

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

### Introduction

An Altera® Quartus® II software subscription includes a no-cost entry-level version of the ModelSim-Altera software on a PC or UNIX platform. Altera also offers the ModelSim-Altera Subscription Edition software with full support for Altera devices. You can use the ModelSim-Altera Starter Edition software to perform register transfer level (RTL) functional, post-synthesis, and gate-level timing simulations for either Verilog HDL or VHDL designs that target an Altera FPGA.

This chapter provides details about the specific libraries that are needed for an RTL functional, post-synthesis, and gate-level timing simulation.

This chapter includes the following topics:

- “Software Compatibility” on page 3-3
- “Altera Design Flow with ModelSim-Altera or ModelSim Software” on page 3-3
- “Simulation Libraries” on page 3-3
- “Performing Simulation Using the ModelSim-Altera Software” on page 3-7
- “Performing Simulation Using the ModelSim Software” on page 3-26
- “Simulating Designs that Include Transceivers” on page 3-53
- “Using the NativeLink Feature with ModelSim-Altera or ModelSim Software” on page 3-61
- “Generating a Timing Value Change Dump (.vcd) File for the PowerPlay Power Analyzer” on page 3-61
- “Viewing a Waveform from a .wlf File” on page 3-62
- “Scripting Support” on page 3-63
- “Software Licensing and Licensing Setup in ModelSim-Altera Subscription Edition” on page 3-64



For more information about the current Quartus II software version, refer to the Altera website at [www.altera.com](http://www.altera.com).

ModelSim-Altera software is included with your Altera software subscription and can be licensed for PC, UNIX, or Linux platforms to support either Verilog HDL or VHDL simulation. ModelSim-Altera software supports RTL functional, post-synthesis, and gate-level timing simulations for all Altera devices.

Table 3-1 lists the differences between the Mentor Graphics ModelSim SE/PE and ModelSim-Altera software versions.

**Table 3-1.** Comparison of ModelSim Software Versions

Product Feature	ModelSim SE	ModelSim PE	ModelSim-Altera	ModelSim-Altera Starter Edition
100% VHDL, Verilog HDL, mixed-HDL support	Optional	Optional	Supports only single-HDL simulation	Supports only single-HDL simulation
Complete HDL debugging environment	✓	✓	✓	✓
Optimized direct compile architecture	✓	✓	✓	✓
Industry-standard scripting	✓	✓	✓	✓
Flexible licensing	✓	Optional	✓	—
Verilog HDL PLI support. Interfaces Verilog HDL designs to customer C code and third-party software	✓	✓	✓	✓
VHDL FLI support. Interfaces VHDL designs to customer C code and third-party software	✓	—	—	—
Standard Delay Output Format File annotation	✓	✓	✓ <sup>(1)</sup>	✓ <sup>(1)</sup>
Advanced debugging features and language-neutral licensing	✓	—	—	—
Customizable, user-expandable GUI and integrated simulation performance analyzer	✓	—	—	—
Integrated code coverage analysis and SWIFT support	✓	—	—	—
Accelerated VITAL and Verilog HDL primitives (3 times faster), and register transfer level (RTL) acceleration (5 times faster)	✓	—	—	—
Platform support	PC, UNIX, Linux	PC only	PC, UNIX, Linux	PC, UNIX, Linux
Precompiled libraries	No	No	Yes	Yes

**Note to Table 3-1:**

(1) ModelSim-Altera software only allows SDF annotation to modules in the Altera library.

## Software Compatibility

Table 3–2 shows which ModelSim-Altera and ModelSim software version is compatible with the Quartus II software versions. ModelSim versions provided directly from Mentor Graphics do not correspond to specific Quartus II software versions.

For help with the ModelSim-Altera licensing setup, refer to “Software Licensing and Licensing Setup in ModelSim-Altera Subscription Edition” on page 3–64.

**Table 3–2.** Compatibility Between Software Versions

ModelSim-Altera Software	Quartus II Software (1)
ModelSim-Altera and ModelSim software version 6.5b	Quartus II software version 9.1
ModelSim-Altera and ModelSim software version 6.4a	Quartus II software version 9.0
ModelSim-Altera 6.3g_p1 and ModelSim software version 6.3g	Quartus II software version 8.1
ModelSim-Altera and ModelSim software version 6.1g	Quartus II software version 6.1, 7.0, 7.1, 7.2, and 8.0
ModelSim-Altera and ModelSim software version 6.1d	Quartus II software version 6.0
ModelSim-Altera and ModelSim software version 6.0e	Quartus II software version 5.1
ModelSim-Altera and ModelSim software version 6.0c	Quartus II software version 5.0
ModelSim-Altera and ModelSim software version 5.8.e	Quartus II software version 4.2
ModelSim-Altera and ModelSim software version 5.8	

**Note to Table 3–2:**

(1) Updated ModelSim-Altera precompiled libraries are available for download on Altera’s website for each release of the Quartus II service pack.

## Altera Design Flow with ModelSim-Altera or ModelSim Software

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

- RTL functional simulation
- Post-synthesis simulation
- Gate-level timing simulation

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*.

## Simulation Libraries

Simulation model libraries are required to run a simulation whether you are running an RTL functional simulation, post-synthesis simulation, or gate-level timing simulation. In general, running an RTL functional simulation requires the RTL functional simulation model libraries, while running a post-synthesis or gate-level timing simulation requires the gate-level timing simulation model libraries. You must compile the necessary library files before you can run the simulation.

There are a few exceptions where you must compile gate-level timing simulation library files to perform RTL functional simulation. For example, the following list shows some of the Altera megafunctions' gate-level libraries required to perform an RTL functional simulation using third-party simulators:

- ALTCLKBUF
- ALTCLKCTRL
- ALTDQS
- ALTDQ
- ALTDDIO\_IN
- ALTDDIO\_OUT
- ALTDDIO\_BIDIR
- ALTUFM\_NONE
- ALTUFM\_PARALLEL
- ALTUFM\_SPI
- ALTMEMMULT
- ALTREMOTE\_UPDATE



To identify which type of simulation libraries are required to run the simulation for a specified Altera megafunction, refer to the last page in the Altera megafunction MegaWizard™ Plug-In Manager. This page explains which simulation library files are required to perform an RTL functional simulation for that particular megafunction.

Simulation of the transceiver megafunction (for example, ALT2GXB) is another exception that requires the gate-level libraries to perform RTL functional simulation and vice versa.

For detailed, step-by-step instructions about how to simulate the transceiver megafunction, refer to [“Simulating Designs that Include Transceivers”](#) on page 3-53.

## Precompiled Simulation Libraries in the ModelSim-Altera Software

Precompiled libraries for both functional and gate-level simulations are available in the ModelSim-Altera software. You do not have to explicitly compile these library files before running the 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.

### RTL Functional Simulation Libraries

RTL functional simulation libraries include the LPM simulation model, Altera megafunction simulation model, and the low-level primitive simulation model. Table 3-3 shows the precompiled library name and the location in the ModelSim-Altera software for all RTL functional simulation models.

**Table 3-3.** Precompiled RTL Functional Simulation Libraries in the ModelSim-Altera Software

RTL Simulation Model	Precompiled Library Name (Verilog)	Location in ModelSim-Altera (Verilog)	Precompiled Library Name (VHDL)	Location in ModelSim-Altera (VHDL)
LPM	lpm_ver	<compiled>\verilog\220model	lpm	<compiled>\vhd\220model
Altera Megafunction	altera_mf_ver	<compiled>\verilog\altera_mf	altera_mf	<compiled>\vhd\altera_mf
Low-Level Primitives	altera_ver	<compiled>\verilog\altera	altera	<compiled>\vhd\altera
ALTGXB Megafunction (Stratix GX)	altgxb_ver	<compiled>\verilog\altgxb	altgxb	<compiled>\vhd\altgxb
High-Level Primitives	sgate_ver	<compiled>\verilog\sgate	sgate	<compiled>\vhd\sgate
Low-Level Primitives	alt_vtl_ver	<compiled>\verilog\alt_vtl	alt_vtl	<compiled>\vhd\alt_vtl

 <compiled> refers to <ModelSim-Altera installation directory>\altera

### Gate-Level Simulation Libraries

Gate-level simulation libraries include the supported Altera device atom simulation models. Table 3-4 shows the precompiled library name and location in the ModelSim-Altera software for all gate-level simulation models.

**Table 3-4.** Precompiled Gate-Level Simulation Libraries in the ModelSim-Altera Software (Part 1 of 2)

Device Simulation Model	Precompiled Library Name (Verilog)	Location in ModelSim-Altera (Verilog)	Precompiled Library Name (VHDL)	Location in ModelSim-Altera (VHDL)
Arria® II (without transceiver block)	arriaii_ver	<compiled>\verilog\arriaii	arriaii	<compiled>\vhd\arriaii
Arria II GX (with transceiver block)	arriaii_hssi_ver	<compiled>\verilog\arriaii_hssi	arriaii_hssi	<compiled>\vhd\arriaii_hssi
Arria II (with PCI Express)	arriaii_pcie_hip_ver	<compiled>\verilog\arriaii_pcie_hip	arriaii_pcie_hip	<compiled>\vhd\arriaii_pcie_hip
Arria GX (without transceiver block)	arriagx_ver	<compiled>\verilog\arriagx	arriagx	<compiled>\vhd\arriagx
Arria GX (with transceiver block)	arriagx_hssi_ver	<compiled>\verilog\arriagx_hssi	arriagx_hssi	<compiled>\vhd\arriagx_hssi
Stratix® IV	stratixiv_ver	<compiled>\verilog\stratixiv	stratixiv	<compiled>\vhd\stratixiv
Stratix IV (with transceiver block)	stratixiv_hssi_ver	<compiled>\verilog\stratixiv_hssi	stratixiv_hssi	<compiled>\vhd\stratixiv_hssi
Stratix IV (with PCI Express)	stratixiv_pcie_hip_ver	<compiled>\verilog\stratixiv_pcie_hip	stratixiv_pcie_hip	<compiled>\vhd\stratixiv_pcie_hip
Stratix III	stratixiii_ver	<compiled>\verilog\stratixiii	stratixiii	<compiled>\vhd\stratixiii
Stratix II	stratixii_ver	<compiled>\verilog\stratixii	stratixii	<compiled>\vhd\stratixii
Stratix II GX (without transceiver block)	stratixiigx_ver	<compiled>\verilog\stratixiigx	stratixiigx	<compiled>\vhd\stratixiigx

**Table 3-4.** Precompiled Gate-Level Simulation Libraries in the ModelSim-Altera Software (Part 2 of 2)

Device Simulation Model	Precompiled Library Name (Verilog)	Location in ModelSim-Altera (Verilog)	Precompiled Library Name (VHDL)	Location in ModelSim-Altera (VHDL)
Stratix II GX (with transceiver block)	stratixiigx_hssi_ver	<compiled>\verilog\stratixiigx_hssi	stratixiigx_hssi	<compiled>\vhd\stratixiigx_hssi
Stratix	stratix_ver	<compiled>\verilog\stratix	stratix	<compiled>\vhd\stratix
Stratix GX (without transceiver block)	stratixgx_ver	<compiled>\verilog\stratixgx	stratixgx	<compiled>\vhd\stratixgx
Stratix GX (with transceiver block)	stratixgx_gxb_ver	<compiled>\verilog\stratixgx_gxb	stratixgx_gxb	<compiled>\vhd\stratixgx_gxb
HardCopy® IV	hardcopyiv_ver	<compiled>\verilog\hardcopyiv	hardcopyiv	<compiled>\vhd\hardcopyiv
HardCopy III	hardcopyiii_ver	<compiled>\verilog\hardcopyiii	hardcopyiii	<compiled>\verilog\hardcopyiii
HardCopy II	hardcopyii_ver	<compiled>\verilog\hardcopyii	hardcopyii	<compiled>\vhd\hardcopyii
Cyclone® III LS	cycloneiii_ls_ver	<compiled>\verilog\cycloneiii_ls	cycloneiii_ls	<compiled>\vhd\cycloneiii_ls
Cyclone IV (without transceiver block)	cycloneiv_ver	<compiled>\verilog\cycloneiv	cycloneiv	<compiled>\vhd\cycloneiv
Cyclone IV (with transceiver block)	cycloneiv_hssi_ver	<compiled>\verilog\cycloneiv_hssi	cycloneiv_hssi	<compiled>\vhd\cycloneiv_hssi
Cyclone IV (with PCI Express)	cycloneiv_pcie_hip_ver	<compiled>\verilog\cycloneiv_pcie_hip	cycloneiv_pcie_hip	<compiled>\vhd\cycloneiv_pcie_hip
Cyclone III	cycloneiii_ver	<compiled>\verilog\cycloneiii	cycloneiii	<compiled>\vhd\cycloneiii
Cyclone II	cycloneii_ver	<compiled>\verilog\cycloneii	cycloneii	<compiled>\vhd\cycloneii
Cyclone	cyclone_ver	<compiled>\verilog\cyclone	cyclone	<compiled>\vhd\cyclone
MAX® II	maxii_ver	<compiled>\verilog\maxii	maxii	<compiled>\vhd\maxii
MAX 7000 MAX 3000	max_ver	<compiled>\verilog\max	max	<compiled>\vhd\max

## Simulation Library Files in the Quartus II Software

In ModelSim SE/PE, no precompiled libraries are available. You must compile the necessary libraries to perform RTL functional or gate-level simulation. For information about the required libraries, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

## 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>
```

## Compiling Libraries Using the EDA Simulation Library Compiler

The EDA Simulation Library Compiler can compile Verilog HDL and VHDL simulation libraries for all Altera devices and supported EDA simulators. You can use this tool to compile all libraries required for RTL and gate-level simulation.



For more information about this tool, refer to the “EDA Simulation Library Compiler” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

## Performing Simulation Using the ModelSim-Altera Software

Simulation of Verilog HDL or VHDL designs with the ModelSim-Altera software can be performed at various levels to verify designs from different aspects. Simulation is divided into three categories: RTL functional simulation, post-synthesis simulation, and gate-level timing simulation. Simulation helps you verify your designs and debug them.

The following sections provide step-by-step instructions for performing the simulation through the GUI and from the command line.

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 the VHDL Designs Using the GUI

Simulating the VHDL design using the ModelSim-Altera GUI is user-friendly. You do not have to remember the commands to compile the libraries, or load and simulate the VHDL design files. You can use the ModelSim-Altera GUI to perform RTL functional simulation, post-synthesis simulation, and gate-level timing simulation. The following sections show how to perform simulation at various levels through the ModelSim-Altera GUI.

#### Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in the ModelSim-Altera software for VHDL designs.



The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

#### Compiling Testbench and Design Files into the Work Library

The following instructions show you how to compile your testbench and design files into the work library using the ModelSim-Altera GUI.

1. Browse to locate your designs.
2. Click **OK**.

To create the work library, perform the following steps:

1. In the ModelSim-Altera software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.

2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name `Work` in the text box.
4. Click **OK**.

To compile the testbench and design files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The design files and testbench file should be compiled into the **Work** library.
3. Select the design files and testbench file, and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. For VHDL designs, if you have not included the mapped name of the libraries in your design files or subfiles, perform the following steps:
  - a. Click the **Libraries** tab.
  - b. In the **Search Libraries** text box, click the **Add** button.
  - c. Browse to the required precompiled library in the ModelSim-Altera software. You can either browse to locate the path `<ModelSim-Altera installation directory>/altera/vhdl/<precompiled library>` or you can just click the arrow button to select the `<precompiled library mapped name>`.  
  
Examples of `<precompiled library>` or `<precompiled library mapped name>` are **altera\_mf** and **lpm**. The functional RTL simulation libraries are required for performing RTL functional simulation. For the complete set of libraries, refer to [“Precompiled Simulation Libraries in the ModelSim-Altera Software” on page 3-4](#).
- d. Click **OK** to add the libraries to the **Search Libraries** text box.
7. Click **OK**.


### Running the Simulation

To run a simulation, perform the following steps:


1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

### 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 following sections help you perform a post-synthesis simulation for a VHDL design in the ModelSim-Altera software.

 The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and VHDL Output File into the Work Library

The following instructions show you how to compile your testbench and \*.vho file into the work library using the ModelSim-Altera GUI.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your testbench or \*.vho file. By default, the \*.vho file is located in *<project directory>/simulation/modelsim*.
2. Click **OK**.

To create the work library, perform the following steps:

1. In the ModelSim-Altera software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name **Work** in the text box.
4. Click **OK**.

To compile the testbench and VHDL output (\*.vho) files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.

2. Select the library **Work**. The testbench and VHDL output (\*.vho) files should be compiled into the **Work** library.
3. Select the testbench and \*.vho design files and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
  2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
  3. In the **Library** list, select and expand the **Work** library.
  4. Select the top-level design unit (your testbench).
  5. In the **Resolution** list, select **ps**.
  6. For VHDL designs, if you have not included the mapped name of the libraries in your design files or subfiles, perform the following steps:
    - a. Click the **Libraries** tab.
    - b. In the **Search Libraries** text box, click the **Add** button.
    - c. Browse to the required precompiled library in the ModelSim-Altera software. You can either browse to locate the path *<ModelSim-Altera installation directory>/altera/vhdl/<precompiled library>* or you can just click the arrow button to select the *<precompiled library mapped name>*.  
  
Examples of *<precompiled library>* or *<precompiled library mapped name>* are **stratixiii** and **cycloneiii**. The gate-level simulation libraries are required for performing post-synthesis simulation. For the complete set of libraries, refer to ["Precompiled Simulation Libraries in the ModelSim-Altera Software"](#) on page 3-4.
  - d. Click **OK** to add the libraries to the **Search Libraries** text box.
7. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

## Performing Gate-Level Simulation

Gate-level 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 simulation with the ModelSim-Altera software.



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*.

The following sections help you perform a gate-level simulation for a VHDL design in the ModelSim-Altera software.



The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and VHDL Output File into the Work Library

The following instructions show how you can compile your testbench and \*.vho file into the work library using the ModelSim-Altera GUI.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your testbench or \*.vho file. By default, the \*.vho file is located in *<project directory>/simulation/modelsim*.
2. Click **OK**.

To create the work library, perform the following steps:

1. In the ModelSim-Altera software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name `work` in the text box.
4. Click **OK**.

To compile the testbench and VHDL output (\*.vho) files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and VHDL output (\*.vho) files should be compiled into the **Work** library.
3. Select the testbench and VHDL output (\*.vho) design files, and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. Click the **SDF** tab, and click **Add**. The **Add SDF Entry** dialog box appears.
3. In the **Add SDF Entry** dialog box, click **Browse** and select the \*.sdo file. By default, the \*.sdo file is located in *<project directory>/simulation/modelsim*.
4. In the **Apply to Region** dialog box, type the instance path to which the \*.sdo file is to be applied. For example, if you are using a testbench exported into the Quartus II software from a Vector Waveform File, the instance path is set to */i1*.



You do not have to choose from the **Delay** list because the Quartus II EDA Netlist Writer generates the \*.sdo file using the same value for the triplet (minimum, typical, and maximum timing values).

5. Click **OK**.
6. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
7. In the **Library** list, select and expand the **Work** library.
8. Select the top-level design unit (your testbench).
9. In the **Resolution** list, select **ps**.
10. For VHDL designs, if you have not included the mapped name of the libraries in your design files or subfiles, perform the following steps:
  - a. Click the **Libraries** tab.
  - b. In the **Search Libraries** text box, click the **Add** button.
  - c. Browse to the required precompiled library in the ModelSim-Altera software. You can either browse to locate the path *<ModelSim-Altera installation directory>/altera/vhdl/<precompiled library>* or you can just click the arrow button to select the *<precompiled library mapped name>*.  
  
Examples of *<precompiled library>* or *<precompiled library mapped name>* are **stratixiii** and **cycloneiii**. The gate-level simulation libraries are required for performing gate-level simulation. For the complete set of libraries, refer to [“Precompiled Simulation Libraries in the ModelSim-Altera Software” on page 3–4](#).
  - d. Click **OK** to add the libraries to the **Search Libraries** text box.
11. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.

3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

## Simulating Verilog HDL Designs through the GUI

You do not have to remember the commands to compile the libraries or load and simulate the Verilog HDL design files. You can use the ModelSim-Altera GUI to perform RTL functional simulation, post-synthesis simulation, and gate-level timing simulation. The following sections show how to perform simulation at various levels through the ModelSim-Altera GUI.

### Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in the ModelSim-Altera software for Verilog HDL designs.



The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and Design Files into the Work Library

The following instructions show you how to compile your testbench and design files into the work library using the ModelSim-Altera GUI.

To change to the design directory, perform the following steps:

1. Browse to locate your designs.
2. Click **OK**.

To create the work library, perform the following steps:

1. In the ModelSim-Altera software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name `work` in the text box.
4. Click **OK**.

To compile the testbench and design files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The design files and testbench file should be compiled into the **Work** library.
3. Select the design files and testbench file and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. Click the **Libraries** tab.
7. In the **Search Libraries** text box, click the **Add** button.
8. Browse to the required precompiled library in the ModelSim-Altera software. You can either browse to locate the path *<ModelSim-Altera installation directory>/altera/verilog/<precompiled library>* or you can just click the arrow button to select the *<precompiled library mapped name>*.

Examples of *<precompiled library>* or *<precompiled library mapped name>* are **altera\_mf\_ver** and **lpm\_ver**. The RTL simulation libraries are required for performing RTL functional simulation. For the complete set of libraries, refer to [“Precompiled Simulation Libraries in the ModelSim-Altera Software” on page 3–4](#).

9. Click **OK** to add the libraries to the **Search Libraries** text box.
10. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

### 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 following sections help you perform a post-synthesis simulation for a Verilog HDL design in the ModelSim-Altera software.



The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and Verilog HDL Output File into the Work Library

The following instructions show you how to compile your testbench and Verilog HDL output file (\*.vo) into the work library using the ModelSim-Altera GUI.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your testbench or \*.vo file. By default, the \*.vho file is located in *<project directory>/simulation/modelsim*.
2. Click **OK**.

To create the work library, perform the following steps:

1. In the ModelSim-Altera software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name **work** in the text box.
4. Click **OK**.

To compile the testbench and Verilog HDL output (\*.vo) files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and Verilog HDL output (\*.vo) files should be compiled into the **Work** library.
3. Select the testbench and \*.vo design files, and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. Click the **Libraries** tab.
7. In the **Search Libraries** text box, click the **Add** button.

- Browse to the required precompiled library in the ModelSim-Altera software. You can either browse to locate the path `<ModelSim-Altera installation directory>/altera/verilog/<precompiled library>` or you can just click the arrow button to select the `<precompiled library mapped name>`.

Examples of `<precompiled library>` or `<precompiled library mapped name>` are `stratixiii_ver` and `cycloneiii_ver`. The gate-level simulation libraries are required for performing post-synthesis simulation. For the complete set of libraries, refer to “[Precompiled Simulation Libraries in the ModelSim-Altera Software](#)” on page 3-4.

- Click **OK** to add the libraries to the **Search Libraries** text box.
- Click **OK**.


### Running the Simulation

To run a simulation, perform the following steps:


- On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
- On the View menu, point to **Debug Windows** and click **Wave**.
- Drag signals to monitor from the Objects window and drop them into the Wave window.
- On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

### 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-Altera software.

 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” in the *Simulating Designs with EDA Tools* section chapter in volume 3 of the *Quartus II Handbook*.

The following sections help you perform a gate-level simulation for a Verilog HDL design in the ModelSim-Altera software.

 The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and Verilog HDL Output File into the Work Library

The following instructions show you how to compile your testbench and \*.vo file into the work library using the ModelSim-Altera GUI.

To change to the simulation output directory, perform the following steps:

- Browse to locate your testbench or \*.vo file. By default, the \*.vo file is located in `<project directory>/simulation/modelsim`.
- Click **OK**.

To create the work library, perform the following steps:

1. In the ModelSim-Altera software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name `work` in the text box.
4. Click **OK**.

To compile the testbench and `*.vo` files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and `*.vo` files should be compiled into the **Work** library.
3. Select the testbench and `*.vo` design files, and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

When simulating in Verilog HDL, you do not have to specify the Standard Delay Output Format File (`*.sdo`) file. In the `$sdf_annotate` task, when the Quartus II software generates the `*.vo` file, the ModelSim-Altera software looks for the `*.sdo` file in the directory in which the VSIM was run. If your `*.sdo` file is not in this directory, copy the `*.sdo` file into your current directory.

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. Click the **Libraries** tab.
7. In the **Search Libraries** text box, click the **Add** button.
8. Browse to the required precompiled library in the ModelSim-Altera software. You can either browse to locate the path `<ModelSim-Altera installation directory>/altera/verilog/<precompiled library>` or you can just click the arrow button to select the `<precompiled library mapped name>`.

Examples of `<precompiled library>` are `altera_mf_ver` and `lpm_ver`. The gate-level simulation libraries are required for performing gate-level simulation. For the complete set of libraries, refer to [“Precompiled Simulation Libraries in the ModelSim-Altera Software” on page 3-4](#).

9. Click **OK** to add the libraries to the **Search Libraries** text box.

10. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

## Simulating the VHDL Designs from the Command Line

Simulating VHDL designs from the ModelSim-Altera command line gives you more flexibility and control to compile the libraries, and load and simulate the VHDL design files. All simulation commands are Tcl commands that can be included in the ModelSim Macro File (\*.do). Using the \*.do file allows you to run simulation in batch mode. You have to execute only the \*.do file, and the ModelSim-Altera tool automatically executes all commands in the \*.do script macro file.

You can use the ModelSim-Altera command line to perform RTL functional simulation, post-synthesis simulation, and gate-level simulation. The following sections show how to perform simulation at various levels from the ModelSim-Altera command line.

### Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in ModelSim-Altera for VHDL designs.



The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and Design Files into the Work Library

Use the following commands to compile your testbench and design files into the work library in the ModelSim-Altera software.

To change to the design library, type the following command:


```
cd <your_design_directory> ←  
(for example, cd:/designs)
```

To create the work library, type the following commands:

```
vlib work ←  
vmap work work ←
```

To compile the testbench and design files into the work library, type the following command:

```
vcom -work work <my_testbench.vhd> <my_design_files.vhd> ←
```

 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:

```
vsim -t ps -L <precompiled-library1> -L <precompiled-library2> work.<my_testbench> ←
```

The *<precompiled-library1>* and *<precompiled-library2>* variables are the libraries required to compile your testbench. If you have multiple libraries, use the `-L` option for each library in the `vsim` command.

Examples of *<precompiled library>* are `altera_mf` and `lpm`. The functional RTL simulation libraries are required for performing RTL functional simulation. For the complete set of libraries, refer to “[Precompiled Simulation Libraries in the ModelSim-Altera Software](#)” on page 3–4.

You can choose not to invoke `-L` in the `vsim` command for VHDL designs if you have already included the mapped name of the libraries in your design files or subfiles.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←  
run <time period> ←
```

For example, to add all signals in your testbench hierarchy, type the following command:


```
add wave * ←
```

To run the simulation for 100 ps, type the following command:


```
run 100 ps ←
```

### 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 following sections help you perform a post-synthesis simulation for a VHDL design in the ModelSim-Altera software.

 The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and VHDL Output File into the Work Library

Use the following commands to compile your testbench and `*.vho` file into the work library in the ModelSim-Altera software.

To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ↵  
(for example, cd:/designs/modelsim/simulation)
```


 This directory contains the \*.vho file, which is generated by the netlist writer.

To create the work library, type the following commands:

```
vlib work ↵  
vmap work work ↵
```

To compile the testbench and \*.vho files into the work library, type the following command:

```
vcom -work work <my_testbench.vhd> <my_design_netlists.vho> ↵
```

 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:

```
vsim -t ps -L <precompiled-library1> -L <precompiled-library2> work.<my_testbench> ↵
```

The <precompiled-library1> and <precompiled-library2> variables are the libraries required to compile your testbench. If you have multiple libraries, use the -L option for each library in the vsim command.


Examples of <precompiled library> are **stratixiii** and **cycloneiii**. The gate-level simulation libraries are required for performing post-synthesis simulation. For the complete set of libraries, refer to “[Precompiled Simulation Libraries in the ModelSim-Altera Software](#)” on page 3–4.

You can choose not to invoke -L in the vsim command for VHDL designs if you have already included the mapped name of the libraries in your design files or subfiles.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ↵  
run <time period> ↵
```

 To add all signals in your testbench hierarchy, type the following command:


```
add wave * ↵
```

To run the simulation for 100 ps, type the following command:


```
run 100 ps ↵
```

### Performing Gate-Level Simulation

Gate-level simulation is a very important step in ensuring that the FPGA device 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-Altera software.

-  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*.

The following sections help you perform a gate-level simulation for a VHDL design in the ModelSim-Altera software.

-  The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and VHDL Output File into the Work Library

Use the following commands to compile your testbench and \*.vho file into the work library in the ModelSim-Altera software.

To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ←  
(for example, cd:/designs/modelsim/simulation)
```


-  This directory contains the \*.vho file, which is generated by the netlist writer.

To create the work library, type the following commands:

```
vlib work ←  
vmap work work ←
```

To compile the testbench and \*.vho files into the work library, type the following command:

```
vcom -work work <my_testbench.vhd> <my_design_netlists.vho> ←
```

-  Resolve compile-time errors before proceeding to the next section.

### Loading the Design


To load a design, type the following command:


```
vsim -t ps -sdftyp <design_instance> = <path to *.sdo file> -L \  
<precompiled-library1> -L <precompiled-library2> work.<my_testbench> ←
```

The <precompiled-library1> and <precompiled-library2> variables are the libraries required to compile your testbench. If you have multiple libraries, use the -L option for each library in the vsim command.

Examples of <precompiled library> are **stratixiii** and **cycloneiii**. The gate-level simulation libraries are required for performing gate-level simulation. For the complete set of libraries, refer to “Precompiled Simulation Libraries in the ModelSim-Altera Software” on page 3-4.

You can choose not to invoke -L in the vsim command for VHDL designs if you have already included the mapped name of the libraries in your design files or subfiles.


-  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`).

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←
run <time period> ←
```

 To add all signals in your testbench hierarchy, type the following command:

```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```


## Simulating the Verilog HDL Designs on the Command Line

Simulating Verilog HDL design from the ModelSim-Altera command line gives you more flexibility and control to compile the libraries and load the design files. All simulation commands are Tcl commands that can be put into the ModelSim Macro File (`*.do`). Using the `*.do` file allows you to run simulation in batch mode. You only have to execute the `*.do` file and the ModelSim-Altera tool automatically executes all commands in the `*.do` script macro file.

You can use the ModelSim-Altera command line to perform the RTL functional simulation, post-synthesis simulation, and gate-level simulation. The following sections show how to perform simulation at various levels from the ModelSim-Altera command line.

### Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in ModelSim-Altera for Verilog HDL designs.

 The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and Design Files into the Work Library

Use the following commands to compile your testbench and design files into the work library in the ModelSim-Altera software.

To change to the design library, type the following command:


```
cd <your_design_directory> ←
(for example, cd:/designs)
```

To create the work library, type the following commands:

```
vlib work ←
vmap work work ←
```

To compile the testbench and design files into the work library, type the following command:


```
vlog -work work <my_testbench.v> <my_design_files.v> ←
```


 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:

```
vsim -t ps -L <library1> -L <library2> work.<my_testbench> ↵
```

 The *<library1>* and *<library2>* variables are required libraries to compile your testbench. If you have multiple libraries, use the `-L` option multiple times in the `vsim` command.

 Examples of *<precompiled library>* are `altera_mf_ver` and `lpm_ver`. The functional RTL simulation libraries are required for performing RTL functional simulation. For the complete set of libraries, refer to “[Precompiled Simulation Libraries in the ModelSim-Altera Software](#)” on page 3–4.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ↵  
run <time period> ↵
```

 To add all signals in your testbench hierarchy, type the following command:


```
add wave * ↵
```

To run the simulation for 100 ps, type the following command:


```
run 100 ps ↵
```

### 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 following sections help you perform a post-synthesis simulation for a Verilog HDL design in the ModelSim-Altera software.


 The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.

### Compiling Testbench and Verilog HDL Output File into the Work Library

Use the following commands to compile your testbench and `*.vo` file into the work library in the ModelSim-Altera software.

To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ←
    (for example, cd:/designs/modelsim/simulation)
```


 This directory contains the \*.vo file, which is generated by the netlist writer.

To create the work library, type the following commands:

```
vlib work ←
vmap work work ←
```

To compile the testbench and \*.v files into the work library, type the following command:


```
vlog -work work <my_testbench.v> <my_design_netlists.vo> ←
```

 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:


```
vsim -t ps -L <library1> -L <library2> work.<my_testbench> ←
```

 The <library1> and <library2> variables are libraries required to compile your testbench. Examples of <precompiled library> are **stratixiii\_ver**, **stratixii\_ver**, and **stratixiigx\_ver**. Gate-level libraries are required for performing post-synthesis simulation. If you have multiple libraries, use the -L option multiple times in the vsim command. For the complete set of libraries, refer to “[Precompiled Simulation Libraries in the ModelSim-Altera Software](#)” on page 3-4.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←
run <time period> ←
```

 To add all signals in your testbench hierarchy, type the following command:


```
add wave * ←
```

To run the simulation for 100 ps, type the following command:


```
run 100 ps ←
```

### 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-Altera software.

 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*.

The following sections help you perform a gate-level simulation for a Verilog HDL design in the ModelSim-Altera software.

-  The ModelSim-Altera software includes precompiled simulation libraries. It is not necessary to create simulation libraries and compile simulation models.


### Compiling Testbench and Verilog HDL Output File into the Work Library

Use the following commands to compile your testbench and \*.vo file into the work library in the ModelSim-Altera software.

To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ←
```

(for example, **cd:/designs/modelsim/simulation**)


-  This directory contains the \*.vho file, which is generated by the netlist writer.

To create the work library, type the following commands:

```
vlib work ←  
vmap work work ←
```

To compile the testbench and VHDL output (\*.vho) files into the work library, type the following command:

```
vlog -work work <my_testbench.v> <my_design_netlists.vo> ←
```


-  Resolve compile-time errors before proceeding to the next section.

### Loading the Design

When simulating in Verilog HDL, you do not have to specify the \*.sdo file. In the \$sdf\_annotate task, when the Quartus II software generates the \*.vo file, the ModelSim-Altera software looks for the \*.sdo file in the directory in which the VSIM was run. If your \*.sdo file is not in this directory, copy the \*.sdo file into your current directory.

To load a design, type the following command:


```
vsim -t ps -L <library1> -L <library2> work.<my_testbench> ←
```

-  The <library1> and <library2> variables are required libraries to compile your testbench. Examples of precompiled libraries are **stratixiii\_ver**, **stratixii\_ver**, and **stratixiigx\_ver**. Gate-level libraries are required for performing gate-level timing simulation. If you have multiple libraries, use the -L option multiple times in the vsim command. For the complete set of libraries, refer to [“Precompiled Simulation Libraries in the ModelSim-Altera Software”](#) on page 3-4.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←  
run <time period> ←
```

-  To add all signals in your testbench hierarchy, type the following command:

```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

## Performing Simulation Using the ModelSim Software

Simulation of Verilog HDL or VHDL designs with ModelSim software can be done at various levels to verify designs from different aspects. Simulation is divided into three categories: RTL functional simulation, post-synthesis simulation, and gate-level simulation. Simulation helps you verify your designs and debug them against any possible errors in the designs.

You can perform the simulation through the GUI or from the command line. The following sections provide step-by-step 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 the VHDL Designs Using the GUI

You do not have to remember the commands to compile the libraries or load and simulate the VHDL design files. You can use the ModelSim GUI to perform RTL functional simulation, post-synthesis simulation, and gate-level timing simulation. The following sections show how to perform simulation at various levels through the ModelSim GUI.

### Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in the ModelSim software for VHDL designs.

#### Creating Simulation Libraries

Simulation libraries are required to simulate a design that contains an Altera primitive, LPM function, or Altera megafunction. Depending on your design, you must create the required simulation libraries and link them to your design correctly.

To change to the design directory, perform the following steps:

1. Browse to locate your designs.
2. Click **OK**.

If you are not using the EDA Simulation Library Compiler, as described in [“Compiling Libraries Using the EDA Simulation Library Compiler” on page 3-7](#), perform the following steps to create the simulation library. (If you are using this utility, you can skip these steps.)

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.

3. In the **Library Name** box, type the library name of the newly created library.



For example, the library name for Altera megafunctions is **altera\_mf**, and the library name for LPM is **lpm**. To see all of the functional simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

4. Click **OK**.

### Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in “Compiling Libraries Using the EDA Simulation Library Compiler” on page 3-7, perform the following steps to compile simulation models into simulation libraries. (If you are using this utility, you can skip these steps.)

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**.



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**.

### Compiling Testbench and Design Files into the Work Library

The following instructions show you how to compile your testbench and design files into the work library with the ModelSim GUI.

To create the work library, perform the following steps:

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name **Work** in the text box.
4. Click **OK**.

To compile the testbench and design files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The design files and testbench file should be compiled into the **Work** library.
3. Select the design files and the testbench file and click **Compile**.

4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. For VHDL designs, if you have not included the mapped name of the libraries in your design files or subfiles, perform the following steps:
  - a. Click the **Libraries** tab.
  - b. In the **Search Libraries (-L)** text box, click the **Add** button.
  - c. Browse to the required simulation library that you previously compiled (for example, **altera\_mf**, **lpm**, or **altera**).
  - d. Click **OK** to add the libraries to the **Search Libraries (-L)** text box.
7. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

### 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 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 following sections help you perform a post-synthesis simulation for a VHDL design in the ModelSim software.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. You must create the required simulation libraries for the device family you are using and link them to your design correctly.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your testbench or \*.vho file. By default, the \*.vho file is located in *<project directory>/simulation/modelsim*.
2. Click **OK**.

If you are not using the EDA Simulation Library Compiler as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3–7, perform the following steps to create the simulation library. (If you are using this utility, you can skip these steps.)

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name of the newly created library.



For example, the library name for Stratix III family is **stratixiii**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

4. Click **OK**.

### Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3–7, perform the following steps to compile simulation models into simulation libraries. (If you are using this utility, you can skip these steps.)

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library that you created (for example, **stratixiii**).
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**.



The **stratixiii\_components.vhd** and **stratixiii.vhd** model files are compiled into the **stratixiii** library.

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

### Compiling Testbench and VHDL Output File into the Work Library

The following instructions show you how to compile your testbench and \*.vho file into the work library using the ModelSim GUI.

To create the work library, perform the following steps:

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name **Work** in the text box.
4. Click **OK**.

To compile the testbench file and \*.vho file into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and \*.vho file should be compiled into the **Work** library.
3. Select the testbench and \*.vho file design files and click **Compile**.
4. Click **Done**.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. In the **Start Simulation** dialog box, click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. For VHDL designs, if you have not included the mapped name of the libraries in your design files or subfiles, perform the following steps:
  - a. Click the **Libraries** tab.
  - b. In the **Search Libraries (-L)** text box, click the **Add** button.
  - c. Browse to the required simulation library that you previously compiled (for example, **stratixii**, **stratixiii**, or **cycloneiii**).
  - d. Click **OK** to add the libraries to the **Search Libraries (-L)** text box.
7. Click **OK**.

### Running the Simulation


To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.

2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

### 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 software.

 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*.

The following sections help you perform a gate-level simulation for a VHDL design in the ModelSim software.

### Creating Simulation Libraries


Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. You must create the required simulation libraries for the device family you are using and link them to your design correctly.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your testbench or \*.vho file. By default, the \*.vho file is located in `<project directory>/simulation/modelsim<`.
2. Click **OK**.

If you are not using the EDA Simulation Library Compiler, as described in "Compiling Libraries Using the EDA Simulation Library Compiler" on page 3-7, perform the following steps to create simulation libraries. (If you are using this utility, you can skip these steps.)

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name of the newly created library.

 For example, the library name for the Stratix III family is **stratixiii**. To see all gate-level timing simulation library files, refer to the "Simulation Libraries" section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

4. Click **OK**.

### Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, perform the following steps to compile simulation models into simulation libraries. (If you are using this utility, you can skip these steps.)

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library that you created (for example, **stratixiii**).
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**.



The **stratixiii\_components.vhd** and **stratixiii.vhd** model files should be compiled into the **stratixiii** library.

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

### Compiling Testbench and Design Files into the Work Library

The following instructions show you how to compile your testbench and \*.vho file into the work library using the ModelSim GUI.

To create the work library, perform the following steps:

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name `Work` in the text box.
4. Click **OK**.

To compile the testbench file and \*.vho file into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and \*.vho file should be compiled into the **Work** library.
3. Select the testbench and \*.vho design files and click **Compile**.
4. Click **Done**.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. Click the **SDF** tab and click **Add**.

3. In the **Add SDF Entry** dialog box, click **Browse** and select the \*.sdo file. By default, the \*.sdo file is located in *<project directory>/simulation/modelsim*.
4. In the **Apply to Region** dialog box, type in the instance path to which the \*.sdo file is to be applied. For example, if you are using a testbench exported into the Quartus II software from a Vector Waveform File, the instance path is set to */i1*.



You do not have to choose from the **Delay** list because the Quartus II EDA Netlist Writer generates the \*.sdo file using the same value for the triplet (minimum, typical, and maximum timing values).

5. Click **OK**.
6. Click the **Design** tab. In the **Resolution** list, select **ps**.
7. In the **Library** list, select the **Work** library.
8. In the **Start Simulation** dialog box, expand the **Work** library.
9. Select the top-level design unit (your testbench).
10. In the **Resolution** list, select **ps**.
11. For VHDL designs, if you have not included the mapped name of the libraries in your design files or subfiles, perform the following steps:
  - a. Click the **Libraries** tab.
  - b. In the **Search Libraries (-L)** text box, click the **Add** button.
  - c. Browse to the required simulation library that you previously compiled (for example, *stratixii*, *stratixiii*, or *cycloneiii*).
  - d. Click **OK** to add the libraries to the **Search Libraries (-L)** text box.
12. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

## Simulating the Verilog HDL Designs Using the GUI

Simulating Verilog HDL design using the ModelSim GUI is user-friendly. You do not have to remember the commands to compile the libraries or load and simulate the Verilog HDL design files. You can use the ModelSim GUI to perform RTL functional simulation, post-synthesis simulation, and gate-level timing simulation. The following sections show how to perform simulation at various levels through the ModelSim GUI.

## Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in the ModelSim software for Verilog HDL designs.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that contains an Altera primitive, LPM function, or Altera megafunction. Depending on your design, you must create the required simulation libraries and link them to your design correctly.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your designs.
2. Click **OK**.

If you are not using the EDA Simulation Library Compiler, as described in [“Compiling Libraries Using the EDA Simulation Library Compiler” on page 3-7](#), perform the following steps to create simulation libraries. (If you are using this utility, you can skip these steps.)

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name of the newly created library.



For example, the library name for Altera megafunctions is **altera\_mf\_ver**, and the library name for LPM is **lpm\_ver**. To see all of the functional simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

4. Click **OK**.

### Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in [“Compiling Libraries Using the EDA Simulation Library Compiler” on page 3-7](#), perform the following steps to compile simulation models into simulation libraries. (If you are using this utility, you can skip these steps.)

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**.



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**.

### Compiling Testbench and Design Files into the Work Library

The following instructions show you how to compile your testbench and design files into the work library using the ModelSim GUI.

To change to the design directory, perform the following steps:

1. Browse to locate your designs.
2. Click **OK**.

To compile the testbench and design files into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The design files and testbench file should be compiled into the **Work** library.
3. Select the design files and the testbench file and click **Compile**.
4. Click **Done**.



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. Click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. Click the **Libraries** tab.
7. In the **Search Libraries (-L)** text box, click the **Add** button to browse to the required simulation library that you previously compiled (for example, **altera\_mf\_ver**, **lpm\_ver**, or **altera\_ver**) and click **OK** to add them into the **Search Libraries (-L)** text box.
8. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

## 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 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 following sections help you perform a post-synthesis simulation for a Verilog HDL design in the ModelSim software.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. Depending on the device family you are using, you must create the required simulation libraries and link them to your design correctly.

To change to the design directory, perform the following steps:

1. Browse to locate your testbench or \*.vo file. By default, the \*.vo file is located in *<project directory>/simulation/modelsim*.
2. Click **OK**.

If you are not using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, perform the following steps to create the simulation library. (If you are using the utility, you can skip these steps.)

To create the simulation library, perform the following steps:

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name of the newly created library.



For example, the library name for the Stratix III family is **stratixiii**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

4. Click **OK**.

### Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, perform the following steps to compile simulation models into simulation libraries. (If you are using this utility, you can skip these steps.)

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library that you created (for example, **stratixiii\_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**.



The **stratixiii\_atoms.v** model files should be compiled into the **stratixiii\_ver** library.

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

### Compiling Testbench and Verilog HDL Output File into the Work Library

The following instructions show you how to compile your testbench and \*.vo into the work library using the ModelSim GUI.

To create the work library, perform the following steps:

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name **Work** in the text box.
4. Click **OK**.

To compile the testbench file and \*.vo file into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and \*.vo file should be compiled into the **Work** library.
3. Select the testbench and \*.vo design files, and click **Compile**.
4. Click **Done**.

### Loading the Design

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. Click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select the **Work** library.
4. In the **Start Simulation** dialog box, expand the **Work** library.
5. Select the top-level design unit (your testbench).
6. In the **Resolution** list, select **ps**.
7. Click the **Libraries** tab.
8. In the **Search Libraries (-L)** text box, click the **Add** button to browse to the required simulation library that you previously compiled (for example, **stratixiii\_ver** or **stratixiii\_gx\_ver**) and click **OK** to add them into the **Search Libraries (-L)** text box.

9. Click **OK**.


### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

### 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 software.

 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*.

The following sections help you perform a gate-level simulation for a Verilog HDL design in the ModelSim software.

### Creating Simulation Libraries


Simulation libraries are required to simulate a design that contains an Altera primitive, LPM function, or Altera megafunction. Depending on your design, you must create the required simulation libraries and link them to your design correctly.

To change to the simulation output directory, perform the following steps:

1. Browse to locate your testbench or \*.vo file. By default, the \*.vo file is located in *<project directory>/simulation/modelsim*.
2. Click **OK**.

If you are not using the EDA Simulation Library Compiler, as described in "Compiling Libraries Using the EDA Simulation Library Compiler" on page 3-7, perform the following steps to create the simulation library. (If you are using the utility, you can skip these steps.)

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name of the newly created library.


 For example, the library name for the Stratix III family is **stratixiii**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

4. Click **OK**.

### Compiling Simulation Models into Simulation Libraries

If you are not using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, perform the following steps to compile simulation models into simulation libraries. (If you are using this utility, you can skip these steps.)

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library that you created (for example, **stratixiii\_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**.

 The **stratixiii\_atoms.v** model files should be compiled into the **stratixiii\_ver** library.

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

### Compiling Testbench and Verilog HDL Output File into the Work Library


The following instructions show you how to compile your testbench and \*.vo file into the work library using the ModelSim GUI.

To create the work library, perform the following steps:

1. In the ModelSim software, on the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box appears.
2. Select a new library and a logical mapping to it.
3. In the **Library Name** box, type the library name `work` in the text box.
4. Click **OK**.

To compile the testbench file and Verilog HDL output file (\*.vo) into the work library, perform the following steps:

1. On the Compile menu, click **Compile**. The **Compile Source Files** dialog box appears.
2. Select the library **Work**. The testbench and \*.vo file should be compiled into the **Work** library.
3. Select the testbench and \*.vo design files and click **Compile**.
4. Click **Done**.

 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

When simulating in Verilog HDL, you do not have to specify the \*.sdo file. In the \$sdf\_annotate task, when the Quartus II software generates the \*.vo file, the ModelSim-Altera software looks for the \*.sdo file in the directory in which the VSIM was run. If your \*.sdo file is not in this directory, copy the \*.sdo file into your current directory.

To load a design, perform the following steps:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box appears.
2. Click the **Design** tab. In the **Resolution** list, select **ps**.
3. In the **Library** list, select and expand the **Work** library.
4. Select the top-level design unit (your testbench).
5. In the **Resolution** list, select **ps**.
6. Click the **Libraries** tab.
7. In the **Search Libraries (-L)** text box, click the **Add** button to browse to the required simulation library that you previously compiled (for example, **stratixiii\_ver** or **stratixiiiigx\_ver**) and click **OK** to add them into the **Search Libraries (-L)** text box.
8. Click **OK**.

### Running the Simulation

To run a simulation, perform the following steps:

1. On the View menu, point to **Debug Windows** and click **Objects**. This command displays all objects in the current scope.
2. On the View menu, point to **Debug Windows** and click **Wave**.
3. Drag signals to monitor from the Objects window and drop them into the Wave window.
4. On the Processing menu, point to **Run** and click **Run 100 ps** to run the simulation for 100 ps.

## Simulating the VHDL Designs from the Command Line

Simulating VHDL designs from the ModelSim command line gives you more flexibility and control to compile the libraries and load and simulate the VHDL design files. All simulation commands are Tcl commands that can be included in the ModelSim Macro File (\*.do). Using the \*.do file allows you to run simulation in batch mode. You have to execute only the \*.do file, and the ModelSim tool automatically executes all commands in the \*.do script macro file.

You can use the ModelSim command line to perform RTL functional simulation, post-synthesis simulation, and gate-level simulation. The following sections show how to perform simulation at various levels from the ModelSim command line.

## Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections describe how to perform RTL functional simulation in ModelSim for VHDL designs.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that contains an Altera primitive, LPM function, or Altera megafunction. Depending on your design, you must create the required simulation libraries and link them to your design correctly.

To change to the design library, type the following command:

```
cd <your_design_directory> ←
```

(for example, **cd:/designs**)

To create the simulation libraries, type the following commands:

```
vlib <library_name> ←  
vmap <logical_library_name> <library_name> ←
```



For example, the library name for Altera megafunctions is **altera\_mf**, and the library name for LPM is **lpm**. To see all of the functional simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

To create simulation libraries for **altera\_mf**, **lpm**, and **altera**, type the following commands:

```
vlib altera_mf ←  
vmap altera_mf altera_mf ←  
vlib lpm ←  
vmap lpm lpm ←  
vlib altera ←  
vmap altera altera ←
```

### Compiling Simulation Models into Simulation Libraries

To compile simulation models into simulation libraries, type the following command:

```
vcom -work <simulation_library> <Quartus II installation directory> \  
/eda/sim_lib/<simulation_library_files> ←
```

For example, 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.

Use [Example 3-1](#) to compile the simulation model files to the simulation libraries for **altera\_mf**, **lpm**, and **altera**.

**Example 3-1.**

```
vcom -work altera_mf <Quartus II installation directory> \
/eda/sim_lib/altera_mf_components.v <Quartus II installation directory> \
/eda/sim_lib/altera_mf.vhd ←

vcom -work lpm <Quartus II installation directory>/eda/sim_lib/220model.vhd \
<Quartus II installation dir>/eda/sim_lib/220model.vhd ←

vcom -work altera <Quartus II installation directory> \
/eda/sim_lib/altera_primitives_components.vhd \
<Quartus II installation directory>/eda/sim_lib/altera_primitives.vhd ←
```

If you are using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, skip the steps for creating and compiling the simulation libraries.

**Compiling Testbench and Design Files into the Work Library**

Use the following commands to compile your testbench and design files into the work library in the ModelSim software.

To create the work library, type the following commands:

```
vlib work ←
vmap work work ←
```

To compile the testbench and design files into the work library, type the following command:

```
vcom -work work <my_testbench.vhd> <my_design_files.vhd> ←
```



Resolve compile-time errors before proceeding to the next section.

**Loading the Design**

To load a design, type the following command:

```
vsim -t ps -L <compiled-library1> -L <compiled-library2> work.<my_testbench> ←
```

The *<compiled-library1>* and *<compiled-library2>* variables are the libraries you compiled previously (for example, **altera\_mf** or **lpm**) that are required to compile your testbench. RTL libraries are required for performing RTL simulation. If you have multiple libraries, use the **-L** option for each library in the **vsim** command.

**Running the Simulation**

To run a simulation, type the following commands:

```
add wave <signal name> ←
run <time period> ←
```



To add all signals in your testbench hierarchy, type the following command:


```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

## 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 software.

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

The following sections help you perform a post-synthesis simulation for a VHDL design in the ModelSim software.

### Creating Simulation Libraries


Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. Depending on the device family you are using, you must create the required simulation libraries and link them to your design correctly.

To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ←  
(for example, cd:/designs/modelsim/simulation)
```

To create the simulation libraries, type the following commands:

```
vlib <library_name> ←  
vmap <logical_library_name> <library_name> ←
```

 For example, the library name for the Stratix III family is **stratixiii**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

To create simulation libraries for **stratixiii**, type the following commands:

```
vlib stratixiii ←  
vmap stratixiii stratixiii ←
```

### Compiling Simulation Models into Simulation Libraries

To compile simulation models into simulation libraries, type the following command:

```
vcom -work <simulation_library> <Quartus II installation directory> \  
/eda/sim_lib/<simulation_library_files> ←
```

For example, the **stratixiii\_atoms\_components.vhd** and **stratixiii\_atoms.vhd** model files should be compiled into the **stratixiii** library.

Use [Example 3-2](#) to compile the simulation model files to the simulation libraries for **stratixiii**:

#### Example 3-2.

---

```
vcom -work stratixiii <Quartus II installation directory> \  
/eda/sim_lib/stratixiii_atoms_components.vhd <Quartus II installation directory> \  
/eda/sim_lib/stratixiii_atoms.vhd ←
```

---

If you are using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, skip the steps for creating and compiling the simulation libraries.

### Compiling Testbench and VHDL Output Files into the Work Library

Use the following commands to compile your testbench and \*.vho file into the work library in the ModelSim software.

To create the work library, type the following commands:

```
vlib work ←  
vmap work work ←
```

To compile the testbench and \*.vho file into the work library, type the following command:

```
vcom -work work <my_testbench.vhd> <my_design_files.vho> ←
```



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:

```
vsim -t ps -L <compiled-library1> -L <compiled-library2> work.<my_testbench> ←
```

The <compiled-library1> and <compiled-library2> variables are the libraries you compiled previously (for example, **stratixiii** or **cycloneiii**) that are required to compile your testbench. Gate-level libraries are required for performing post-synthesis simulation. If you have multiple libraries, use the -L option for each library in the vsim command.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←  
run <time period> ←
```



To add all signals in your testbench hierarchy, type the following command:

```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

### 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 software.



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*.

The following sections help you perform a gate-level simulation for a VHDL design in the ModelSim software.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. Depending on the device family you are using, you must create the required simulation libraries and link them to your design correctly.

To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ↵
```

(for example, **cd:/designs/modelsim/simulation**)

To create the simulation libraries, type the following commands:

```
vlib <library_name> ↵  
vmap <logical_library_name> <library_name> ↵
```



For example, the library name for the Stratix III family is **stratixiii**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

To create simulation libraries for **stratixiii**, type the following commands:

```
vlib stratixiii ↵  
vmap stratixiii stratixiii ↵
```

### Compiling Simulation Models into Simulation Libraries

To compile simulation models into simulation libraries, type the following command:

```
vcom -work <simulation_library> <Quartus II installation directory> \  
/eda/sim_lib/<simulation_library_files> ↵
```

For example, the **stratixiii\_atoms\_components.vhd** and **stratixiii\_atoms.vhd** model files should be compiled into the **stratixiii** library.

Use [Example 3-3](#) to compile the simulation model files to the simulation libraries for **stratixiii**.

#### Example 3-3.

```
vcom -work stratixiii <Quartus II installation directory> \  
/eda/sim_lib/stratixiii_atoms_components.vhd <Quartus II installation directory> \  
/eda/sim_lib/stratixiii_atoms.vhd ↵
```

If you are using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, skip the steps for creating and compiling the simulation libraries.

Be sure that the user-compiled libraries are stored in the same directory as the design and testbench files you want to compile.

### Compiling Testbench and VHDL Output Files into the Work Library

Use the following commands to compile your testbench and \*.vho file into the work library in the ModelSim software.

To create the work library, type the following commands:

```
vlib work ←
vmap work work ←
```

To compile the testbench and \*.vho file into the work library, type the following command:

```
vcom -work work <my_testbench.vht> <my_design_files.vho> ←
```



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:

```
vsim -t ps -sdftyp <design instance> = <path to *.sdo file> -L <compiled-library1> \
-L <compiled-library2> work.<my_testbench> ←
```

The <compiled-library1> and <compiled-library2> variables are the libraries you compiled previously (for example, **stratixiii** or **cycloneiii**) that are required to compile your testbench. Gate-level libraries are required for performing gate-level simulation. If you have multiple libraries, use the -L option for each library in the vsim command.



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).

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←
run <time period> ←
```



To add all signals in your testbench hierarchy, type the following command:

```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

## Simulating the Verilog HDL Designs from the Command Line

Simulating Verilog HDL design from the ModelSim command line gives you more flexibility and control to compile the libraries, and load and simulate 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 tool automatically executes all commands in the \*.do script macro file.

You can use the ModelSim command line to perform RTL functional simulation, post-synthesis simulation, and gate-level simulation. The following sections show how to perform simulation at various levels from the ModelSim command line.

## Performing RTL Functional Simulation

RTL functional simulation verifies code syntax and design functionality. The following sections show how to perform RTL functional simulation in ModelSim for Verilog HDL designs.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that contains an Altera primitive, LPM function, or Altera megafunction. Depending on your design, you must create the required simulation libraries and link them to your design correctly.

To change to the design library, type the following command:

```
cd <your_design_directory> ←
```

(for example, **cd:/designs**)

To create the simulation libraries, type the following commands:

```
vlib <library_name> ←  
vmap <logical_library_name> <library_name> ←
```



For example, the library name for Altera megafunction is **altera\_mf\_ver**, and the library name for LPM is **lpm\_ver**. To see all of the functional simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

To create simulation libraries for **altera\_mf**, **lpm\_ver**, and **altera\_ver**, type the following commands:

```
vlib altera_mf_ver ←  
vmap altera_mf altera_mf_ver ←  
vlib lpm_ver ←  
vmap lpm lpm_ver ←  
vlib altera_ver ←  
vmap altera altera_ver ←
```

If you are using the EDA Simulation Library Compiler, skip the steps for creating and compiling the simulation libraries.

### Compiling Simulation Models into Simulation Libraries

If you are using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, skip the steps for creating and compiling the simulation libraries.

To compile simulation models into simulation libraries, type the following command:

```
vlog -work <simulation_library> <Quartus II installation directory>\  
/eda/sim_lib/<simulation_library_files> ←
```

For example, 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.

To compile the simulation model files to the simulation libraries for **altera\_mf\_ver**, **lpm\_ver**, and **altera\_ver**, type the following commands:

```
vlog -work altera_mf_ver \  
<Quartus II installation directory>/eda/sim_lib/altera_mf.v ←  
  
vlog -work lpm_ver \  
<Quartus II installation directory>/eda/sim_lib/220model.v ←  
  
vlog -work altera_ver \  
<Quartus II installation directory>/eda/sim_lib/altera_primitives.v ←
```

### Compiling Testbench and Design Files into the Work Library


Use the following commands to compile your testbench and design files into the work library in the ModelSim software.

To create the work library, type the following commands:

```
vlib work ←  
vmap work work ←
```

To compile the testbench and design files into the work library, type the following command:


```
vlog -work work <my_testbench.v> <my_design_files.v> ←
```

 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:


```
vsim -t ps -L <library1> -L <library2> work.<my_testbench> ←
```

 The *<library1>* and *<library2>* variables are required libraries to compile your testbench. If you have multiple libraries, use the *-L* option multiple times in the *vsim* command.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←  
run <time period> ←
```

 To add all signals in your testbench hierarchy, type the following command:


```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

### 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 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 following sections help you perform a post-synthesis simulation for a Verilog HDL design in the ModelSim software.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. Depending on the device family you are using, you must create the required simulation libraries and link them to your design correctly.


To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ↵
```

(for example, `cd:/designs/modelsim/simulation`)

To create the simulation libraries, type the following commands:

```
vlib <library_name> ↵  
vmap <logical_library_name> <library_name> ↵
```

 For example, the library name for the Stratix III family is **stratixiii\_ver**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

To create simulation libraries for **stratixiii\_ver**, type the following commands:

```
vlib stratixiii_ver ↵  
vmap stratixiii_ver stratixiii_ver ↵
```

If you are using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, skip the steps for creating and compiling the simulation libraries.

### Compiling Simulation Models into Simulation Libraries

To compile simulation models into simulation libraries, type the following command:

```
vlog -work <simulation_library> <Quartus II installation dir> \  
/eda/sim_lib/<simulation_library_files> ↵
```

For example, the **stratixiii\_atoms.v** model files should be compiled into the **stratixiii\_ver** library.

To compile the simulation model files to the simulation libraries for **stratixiii\_ver**, type the following command:

```
vlog -work stratixiii_ver <Quartus II installation directory> \  
/eda/sim_lib/stratixiii_atoms.v ↵
```

If you are using the EDA Simulation Library Compiler, skip the steps for creating and compiling the simulation libraries.

### Compiling Testbench and Verilog HDL Output Files into the Work Library

Use the following commands to compile your testbench and \*.vo file into the work library in the ModelSim software.

To create the work library, type the following commands:

```
vlib work ←  
vmap work work ←
```

To compile the testbench and \*.vo file into the work library, type the following command:

```
vlog -work work <my_testbench.vt> <my_design_netlists.vo> ←
```



Resolve compile-time errors before proceeding to the next section.

### Loading the Design

To load a design, type the following command:

```
vsim -t ps -L <library1> -L <library2> work.<my_testbench> ←
```



The <library1> and <library2> variables are the libraries you compiled previously (for example, **stratixiii** or **stratixiigx**) that are required to compile your testbench. Gate-level libraries are required for performing post-synthesis simulation. If you have multiple libraries, use the -L option multiple times in the vsim command.

### Running the Simulation

To run a simulation, type the following commands:

```
add wave <signal name> ←  
run <time period> ←
```



To add all signals in your testbench hierarchy, type the following command:

```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

### Performing Gate-Level Simulation

Gate-level simulation is a necessary step in ensuring that the functionality of the FPGA device 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 software.



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*.

The following sections help you perform a gate-level simulation for a Verilog HDL design in the ModelSim software.

### Creating Simulation Libraries

Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. Depending on the device family you are using, you must create the required simulation libraries and link them to your design correctly.


To change to the simulation output directory, type the following command:

```
cd <simulation_output_directory> ↵  
(for example, cd:/designs/modelsim/simulation)
```

 This directory contains the \*.vo file, which is generated by the netlist writer.

To create the simulation libraries, type the following commands:

```
vlib <library_name> ↵  
vmap <logical_library_name> <library_name> ↵
```

 For example, the library name for the Stratix III family is **stratixiii\_ver**. To see all of the gate-level timing simulation library files, refer to the “Simulation Libraries” section in the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

To create simulation libraries for **stratixiii\_ver**, type the following commands:

```
vlib stratixiii_ver ↵  
vmap stratixiii_ver stratixiii_ver ↵
```

### Compiling Simulation Models into Simulation Libraries

To compile simulation models into simulation libraries, type the following command:

```
vlog -work <simulation_library> <Quartus II installation directory> \  
/eda/sim_lib/<simulation_library_files> ↵
```

For example, the **stratixiii\_atoms.v** model files should be compiled into the **stratixiii\_ver** library.

Use the following example to compile the simulation model files to the simulation libraries for **stratixiii\_ver**:

```
vlog -work stratixiii <Quartus II installation directory> \  
/eda/sim_lib/stratixiii_atoms.v ↵
```

If you are using the EDA Simulation Library Compiler, as described in “[Compiling Libraries Using the EDA Simulation Library Compiler](#)” on page 3-7, skip the steps for creating and compiling the simulation libraries.

### Compiling Testbench and Verilog HDL Output Files into the Work Library


Use the following commands to compile your testbench and \*.vo file into the work library in the ModelSim software.

To create the work library, type the following commands:

```
vlib work ↵  
vmap work work ↵
```

To compile the testbench and \*.vo files into the work library, type the following command:

```
vlog -work work <my_testbench.v> <my_design_netlists.vo> ↵
```

 Resolve compile-time errors before proceeding to the next section.

### Loading the Design

When simulating in Verilog HDL, you do not have to specify the \*.sdo file. In the \$sdf\_annotate task, when the Quartus II software generates the \*.vo file, the ModelSim-Altera software looks for the \*.sdo file in the directory in which the VSIM was run. If your \*.sdo file is not in this directory, copy the \*.sdo file into your current directory.

To load a design, type the following command:

```
vsim -t ps -L <library1> -L <library2> work.<my_testbench> ←
```



The <library1> and <library2> variables are the libraries you compiled previously (for example, stratixiii\_ver or stratixiigx\_ver) that are required to compile your testbench. Gate-level libraries are required for performing gate-level timing simulation. If you have multiple libraries, use the -L option multiple times in the vsim command.

### Running the Simulation

Perform the following commands to run a simulation:

```
add wave <signal name> ←
run <time period> ←
```



To add all signals in your testbench hierarchy, type the following command:

```
add wave * ←
```

To run the simulation for 100 ps, type the following command:

```
run 100 ps ←
```

## Passing Parameter Information from Verilog to VHDL

You must use in-line parameters to pass values from Verilog HDL to VHDL. Using the defparam command can cause an error in simulation.

The following error message appears:


```
# ** 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/i1/b2v_inst
```

The following example converts the defparam command to in-line parameters:

```
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;
```

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 at the VHDL component declaration.

## Speeding Up Simulation

By default, the ModelSim software runs in a debug-optimized mode. To run the ModelSim 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, or Stratix IV transceiver, you must compile additional library files to perform RTL functional or gate-level timing simulations. The following example shows how to perform simulation on designs that include transceivers in Stratix GX, Stratix II GX, and Stratix IV devices.


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 **hardcopyiv\_atoms** and **hardcopyiv\_hssi\_atoms** model files for HardCopy 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*.

## RTL Functional Simulation for Stratix IV Devices

To perform an RTL 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 **stratixiigx\_mf** model file references the **lpm** and **sgate** libraries. If you are using ModelSim PE/SE, you must create these libraries to perform a simulation.

### Performing RTL Functional Simulation in VHDL (ModelSim-Altera)

To perform RTL functional simulation for Stratix GX devices in VHDL (ModelSim-Altera), 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 RTL Functional Simulation in VHDL (ModelSim SE/PE)

To perform RTL functional simulation for Stratix GX devices in VHDL (ModelSim SE/PE), 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 RTL Functional Simulation in Verilog HDL (ModelSim-Altera)

To perform RTL functional simulation for Stratix GX devices in Verilog HDL (ModelSim-Altera), 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 RTL Functional Simulation in Verilog HDL (ModelSim SE/PE)

To perform RTL functional simulation for Stratix GX devices in Verilog HDL (ModelSim SE/PE), 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 IV 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 PE/SE, 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 (ModelSim-Altera), 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 SE/PE)

To perform gate-level timing simulation for Stratix GX devices in VHDL (ModelSim SE/PE), 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 (ModelSim-Altera), 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 SE/PE)

To perform gate-level timing simulation for Stratix GX devices in Verilog HDL (ModelSim SE/PE), 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 ↵
```

## RTL Functional Simulation for Stratix II GX Devices

To perform a 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 PE/SE, 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 (Figure 3-1). The `<alt2gxb entity name>.vho` or `<alt2gxb module name>.vo` is generated in the current project directory.



### Performing RTL Functional Simulation in VHDL (ModelSim-Altera)

To perform RTL functional simulation for Stratix II GX devices in VHDL (ModelSim-Altera), 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 RTL Functional Simulation in VHDL (ModelSim SE/PE)

To perform RTL functional simulation for Stratix II GX devices in VHDL (ModelSim SE/PE), 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 RTL Functional Simulation in Verilog HDL (ModelSim-Altera)

To perform RTL functional simulation for Stratix II GX devices in Verilog HDL (ModelSim-Altera), 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 RTL Functional Simulation in Verilog HDL (ModelSim SE/PE)

To perform RTL functional simulation for Stratix II GX devices in Verilog HDL (ModelSim SE/PE), 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 PE/SE, 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 (ModelSim-Altera), 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 SE/PE)

To perform gate-level timing simulation for Stratix II GX devices in VHDL (ModelSim SE/PE), 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 (ModelSim-Altera), 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 SE/PE)

To perform gate-level timing simulation for Stratix II GX devices in Verilog HDL (ModelSim SE/PE), 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 ←
```

## RTL Functional Simulation for Stratix IV Devices

To perform an RTL 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 PE/SE, you must create these libraries to perform a simulation.

### Performing RTL Functional Simulation in VHDL (ModelSim-Altera)

To perform RTL functional simulation for Stratix IV devices in VHDL (ModelSim-Altera), 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 RTL Functional Simulation in VHDL (ModelSim SE/PE)

To perform RTL functional simulation for Stratix IV devices in VHDL (ModelSim SE/PE), 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 RTL Functional Simulation in Verilog HDL (ModelSim-Altera)

To perform RTL functional simulation for Stratix IV devices in Verilog HDL (ModelSim-Altera), 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 RTL Functional Simulation in Verilog HDL (ModelSim SE/PE)

To perform RTL functional simulation for Stratix IV devices in Verilog HDL (ModelSim SE/PE), 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 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 PE/SE, 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 (ModelSim-Altera), 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 SE/PE)

To perform gate-level timing simulation for Stratix IV devices in VHDL (ModelSim SE/PE), 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 (ModelSim-Altera), 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 SE/PE)

To perform gate-level timing simulation for Stratix IV devices in Verilog HDL (ModelSim SE/PE), 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 ←
```

## Transport Delays

By default, the ModelSim software filters out all pulses that are shorter than the propagation delay between primitives. Turning on the transport delay options in the ModelSim 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 software.

The following ModelSim 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 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 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 Error Message Verification

ModelSim 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 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 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 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 software, perform the following steps:

1. In the ModelSim 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 or ModelSim-Altera software. If you do not exit the software, the ModelSim 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 not readable. It is used for generating the waveform view through ModelSim.

To view a waveform from a \*.wlf file through ModelSim, perform the following steps:

1. Type `vsim` on the command line. The **ModelSim** dialog box appears.
2. On the View 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 directly. The \*.vcd file must first be converted to a \*.wlf file.

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


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


2. After you convert the `.vcd` file to a `.wlf` file, follow the procedures for viewing a waveform from a `.wlf` file through ModelSim.

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 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

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 (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 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 ←
```

### Command Prompt

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

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

## Generating a Gate-Level Timing Simulation Netlist for ModelSim

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)" ←`

- `set_global_assignment -name EDA_SIMULATION_TOOL \`  
`"ModelSim-Altera (VHDL)"` ↵
- `set_global_assignment -name EDA_SIMULATION_TOOL \`  
`"ModelSim (Verilog)"` ↵
- `set_global_assignment -name EDA_SIMULATION_TOOL \`  
`"ModelSim (VHDL)"` ↵

### Command Line

Generate a simulation output file for the ModelSim 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 ↵
```

## 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.

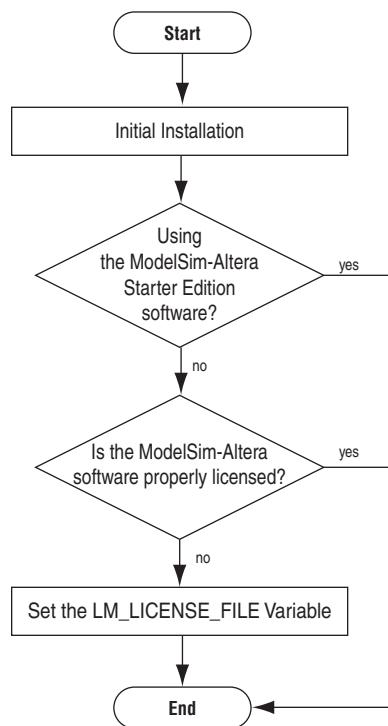
You can obtain a license for the ModelSim-Altera Subscription Edition software from the Altera website at [www.altera.com](http://www.altera.com). Get licensing information for the Mentor Graphics ModelSim software directly from Mentor Graphics. Refer to [Figure 3-2](#) for the set-up process.



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 3-2. 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 and ModelSim-Altera simulation software within the Altera FPGA design flow enables Altera software users to easily and accurately perform RTL 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 and ModelSim-Altera software helps ensure design functionality and success and, ultimately, a quick time-to-market.

## Referenced Documents

This chapter references the following documents:

- *Altera Software Installation and Licensing* manual
- *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*
- *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*

- *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*
- *Quartus II Classic Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*
- *Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*
- *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*

## Document Revision History


Table 3-5 shows the revision history for this chapter.

**Table 3-5.** Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2009 v9.1.0	<ul style="list-style-type: none"> <li>■ Removed NativeLink information and referenced new <i>Simulating Designs with EDA Tools</i> chapter</li> <li>■ Added Stratix IV transceiver simulation section</li> <li>■ Reformatted transceiver simulation sections</li> <li>■ Text edits throughout chapter</li> </ul>	Updated for the Quartus II software version 9.1 release.
March 2009 v9.0.0	<p>Added the following sections:</p> <ul style="list-style-type: none"> <li>■ “Compile Libraries Using the EDA Simulation Library Compiler” on page 2-17</li> <li>■ “Generate Simulation Script from EDA Netlist Writer” on page 2-77</li> <li>■ “Viewing a Waveform from a .wlf File” on page 2-78</li> </ul> <p>Updated the following:</p> <ul style="list-style-type: none"> <li>■ Table 2-1, Table 2-2, Table 2-5, Table 2-6, Table 2-7, Table 2-8, Table 2-9, Table 2-10</li> <li>■ Figure 2-4 on page 2-81</li> <li>■ All sections titled “Loading the Design”</li> </ul>	Updated for the Quartus II software version 9.0 release.

**Table 3-5.** Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2008 v8.1.0	Updated the following: <ul style="list-style-type: none"> <li>■ Table 2-2, Table 2-3, Table 2-4, Table 2-5, Table 2-6</li> <li>■ Removed <code>--zero_ic_delays</code> from <code>quartus_sta</code> option in “Generate Post-Synthesis Simulation Netlist Files” on page 2-11</li> <li>■ Removed steps to include the library when the simulation is run in VHDL mode from all procedures; this is no longer necessary</li> <li>■ Added information about the Altera Simulation Library Compiler throughout the chapter</li> <li>■ Added “Compile Libraries Using the Altera Simulation Library Compiler” on page 2-15</li> <li>■ Added “Disabling Simulation” on page 2-72</li> <li>■ Minor editorial updates</li> <li>■ Updated entire chapter using 8½” × 11” chapter template</li> </ul>	Updated for the Quartus II software version 8.1 release.
May 2008 v8.0.0	<ul style="list-style-type: none"> <li>■ Updated “Altera Design Flow with ModelSim-Altera or ModelSim Software” on page 2-3</li> <li>■ Updated “Simulation Libraries” on page 2-4</li> <li>■ Updated “Simulation Netlist Files” on page 2-11</li> <li>■ Updated “Perform Simulation Using ModelSim-Altera Software” on page 2-15</li> <li>■ Updated “Perform Simulation Using ModelSim Software” on page 2-33</li> <li>■ Updated “Simulating Designs that Include Transceivers” on page 2-57</li> <li>■ Updated “Using the NativeLink Feature with ModelSim-Altera or ModelSim Software” on page 2-63</li> <li>■ Updated “Generating a Timing VCD File for PowerPlay” on page 2-68</li> </ul>	Updated for the Quartus II software version 8.0.

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

