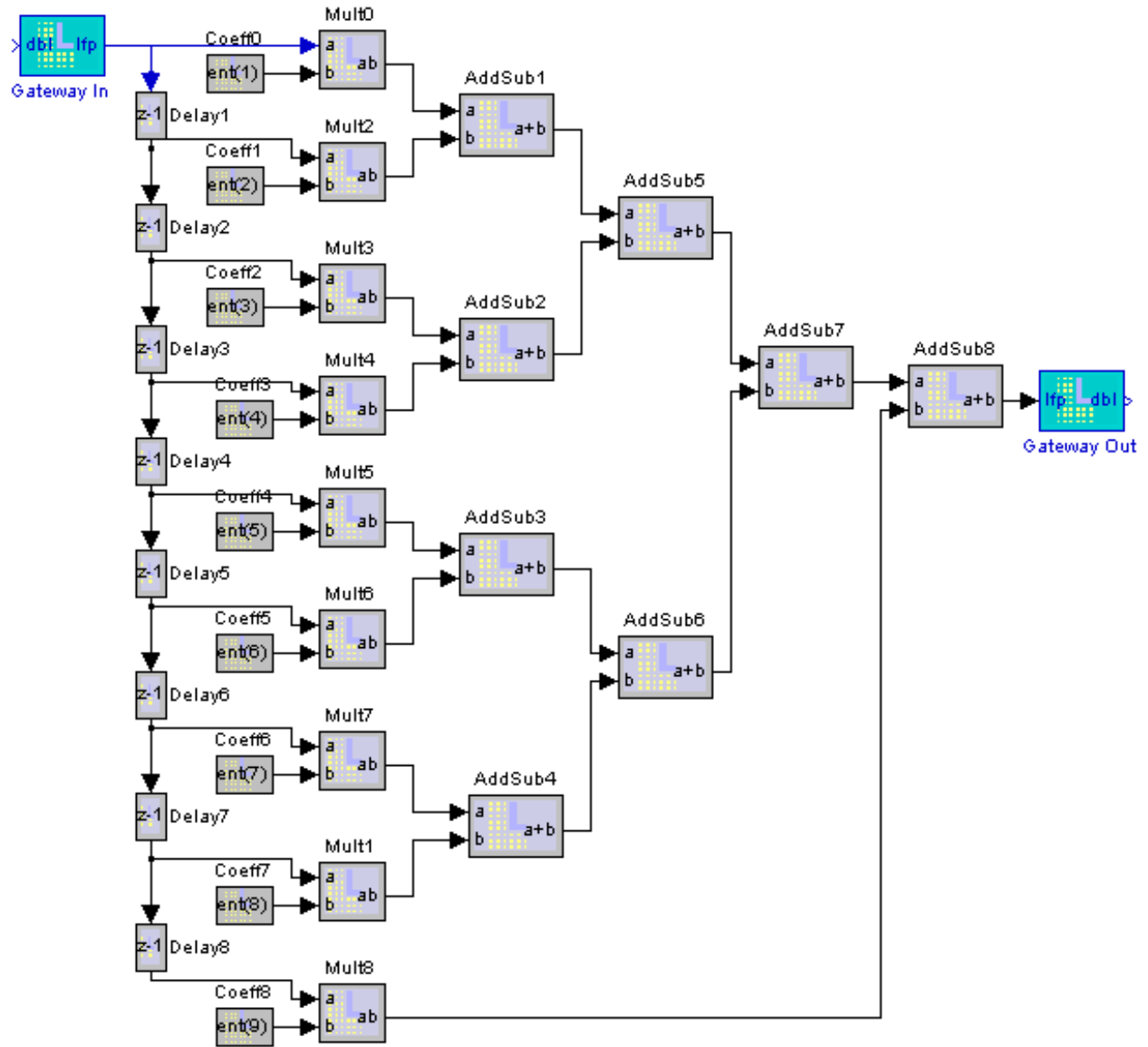
```
<ispLEVER_install_path>\ispLeverDSP  
<ispLEVER_install_path>\ispFPGA\bin\nt
```

3. From the MATLAB startup window, set the Current Directory to your working directory. For example:

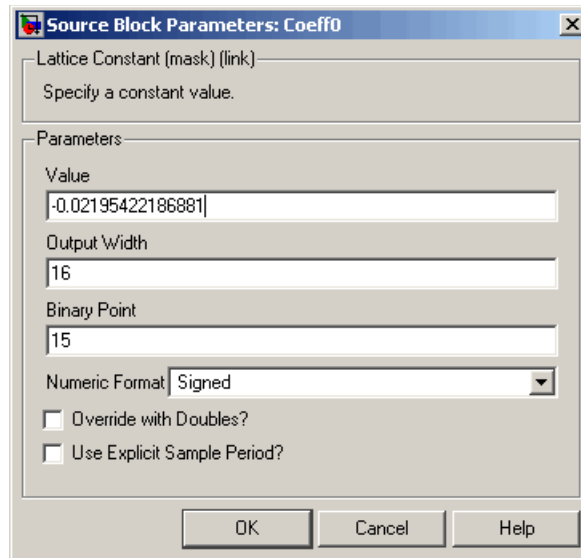
```
<ispLEVER_install_path>\ispLeverDSP\examples
```

4. Choose **File > New > Model** to open a blank model window.
5. Select the Simulink icon from the toolbar, or type **simulink** in the MATLAB Command window. This opens the Simulink Library Browser.
6. In the Simulink Library window, double-click **Lattice Blockset** to open the library and display its contents.

7. Create a FIR Filter design, as shown in the following figure, or open the file DSP\_tutorial3.mdl.



- Each Constant block will be assigned an element from coefficient vector. Double-click the first Constant block, and enter the coefficient  $-0.02195422186881$  into the parameter window for the first constant.



- For the remaining constants, enter the remaining coefficient values as follows:

-0.03832099116987

0.06575727134164

0.28642510883572

0.41152790318546

0.28642510883572

0.06575727134164

-0.03832099116987

-0.02195422186881

It is not necessary to modify any other parameters; the default values will suffice.

## Task 2: Create a Subsystem

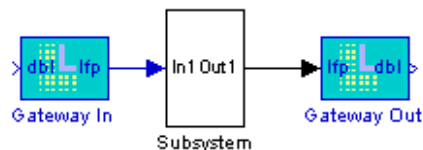
Once this design has been entered, create a subsystem from it.

1. Using your mouse, drag a rectangle around the blocks inside the Gateway-In and Gateway-Out blocks. Do not include the Gateway-In and Gateway-Out blocks in the rectangle you draw. Release the mouse button when all the blocks are selected in the rectangle.
2. Choose **Edit > Create Subsystem**. All the blocks you selected are replaced by a subsystem block. You can undo the subsystem by choosing **Edit > Undo Create Subsystem**.

### Note

If you save the model after creating the subsystem, you will not be able to undo the subsystem later.

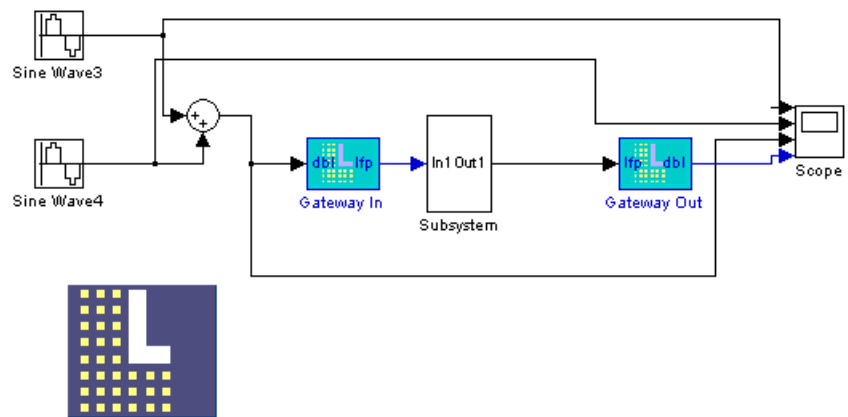
3. You will create a design as shown below, or you can open the file DSP\_tutorial4.mdl.



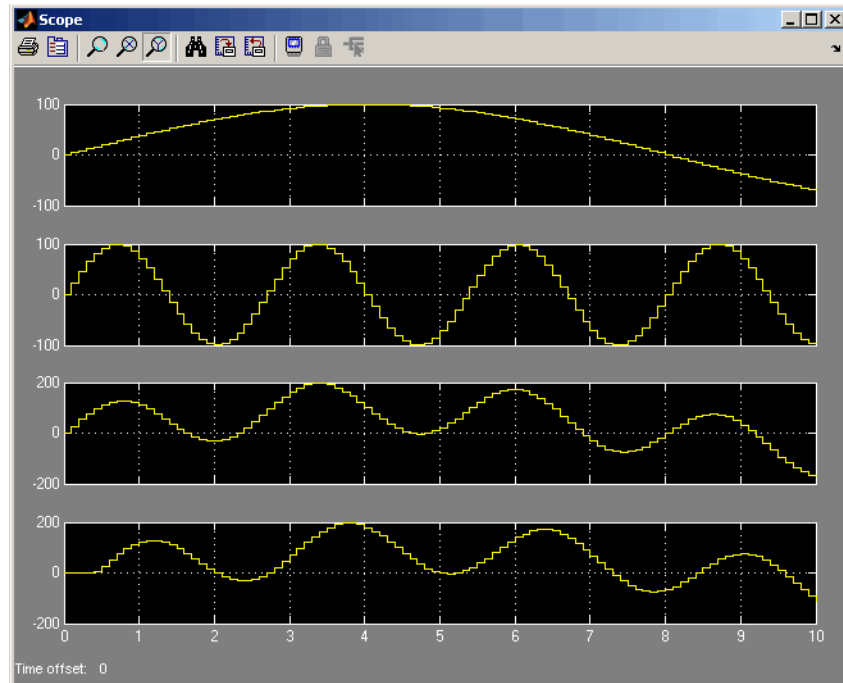
## Task 3: Complete the Testbench

The testbench includes two sine wave sources.

1. Create the model as shown in the following figure, or open the file DSP\_tutorial5.mdl.



2. Double-click on the Lattice Generator block. Select **Override with Doubles** in the Block Parameters dialog box. This directs all the Lattice blocks to operate in floating-point mode.
3. Simulate the design and double-click the Scope to display the results. You should see the following in the display.

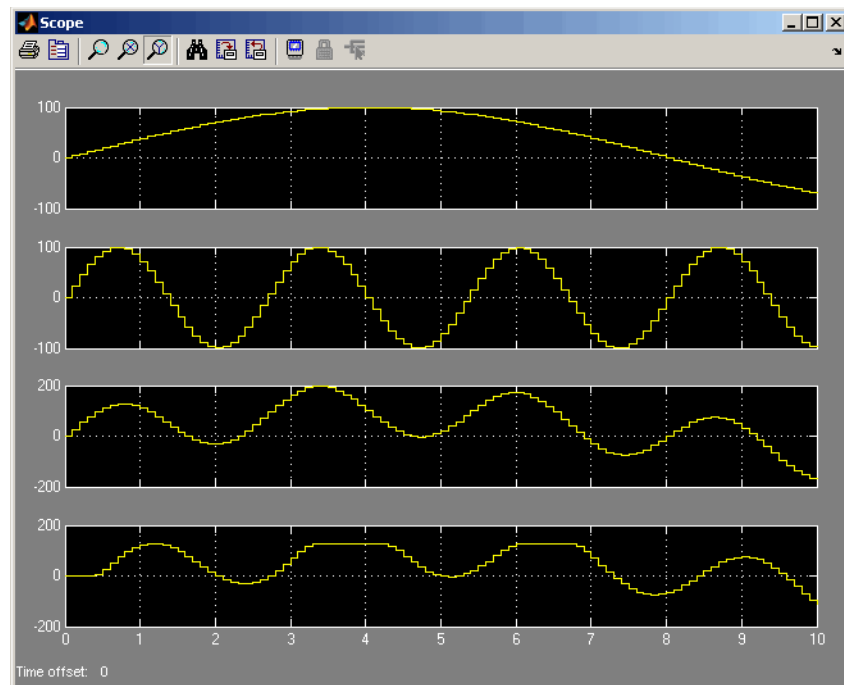


The top two traces are the input signals; the third trace is the combined signal, and the fourth trace is the filter output.

#### Note

If you encounter Block error on the Gateway In block during simulation, double-click the Gateway In block and change Sample Period to a positive number.

4. Double-click on the Lattice Generator block. Clear the **Override with Doubles** option. This directs all the Lattice blocks to operate in fixed-point mode.
5. Simulate the design and double-click the Scope to display the results. You should see the following in the display.



A comparison of the results show that the second scope has waveforms similar to the first scope. The second scope results are fixed-point and are therefore coarser than the initial floating-point results.

---

## Summary

---

You have completed the “DSP Floating Point to Fixed Point Conversion Tutorial.” In this tutorial, you have learned how to do the following:

- ◆ Create a floating-point design in Simulink that achieves the desired system performance criteria.
- ◆ Create a testbench to compare the two versions of the design.
- ◆ Simulate floating-point and fixed-point versions of the design and compare the results.