

This chapter provides detailed instructions about how to simulate your design in the ModelSim-Altera® software, Mentor Graphics® ModelSim software, and Mentor Graphics QuestaSim software.

An Altera Quartus® II software subscription includes the ModelSim-Altera Starter Edition, which is a no-cost entry-level version of the ModelSim-Altera Subscription Edition software. The ModelSim-Altera Subscription Edition software offers support for all Altera devices. Both versions are available on PC and Linux platforms. You can use the ModelSim-Altera software to perform functional, post-synthesis, and gate-level timing simulations for either Verilog HDL or VHDL designs that target an Altera FPGA.



In this chapter, ModelSim refers to ModelSim SE, PE, DE, and QuestaSim. ModelSim-Altera refers to ModelSim-Altera Starter Edition and ModelSim-Altera Subscription Edition software.

This chapter includes the following topics:

- “Software Requirements”
- “Design Flow with ModelSim-Altera or ModelSim/QuestaSim Software”
- “Simulation Libraries” on page 2–2
- “Performing Simulation Using the ModelSim-Altera Software” on page 2–3
- “Performing Simulation Using the ModelSim/QuestaSim Software” on page 2–5
- “Simulating Designs that Include Transceivers” on page 2–16
- “Using the NativeLink Feature with ModelSim-Altera or ModelSim/QuestaSim Software” on page 2–24
- “Generating a Timing Value Change Dump (.vcd) File for the PowerPlay Power Analyzer” on page 2–25
- “Viewing a Waveform from a .wlf File” on page 2–26
- “Scripting Support” on page 2–26
- “Software Licensing and Licensing Setup in ModelSim-Altera Subscription Edition” on page 2–28

Software Requirements

To simulate your design using the ModelSim-Altera and ModelSim/QuestaSim software, you must first set up the Altera libraries. These libraries are installed with the Quartus II software.



For more information about installing the software and directories created during the Quartus II software installation, refer to the *Altera Software Installation and Licensing* manual.

Design Flow with ModelSim-Altera or ModelSim/QuestaSim Software

You can perform the following types of simulations using the ModelSim-Altera and ModelSim/QuestaSim software:

- Functional simulation
- Post-synthesis simulation
- Gate-level timing simulation



Some versions of ModelSim and QuestaSim support SystemVerilog, PSL assertions, SystemC, and more. Refer to Mentor Graphics literature or your salesperson to learn more about the features supported in the different versions of ModelSim and QuestaSim.



The VHDL version of ModelSim-Altera and other single-language VHDL versions of ModelSim cannot simulate designs that target the Stratix V device family if you are using transceivers.

You need a version of ModelSim that supports VHDL/Verilog co-simulation to simulate designs that use Stratix V transceivers.



For more information about the Quartus II software design flow, refer to the “PLD Design Flow” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.



For additional documentation about ModelSim-Altera, refer to the ModelSim-Altera Help that ships with the product. Click the **Help** button on the ModelSim-Altera toolbar.

Simulation Libraries


Simulation model libraries are required to run a simulation whether you are running a functional simulation, post-synthesis simulation, or gate-level timing simulation. In general, running a functional simulation requires the functional simulation model libraries, while running a post-synthesis or gate-level timing simulation requires the gate-level timing simulation model libraries. Unless you are using ModelSim-Altera, you must compile the necessary library files before you can run the simulation. The ModelSim-Altera software has the Altera libraries pre-compiled and built in. Do not compile these libraries again.

There are a few exceptions where you must compile gate-level timing simulation library files to perform functional simulation. For example, some Altera megafunctions require gate-level libraries to perform a functional simulation using third-party simulators.

Precompiled Simulation Libraries in the ModelSim-Altera Software


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.

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 Precompiled Libraries* in Quartus II Help.

Simulation Library Files in the Quartus II Software

In ModelSim/QuestaSim, no precompiled libraries are available. You must compile the necessary libraries to perform functional or gate-level simulation.

 For a list of all functional simulation library files in the Quartus II directory, refer to *Altera Functional Simulation Libraries* in Quartus II Help. For a list of all post-synthesis and post-fit (gate-level) library files in the Quartus II directory, refer to *Altera Post-Fit Libraries* in Quartus II Help.

Disabling Timing Violation on Registers

In certain situations, a timing violation can be ignored and you can disable timing violations on registers (for example, timing violations that occur in internal synchronization registers used for asynchronous clock domain crossing).

By default, the `x_on_violation_option` logic option is **On**, which means simulation shows “x” whenever a timing violation occurs. To disable showing the timing violation on certain registers, set the `x_on_violation_option` logic option to **Off** on those registers. The following Quartus II Tcl command disables timing violation on registers. This Tcl command is also stored in the `.qsf` file.

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


Performing Simulation Using the ModelSim-Altera Software

You can perform simulation of Verilog HDL or VHDL designs with the ModelSim-Altera software at three levels: functional, post-synthesis, and gate-level.

For high-speed simulation, you must select **ps** in the **Resolution** list for your simulator resolutions (**Design** tab of the **Start Simulation** dialog box). If you choose slower than **ps**, the high-speed simulation may fail.

Performing Functional Simulation

Functional simulation verifies code syntax and design functionality. The following sections describe how to perform functional simulation in the ModelSim-Altera software for a Verilog HDL or VHDL design.

 The ModelSim-Altera software includes precompiled simulation libraries for Altera-provided models. You should not create simulation libraries and compile simulation models for the pre-compiled Altera libraries.

Setting Up a Quartus II Project for the ModelSim-Altera Software

The first steps in performing a simulation are starting the ModelSim-Altera software, changing to your project/simulation directory, and creating libraries for your design.

- For more information, refer to *Setting Up a Project with the ModelSim-Altera Software* in Quartus II Help.

Compiling and Loading Designs with the ModelSim-Altera Software

- For information about compiling and loading your design files and testbench files, refer to *Mapping to Libraries and Compiling Design Files with the ModelSim-Altera Software* in Quartus II Help.

Performing the Simulation

- For information about performing a functional simulation with the ModelSim-Altera software, refer to *Performing a Functional Simulation with the ModelSim-Altera Software* in Quartus II Help.

Performing Post-Synthesis Simulation




You can perform post-synthesis simulation to verify that design functionality is preserved after synthesis. You can create the post-synthesis netlist in the Quartus II software and use the netlist to perform post-synthesis simulation with the ModelSim-Altera software.

- Before running post-synthesis simulation, generate post-synthesis simulation netlist files. For more information, refer to the “Generating Post-Synthesis Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.
- The ModelSim-Altera software includes precompiled simulation libraries. You should not create simulation libraries and compile simulation models.
- For information about performing a post-synthesis simulation with the ModelSim-Altera software, refer to *Performing a Timing Simulation with the ModelSim-Altera Software* in Quartus II Help.

Performing Gate-Level Timing Simulation

Gate-level timing simulation is an important step in ensuring that the device functionality is correct and meets all timing requirements following place and route. You can create the gate-level netlist in the Quartus II software and use the netlist to perform gate-level timing simulation with the ModelSim-Altera software.

- Before running gate-level timing simulation, generate gate-level timing simulation netlist files. For more information, refer to the “Generating Gate-Level Timing Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

-  The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.
-  For information about performing a gate-level simulation with the ModelSim-Altera software, refer to *Performing a Timing Simulation with the ModelSim-Altera Software* in Quartus II Help.
-  For additional documentation about ModelSim-Altera, refer to the ModelSim-Altera Help that ships with the product. Click the **Help** button on the ModelSim-Altera toolbar.

Performing Simulation Using the ModelSim/ QuestaSim Software

You can perform simulation of Verilog HDL or VHDL designs with the ModelSim/ QuestaSim software at three levels: functional, post-synthesis, and gate-level.

You can perform the simulation through the GUI or from the command line. The following sections provide instructions to perform the simulation through the GUI and from the command line. You can proceed to the specific section that meets your needs.

For high-speed simulation, you must select **ps** in the **Resolution** list for your simulator resolutions (**Design** tab of the **Start Simulation** dialog box). If you choose slower than **ps**, the high-speed simulation may fail.

Simulating VHDL Designs Using the GUI

This section provides information about performing functional, post-synthesis, and gate-level simulations of VHDL designs using the GUI.

Performing Functional Simulation

This section provides information about compiling simulation models and performing a functional simulation.

Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*, perform the following steps to compile simulation models into simulation libraries:


1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library that you created (for example, **altera_mf**, **lpm**).
3. Browse to the `<Quartus II installation directory>/eda/sim_lib` and add the necessary simulation model files to your project. Select the simulation model files and click **Compile**.

If you are targeting a Stratix V device, compile the following files in the **mentor** subdirectory:

```
stratixv_atoms_ncrypt.v
stratixv_hssi_atoms_ncrypt.v
stratixv_pcie_hip_atoms_ncrypt.v
```


These files contain IEEE encrypted Verilog models suitable for VHDL/Verilog co-simulation. You need a co-simulation license from Mentor Graphics to use these models.


 Compile these encrypted Verilog files before you compile any VHDL files.

 Compile **stratixv_pcie_hip_atoms_ncrypt.v** with the SystemVerilog option.

Also, compile the following files in the **quartus/eda/sim_lib** directory:

```
stratixv_atoms.vhd
stratixv_components.vhd
stratixv_hssi_components.vhd
stratixv_pcie_hip_components.vhd
stratixv_hssi_atoms.vhd
stratixv_pcie_hip_atoms.vhd
```


 The PCIe® files are required only if you are using the PCIe HIP.

 The **altera_mf_components.vhd** and **altera_mf.vhd** model files should be compiled into the **altera_mf** library. The **220pack.vhd** and **220model.vhd** model files should be compiled into the **lpm** library.

4. Repeat step 2 and step 3 to compile other simulation models.
5. Click **Done**.


Performing the Simulation


 For information about simulating VHDL designs using the GUI, refer to *Performing a Functional Simulation with the ModelSim Software* and *Performing a Functional Simulation with the QuestaSim Software* in Quartus II Help.


 To see all of the functional simulation library files, refer to *Altera Functional Simulation Libraries* in Quartus II Help.

Performing Post-Synthesis Simulation

You can perform post-synthesis simulation to verify that design functionality is preserved after synthesis. You can create the post-synthesis netlist in the Quartus II software and use the netlist to perform post-synthesis simulation with the ModelSim/QuestaSim software.


 Before running post-synthesis simulation, generate post-synthesis simulation netlist files. For more information, refer to the “Generating Post-Synthesis Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.


 You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.


 For information about performing a post-synthesis simulation using the GUI, refer to *Performing a Timing Simulation with the ModelSim Software* and *Performing a Timing Simulation with the QuestaSim Software* in Quartus II Help.

Performing Gate-Level Simulation

Gate-level simulation is a very important step in ensuring that the FPGA device's functionality is still correct and meets all required timing requirements after the design was placed and routed. You can create the gate-level netlist in the Quartus II software and use the netlist to perform gate-level simulation with the ModelSim/QuestaSim software.

 You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.

 Before running gate-level simulation, generate gate-level timing simulation netlist files. For more information, refer to the “Generating Gate-Level Timing Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

 For information about performing a gate-level simulation using the GUI, refer to *Performing a Timing Simulation with the ModelSim Software* and *Performing a Timing Simulation with the QuestaSim Software* in Quartus II Help.

Simulating Verilog HDL Designs Using the GUI

This section provides information about performing functional, post-synthesis, and gate-level simulations of Verilog HDL designs using the GUI.

Performing Functional Simulation

This section provides information about compiling simulation models and performing a functional simulation.

Compiling Simulation Models into Simulation Libraries


If you are not using the EDA Simulation Library Compiler, as described in the section “EDA Simulation Library Compiler” in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*, perform the following steps to compile simulation models into simulation libraries:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library that you created (for example, **altera_mf_ver** or **lpm_ver**).
3. Browse to the `<Quartus II installation directory>/eda/sim_lib` and add the necessary simulation model files to your project. Select the simulation model files and click **Compile**.

If you are targeting a Stratix V device, compile the following files in the **mentor** subdirectory:


```
stratixv_atoms_ncrypt.v
stratixv_hssi_atoms_ncrypt.v
stratixv_pcie_hip_atoms_ncrypt.v
```


These files contain IEEE encrypted Verilog models.

 Compile **stratixv_pcie_hip_atoms_ncrypt.v** with the SystemVerilog option.

Also, compile the following files in the **quartus/eda/sim_lib** directory:


```
stratixv_atoms.v
stratixv_hssi_atoms.v
stratixv_pcie_hip_atoms.v
```


 The PCIe file is required only if you are using the PCIe HIP.

 The **altera_mf.v** model files should be compiled into the **altera_mf_ver** library. The **220model.v** model files should be compiled into the **lpm_ver** library.

4. Repeat step 2 and step 3 to compile other simulation models.
5. Click **Done**.


Performing the Simulation


 For information about performing a functional simulation using the GUI, refer to *Performing a Functional Simulation with the ModelSim Software* and *Performing a Functional Simulation with the QuestaSim Software* in Quartus II Help.


 To see all of the functional simulation library files, refer to *Altera Functional Simulation Libraries* in Quartus II Help.

Performing Post-Synthesis Simulation

Perform post-synthesis simulation to verify that design functionality is preserved after synthesis. Create the post-synthesis netlist in the Quartus II software and use the netlist to perform post-synthesis simulation with the ModelSim/QuestaSim software.

 Before running post-synthesis simulation, generate post-synthesis simulation netlist files. For more information, refer to the “Generating Post-Synthesis Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

 You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.

 For information about performing a post-synthesis simulation using the GUI, refer to *Performing a Timing Simulation with the ModelSim Software* and *Performing a Timing Simulation with the QuestaSim Software* in Quartus II Help.

Performing Gate-Level Simulation

Gate-level simulation is a very important step in ensuring that the FPGA device's functionality is still correct and meets all required timing requirements after the design was placed and routed. You can create the gate-level netlist in the Quartus II software and use the netlist to perform gate-level simulation with the ModelSim/QuestaSim software.



You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.



Before running gate-level simulation, generate gate-level timing simulation netlist files. For more information, refer to the "Generating Gate-Level Timing Simulation Netlist Files" section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.



For information about performing a gate-level simulation using the GUI, refer to *Performing a Timing Simulation with the ModelSim Software* and *Performing a Timing Simulation with the QuestaSim Software* in Quartus II Help.

Simulating VHDL Designs From the Command Line

This section provides information about performing functional, post-synthesis, and gate-level simulations of VHDL designs from the command line.

Simulating VHDL designs from the ModelSim/QuestaSim command line gives you more flexibility and control in compiling the libraries and loading and simulating the VHDL design files. All simulation commands are Tcl commands that can be included in the *.do file. Using the *.do file allows you to run simulation in batch mode. You have to execute only the *.do file, and the ModelSim/QuestaSim tool automatically executes all commands in the *.do script macro file.

Performing Functional Simulation

Function simulation verifies code syntax and design functionality.

Type the following commands to perform a functional simulation for VHDL designs with one of the libraries (lib1) listed in the *Altera Functional Simulation Libraries* in Quartus II Help.

To create and compile Altera libraries, type the following commands:

```
vlib <lib1> ←  
vmap <lib1> <lib1> ←  
vcom -work <lib1> <lib1>.vhd ←  
vlib <lib2> ←  
vmap <lib2> <lib2> ←  
vcom -work <lib2> <lib2>.vhd ←
```

To create the work library and compile the design and testbench files, type the following commands:

```
vlib work ←  
vmap work work ←  
vcom -work work <design_file1>.vhd <design_file2>.vhd <testbench \  
file>.vhd ←
```

To load the design, type the following command:

```
vsim -L work -L <lib1> -L <lib2> work.<testbench module name> ↵
```

To add signals to the waveform viewer and run the simulation, type the following commands:

```
add wave * ↵
run ↵
```

Example

```
# Create and compile Altera libraries
vlib altera_mf
vmap altera_mf altera_mf
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd
vlib lpm
vmap lpm lpm
vcom -work lpm 220pack.vhd 220model.vhd

# Create work library and compile design files and testbench file
vlib work
vmap work work
vcom -work work top_level.vhd adder.vhd testbench.vhd

# Load design
vsim -L work -L altera_mf -L lpm work.testbench

# add signals to the waveform viewer and run simulation
add wave *
run
```

If you are targeting a Stratix V device, compile the following files in the **quartus/eda/sim_lib/mentor** directory:

```
stratixv_atoms_ncrypt.v
stratixv_hssi_atoms_ncrypt.v
stratixv_pcie_hip_atoms_ncrypt.v
```

These files contain IEEE encrypted Verilog models suitable for VHDL/Verilog co-simulation. You need a co-simulation license from Mentor Graphics to use these models.

Compile these encrypted Verilog files before you compile any VHDL files.

Compile **stratixv_pcie_hip_atoms_ncrypt.v** with the SystemVerilog option.

Also, compile the following files in the **quartus/eda/sim_lib** directory:

```
stratixv_atoms.vhd
stratixv_components.vhd
stratixv_hssi_components.vhd
stratixv_pcie_hip_components.vhd
stratixv_hssi_atoms.vhd
stratixv_pcie_hip_atoms.vhd
```

The PCIe® files are required only if you are using the PCIe HIP.

Performing Post-Synthesis Simulation

Perform post-synthesis simulation to verify that design functionality is preserved after synthesis. Create the post-synthesis netlist in the Quartus II software and use the netlist to perform post-synthesis simulation with the ModelSim/QuestaSim software. Before running post-synthesis simulation, generate post-synthesis simulation netlist files.

- For more information, refer to the “Generating Post-Synthesis Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the Quartus II Handbook.
- You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.

Type the following commands to perform a post-synthesis simulation for VHDL designs with one of the libraries (lib1) listed in *Altera Post-Fit Libraries* in Quartus II Help.

To create and compile Altera libraries, type the following commands:

```
vlib <lib1> ↵  
vmap <lib1> <lib1> ↵  
vcom -work <lib1> <lib1>.vhd ↵  
vlib <lib2> ↵  
vmap <lib2> <lib2> ↵  
vcom -work <lib2> <lib2>.vhd ↵
```

To create the work library and compile design and testbench files, type the following commands:

```
vlib work ↵  
vmap work work ↵  
vcom -work work <output_netlist>.vho <testbench file>.vhd ↵
```

To load the design, type the following command:

```
vsim +transport_int_delays +transport_path_delays -L work -L \ <lib1> -L  
<lib2> work.<testbench module name> ↵
```

To add signals to the waveform viewer and run simulation, type the following commands:

```
add wave * ↵  
run ↵
```

Example

```
# Create and compile Altera libraries
vlib altera
vmap altera altera
vcom -work altera altera_primitives_components.vhd \
altera_primitives.vhd
vlib stratixiii
vmap stratixiii stratixiii
vcom -work stratixiii stratixiii.atoms.vhd stratixiii_components.vhd

# Create work library and compile design files and testbench file
vlib work
vmap work work
vcom -work work top_level.vho testbench.vhd

# Load design
vsim +transport_int_delays +transport_path_delays -L work -L \ altera
-L stratixiii work.testbench

# add signals to the waveform viewer and run simulation
add wave *
run
```

Performing Gate-Level Simulation

The steps for gate-level timing simulation are almost same as the steps for post-synthesis simulation.

The only difference is that the **.sdo** file must be back-annotated for gate level-timing simulation.

For VHDL designs, add the **-sdftyp** option for back-annotating.

Example

```
vsim +transport_int_delays +transport_path_delays -sdftyp \ <instance
path to design>= <path to SDO file> -L work -L stratixiii -L \ altera
work.testbench
```

You do not have to set the value (minimum, average, maximum) for the ***.sdo** file, because the Quartus II EDA Netlist Writer generates the ***.sdo** file using the same value for the triplet (minimum, average, and maximum timing values).

If your design under test is instantiated in the testbench file under the **i1** label, the **<design instance>** should be "i1" (for example, /i1=<my design>.sdo).



You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.

Simulating Verilog HDL Designs from the Command Line

This section provides information about performing functional, post-synthesis, and gate-level simulations of Verilog HDL designs from the command line.

Simulating Verilog HDL designs from the ModelSim/QuestaSim command line gives you more flexibility and control in compiling the libraries and loading and simulating the Verilog HDL design files. All simulation commands are Tcl commands that can be included in the *.do file. Using the *.do file allows you to run simulation in batch mode. You have to execute only the *.do file, and the ModelSim/QuestaSim tool automatically executes all commands in the *.do script macro file.

Performing Functional Simulation

Functional simulation verifies code syntax and design functionality.

Type the following commands to perform a functional simulation for Verilog HDL designs with one of the libraries (lib1) listed in *Altera Functional Simulation Libraries* in Quartus II Help.

To create and compile Altera libraries, type the following commands:

```
vlib <lib1> ↵  
vmap <lib1> <lib1> ↵  
vlog -work <lib1> <lib1>.v ↵  
vlib <lib2> ↵  
vmap <lib2> <lib2> ↵  
vlog -work <lib2> <lib2>.v ↵
```

To create the work library and compile design and testbench files, type the following commands:

```
vlib work ↵  
vmap work work ↵  
vlog -work work <design_file1>.v <design_file2>.v <testbench_file>.v ↵
```

To load the design, type the following command:

```
vsim -L work -L <lib1> -L <lib2> work.<testbench module name> ↵
```

To add signals to the waveform viewer and run simulation, type the following commands:

```
add wave * ↵  
run ↵
```

Example


```
# Create and compile Altera libraries  
  
vlib altera_mf_ver  
vmap altera_mf_ver altera_mf_ver  
vlog -work altera_mf_ver altera_mf.v  
vlib lpm_ver  
vmap lpm_ver lpm_ver  
vlog -work lpm_ver 220model.v  
  
# Create work library and compile design files and testbench file  
  
vlib work  
vmap work work  
vlog -work work top_level.v adder.v testbench.v  
  
# Load design
```

```
vsim -L work -L altera_mf_ver -L lpm_ver work.testbench
# add signals to the waveform viewer and run simulation
add wave *
run
```

If you are targeting a Stratix V device, compile the following files in the **quartus/eda/sim_lib/mentor** directory:


```
stratixv_atoms_ncrypt.v
stratixv_hssi_atoms_ncrypt.v
stratixv_pcie_hip_atoms_ncrypt.v
```

These files contain IEEE encrypted Verilog models.

 Compile **stratixv_pcie_hip_atoms_ncrypt.v** with the SystemVerilog option.

Also, compile the following files in the **quartus/eda/sim_lib** directory:


```
stratixv_atoms.v
stratixv_hssi_atoms.v
stratixv_pcie_hip_atoms.v
```

 The PCIe file is required only if you are using the PCIe HIP.

Performing Post-Synthesis Simulation

Perform post-synthesis simulation to verify that design functionality is preserved after synthesis. Create the post-synthesis netlist in the Quartus II software and use the netlist to perform post-synthesis simulation with the ModelSim/QuestaSim software. Before running post-synthesis simulation, generate post-synthesis simulation netlist files.

 For more information, refer to the “Generating Post-Synthesis Simulation Netlist Files” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the Quartus II Handbook.

 You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.

Type the following commands to perform a post-synthesis simulation for Verilog HDL designs with one of the libraries (lib1) listed in *Altera Post-Fit Libraries* in Quartus II Help.

To create and compile Altera libraries, type the following commands:

```
vlib <lib1> ↵
vmap <lib1> <lib1> ↵
vlog -work <lib1> <lib1>.v ↵
vlib <lib2> ↵
vmap <lib2> <lib2> ↵
vlog -work <lib2> <lib2>.v ↵
```

To create the work library and compile design and testbench files, type the following commands:

```
vlib work ↵
vmap work work ↵
vlog -work work <output_netlist>.vo <testbench file>.v ↵
```

To load the design, type the following command:

```
vsim -t ps +transport_int_delays +transport_path_delays -L work -L \  
<lib1> -L <lib2> work.<testbench module name> ←
```

To add signals to the waveform viewer and run the simulation, type the following commands:

```
add wave * ←  
run ←
```

Example

```
# Create and compile Altera libraries  
  
vlib altera_ver  
vmap altera_ver altera_ver  
vlog -work altera_ver altera_primitives.v  
vlib stratixiii_ver  
  
vmap stratixiii_ver stratixiii_ver  
vlog -work stratixiii_ver stratixiii_atoms.v  
  
# Create work library and compile design files and testbench file  
  
vlib work  
vmap work work  
vlog -work work top_level.vo testbench.v  
  
# Load design  
  
vsim +transport_int_delays +transport_path_delays -L work -L  
altera_ver -L stratixiii_ver work.testbench  
  
#add signals to the waveform viwer and run simulation  
  
add wave *  
run
```

Performing Gate-Level Simulation

The steps for gate-level timing simulation are almost same as the steps for post-synthesis simulation.

The only difference is that the `.sdo` file must be back-annotated for gate level-timing simulation.

For Verilog HDL designs, the back-annotating process is done within the `output_netlist.vo` script. Therefore, you are not required to back-annotate the SDO file again.



You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.

Passing Parameter Information from Verilog to VHDL

You must use in-line parameters to pass values from Verilog HDL to VHDL. Using the `defparam` construct will cause an error in simulation. In the example below:

```
lpm_add_sub_component (  
  .dataa (dataa),  
  .datab (datab),  
  .result (sub_wire0)  
);  
defparam
```

```
lpm_add_sub_component.lpm_direction = "ADD",
lpm_add_sub_component.lpm_hint =
"ONE_INPUT_IS_CONSTANT=NO,CIN_USED=NO",
lpm_add_sub_component.lpm_type = "LPM_ADD_SUB",
lpm_add_sub_component.lpm_width = 12;
```

You will see the following error message:

```
# ** Error: (vsim-3043)
/apps2/home/users/bhlee/SPR_ADOQS/ADOQS10000935_IN_LINE_PARAMETER/lpm_
add_sub1.v(67): Unresolved reference to 'lpm_add_sub_component' in
lpm_add_sub_component.lpm_direction.
# Region: /IN_LINE_PARAMETER_vlg_vec_tst/il/b2v_inst
```

This megafunction instantiation has been modified to use in-line parameters:

```
lpm_add_sub#(12,"SIGNED","ADD",0,"LPM_ADD_SUB","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.

Speeding Up Simulation

By default, the ModelSim/QuartaSim software runs in a debug-optimized mode. To run the ModelSim/QuartaSim 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. This eliminates levels of debugging hierarchy, which may result in faster simulation. This switch is not supported in the ModelSim-Altera simulator.

Simulating Designs that Include Transceivers

If your design includes an Arria GX, Arria II GX, Cyclone IV, HardCopy IV, Stratix GX, Stratix II GX, Stratix IV, or Stratix V transceiver, you must compile additional library files to perform functional or gate-level timing simulations.



You cannot perform post-synthesis or post-fit (gate-level) simulation if you are targeting the Stratix V device family.




For Stratix V, you must compile the libraries listed in *Compiling Stratix V Libraries* in Quartus II Help.

Performing simulation with transceivers in Arria II, Cyclone IV, HardCopy IV, and Stratix IV device families are very similar. You have to replace only **stratixiigx_atoms** and **stratixiigx_hssi_atoms** model files with **arriaii_atoms** and **arriaii_hssi_atoms** model files for Arria II devices, **cycloneiv_atoms** and **cycloneiv_hssi_atoms** model files for Cyclone IV devices, and **stratixiv_atoms** and **stratixiv_hssi_atoms** model files for Stratix IV devices.

For high-speed simulation, you must select **ps** in the **Resolution** list for your simulator resolutions (**Design** tab of the **Start Simulation** dialog box). If you choose slower than **ps**, the high-speed simulation may fail.


 If your design contains PCI Express® hard IP, refer to the “Simulate the Design” section in the *PCI Express Compiler User Guide*.

 The VHDL version of ModelSim-Altera and other single language VHDL versions of ModelSim cannot simulate designs that target the Stratix V device family.

You need a version of ModelSim that supports VHDL/Verilog co-simulation to simulate designs that use Stratix V transceivers.

Functional Simulation for Stratix GX Devices

To perform a functional simulation of your design that instantiates the ALTGXB megafunction, which enables the gigabit transceiver block on Stratix GX devices, compile the **stratixgx_mf** model file into the **altgxb** library.

 The **stratixgx_mf** model file references the **lpm** and **sgate** libraries. If you are using ModelSim/QuestaSim, you must create these libraries to perform a simulation.

Performing Functional Simulation in VHDL (ModelSim-Altera)

To perform functional simulation for Stratix GX devices in VHDL, type the following commands:

```
vcom -work <my_design>.vhd <my_testbench>.vhd ↵  
vsim -L lpm -L altera_mf -L sgate -L altgxb work.<my_testbench> ↵
```

Performing Functional Simulation in VHDL (ModelSim/QuestaSim)

To perform functional simulation for Stratix GX devices in VHDL, type the following commands:

```
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd ↵  
vcom -work lpm 220pack.vhd 220model.vhd ↵  
vcom -work sgate sgate_pack.vhd sgate.vhd ↵  
vcom -work altgxb stratixgx_mf.vhd stratixgx_mf_components.vhd ↵  
vcom -work <my_design>.vhd <my_testbench>.vhd ↵  
vsim -L lpm -L altera_mf -L sgate -L altgxb work.<my_testbench> ↵
```

Performing Functional Simulation in Verilog HDL (ModelSim-Altera)

To perform functional simulation for Stratix GX devices in Verilog HDL, type the following commands:

```
vlog -work <my design>.v <my_testbench>.v ↵  
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L altgxb work.<my_testbench> ↵
```

Performing Functional Simulation in Verilog HDL (ModelSim/QuestaSim)

To perform functional simulation for Stratix GX devices in Verilog HDL, type the following commands:

```
vlib work_ver ↵  
vlib lpm_ver ↵  
vlib altera_mf_ver ↵  
vlib sgate_ver ↵  
vlib altgxb_ver ↵
```

```
vlog -work lpm_ver 220model.v ␣
vlog -work altera_mf_ver altera_mf.v ␣
vlog -work sgate_ver sgate.v ␣
vlog -work altgxb_ver stratixgx_mf.v ␣
vlog -work <my design>.v <my testbench>.v ␣
vsim -L lpm_ver -L sgate_ver -L altgxb_ver work.<my testbench> ␣
```

Gate-Level Timing Simulation for Stratix GX Devices

Perform a gate-level timing simulation of your design that includes a Stratix GX transceiver by compiling the `stratixgx_atoms` and `stratixgx_hssi_atoms` model files into the `stratixgx` and `stratixgx_gxb` libraries, respectively.



The `stratixgx_hssi_atoms` model file references the `lpm` and `sgate` libraries. If you are using ModelSim/ QuestaSim, you must create these libraries to perform a simulation.

Performing Gate-Level Timing Simulation in VHDL (ModelSim-Altera)

To perform gate-level timing simulation for Stratix GX devices in VHDL, type the following commands:

```
vcom -work <my design>.vho <my testbench>.vhd ␣
vsim -L lpm -L altera_mf -L sgate -L stratixgx -L stratixgx_gxb \
-sdftyp <design instance>=<path to .sdo file>.sdo work.<my testbench> \
-t ps - +transport_int_delays+transport_path_delays ␣
```

Performing Gate-Level Timing Simulation in VHDL (ModelSim/ QuestaSim)

To perform gate-level timing simulation for Stratix GX devices in VHDL, type the following commands:

```
vcom -work lpm 220pack.vhd 220model.vhd ␣
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd ␣
vcom -work sgate sgate_pack.vhd sgate.vhd ␣
vcom -work stratixgx stratixgx_atoms.vhd stratixgx_components.vhd ␣
vcom -work stratixgx_gxb stratixgx_hssi_atoms.vhd \
stratixgx_hssi_components.vhd ␣
vcom -work <my design>.vho <my testbench>.vhd ␣
vsim -L lpm -L altera_mf -L sgate -L stratixgx -L stratixgx_gxb \
-sdftyp <design instance>=<path to .sdo file>.sdo work.<my testbench> \
-t ps +transport_int_delays +transport_path_delays ␣
```

Performing Gate-Level Timing Simulation in Verilog HDL (ModelSim-Altera)

To perform gate-level timing simulation for Stratix GX devices in Verilog HDL, type the following commands:

```
vlog -work <my design>.vo <my testbench>.v ␣
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixgx_ver -L \
stratixgx_gxb_ver work.<my testbench> -t ps +transport_int_delays \
+transport_path_delays ␣
```

Performing Gate-Level Timing Simulation in Verilog HDL (ModelSim/QuestaSim)

To perform gate-level timing simulation for Stratix GX devices in Verilog HDL, type the following commands:

```
vlog -work lpm_ver 220model.v ←
vlog -work altera_mf_ver altera_mf.v ←
vlog -work sgate_ver sgate.v ←
vlog -work stratixgx_ver stratixgx_atoms.v ←
vlog -work stratixgx_gxb_ver stratixgx_hssi_atoms.v ←
vlog -work <my design>.vo <my testbench>.v ←
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixgx_ver \
-L stratixgx_gxb_ver work.<my testbench> -t ps +transport_int_delays \
+transport_path_delays ←
```

Functional Simulation for Stratix II GX Devices

To perform functional simulation of your design that instantiates the ALT2GXB megafunction, which enables the gigabit transceiver block on Stratix II GX devices, compile the **stratixiigx_hssi** model file into the **stratixiigx_hssi** library.



The **stratixiigx_hssi_atoms** model file references the **lpm** and **sgate** libraries. If you are using ModelSim/QuestaSim, you must create these libraries to perform a simulation.

Generate a functional simulation netlist by turning on **Generate Simulation Model** in the **Simulation Library** tab of the ALT2GXB MegaWizard Plug-In Manager. The **<alt2gxb entity name>.vho** or **<alt2gxb module name>.vo** is generated in the current project directory.



The ALT2GXB functional simulation library file generated by the Quartus II software references **stratixiigx_hssi** WYSIWYG atoms.

Performing Functional Simulation in VHDL (ModelSim-Altera)

To perform functional simulation for Stratix II GX devices in VHDL, type the following commands:

```
vcom -work work <alt2gxb entity name>.vho ←
vcom -work <my design>.vhd <my testbench>.vhd ←
vsim -L lpm -L altera_mf -L sgate -L stratixgx_hssi work.<my design> ←
```

Performing Functional Simulation in VHDL (ModelSim/QuestaSim)

To perform functional simulation for Stratix II GX devices in VHDL, type the following commands:

```
vcom -work lpm 220pack.vhd 220model.vhd ←
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd ←
vcom -work sgate sgate_pack.vhd sgate.vhd ←
vcom -work stratixiigx_hssi stratixiigx_hssi_components.vhd \
stratixiigx_hssi_atoms.vhd ←
vcom -work work <alt2gxb entity name>.vho ←
vcom -work <my design>.vhd <my testbench>.vhd ←
vsim -L lpm -L altera_mf -L sgate -L stratixgx_hssi work.<my testbench> ←
```

Performing Functional Simulation in Verilog HDL (ModelSim-Altera)

To perform functional simulation for Stratix II GX devices in Verilog HDL, type the following commands:

```
vlog -work work <alt2gxb module name>.vo ↵
vlog -work <my design>.v <my testbench>.v ↵
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixgx_hssi_ver \
work.<my testbench> ↵
```

Performing Functional Simulation in Verilog HDL (ModelSim/QuestaSim)

To perform functional simulation for Stratix II GX devices in Verilog HDL, type the following commands:

```
vlog -work lpm_ver 220model.v ↵
vlog -work altera_mf_ver altera_mf.v ↵
vlog -work sgate_ver sgate.v ↵
vlog -work stratixiigx_hssi_ver stratixiigx_hssi_atoms.v ↵
vlog -work work <alt2gxb module name>.vo ↵
vlog -work <my design>.v <my testbench>.v ↵
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixgx_hssi_ver \
work.<my testbench> ↵
```

Gate-Level Timing Simulation for Stratix II GX Devices

To perform a gate-level timing simulation of your design that includes a Stratix II GX transceiver, compile **stratixiigx_atoms** and **stratixiigx_hssi_atoms** into the **stratixiigx** and **stratixiigx_hssi** libraries, respectively.



The **stratixiigx_hssi_atoms** model file references the **lpm** and **sgate** libraries. If you are using ModelSim/QuestaSim, you must create these libraries to perform a simulation.

Performing Gate-Level Timing Simulation in VHDL (ModelSim-Altera)

To perform gate-level timing simulation for Stratix II GX devices in VHDL, type the following commands:

```
vcom -work <my design>.vho <my testbench>.vhd ↵
vsim -L lpm -L altera_mf -L sgate -L stratixiigx -L stratixiigx_hssi \
-sdftyp <design instance>=<path to .sdo file>.sdo work.<my testbench> \
-t ps +transport_int_delays +transport_path_delays ↵
```

Performing Gate-Level Timing Simulation in VHDL (ModelSim/QuestaSim)

To perform gate-level timing simulation for Stratix II GX devices in VHDL, type the following commands:

```
vcom -work lpm 220pack.vhd 220model.vhd ↵
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd ↵
vcom -work sgate sgate_pack.vhd sgate.vhd ↵
vcom -work stratixiigx stratixiigx_atoms.vhd \
stratixiigx_components.vhd ↵
vcom -work stratixiigx_hssi stratixiigx_hssi_components.vhd \
stratixiigx_hssi_atoms.vhd ↵
vcom -work <my design>.vho <my testbench>.vhd ↵
vsim -L lpm -L altera_mf -L sgate -L stratixiigx -L stratixiigx_hssi \
-sdftyp <design instance>=<path to .sdo file>.sdo work.<my testbench> \
-t ps +transport_int_delays +transport_path_delays ↵
```

Performing Gate-Level Timing Simulation in Verilog HDL (ModelSim-Altera)

To perform gate-level timing simulation for Stratix II GX devices in Verilog HDL, type the following commands:

```
vlog -work <my design>.vo <my testbench>.v ␣
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixiigx_ver \
-L stratixiigx_hssi_ver work.<my testbench> -t ps \
+transport_int_delays +transport_path_delays ␣
```

Performing Gate-Level Timing Simulation in Verilog HDL (ModelSim/QuestaSim)

To perform gate-level timing simulation for Stratix II GX devices in Verilog HDL, type the following commands:

```
vlog -work lpm_ver 220model.v ␣
vlog -work altera_mf_ver altera_mf.v ␣
vlog -work sgate_ver sgate.v ␣
vlog -work stratixiigx_ver stratixiigx_atoms.v ␣
vlog -work stratixiigx_hssi_ver stratixiigx_hssi_atoms.v ␣
vlog -work <my design>.vo <my testbench>.v ␣
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixiigx_ver \
-L stratixiigx_hssi_ver work.<my testbench> -t ps \
+transport_int_delays +transport_path_delays ␣
```

Functional Simulation for Stratix IV GX Devices

To perform a functional simulation of your design that instantiates the ALTGX megafunction, which enables the gigabit transceiver block on Stratix IV devices, compile the `stratixiv_hssi` model file into the `altgx` library.



The `stratixiv_hssi` model file references the `lpm` and `sgate` libraries. If you are using ModelSim/QuestaSim, you must create these libraries to perform a simulation.

Performing Functional Simulation in VHDL (ModelSim-Altera)

To perform functional simulation for Stratix IV devices in VHDL, type the following commands:

```
vcom -work <my design>.vhd <my testbench>.vhd ␣
vsim -L lpm -L altera_mf -L sgate -L stratixiv_hssi work.<my testbench>␣
```

Performing Functional Simulation in VHDL (ModelSim/QuestaSim)

To perform functional simulation for Stratix IV devices in VHDL, type the following commands:

```
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd ␣
vcom -work lpm 220pack.vhd 220model.vhd ␣
vcom -work sgate sgate_pack.vhd sgate.vhd ␣
vcom -work stratixiv_hssi \
stratixiv_hssi_atoms.vhd stratixiv_hssi_components.vhd ␣
vcom -work <my design>.vhd <my testbench>.vhd ␣
vsim -L lpm -L altera_mf -L sgate -L stratixiv_hssi work.<my testbench> ␣
```

Performing Functional Simulation in Verilog HDL (ModelSim-Altera)

To perform functional simulation for Stratix IV devices in Verilog HDL, type the following commands:

```
vlog -work <my design>.v <my testbench>.v ←
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L \
stratixiv_hssi_ver work.<my testbench> ←
```

Performing Functional Simulation in Verilog HDL (ModelSim/QuestaSim)

To perform functional simulation for Stratix IV devices in Verilog HDL, type the following commands:

```
vlog -work lpm_ver 220model.v ←
vlog -work altera_mf_ver altera_mf.v ←
vlog -work sgate_ver sgate.v ←
vlog -work stratixiv_hssi_ver stratixiv_hssi_atoms.v ←
vlog -work <my design>.v <my testbench>.v ←
vsim -L lpm_ver -L sgate_ver-L stratixiv_hssi_ver work.<my testbench> ←
```

Gate-Level Timing Simulation for Stratix IV GX Devices

Perform a gate-level timing simulation of your design that includes a Stratix IV transceiver by compiling the **stratixiv_atoms** and **stratixiv_hssi_atoms** model files into the **stratixiv** and **stratixiv_hssi** libraries, respectively.



The **stratixgx_hssi_atoms** model file references the **lpm** and **sgate** libraries. If you are using ModelSim/QuestaSim, you must create these libraries to perform a simulation.

Performing Gate-Level Timing Simulation in VHDL (ModelSim-Altera)

To perform gate-level timing simulation for Stratix IV devices in VHDL, type the following commands:

```
vcom -work <my design>.vho <my testbench>.vhd ←
vsim -L lpm -L altera_mf -L sgate -L stratixiv -L stratixiv_hssi \
-sdftyp <design instance>=<path to .sdo file>.sdo work.<my testbench> \
-t ps - +transport_int_delays +transport_path_delays ←
```

Performing Gate-Level Timing Simulation in VHDL (ModelSim/QuestaSim)

To perform gate-level timing simulation for Stratix IV devices in VHDL, type the following commands:

```
vcom -work lpm 220pack.vhd 220model.vhd ←
vcom -work altera_mf altera_mf_components.vhd altera_mf.vhd ←
vcom -work sgate sgate_pack.vhd sgate.vhd ←
vcom -work stratixiv stratixiv_atoms.vhd stratixiv_components.vhd ←
vcom -work stratixiv_hssi stratixiv_hssi_atoms.vhd \
stratixiv_hssi_components.vhd ←
vcom -work <my design>.vho <my testbench>.vhd ←
vsim -L lpm -L altera_mf -L sgate -L stratixiv -L stratixiv_hssi \
-sdftyp <design instance>=<path to .sdo file>.sdo work.<my testbench> \
-t ps +transport_int_delays +transport_path_delays ←
```

Performing Gate-Level Timing Simulation in Verilog HDL (ModelSim-Altera)

To perform gate-level timing simulation for Stratix IV devices in Verilog HDL, type the following commands:

```
vlog -work <my design>.vo <my testbench>.v ␣  
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixiv_ver -L \  
stratixiv_hssi_ver work.<my testbench> -t ps +transport_int_delays \  
+transport_path_delays ␣
```

Performing Gate-Level Timing Simulation in Verilog HDL (ModelSim/QuestaSim)

To perform gate-level timing simulation for Stratix IV devices in Verilog HDL, type the following commands:

```
vlog -work lpm_ver 220model.v ␣  
vlog -work altera_mf_ver altera_mf.v ␣  
vlog -work sgate_ver sgate.v ␣  
vlog -work stratixiv_ver stratixiv_atoms.v ␣  
vlog -work stratixiv_hssi_ver stratixiv_hssi_atoms.v ␣  
vlog -work <my design>.vo <my testbench>.v ␣  
vsim -L lpm_ver -L altera_mf_ver -L sgate_ver -L stratixiv_ver \  
-L stratixiv_hssi_ver work.<my testbench> -t ps +transport_int_delays \  
+transport_path_delays ␣
```

Functional Simulation for Stratix V GX Devices

To perform a functional simulation of your design that instantiates the Custom PHY megafunction, which enables the gigabit transceiver block on Stratix V devices, compile the `stratixv_hssi` model file.



The `stratixv_hssi` model file references the `lpm` and `sgate` libraries. You must create these libraries to perform a simulation using ModelSim/QuestaSim.



The transceiver module from the MegaWizard Plug-In Manager is created in **Interfaces/Transceiver PHY**. Select **Custom PHY**.

Performing Functional Simulation in VHDL (ModelSim/QuestaSim)

For information about how to perform functional simulation for Stratix V devices in VHDL, refer to “[Performing Functional Simulation](#)” on page 2–9.

Performing Functional Simulation in Verilog HDL (ModelSim-Altera)

For information about how to perform functional simulation for Stratix V devices in Verilog HDL, refer to *Performing a Functional Simulation with the ModelSim-Altera Software* and *Compiling Stratix V Libraries* in Quartus II Help.

Performing Functional Simulation in Verilog HDL (ModelSim/QuestaSim)

For information about how to perform functional simulation for Stratix V devices in Verilog HDL, refer to “[Performing Functional Simulation](#)” on page 2–13.

Transport Delays

By default, the ModelSim/QuestaSim software filters out all pulses that are shorter than the propagation delay between primitives. Turning on the transport delay options in the ModelSim/QuestaSim software prevents the simulation tool from filtering out these pulses. Use the following options to ensure that all signal pulses are seen in the simulation results.

+transport_path_delays

Use this option when the pulses in your simulation are shorter than the delay within a gate-level primitive.

+transport_int_delays

Use this option when the pulses in your simulation are shorter than the interconnect delay between gate-level primitives.



The **+transport_path_delays** and **+transport_int_delays** options are also used by default in the NativeLink feature for gate-level timing simulation.



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

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

```
vsim -t lps -L stratixii -sdftyp /il=filtref_vhd.sdo work.filtref_vhd_vec_tst \  
+transport_int_delays +transport_path_delays
```

Using the NativeLink Feature with ModelSim-Altera or ModelSim/QuestaSim Software

The NativeLink feature in the Quartus II software facilitates the seamless transfer of information between the Quartus II software and EDA tools and allows you to run ModelSim/QuestaSim within the Quartus II software.



For more information, refer to the “Using the NativeLink Feature” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

ModelSim/QuestaSim Error Message Verification

ModelSim/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/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 ←
```

At that point, 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 a Timing Value Change Dump (.vcd) File for the PowerPlay Power Analyzer

To generate a timing Value Change Dump (*.vcd) file for the PowerPlay Power Analyzer, you must first generate a *.vcd script file in the Quartus II software and run the *.vcd script file from the ModelSim/QuestaSim or ModelSim-Altera software to generate a timing *.vcd file. This timing *.vcd file can then be used by PowerPlay for power analysis. The following instructions show you step-by-step how to generate a timing *.vcd file.

To generate timing VCD Scripts in the Quartus II software, perform the following steps:


1. In the Quartus II software, on the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, click the "+" icon to expand **EDA Tool Settings** and select **Simulation**. The **Simulation** page appears.
3. Choose the appropriate third-party simulation tool (ModelSim/QuestaSim or ModelSim-Altera) in the **Tool name** list. Turn on the **Generate Value Change Dump (VCD) file script** option.
4. To generate the *.vcd script file, perform a full compilation.

To generate a timing *.vcd file in the ModelSim-Altera or ModelSim/QuestaSim software, perform the following steps:

1. In the ModelSim/QuestaSim or ModelSim-Altera software, before simulating your design, source the `<revision_name>_dump_all_vcd_nodes.tcl` script. To source the Tcl script, run the following command before running the `vsim` command. For example:

```
source <revision_name>_dump_all_vcd_nodes.tcl ←
```

2. Continue to run the simulation as usual until the end of the simulation. Exit the ModelSim/QuestaSim or ModelSim-Altera software. If you do not exit the software, the ModelSim/QuestaSim software may end the writing process of the timing *.vcd files improperly, resulting in a corrupted timing *.vcd file.

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

Viewing a Waveform from a .wlf File

A *.wlf file is automatically generated when your simulation is done. The *.wlf file is used for generating the waveform view through ModelSim-Altera or ModelSim/QuestaSim.

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

1. Type `vsim` on 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 file.
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 file in ModelSim-Altera or ModelSim/QuestaSim directly. The *.vcd file must first be converted to a *.wlf file.

8. Use the `vcd2wlf` command to convert the file. For example, type the following on a command-line:


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


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

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

Scripting Support

You can run procedures and create settings described in this chapter in a Tcl script. You can also run some procedures at the command line prompt.

 For more information about Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*.

 For more information about command line scripting, refer to the *Command Line Scripting* chapter in volume 2 of the *Quartus II Handbook*.

For detailed information about scripting command options, refer to the Quartus II Help command line and Tcl API help browser. To access this information, type the following command to start a help browser:

```
quartus_sh --qhelp ←
```

Generating a Post-Synthesis Simulation Netlist for ModelSim/QuestaSim

You can use the Quartus II software to generate a post-synthesis simulation netlist with Tcl commands or with a command at the command-line prompt. The following example assumes that you are selecting ModelSim/QuestaSim (Verilog HDL output from the Quartus II software).

Tcl Commands

Use the following Tcl commands to set the output format to Verilog HDL, the simulation tool to ModelSim/QuestaSim for Verilog HDL, and to generate a functional netlist:

```
set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim (Verilog)" ←  
set_global_assignment -name EDA_GENERATE_FUNCTIONAL_NETLIST ON ←
```

or

```
set_global_assignment -name EDA_SIMULATION_TOOL "QuestaSim (Verilog)" ←  
set_global_assignment -name EDA_GENERATE_FUNCTIONAL_NETLIST ON ←
```

Command Prompt

Use the following command to generate a simulation output file for the ModelSim/QuestaSim simulator. Specify VHDL or Verilog HDL for the format:

```
quartus_eda <project name> --simulation=on --format=<format> \  
--tool=ModelSim --functional ←
```

or

```
quartus_eda <project name> --simulation=on --format=<format> \  
--tool=QuestaSim --functional ←
```

Generating a Gate-Level Timing Simulation Netlist for ModelSim/QuestaSim

Use the Quartus II software to generate a gate-level timing simulation netlist with Tcl commands or with a command at the command prompt.

Tcl Commands

Use one of the following Tcl commands:

```
■ set_global_assignment -name EDA_SIMULATION_TOOL \  
"ModelSim-Altera (Verilog)" ←
```

or

```
set_global_assignment -name EDA_SIMULATION_TOOL \  
"QuestaSim-Altera (Verilog)" ←
```

```
■ set_global_assignment -name EDA_SIMULATION_TOOL \  
"ModelSim-Altera (VHDL)" ←
```

or

```
set_global_assignment -name EDA_SIMULATION_TOOL \
"QuestaSim-Altera (VHDL)" ←
```

```
■ set_global_assignment -name EDA_SIMULATION_TOOL \
"ModelSim (Verilog)" ←
```

or

```
set_global_assignment -name EDA_SIMULATION_TOOL \
"QuestaSim (Verilog)" ←
```

```
■ set_global_assignment -name EDA_SIMULATION_TOOL \
"ModelSim (VHDL)" ←
```

or

```
set_global_assignment -name EDA_SIMULATION_TOOL \
"QuestaSim (VHDL)" ←
```

Command Line

Generate a simulation output file for the ModelSim/QuestaSim simulator by specifying VHDL or Verilog HDL for the format by typing the following command at the command prompt:

```
quartus_eda <project name> --simulation=on --format=<format> \
--tool=ModelSim ←
```

or


```
quartus_eda <project name> --simulation=on --format=<format> \
--tool=QuestaSim ←
```

Software Licensing and Licensing Setup in ModelSim-Altera Subscription Edition

License the ModelSim-Altera Subscription Edition software subscription with a parallel port FIXEDPC license, or a network FLOATNET or FLOATPC license. Each Altera software subscription includes a license for both VHDL and Verilog HDL. The ModelSim-Altera Subscription Edition software supports both VHDL and Verilog HDL, but the software does not support mixed language simulation.

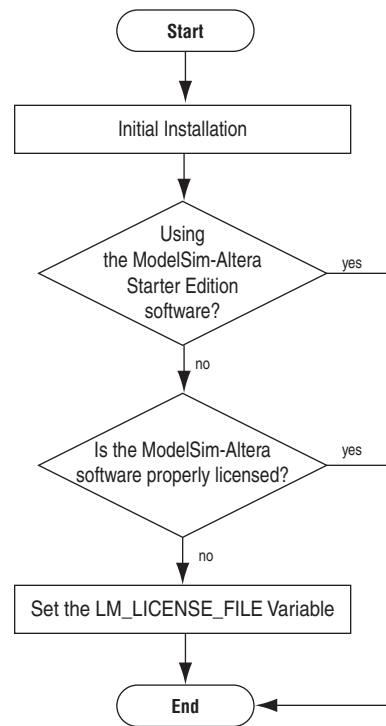
 The USB software guard is not supported by versions earlier than Mentor Graphics ModelSim software 5.8d.

You can obtain a license for the ModelSim-Altera Subscription Edition software from the Altera website at www.altera.com. Get licensing information for the Mentor Graphics ModelSim software directly from Mentor Graphics. Refer to [Figure 2-1](#) for the set-up process.

 For ModelSim-Altera software versions prior to 5.5b, use the **PCLS** utility included with the software to set up the license.

For the Quartus II software version 8.1 and later, the no-cost entry level of the ModelSim-Altera software does not require a license file. However, you must request a license file to use the ModelSim-Altera Subscription Edition software.

Figure 2-1. ModelSim-Altera Subscription Edition Software Licensing Set Up Process



LM_LICENSE_FILE Variable

Altera recommends setting the `LM_LICENSE_FILE` environment variable to the location of the license file. For example, the value for the `LM_LICENSE_FILE` environment variable should point to `<path to license file>\license.dat`.



For more information about setting up the license for ModelSim-Altera Subscription Edition software, refer to the [Altera Software Installation and Licensing](#) manual.

Conclusion

Using the ModelSim/QuestaSim and ModelSim-Altera simulation software within the Altera FPGA design flow enables Altera software users to easily and accurately perform functional simulations, post-synthesis simulations, and gate-level simulations on their designs. Proper verification of designs at the functional, post-synthesis, and post place-and-route stages using the ModelSim/QuestaSim and ModelSim-Altera software helps ensure design functionality and success and, ultimately, a quick time-to-market.

Document Revision History


Table 2-1 shows the revision history for this chapter.


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

Date	Version	Changes Made
July 2010	10.0.0	<ul style="list-style-type: none"> ■ 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
November 2009	9.1.0	<ul style="list-style-type: none"> ■ Removed NativeLink information and referenced new <i>Simulating Designs with EDA Tools</i> chapter ■ Added Stratix IV transceiver simulation section ■ Reformatted transceiver simulation sections ■ Text edits throughout chapter
March 2009	9.0.0	<p>Added the following sections:</p> <ul style="list-style-type: none"> ■ “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 <p>Updated the following:</p> <ul style="list-style-type: none"> ■ 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	<p>Updated the following:</p> <ul style="list-style-type: none"> ■ Table 2-2, Table 2-3, Table 2-4, Table 2-5, Table 2-6 ■ Removed <code>--zero_ic_delays</code> from <code>quartus_sta</code> 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

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

Date	Version	Changes Made
May 2008	8.0.0	Updated the following: <ul style="list-style-type: none">■ “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

 For previous versions of the *Quartus II Handbook*, refer to the [Quartus II Handbook Archive](#).

 Take an [online survey](#) to provide feedback about this handbook chapter.

