Getting Started with Quartus II Simulation Using the ModelSim- Altera Software

User Guide
The NativeLink feature of the Quartus II software allows you to control the ModelSim-Altera software from within the Quartus II software. By connecting the two tools with the NativeLink feature, the process of setting up your simulation is greatly streamlined, allowing you to focus on the simulation itself.

This user guide provides instructions for getting started with Quartus II project simulation using the ModelSim-Altera Starter Edition or ModelSim-Altera Edition software, and includes the following sections:

- “Prerequisites”
- “Starting the ModelSim-Altera Software with the Quartus II Software”
- “Setting Up EDA Tool Options” on page 1–1
- “Setting Up the Simulation” on page 1–3
- “Running ModelSim-Altera from the Quartus II Software” on page 1–4
- “Creating Stimulus Waveforms” on page 1–4
- “Modifying Stimulus Waveforms” on page 1–6
- “Starting Simulation” on page 1–7
- “Exporting Created Stimulus Waveforms as an HDL Testbench” on page 1–8

Prerequisites

You should have met the following prerequisites before performing a simulation:

- Understanding of Verilog, SystemVerilog, or VHDL hardware description language
- Completion of the “Getting Started Tutorial” in the Quartus II software

Starting the ModelSim-Altera Software with the Quartus II Software

To start the ModelSim-Altera software, follow these steps:

1. Unzip the provided Quartus II design example project counter.zip.
2. Start the Quartus II software and open the design example Quartus II project file counter.qpf.

Setting Up EDA Tool Options

You can specify where your third-party EDA simulators are installed with the EDA tool options settings in the Quartus II software. These settings enable you to start third-party EDA simulators from the Quartus II software.
For Windows users, the ModelSim-Altera tool path is automatically added during installation. Linux users must perform the following two steps to add the ModelSim-Altera tool path.

To set up the EDA tool options for ModelSim-Altera, follow these steps:

1. On the Tools menu of the Quartus II GUI, click Options. The Options dialog box appears. In the Category list, under General, click EDA Tool Options. The EDA Tool Options page appears (Figure 1–1).

2. Click the Browse button to the right of ModelSim-Altera in the EDA Tool list to specify the location of the ModelSim-Altera executable. The path is `<ModelSim-Altera installation path>\bin`.
Setting Up the Simulation

This section guides you in setting up the simulation environment to run the ModelSim-Altera simulator on your Quartus II project.

To set up the simulation environment, follow these steps:

1. On the Assignments menu, click Settings.

2. In the Settings dialog box, under EDA Tool Settings, select Simulation. The Simulation page appears. (Figure 1–2).

3. In the Tool name list, select ModelSim-Altera. Ensure that Run gate-level simulation automatically after compilation box is turned off.

4. Under EDA Netlist Writer settings, in the Format for output netlist list, select Verilog HDL. Ensure that the Map illegal HDL characters, Enable glitch filtering, and Generate Value Change Dump (VCD) file script boxes are turned off.

5. Under NativeLink settings, select None.

For more information about the Quartus II NativeLink feature, refer to the Simulating Altera Designs chapter in volume 3 of the Quartus II Handbook.
Running ModelSim-Altera from the Quartus II Software

In this section, you will use Quartus II software to generate the ModelSim-Altera automation script (.do file) and start the ModelSim-Altera software. The script then compiles the design files, and you are ready for simulation. Follow these steps:

1. On the Processing menu, point to Start, and then click Start Analysis & Elaboration.
2. On the Tools menu, point to Run EDA Simulation Tool and then click EDA RTL Simulation. The Quartus II software starts the ModelSim-Altera simulator.

Creating Stimulus Waveforms

This section guides you in adding signals to the Wave window, creating the clock waveform, and creating the reset waveform.

Adding Signals to the Wave Window

To add signals to the Wave window, follow these steps:

1. In the ModelSim-Altera GUI, expand Work in the Library window, and then right-click counter.
2. Click Create Wave (Figure 1–3).

Three signals (clk, reset, and count) from the counter design example are added in the Wave window, where you can create stimulus waveforms for each signal to simulate the design. In this example, you will create stimulus waveforms for the clk and reset signals. The count signal is not needed and can be deleted.
Creating Stimulus Waveforms

3. Right-click the `count` signal in the Wave window, and then click Delete.

**Creating the Clock Waveform**

1. Right-click the `clk` signal in the Wave window, and then click Create/Modify Waveform. The Create Pattern Wizard appears (Figure 1–4).

**Figure 1–4. Create Pattern Wizard**

2. Under Patterns, select Clock.
3. For Start Time, enter 0, for End Time, enter 5000, and for Time Unit, enter ns.
4. Click Next.
5. For Clock Period, enter 100, for Time Unit enter ns, and for Duty Cycle, enter 50.
6. Click Finish.

Creating the Reset Waveform

1. Right-click the reset signal in the Wave window, and then click Create/Modify Waveform.
2. In the Create Pattern Wizard, under Patterns, select Constant.
3. For Start Time, enter 0, for End Time, enter 5000, and for Time Unit, enter ns.
4. Click Next.
5. Enter St0 (Strong 0) for Value.
6. Click Finish.

For more information, refer to Chapter 12, Generating Stimulus with Waveform Editor in the ModelSim User’s Manual. In the ModelSim-Altera software, on the Help menu, point to PDF Documentation, and then click User’s Manual.

Modifying Stimulus Waveforms

To modify stimulus waveforms, follow these steps:

1. On the Wave menu, point to Mouse Mode and then click Edit Mode. Make sure the Wave window is activated. To activate it, click in the Wave window.
2. Select the reset signal from 0 ns to approximately 120 ns. The selection does not need to be exact (Figure 1–5).

Figure 1–5. Inverted Waveform for reset Signal
3. On the Wave menu, point to Wave Editor, and then click Invert. The Edit Invert dialog box appears (Figure 1–6).

![Figure 1–6. Edit Invert Dialog Box](image)

4. For Start Time, enter 0, for End Time, enter 120, and for Time Unit, enter ns.

5. Click OK to invert the waveform. Figure 1–7 shows the inverted waveform.

![Figure 1–7. Inverted Waveform for reset Signal (0–120 ns)](image)

### Starting Simulation

To start the simulation, follow these steps:

1. Expand the Work library, right-click counter and click Simulate.
2. Drag the count signal from the Objects window to the Wave window.
3. In the Transcript window, type the command `run -all`.
After you type the `run -all` command, the example counter design is simulated with the created stimulus waveforms for the clk and reset signals. The clk signal is a continuous clock waveform. The reset signal is asserted for the first 120 ns. When the reset signal is deasserted after 120 ns, the counter begins to increment, once on each rising edge of clk. The output signal count produces the simulated waveform, in the Wave window, adjacent to the stimulus waveforms.

### Exporting Created Stimulus Waveforms as an HDL Testbench

An HDL testbench file is typically written in the same hardware description language as your design, and interacts with your design as an instantiated module. An HDL testbench file is conceptually similar to a waveform like the one created in “Creating Stimulus Waveforms” on page 1–4, but because it is written in HDL, it is able to accommodate more complexity and flexibility than a graphical waveform.

The stimulus waveforms you created for the simulation can be exported as a HDL testbench file. You can then use the HDL testbench file to simulate your design again without manually recreating the stimulus waveforms.

To export the stimulus waveform as an HDL testbench, perform the following steps:

1. Click in the Wave window to ensure it is active.
2. On the File menu, point to Export and then click Waveform. The Export Waveform dialog box appears (Figure 1–8).
3. In the Export Waveform dialog box, under Save As, select Verilog Testbench.
4. For Start Time, enter 0, for End Time, enter 5000, and for Time Unit, enter ns.
5. In the File Name box, type the name of your testbench or click the Browse button.
6. Click OK.

Your testbench will be created in your `<Quartus II project folder>\simulation\modelsim` folder. You can use the testbench file for the simulation of your Quartus II design.
You can use NativeLink in the Quartus II software to start the ModelSim-Altera software, where you simulate your design with the exported testbench file.

For more information about setting up the testbench with NativeLink, refer to “Setting Up Testbench Files Using the NativeLink Feature” in *Simulating Altera Designs*.
This section provides additional information about the document and Altera.

**Document Revision History**

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2011</td>
<td>1.0</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

**How to Contact Altera**

To locate the most up-to-date information about Altera products, refer to the following table.

<table>
<thead>
<tr>
<th>Contact (1)</th>
<th>Contact Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical support</td>
<td>Website</td>
<td><a href="http://www.altera.com/support">www.altera.com/support</a></td>
</tr>
<tr>
<td>Technical training</td>
<td>Website</td>
<td><a href="http://www.altera.com/training">www.altera.com/training</a></td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td><a href="mailto:custrain@altera.com">custrain@altera.com</a></td>
</tr>
<tr>
<td>Product literature</td>
<td>Website</td>
<td><a href="http://www.altera.com/literature">www.altera.com/literature</a></td>
</tr>
<tr>
<td>Non-technical support (General)</td>
<td>Email</td>
<td><a href="mailto:nacomp@altera.com">nacomp@altera.com</a></td>
</tr>
<tr>
<td>(Software Licensing)</td>
<td>Email</td>
<td><a href="mailto:authorization@altera.com">authorization@altera.com</a></td>
</tr>
</tbody>
</table>

**Note to Table:**
(1) You can also contact your local Altera sales office or sales representative.

**Typographic Conventions**

The following table shows the typographic conventions this document uses.

<table>
<thead>
<tr>
<th><strong>Visual Cue</strong></th>
<th><strong>Meaning</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold Type with Initial Capital Letters</td>
<td>Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, Save As dialog box. For GUI elements, capitalization matches the GUI.</td>
</tr>
<tr>
<td><strong>bold type</strong></td>
<td>Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, \texttt{\textbackslash qdesigns} directory, \texttt{D:} drive, and \texttt{chiptrip.pdf} file.</td>
</tr>
<tr>
<td><strong>Italic Type with Initial Capital Letters</strong></td>
<td>Indicate document titles. For example, <em>Stratix IV Design Guidelines</em>.</td>
</tr>
<tr>
<td><strong>italic type</strong></td>
<td>Indicates variables. For example, \texttt{n + 1}. Variable names are enclosed in angle brackets (\texttt{&lt; &gt;}). For example, \texttt{&lt;file name&gt;} and \texttt{&lt;project name&gt;.pof} file.</td>
</tr>
<tr>
<td>Initial Capital Letters</td>
<td>Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.</td>
</tr>
<tr>
<td>Visual Cue</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>“Subheading Title”</td>
<td>Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”</td>
</tr>
<tr>
<td>Courier type</td>
<td>Indicates signal, port, register, bit, block, and primitive names. For example, data1, tdi, and input. The suffix n denotes an active-low signal. For example, resetn. Indicates command line commands and anything that must be typed exactly as it appears. For example, c:\qdesigns\tutorial\chiptrip.gdf. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword SUBDESIGN), and logic function names (for example, TRI).</td>
</tr>
<tr>
<td>↩</td>
<td>An angled arrow instructs you to press the Enter key.</td>
</tr>
<tr>
<td>1., 2., 3., and a., b., c., and so on</td>
<td>Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.</td>
</tr>
<tr>
<td>■ ■ ■</td>
<td>Bullets indicate a list of items when the sequence of the items is not important.</td>
</tr>
<tr>
<td>❗</td>
<td>The hand points to information that requires special attention.</td>
</tr>
<tr>
<td>❓</td>
<td>A question mark directs you to a software help system with related information.</td>
</tr>
<tr>
<td>🔴</td>
<td>The feet direct you to another document or website with related information.</td>
</tr>
<tr>
<td>🔴</td>
<td>A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.</td>
</tr>
<tr>
<td>🔴</td>
<td>A warning calls attention to a condition or possible situation that can cause you injury.</td>
</tr>
<tr>
<td>✉️</td>
<td>The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.</td>
</tr>
</tbody>
</table>