

# TUTORIAL: QUARTUS FIR COMPILER

## Goal

Create a basic FIR filter using the Quartus II MegaWizard FIR II Compiler. The filter has the following properties:

- Low-pass filter with cut-off at  $\pi/4$
- Signed 24-bit in-/outputs
- Output is the 24 MSb
- Interface is Avalon ST (8-bit symbols, 24-bits/cycle, no back pressure)

## Prerequisites

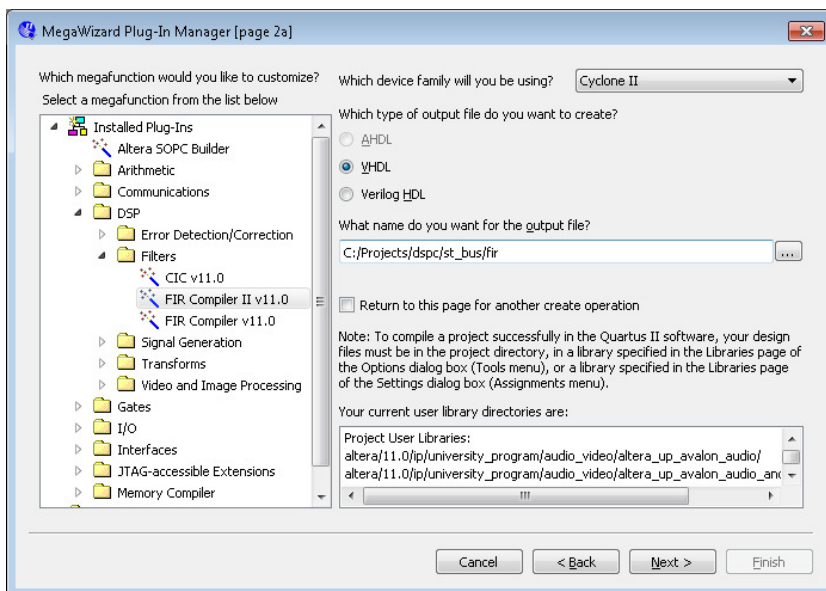
- Quartus with appropriate licenses for the FIR Compiler II

Use the [FIR Compiler II MegaCore Function User Guide](#) as reference for this guide

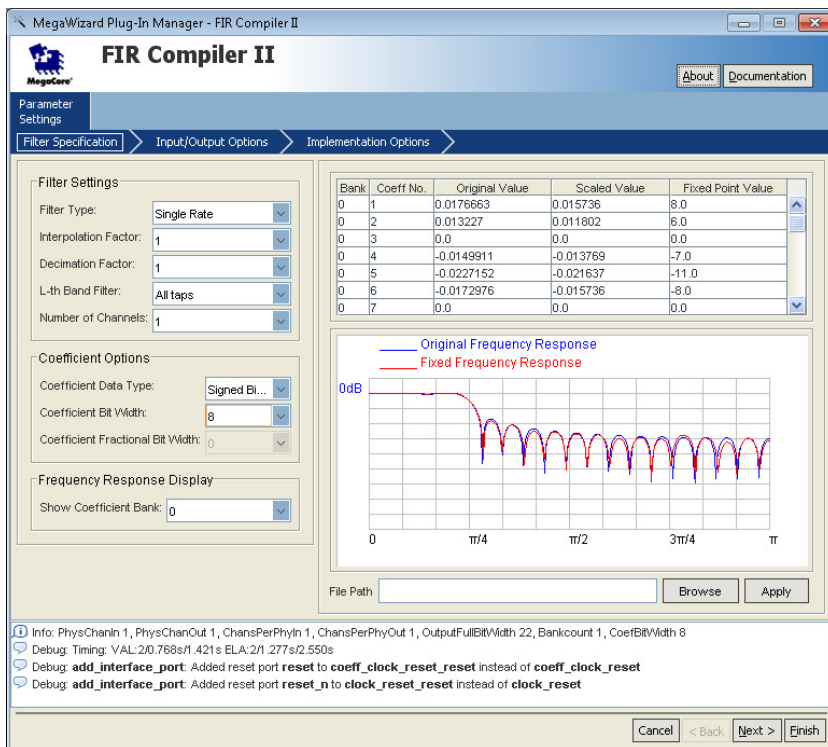
## How To Do It

A) In Quartus II, open the project in which you want to use the filter. When opened, select the Altera Megawizard from the tools menu.

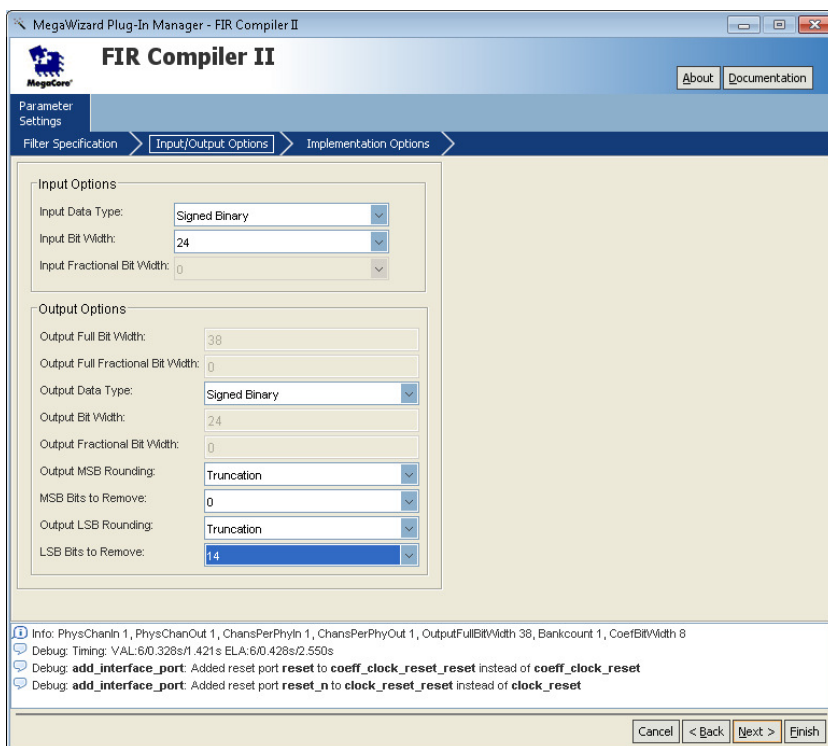
B) Select the "FIR Compiler II" from the DSP subsection. Name your function "fir\_filter".



C) Use the default settings on the first page, note the frequency response, a low-pass function that cuts-off around  $\pi/4$



D) On the next page you have to select the in-/output data widths. Enter a signed 24-bit input data width. For the output, select to remove 14-bits, to get a 24-bit output. The multiplication with the 8-bit coefficients the summation of the many taps is the reason for this. Removing only LSBs ensures that no overflow will occur, but it also removes resolution. This will be significant with low-level input signals.



E) Set the clock frequency to 50 MHz and the sampling frequency likewise. Note that the filters frequency response scales with the sampling frequency ( $=\pi/2$ ). If the clock frequency is higher than the sampling frequency, the FIR compiler may re-use multipliers and adders when building the filter hardware.

The screenshot shows the 'FIR Compiler II' window with the 'Implementation Options' tab selected. The window is titled 'MegaWizard Plug-In Manager - FIR Compiler II'. It has a navigation bar with 'Filter Specification', 'Input/Output Options', and 'Implementation Options'. The 'Implementation Options' section contains several sub-sections: 'Frequency Specification' with fields for Clock Frequency (MHz) set to 50, Clock Slack set to 0, Input Sample Rate (MSPS) set to 50, and Speed Grade set to Medium; 'Symmetry Option' with Symmetry Mode set to Non Symmetry; 'Coefficients Reload Options' with an unchecked checkbox for Coefficients Reload, Base Address set to 0x0, and Read/Write Mode set to Read/Write; and 'Flow Control' with an unchecked checkbox for Back Pressure Support. The 'Resource Optimization Settings' section on the right shows Device Family set to Cyclone II, LEs / Small RAM Block Threshold set to 20, Small / Medium RAM Block Threshold set to 1280, Medium / Large RAM Block Threshold set to 1000000, and LEs / DSP Block Multiplier Threshold set to -1. At the bottom, there is an information section with details about the filter and some debug messages. The bottom right corner has buttons for 'Cancel', '< Back', 'Next >', and 'Finish'.

Implementation Options

Frequency Specification

Clock Frequency (MHz): 50

Clock Slack: 0

Input Sample Rate (MSPS): 50

Speed Grade: Medium

Symmetry Option

Symmetry Mode: Non Symmetry

Coefficients Reload Options

☐ Coefficients Reload

Base Address: 0x0

Read/Write Mode: Read/Write

Flow Control

☐ Back Pressure Support

Resource Optimization Settings

Device Family: Cyclone II

LEs / Small RAM Block Threshold: 20

Small / Medium RAM Block Threshold: 1280

Medium / Large RAM Block Threshold: 1000000

LEs / DSP Block Multiplier Threshold: -1

Info: PhysChanIn 1, PhysChanOut 1, ChansPerPhyIn 1, ChansPerPhyOut 1, OutputFullBitWidth 38, Bankcount 1, CoefBitWidth 8

Debug: Timing: VAL:1/0.112s ELA:1/1.712s

Debug: **add\_interface\_port**: Added reset port **reset** to **coeff\_clock\_reset\_reset** instead of **coeff\_clock\_reset**

Debug: **add\_interface\_port**: Added reset port **reset\_n** to **clock\_reset\_reset** instead of **clock\_reset**

Cancel < Back Next > Finish

F) Press finish and you are done.

Note that you can run the filter with a 48 KHz clock if you want, you will not be using the resources in an optimal fashion, but it will work. Keeping the clock and sampling frequency the same,  $\pi/2$  will now be the 48 KHz instead.