2. Mentor Graphics ModelSim and QuestaSim Support

QII53001-12.1.0

This chapter provides specific guidelines for simulation of Quartus® II designs with Mentor Graphics® ModelSim-Altera®, ModelSim, or QuestaSim software. Altera provides the entry-level ModelSim-Altera software, along with precompiled Altera simulation libraries, to simplify simulation of Altera designs. You can also refer to the following for more information about EDA simulation:

- For overview information, *Simulating Altera Designs* in the *Quartus II Handbook* and *About Using EDA Simulators* in Quartus II Help.
- For detailed GUI steps, *Preparing for EDA Simulation* and *Running EDA Simulators* in Quartus II Help.
- For support information, ModelSim-Altera Software page of the Altera website, Mentor Graphics ModelSim Simulation Design Examples page.

Quick Start Example (ModelSim Verilog)

You can adapt the following RTL simulation example to get started quickly with ModelSim:

1. Specify your EDA simulator and executable path in the Quartus II software:
   `set_user_option -name EDA_TOOL_PATH_MODELSIM <modelsim executable path>`↵
   `set_global_assignment -name EDA_SIMULATION_TOOL "MODELSIM (verilog)"`↵
2. Compile simulation model libraries using one of the following:
   - Run NativeLink RTL simulation to compile required design files, simulation models, and run your simulator. Verify results in your simulator. Skip steps 3 through 5.
   - Use Simulation Library Compiler to compile all required simulation models.
   - Create and map Altera libraries manually:
     `vlib <lib1>_ver`↵
     `vmap <lib1>_ver <lib1>_ver`↵

     Then, compile Altera simulation models manually:
     `vlog -work <lib1> <lib1>`
3. Compile your design and testbench files:
   `vlog -work work <design or testbench name>.v`↵
4. Load the design:
   `vsim -L work -L <lib1>_ver -L <lib2>_ver work.<testbench name>`↵
5. Run the simulation in the ModelSim simulator.

In this chapter, "ModelSim" refers to ModelSim SE, PE, and DE, which share the same commands as QuestaSim. "ModelSim-Altera" refers to ModelSim-Altera Starter Edition and ModelSim-Altera Subscription Edition.

ModelSim, ModelSim-Altera, and QuestaSim Guidelines

The following guidelines apply to simulation of Altera designs in the ModelSim, ModelSim-Altera, or QuestaSim software.

Using ModelSim-Altera Precompiled Libraries

Precompiled libraries for both functional and gate-level simulations are provided for the ModelSim-Altera software. You should not compile these library files before running a simulation. No precompiled libraries are provided for ModelSim or QuestaSim. You must compile the necessary libraries to perform functional or gate-level simulation with these tools.

The precompiled libraries provided in **<***ModelSim-Altera path***>/altera/**must be compatible with the version of the Quartus II software that is used to create the simulation netlist. To check whether the precompiled libraries are compatible with your version of the Quartus II software, refer to the **<***ModelSim-Altera path***>/altera/version.txt** file. This file shows which version and build of the Quartus II software was used to create the precompiled libraries.

For a list of precompiled library names for all functional and gate-level simulation models, refer to *ModelSim-Altera Precompiled Libraries* in Quartus II Help. For a list of all simulation model files, refer to *Altera Simulation Models* in Quartus II Help.

Encrypted Altera simulation model files shipped with the Quartus II software version 10.1 and later can only be read by ModelSim-Altera Edition Software version 6.6c and later. These encrypted simulation model files are located at the <*Quartus II System directory*>/**quartus**/**eda**/**sim_lib**/<*mentor*> directory.

Disabling Timing Violation on Registers

In certain situations, you may want to ignore timing violations on registers and disable the "**X**" propagation that occurs (for example, timing violations in internal synchronization registers in asynchronous clock-domain crossing).

By default, the **x_on_violation_option logic** option applying to all design registers is **On**, resulting in an output of "**X"** at timing violation. To disable "**X**" propagation at timing violations on a specific register, set the **x_on_violation_option logic** option to **Off** for that register. The following command is an example from the Quartus II Settings File (**.qsf**):

```
set_instance_assignment -name X_ON_VIOLATION_OPTION OFF -to \
<register_name>
```

Passing Parameter Information from Verilog HDL to VHDL

You must use in-line parameters to pass values from Verilog HDL to VHDL. Example 2–1 shows modification to use in-line parameters.

Example 2–1. In-Line Parameter Passing Example

```
lpm_add_sub#(.lpm_width(12), .lpm_direction("Add"),
.lpm_type("LPM_ADD_SUB"),
.lpm_hint("ONE_INPUT_IS_CONSTANT=NO,CIN_USED=NO" ))

lpm_add_sub_component (

        .dataa (dataa),

        .datab (datab),

        .result (sub_wire0)

);
```

The sequence of the parameters depends on the sequence of the GENERIC in the VHDL component declaration.

Increasing Simulation Speed

By default, the ModelSim and QuestaSim software runs in a debug-optimized mode. To run the ModelSim and QuestaSim software in speed-optimized mode, add the following two vlog command-line switches:

```
vlog -fast -05
```

In this mode, module boundaries are flattened and loops are optimized, which eliminates levels of debugging hierarchy and may result in faster simulation. This switch is not supported in the ModelSim-Altera simulator.

Simulating Transport Delays

By default, the ModelSim and QuestaSim software filter out all pulses that are shorter than the propagation delay between primitives. Turning on the **transport delay** options in the ModelSim and QuestaSim software prevents the simulator from filtering out these pulses.

Table 2–1 describes the transport delay options.

Table 2–1. Transport Delay Options

Option	Description
`+transport_path_delays`	Use when simulation pulses are shorter than the delay in a gate-level primitive. You must include the `+pulse_e/number` and `+pulse_r/number` options.
`+transport_int_delays`	Use when simulation pulses are shorter than the interconnect delay between gate-level primitives. You must include the `+pulse_int_e/number` and `+pulse_int_r/number` options.

The `+transport_path_delays` and `+transport_path_delays` options apply by default during NativeLink gate-level timing simulation.

For more information about either of these options, refer to the ModelSim-Altera Command Reference installed with the ModelSim and QuestaSim software.

The following ModelSim and QuestaSim software command shows the command line syntax to perform a gate-level timing simulation with the device family library:

```
vsim -t 1ps -L stratixii -sdftyp /i1=filtref_vhd.sdo work.filtref_vhd_vec_tst \
+transport_int_delays +transport_path_delays
```

Viewing Error Messages

ModelSim and QuestaSim error and warning messages are tagged with a `vsim` or `vcom` code. To determine the cause and resolution for a `vsim` or `vcom` error or warning, use the `verror` command.

For example, ModelSim and QuestaSim may display the following error message:

```
# ** Error:
C:/altera_trn/DUALPORT_TRY/simulation/modelsim/DUALPORT_TRY.vho(31):
(vcom-1136) Unknown identifier "stratixiii".
```

In this case, type the following command:

```
verror 1136 ↵
```

A description of the error message appears as follows:

```
# vcom Message # 1136:
# The specified name was referenced but was not found. This indicates
# that either the name specified does not exist or is not visible at
# this point in the code.
```

Generating Power Analysis Files

To generate a timing Value Change Dump File (**.vcd**) for power analysis, you must first generate a *<filename>***_dump_all_vcd_nodes.tcl** script file in the Quartus II software. You can then run the script from the ModelSim, QuestaSim, or ModelSim-Altera software to generate a timing *<filename>***.vcd**. You can use this **.vcd** for power analysis in the Quartus II PowerPlay power analyzer.

To use a**.vcd** for power analysis, follow these steps:

1. In the Quartus II software, click **Settings** on the Assignments menu.
2. Click **Simulation** under **EDA Tool Settings**.
3. Turn on **Generate Value Change Dump file script**, specify the type of output signals to include, and specify the top-level design instance name in your testbench.
4. On the Processing menu, click **Start Compilation**.
5. On the Tools menu, point to **Run EDA Simulation**, and then click **EDA Gate Level Simulation**. The Compiler creates the *<filename>***_dump_all_vcd_nodes.tcl** file, the ModelSim simulation *<filename>***_run_msim_gate_vhdl/verilog.do** file (including the **.vcd** and **.tcl** execution lines), and all other files for simulation. ModelSim then automatically runs the generated **.do** to start the simulation.
6. Break the simulation if your testbench does not have a break point. End the simulation to have ModelSim generate the **.vcd**. You can only generate the **.vcd** after simulation ends with the **End Simulation** function.

For more information about using the timing <*filename*>**.vcd** for power estimation, refer to the *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*.

Viewing a Simulation Waveform

ModelSim-Altera, ModelSim, and QuestaSim automatically generate a Wave Log Format File (**.wlf**) following simulation. You can use the **.wlf** to generate a waveform view.

To view a waveform from a **.wlf** through ModelSim-Altera, ModelSim, or QuestaSim, perform the following steps:

1. Type `vsim` at the command line. The **ModelSim/QuestaSim** or **ModelSim-Altera** dialog box appears.
2. On the File menu, click **Datasets**. The **Datasets Browser** dialog box appears.
3. Click **Open** and browse to the directory that contains your **.wlf**.
4. Select the **.wlf** file and click **Open**, then click **OK**.
5. Click **Done**.
6. In the Object browser, select the signals that you want to observe.
7. On the Add menu, click **Wave** and then click **Selected Signals**.

You cannot view a waveform from a **.vcd** in ModelSim-Altera, ModelSim, or QuestaSim directly. The **.vcd** must first be converted to a **.wlf**.

1. Use the `vcd2wlf` command to convert the file. For example, type the following at the command-line:

   ```
   vcd2wlf <example>.vcd <example>.wlf ↵
   ```

2. After you convert the **.vcd** to a **.wlf**, follow the procedures for viewing a waveform from a **.wlf** through ModelSim and QuestaSim.

You can also convert your **.wlf** to a **.vcd** by using the `wlf2vcd` command.

Simulating with ModelSim-Altera Waveform Editor

You can use the ModelSim-Altera Waveform Editor as a simple method to create stimulus vectors for simulation. You can create this design stimulus via interactive manipulation of waveforms from the wave window in ModelSim-Altera. With the ModelSim-Altera waveform editor, you can create and edit waveforms, drive simulation directly from created waveforms, and save created waveforms into a stimulus file.

For more information, refer to the *Generating Stimulus with Waveform Editor* chapter in the *ModelSim SE User's Manual* available on the ModelSim website (www.model.com).

Simulation Setup Script Example

The Quartus II software can generate a **msim_setup.tcl** simulation setup script for IP cores in your design. The script compiles the required device library models, compiles the design files, and elaborates the design with or without simulator optimization. To run the script, type `source msim_setup.tcl` in the simulator Transcript window. Alternatively, if you are using the simulator at the command line, you can type the following command:

```
vsim -c -do msim_setup.tcl.
```

Example 2–2 shows the **top-level-simulate.do** custom top-level simulation script that sets the hierarchy variable `TOP_LEVEL_NAME` to `top_testbench` for the design, and sets the variable `QSYS_SIMDIR` to the location of the generated simulation files.

Example 2–2. Example Top Level Simulation Script (top-level-simulate.do)

```
# Set hierarchy variables used in the IP-generated files
set TOP_LEVEL_NAME "top_testbench"
set QSYS_SIMDIR "./ip_top_sim"

# Source generated simulation script which defines aliases used below
source $QSYS_SIMDIR/mentor/msim_setup.tcl

# dev_com alias compiles simulation libraries for device library files
dev_com

# com alias compiles IP simulation or Qsys model files and/or Qsys model
files in the correct order
com

# Compile top level testbench that instantiates your IP
vlog -sv ./top_testbench.sv

# elab alias elaborates the top-level design and testbench
elab

# Run the full simulation
run - all
```

In this example, the top-level simulation files are stored in the same directory as the original IP core, so this variable is set to the IP-generated directory structure. The `QSYS_SIMDIR` variable provides the relative hierarchy path for the generated IP simulation files. The script calls the generated **msim_setup.tcl** script and uses the `alias` commands from the script to compile and elaborate the IP files required for simulation along with the top-level simulation testbench. You can specify additional simulator elaboration command options when you run the `elab` command, for example, `elab +nowarnTFMPC`. The last command run in the example starts the simulation.

Unsupported Features

The Quartus II software does not support the following simulation features:

- Altera does not support companion licensing for ModelSim AE.
- The USB software guard is not supported by versions earlier than Mentor Graphics ModelSim software version 5.8d.
- For ModelSim-Altera software versions prior to 5.5b, use the **PCLS** utility included with the software to set up the license.

- Some versions of ModelSim and QuestaSim support SystemVerilog, PSL assertions, SystemC, and more. For more information about specific feature support, refer to Mentor Graphics literature.

For more information about the ModelSim-Altera Subscription Edition software, including pricing, refer to the ModelSim-Altera Software page of the Altera website. For more information about obtaining and setting up the license for the ModelSim-Altera Subscription Edition software, refer to the "Licensing Altera Software" section in the *Altera Software Installation and Licensing Manual*.

Document Revision History

Table 2–2 shows the revision history for this chapter.

Table 2–2. Document Revision History (Part 1 of 2)

Date	Version	Changes
November 2012	12.1.0	Relocated general simulation information to *Simulating Altera Designs.*
June 2012	12.0.0	Removed survey link.
November 2011	11.1.0	■ Added information about encrypted Altera simulation model files. ■ Updated power analysis information.
May 2011	11.0.0	■ Updated "Software Requirements" on page 2–2 ■ Updated "Design Flow with ModelSim-Altera, ModelSim, or QuestaSim Software" on page 2–2 ■ Restructured "Simulating with the ModelSim-Altera Software" on page 2–4 ■ Restructured "Simulating with the ModelSim and QuestaSim Software" on page 2–5 ■ Restructured "Simulating Designs that Include Transceivers" on page 2–12 ■ Changed section name from "ModelSim and QuestaSim Error Message Verification" to "ModelSim and QuestaSim Error Message Information" on page 2–18 ■ Changed section name from "Simulating with ModelSim-Altera Waveform" to "Simulating with ModelSim-Altera Waveform Editor" on page 2–20
December 2010	10.1.0	■ Changed to new document template ■ Referenced Simulating Altera Designs chapter ■ Added new section, "Simulating with ModelSim-Altera Waveform Editor" on page 2–20 ■ Removed Stratix V compilation information and linked to Quartus II Help
July 2010	10.0.0	■ Removed simulation library tables and linked to Quartus II Help ■ Added other links to Quartus II Help and ModelSim-Altera Help where appropriate and removed redundant information ■ Added QuestaSim support ■ Added Stratix V simulation information ■ Minor editorial changes throughout ■ Removed Referenced Documents section

Table 2–2. Document Revision History (Part 2 of 2)

Date	Version	Changes
November 2009	9.1.0	■ Removed NativeLink information and referenced new *Simulating Designs with EDA Tools* chapter ■ Added Stratix IV transceiver simulation section ■ Reformatted transceiver simulation sections ■ Text edits throughout chapter
March 2009	9.0.0	Added the following sections: ■ "Compile Libraries Using the EDA Simulation Library Compiler" on page 2–17 ■ "Generate Simulation Script from EDA Netlist Writer" on page 2–77 ■ "Viewing a Waveform from a .wlf File" on page 2–78 Updated the following: ■ Table 2–1, Table 2–2, Table 2–5, Table 2–6, Table 2–7, Table 2–8, Table 2–9, Table 2–10 ■ Figure 2–4 on page 2–81 ■ All sections titled "Loading the Design"
November 2008	8.1.0	Updated the following: ■ Table 2–2, Table 2–3, Table 2–4, Table 2–5, Table 2–6 ■ Removed `--zero_ic_delays` from `quartus_sta` option in "Generate Post-Synthesis Simulation Netlist Files" on page 2–11 ■ Removed steps to include the library when the simulation is run in VHDL mode from all procedures; this is no longer necessary ■ Added information about the Altera Simulation Library Compiler throughout the chapter ■ Added "Compile Libraries Using the Altera Simulation Library Compiler" on page 2–15 ■ Added "Disabling Simulation" on page 2–72 ■ Minor editorial updates ■ Updated entire chapter using 8½" × 11" chapter template
May 2008	8.0.0	Updated the following: ■ "Altera Design Flow with ModelSim-Altera or ModelSim Software" on page 2–3 ■ "Simulation Libraries" on page 2–4 ■ "Simulation Netlist Files" on page 2–11 ■ "Perform Simulation Using ModelSim-Altera Software" on page 2–15 ■ "Perform Simulation Using ModelSim Software" on page 2–33 ■ "Simulating Designs that Include Transceivers" on page 2–57 ■ "Using the NativeLink Feature with ModelSim-Altera or ModelSim Software" on page 2–63 ■ "Generating a Timing VCD File for PowerPlay" on page 2–68