



# 1. Mentor Graphics ModelSim Support

QII53001-5.1.0

## Introduction

An Altera® software subscription includes a license for the ModelSim-Altera software on a PC or UNIX platform. The ModelSim-Altera software can be used to perform functional register transfer level (RTL), post-synthesis, and gate-level timing simulations for either Verilog HDL or VHDL designs that target an Altera FPGA. This chapter provides detailed instructions on how to simulate your design in the ModelSim-Altera version or the Mentor Graphics® ModelSim® software version. This chapter gives you details on the specific libraries that are needed for a functional RTL simulation or a gate-level timing simulation.

This document describes using ModelSim-Altera software version 6.0e and the Mentor Graphics ModelSim software version 6.0e. It also contains references to features available in the Altera Quartus® II software version 5.1. For more information on the current Quartus II software version, refer to the Altera website available at [www.altera.com/quartus](http://www.altera.com/quartus).

## Background

The ModelSim-Altera software version 6.0e is included with your Altera software subscription, and can be licensed for the PC, Solaris, HP-UX, or Linux platforms to support either Verilog HDL or VHDL hardware description language (HDL) simulation. The ModelSim-Altera software supports VHDL or Verilog functional RTL, post-synthesis, and gate-level timing simulations for all Altera devices. The ModelSim-Altera simulator has an instance limit of 250 instances for each design that you simulate.

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

**Table 1–1. Comparison of ModelSim Software Versions (Part 1 of 2)**

Product Feature	ModelSim SE	ModelSim PE	ModelSim-Altera
VHDL, Verilog HDL, mixed-HDL support	Optional	Optional	Supports only single-HDL simulation
Complete HDL debugging environment	✓	✓	✓
Industry-standard scripting	✓	✓	✓
Flexible licensing	✓	optional	✓

Product Feature	ModelSim SE	ModelSim PE	ModelSim-Altera
Verilog PLI (1) 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 Format File annotation	✓	✓	✓ (2)
Advanced debugging features and language-neutral licensing	✓		
Customizable, user-expandable graphical user interface (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

Note to Table 1–1:

(1) See [www.altera.com/products/software/pld/products/partners/eda-ms.html](http://www.altera.com/products/software/pld/products/partners/eda-ms.html).

(2) Standard Delay Format File annotation is supported only for Altera libraries.

## Software Compatibility

Table 1–2 shows which ModelSim-Altera 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 on ModelSim-Altera licensing set-up, refer to “Software Licensing & Licensing Set-Up” on page 1–44.

ModelSim-Altera Software	Quartus II Software (1)
ModelSim-Altera software version 6.0e	Quartus II software version 5.1
ModelSim-Altera software version 6.0c	Quartus II software version 5.0
ModelSim-Altera software version 5.8.e ModelSim-Altera software version 5.8.d	Quartus II software version 4.2

Note to Table 1–2:

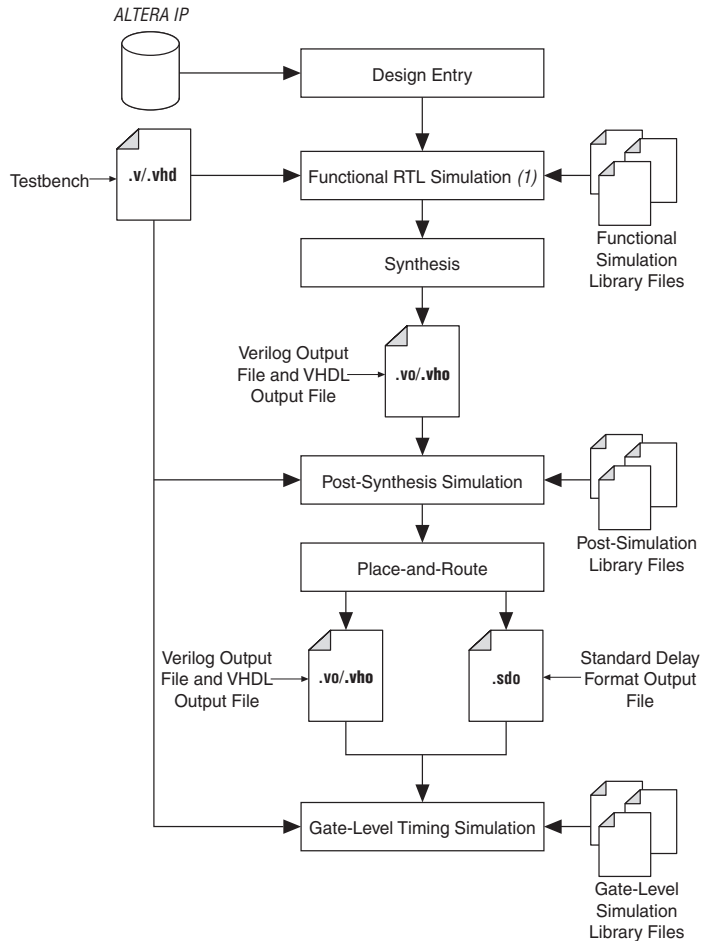
(1) Updated ModelSim-Altera precompiled libraries are available for download on Altera’s website for each release of the Quartus II service pack.

## Altera Design Flow with ModelSim-Altera Software

Figure 1-1 illustrates an Altera design flow using the ModelSim-Altera software or Mentor Graphics ModelSim software version.

- Functional RTL simulations
- Post-synthesis simulations
- Gate-level timing simulations
- Using the NativeLink feature with ModelSim

**Figure 1-1. Altera Design Flow with ModelSim-Altera & Quartus II Software**



**Note to Figure 1-1:**

- (1) If you are performing a functional simulation through NativeLink, you must complete analysis and elaboration first.

## Functional RTL Simulation

A functional RTL simulation is performed before a gate-level simulation or post-synthesis simulation. Functional RTL simulation verifies the functionality of the design before synthesis and place-and-route. This section provides detailed instructions on how to perform a functional RTL simulation in the ModelSim-Altera software, and highlights some of the differences in performing similar steps in the Mentor Graphics ModelSim software versions for Verilog HDL and VHDL designs.

### Functional Simulation Libraries

Pre-compiled libraries are available for functional simulation with the ModelSim-Altera software. These libraries include the **lpm** library and the **altera\_mf** library. To create these libraries for simulation with the ModelSim SE/PE software, compile the library files described in the following sections.

#### *lpm Simulation Models*

To simulate designs containing lpm functions, use the following functional simulation models:

- **220model.v** (for Verilog HDL)
- **220pack.vhd** and **220model.vhd** (for VHDL)



When you are simulating a design that uses VHDL-1987, use the **220model\_87.vhd** model file.

Table 1-3 shows the location of these simulation model files and precompiled libraries in the Quartus II software and the ModelSim-Altera software.

**Table 1-3. Location of lpm Simulation Models Files and Pre-Compiled Libraries**

Software	Location
Quartus II	<Quartus II installation directory>\eda\sim_lib\ (1)
ModelSim-Altera	<ModelSim-Altera installation directory>\altera\<HDL>\220model\ (2), (3)

#### Notes to Table 1-3:

- (1) For ModelSim SE/PE, compile the files provided with the Quartus II software.
- (2) For ModelSim-Altera, use the pre-compiled libraries for simulation.
- (3) <HDL> can be either Verilog HDL or VHDL.



For more information on LPM functions, refer to the Quartus II Help.

### Altera Megafunction Simulation Models

To simulate a design that contains Altera megafunctions, use the following simulation models:

- `altera_mf.v` (for Verilog HDL)
- `altera_mf.vhd` and `altera_mf_components.vhd` (for VHDL)



When you are simulating a design that uses VHDL-1987, use `altera_mf_87.vhd`.

Table 1–4 shows the location of these simulation files and precompiled libraries in the Quartus II software and the ModelSim-Altera software.

Software	Location
Quartus II	<Quartus II installation directory>\eda\sim_lib\ (1)
ModelSim-Altera	<ModelSim-Altera installation directory>\altera\<HDL>\altera_mf (2), (3)

**Notes to Table 1–4:**

- (1) For ModelSim SE/PE, compile the files provided with the Quartus II software.
- (2) For ModelSim-Altera, use the pre-compiled libraries for simulation.
- (3) <HDL> can be either Verilog HDL or VHDL.

The following Altera megafunctions require device atom libraries to perform a functional simulation in a third-party simulator:

- `altclkbuf`
- `altdqs`
- `altdio_in`
- `altdio_out`
- `altdio_bidir`
- `altufm_none`
- `altufm_parallel`
- `altufm_spi`
- `altmemmult`
- `altremote_update`

The device atom library files are located in the following directory:

<Quartus II installation directory>/eda/sim\_lib

### Low-Level Primitives User Guide

You can simulate a design that contains low-level Altera primitives with the following simulation models:

- **altera\_primitives.v** (for Verilog HDL)
- **altera\_primitives.vhd** and **altera\_primitives\_components.vhd** (for VHDL)

Table 1-5 shows the location of these simulation library files and precompiled libraries in the Quartus II software and the ModelSim-Altera software.

Software	Location
Quartus II	<Quartus II installation directory>\eda\sim_lib (1)
ModelSim-Altera	<ModelSim-Altera installation directory>\altera\<HDL>\altera (2), (3)

**Notes to Table 1-5:**

- (1) For ModelSim SE/PE, compile the files provided with the Quartus II software.
- (2) For ModelSim-Altera, use the pre-compiled libraries for simulation.
- (3) <HDL> can be either Verilog HDL or VHDL.

## Simulating VHDL Designs

Use the following instructions to perform a functional RTL simulation for VHDL designs in the ModelSim software.

The steps in the following section assume you have already created a ModelSim project.

### Create Simulation Libraries

Simulation libraries are required to simulate a design that contains an lpm function or an Altera megafunction. If you are using the Mentor Graphics ModelSim software version, you must create the simulation libraries and link them to your design correctly.



Creating a simulation library is not required if you are using the ModelSim-Altera software.

### Create Simulation Libraries Using the ModelSim GUI

Perform the following steps to create simulation libraries:

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



The name of the libraries should be **altera\_mf** (for Altera megafunctions) and **lpm** (for LPM and MegaWizard® Plug-in Manager-generated entities).

4. Click **OK**.

### Create Simulation Libraries Using the ModelSim Command Prompt

Type the following commands at the ModelSim command prompt:

```
vlib altera_mf ↵
vmap altera_mf altera_mf ↵
vlib lpm ↵
vmap lpm lpm ↵
vlib altera ↵
vmap altera altera ↵
```

### *Compile Simulation Models into Simulation Libraries*

The following steps are not required for the ModelSim-Altera software.

### Compile Simulation Models into Simulation Libraries Using the ModelSim GUI

Perform the following steps to compile simulation models into simulation libraries:

1. On the File menu, point to Add to Project and click **Existing File**.
2. Browse to the *<Quartus II installation directory>/eda/sim\_lib* and add the necessary simulation model files to your project.



The **altera\_mf.vhd** model file should be compiled into the **altera\_mf** library. The **220pack.vhd** and **220model.vhd** model file should be compiled into the **lpm** library.

3. In the Workspace window, select the simulation model file, and on the View menu, click **Properties**.

4. Choose the correct library from the **Compile to Library** list.
5. Click **OK**.
6. On the Compile menu, click **Compile selected**.

### Compile Simulation Models into Simulation Libraries at the ModelSim Command Prompt

Type the following command lines at the ModelSim command prompt:

```
vcom -work altera_mf <Quartus II installation directory>/eda/sim_lib/altera_mf_components.vhd ↵
vcom -work altera_mf <Quartus II installation directory>/eda/sim_lib/altera_mf.vhd ↵
vcom -work lpm <Quartus II installation directory>/eda/sim_lib/220pack.vhd ↵
vcom -work lpm <Quartus II installation directory>/eda/sim_lib/220model.vhd ↵
vcom -work altera <Quartus II installation directory>/eda/sim_lib/altera_primitives_components.vhd ↵
vcom -work altera <Quartus II installation directory>/eda/sim_lib/altera_primitives.vhd ↵
```

### *Compile Testbench & Design Files into Work Library*

Compile a testbench and design files into a work library on the Compile menu by clicking **Compile All** or by clicking the **Compile All** toolbar icon.

### Compile Testbench & Design Files into Work Library Using the ModelSim Command Prompt

Type the following command at the ModelSim command prompt:

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



Resolve compile-time errors before proceeding to “**Loading the Design**”.

### *Loading the Design*

Perform the following steps to load a design:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box is shown.
2. Expand the work library in the **Start Simulation** dialog box.
3. Select the top-level design unit (your testbench).
4. In the **Resolution** list, select **ps**.
5. Click **OK**.

### Loading the Design Using the ModelSim Command Prompt

Type the following command at the ModelSim command prompt:

```
vsim work.<my_testbench> -t ps ↵
```

### Running the Simulation

Perform the following steps to run a simulation:

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. At the ModelSim command prompt, type the following:

```
run <time period> ↵
```

### Running the Simulation Using the ModelSim Command Prompt

Type the following command line at the ModelSim command prompt:

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

## Simulating Verilog HDL Designs

The following instructions provide step-by-step instructions to perform functional RTL simulation for Verilog HDL designs in the ModelSim software.



The following steps assume you have already created a ModelSim project.

### Create Simulation Libraries

Simulation libraries are needed to properly simulate a design that contains an lpm function or an Altera megafunction. If you are using the Mentor Graphics ModelSim software version, you need to create the simulation libraries and correctly link them to your design.



Creating a simulation library is not required for the ModelSim-Altera software.

### Create Simulation Libraries Using the ModelSim GUI

Perform the following steps to create simulation libraries:

1. On the File menu, point to **New** and click **Library**. The **Create a New Library** dialog box is shown.
2. Select **a new library and a logical mapping to it**.



Use the **altera\_mf** library (for Altera megafunctions) and the **lpm** library (for lpm and Megawizard-generated entities).

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

### Create Simulation Libraries Using the ModelSim Command Prompt

Type the following command lines at the ModelSim command prompt:

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

### *Compile Simulation Models into Simulation Libraries*

The following steps are not required for the ModelSim-Altera software.

### Compile Simulation Models into Simulation Libraries Using the ModelSim GUI

Perform the following steps to compile simulation models into simulation libraries:

1. On the File menu, point to **Add to Project** and click **Existing File**.
2. Browse to the *<Quartus II installation directory>/eda/sim\_lib* and add the necessary simulation model files to your project.



Compile the **altera\_mf.v** into the **altera\_mf** library. Compile the **220model.v** into the **lpm** library.

3. Select the simulation model file and on the View menu, click **Properties**.
4. Choose the correct library from the **Compile to Library** list.

5. Click **OK**.
6. On the Compile menu, click **Compile selected**.

### Compile Simulation Models into Simulation Libraries Using the ModelSim Command Prompt

Type the following command lines at the ModelSim command prompt:

```
vlog -work altera_mf <Quartus II installation directory>/eda/sim_lib/altera_mf.v ←
vlog -work lpm <Quartus II installation directory>/eda/sim_lib/220model.v ←
vlog -work altera <Quartus II installation directory>/eda/sim_lib/altera_primitives.v ←
```

### *Compile Testbench & Design Files into Work Library*

Compile a testbench and design files into a work library on the Compile menu by clicking **Compile All** or clicking the **Compile All** toolbar icon.

### Compile Testbench & Design Files into Work Library Using the ModelSim Command Prompt

Type the following command at the ModelSim command prompt:

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



Resolve compile-time errors before proceeding to “[Loading the Design](#)” below.

### *Loading the Design*

Perform the following steps to load a design:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box is shown.
2. Click the **Libraries** tab.
3. In the **Search Libraries** box, click **Add**.
4. Specify the location to the **lpm** or **altera\_mf** simulation libraries.



If you are using the ModelSim-Altera version, refer to [Table 1–3](#) and [Table 1–4](#) for the location of the precompiled simulation libraries. If you are using the Mentor Graphics ModelSim software version, browse to the library that was created earlier.

5. In the **Load Design** dialog box, click the **Design** tab and expand the work library.

6. Select the top-level design unit (your testbench).
7. In the **Resolution** list, select **ps**.
8. Click **OK**.

### Loading a Design Using the ModelSim Command Prompt

Type the following command at the ModelSim command prompt:

```
vsim -L <location of the altera_mf library> -L <location of the lpm library> work.<my_testbench> -t ps ↵
```

### Running the Simulation

Perform the following steps to run a simulation:

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. At the ModelSim command prompt, type the following:

```
run <time period> ↵
```

### Running the Simulation Using the ModelSim Command Prompt

Type the following commands at the ModelSim command prompt:

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

### Verilog HDL Functional RTL Simulation with Altera Memory Blocks

Both ModelSim software products support simulating Altera memory megafunctions initialized with Hexadecimal (Intel-Format) File (**.hex**) or RAM initialization files (**.rif**).

Although synthesis is able to read a Memory Initialization File (**.mif**), this memory file is not supported with third-party tools and must be converted to either a Hexadecimal (Intel-Format) File or RAM Initialization File.

[Table 1-6](#) summarizes the different types of memory initialization file formats that are supported with each RTL language.

**Table 1–6. Simulation Support for Memory Initialization Files**

File	Verilog HDL	VHDL
Hexadecimal (Intel-Format) File	Yes (1)	Yes
Memory Initialization File	No	No
RAM Initialization File	Yes (2)	No

**Notes to Table 1–6:**

- (1) For memories and library files from the Quartus II software version 5.0 and earlier, you must use a PLI library containing the `convert_hex2ver` function.
- (2) Requires the `USE_RIF` macro to be defined, described later in this section.

To simulate your design by converting your Memory Initialization File into either Hexadecimal (Intel-Format) File or RAM Initialization File, perform the following steps:

1. Convert a Memory Initialization File to a Hexadecimal (Intel-Format) File or RAM Initialization File in the Quartus II software.

#### Converting a Memory Initialization File to a Hexidecimal (Intel-Format) File

- a. Open the Memory Initialization File. On the File menu, click **Save As**. The **Save As** dialog box is shown.
- b. In the **Save as type** list, select **Hexadecimal (Intel-Format) File (\*.hex)**.
- c. Click **OK**.

#### Convert a Memory Initialization File to a RAM Initialization File

- a. Open the Memory Initialization File and on the File menu, click **Export**. The **Export** dialog box is shown.
- b. In the **Save as type** list, select **RAM Initialization File (\*.rif)**.
- c. Click **OK**.

Alternatively, you can convert a Memory Initialization File to a RAM Initialization File using the `mif2rif.exe` utility located in the `<Quartus II installation>/bin` directory.

```
mif2rif <mif_file> <rif_file> ↵
```

2. Modify the HDL file generated by the MegaWizard Plug-In Manager.

The Altera memory custom megafunction variation file includes the `lpm_file` parameter, for LPM memories such as LPM\_ROM, or `init_file` for Altera specific memories such as an `altsyncram`, to point to the initialization file.

In a text editor, open the custom megafunction variation file and edit the `lpm_file` or `init_file` to point to the Hexadecimal (Intel-Format) File or RAM Initialization File, as shown in the following example:

```
lpm_ram_dp_component.lpm_file = "<path to HEX/RIF>"
```

3. Compile the functional library files with compiler directives

If you use a Hexadecimal (Intel-Format) File, then no compiler directives are required. If you use a RAM Initialization File, then you must define the `USE_RIF` macro when compiling the model library files. For example, you should enter the following when compiling the `altera_mf` library when RAM Initialization File memory initialization files are used:

```
vlog -work altera_mf altera_mf.v +define+USE_RIF=1
```



For the Quartus II software versions 5.0 and earlier, you must define the `NO_PLI` macro instead of `USE_RIF`. The `NO_PLI` macro is forwards compatible with the Quartus II software.

## Post-Synthesis Simulation

A post-synthesis simulation verifies the functionality of a design after synthesis has been performed. You can create a post-synthesis netlist in the Quartus II software and use this netlist to perform a post-synthesis simulation in ModelSim. Once the post-synthesis version of the design has been verified, the next step is to place-and-route the design in the target device using the Quartus II Fitter.

### *Generating a Post-Synthesis Simulation Netlist*

The following steps describe the process of generating a post-synthesis simulation netlist in the Quartus II software:

1. Perform Analysis and Synthesis. On the Processing menu, point to Start and click **Start Analysis & Synthesis**.
2. Turn on the **Generate Netlist for Functional Simulation Only** option by performing the following steps:
  - a. On the Assignments menu, click **EDA Tool Settings**. The **Settings** dialog box is shown.
  - b. In the **Tool Name** list:

If you are using the ModelSim-Altera software, select **ModelSim-Altera (VHDL)** or **ModelSim-Altera (Verilog HDL)**.

If you are using the Mentor Graphics ModelSim software, select **ModelSim (Verilog HDL)** or **ModelSim (VHDL)**.
  - c. Turn on **Generate Netlist for Functional Simulation Only**.
  - d. Click **OK**.
3. Run the EDA Netlist Writer. On the Processing menu, point to Start and click **Start EDA Netlist Writer**.

During the EDA Netlist Writer stage, the Quartus II software produces a Verilog Output File (.vo) or VHDL Output File (.vho) that can be used for post-synthesis simulations in the ModelSim software. This netlist file is mapped to architecture-specific primitives. No timing information is included at this stage. The resulting netlist is located in the *<project directory>/simulation/modelsim* directory.

## Simulating VHDL Designs

The following instructions help you perform a post-synthesis simulation for a VHDL design in the ModelSim software. The following steps assume you have already created a ModelSim project.

### *Create Simulation Libraries*

Simulation libraries are required to simulate a design that is mapped to post-synthesis primitives. If you are using the Mentor Graphics ModelSim software version, you must create the simulation libraries and correctly link them to your design.



This process is not required with the ModelSim-Altera version because a set of pre-compiled libraries is installed with the software.

### **Create Simulation Libraries Using the ModelSim GUI**

Perform the following steps to create simulation libraries:

1. On the File menu, click **New Library**. The **Create a New Library** dialog box is shown.
2. Select a new **Library and a logical linking to it**.
3. In the **Library Name** box, type the name of the newly created library.
4. Click **OK**.

### **Create Simulation Libraries Using the ModelSim Command Prompt**

Type the following commands to create simulation libraries:

```
vlib <library name> ←  
vmap <library name> <device family name> ←
```

### *Compile Simulation Models into Simulation Libraries*

This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software.

### Compile Simulation Models into Simulation Libraries Using the ModelSim GUI

Perform the following steps to compile simulation models into simulation libraries:

1. On the File menu, point to Add to Project and click **Existing File**.
2. Browse to the *<Quartus II installation directory>/eda/sim\_lib* directory and add the necessary gate-level simulation files to your project.
3. Select the simulation model file and on the View menu, click **Properties**.
4. In the **Compile to Library** list, select the correct library.
5. Click OK.
6. On the Compile menu, click **Compile selected**.

### Compile Simulation Models into Simulation Libraries Using the ModelSim Command Prompt

Type the following command lines at the ModelSim command prompt:

```
vcom -work <device family name> <Quartus II installation directory>
/eda/sim_lib/<device family name>_atoms.vhd ←
```

```
vcom -work <device family name> <Quartus II installation directory>
/eda/sim_lib/<device family name>_components.vhd ←
```

### *Compile Testbench & VHDL Output File into Work Library*

To compile testbench and VHDL Output Files into a work library, choose **Compile All** (Compile menu) or click the **Compile All** toolbar icon.

### Compile Testbench & VHDL Output File into Work Library Using ModelSim Command Prompt

Type the following command line at the ModelSim command prompt:

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



Resolve any compilation errors before proceeding to “Loading the Design”.

### *Loading the Design*

Perform the following steps to load a design:

1. On the Simulate menu, click **Simulate**.
2. In the **Library** list (**Design** tab), select the **work** library.
3. In the **Simulate** dialog box, expand the **work** library and select the top-level design unit (your test bench).
4. Click **OK**.

### **Loading the Design Using the ModelSim Command Prompt**

Type the following command line at the ModelSim command prompt:

```
vsim work.<my testbench> -t 1ps↵
```



Set the time scale resolution to 1 ps when simulating Altera FPGA designs.

### *Running the Simulation*

Perform the following steps to run a simulation:

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. At the ModelSim command prompt, type the following:

```
run <time period> ↵
```

### **Running the Simulation Using the ModelSim Command Prompt**

Type the following command lines at the ModelSim command prompt:

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

## **Simulating Verilog HDL Designs**

The following provides step-by-step instructions on performing post-synthesis simulation for Verilog HDL designs in the ModelSim software.

### *Create Simulation Libraries*

The following steps assume you have already created a ModelSim project.



This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software. If you are using the Mentor Graphics ModelSim software version, you need to create the simulation libraries and correctly link them to your design.

#### **Create Simulation Libraries Using the ModelSim GUI**

Perform the following steps to create simulation libraries:

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



The name of the libraries should be **altera\_mf** (for Altera megafunctions) and **lpm** (for lpm and MegaWizard® Plug-in Manager-generated entities).

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

#### **Create Simulation Libraries Using the ModelSim Command Prompt**

Type the following command lines at the ModelSim command prompt:

```
vlib <library name> ←  
vmap <library name> <device family name> ←
```

### *Compile Simulation Models into Simulation Libraries*

This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software.

### Compile Simulation Models into Simulation Libraries Using the ModelSim GUI

Perform the following steps to compile simulation models into simulation libraries:

1. On the File menu, click **Add to Project**, then select **Existing File**.
2. Browse to the *<Quartus II installation directory>/eda/sim\_lib* directory and add the necessary simulation model files to your project.
3. Select the simulation model file and on the View menu, click **Properties**.
4. Specify the correct library in the **Compile to Library** box.

### Compile Simulation Models into Simulation Libraries Using the ModelSim Command Prompt

Type the following command line at the ModelSim command prompt:

```
vlog -work <device family name> <Quartus II installation  
directory> /eda/sim_lib/<device family name>_atoms.v ←
```

### *Compile Testbench & Verilog Output File into Work Library*

To compile testbench and Verilog Output Files into a work library, choose **Compile All** (Compile menu) or click the **Compile All** toolbar icon.

### Compile Testbench & Verilog Output File into Work Library Using the ModelSim Command Prompt

Type the following command line at the ModelSim command prompt:

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



Resolve any compilation errors before proceeding to *Loading the Design*.

### *Loading the Design*

Perform the following steps to load a design:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box is shown.
2. Click the **Libraries** tab.
3. In the **Search Libraries** box, click **Add**.

- Specify the location to the **lpm** or **altera\_mf** simulation libraries.



If you are using the ModelSim-Altera version, refer to [Table 1-3](#) and [Table 1-4](#) for the location of the precompiled simulation libraries. If you are using the Mentor Graphics ModelSim software version, browse to the library that was created earlier.

- In the **Load Design** dialog box, click the **Design** tab and expand the work library.
- Select the top-level design unit (your testbench).
- In the **Resolution** list, select **ps**.
- Click **OK**.

#### Loading the Design Using the ModelSim Command Prompt

Type the following command line at the ModelSim command prompt:

```
vsim -L <gate-level simulation library> work.<my_testbench> -t 1ps ↵
```



Set the time scale resolution to 1 ps when simulating Altera FPGA designs.

#### Running the Simulation

Perform the following steps to run a simulation:

- In 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.
- At the ModelSim command prompt, type the following command:

```
run <time period> ↵
```

### Running the Simulation Using the ModelSim Command Prompt

Type the following commands at the ModelSim command prompt:

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

## Gate-Level Timing Simulation

Gate-level timing simulation is a post place-and-route simulation to verify the operation of the design after the worst-case timing delays have been calculated. This section provides detailed instructions on how to perform gate-level timing simulation in the ModelSim-Altera software and highlights differences in performing similar steps in the Mentor Graphics ModelSim software versions for VHDL and Verilog HDL designs.

### Quartus II Software Output Files for use in the ModelSim-Altera Software

To perform gate-level timing simulation, the ModelSim-Altera software requires information on how the design was placed into device-specific architectural blocks. The Quartus II software provides this information in the form of a Verilog Output File for Verilog HDL designs and a VHDL Output File for VHDL designs. The accompanying timing information is stored in the Standard Delay Format File (.sdf), which annotates the delay for the elements found in the Verilog Output File or VHDL Output File.

To generate the Verilog Output File or VHDL Output Files, perform the following steps:

1. On the Assignments menu, click **EDA Tool Settings**.
2. In the **Tool Name** list:
  - a. If you are using the ModelSim-Altera software, select **ModelSim-Altera (VHDL)** or **ModelSim-Altera (Verilog HDL)**.
  - b. If you are using the Mentor Graphics ModelSim software, select **ModelSim (Verilog HDL)** or **ModelSim (VHDL)**.
3. Click **OK**.
4. Compile the project.
5. The Quartus II output files are located in the *<full path to project>*\simulation\ModelSim\ directory.

## Gate Level Simulation Libraries

Table 1–7 provides a description of the ModelSim-Altera precompiled device libraries.

Library	Description
<b>stratixii</b>	Precompiled library for Stratix II device designs.
<b>stratixiigx</b>	Precompiled library for Stratix II GX device designs.
<b>stratixiigx_hssi</b>	Precompiled library for Stratix II GX device designs using the Gigabit Transceiver Block (a1t2gxb megafunction). This precompiled library is required for both functional and timing simulations.
<b>stratix</b>	Precompiled library for Stratix device designs.
<b>stratixgx</b>	Precompiled library for Stratix GX device designs.
<b>stratixgx_gxb</b>	Precompiled library for Stratix GX device designs using the Gigabit Transceiver Block. This precompiled library should be used for post-fit (timing) simulations.
<b>altgxb</b>	Precompiled library for Stratix GX device designs that include the a1tgxb megafunction. This precompiled library should be used for functional simulations.
<b>cycloneii</b>	Precompiled library for Cyclone™ II device designs.
<b>cyclone</b>	Precompiled library for Cyclone device designs.
<b>maxii</b>	Precompiled library for MAX® II device designs.
<b>max</b>	Precompiled library for MAX 7000 and MAX 3000 device designs.
<b>apexii</b>	Precompiled library for APEX™ II device designs.
<b>apex20k</b>	Precompiled library for APEX 20K device designs.
<b>apex20ke</b>	Precompiled library for APEX 20KC, APEX 20KE, and Excalibur™ device designs.
<b>mercury</b>	Precompiled library for Mercury™ device designs.
<b>flex10ke</b>	Precompiled library for FLEX® 10KE and ACEX® 1K device designs.
<b>flex6000</b>	Precompiled library for FLEX 6000 device designs.

Table 1–8 shows the location of the timing simulation libraries in the ModelSim-Altera software for Verilog HDL.

Library	Verilog HDL
<b>stratixii</b>	<ModelSim-Altera installation directory>\altera\verilog\stratixii\
<b>stratixiigx</b>	<ModelSim-Altera installation directory>\altera\verilog\stratixiigx\
<b>stratixiigx_hssi</b>	<ModelSim-Altera installation directory>\altera\verilog\stratixiigx_hssi\ (1)
<b>stratix</b>	<ModelSim-Altera installation directory>\altera\verilog\stratix\

**Table 1–8. Location of Timing Simulation Libraries for ModelSim-Altera for Verilog HDL**

Library	Verilog HDL
stratixgx	<ModelSim-Altera installation directory>\altera\verilog\stratixgx\
stratixgx_gxb	<ModelSim-Altera installation directory>\altera\verilog\stratixgx_gxb\
cycloneii	<ModelSim-Altera installation directory>\altera\verilog\cycloneii\
cyclone	<ModelSim-Altera installation directory>\altera\verilog\cyclone\
maxii	<ModelSim-Altera installation directory>\altera\verilog\maxii\
max	<ModelSim-Altera installation directory>\altera\verilog\max\
apexii	<ModelSim-Altera installation directory>\altera\verilog\apexii\
apex20k	<ModelSim-Altera installation directory>\altera\verilog\apex20k\
apex20ke	<ModelSim-Altera installation directory>\altera\verilog\apex20ke\
mercury	<ModelSim-Altera installation directory>\altera\verilog\mercury\
flex10ke	<ModelSim-Altera installation directory>\altera\verilog\flex10ke\
flex6000	<ModelSim-Altera installation directory>\altera\verilog\flex6000\

Note to [Table 1–8](#):

- (1) The stratixgx\_hssi precompiled library is required for functional and timing simulations.

[Table 1–9](#) shows the location of the timing simulation libraries in the ModelSim-Altera software for VHDL.

**Table 1–9. Location of Timing Simulation Library Files for ModelSim-Altera for VHDL (Part 1 of 2)**

Library	VHDL
stratixii	<ModelSim-Altera installation directory>\altera\vhdl\stratixii\
stratixiigx	<ModelSim-Altera installation directory>\altera\vhdl\stratixiigx\
stratixiigx_hssi	<ModelSim-Altera installation directory>\altera\vhdl\stratixiigx_hssi\ (1)
stratix	<ModelSim-Altera installation directory>\altera\vhdl\stratix\
stratixgx	<ModelSim-Altera installation directory>\altera\vhdl\stratixgx\
stratixgx_gxb	<ModelSim-Altera installation directory>\altera\vhdl\stratixgx_gxb\
cycloneii	<ModelSim-Altera installation directory>\altera\vhdl\cycloneii\
cyclone	<ModelSim-Altera installation directory>\altera\vhdl\cyclone\
maxii	<ModelSim-Altera installation directory>\altera\vhdl\maxii\
max	<ModelSim-Altera installation directory>\altera\vhdl\max\
apexii	<ModelSim-Altera installation directory>\altera\vhdl\apexii\
apex20ke	<ModelSim-Altera installation directory>\altera\vhdl\apex20ke\
apex20k	<ModelSim-Altera installation directory>\altera\vhdl\apex20k\
flex10ke	<ModelSim-Altera installation directory>\altera\vhdl\flex10ke\

**Table 1–9. Location of Timing Simulation Library Files for ModelSim-Altera for VHDL (Part 2 of 2)**

Library	VHDL
flex6000	<ModelSim-Altera installation directory>\altera\vhdl\flex6000\
mercury	<ModelSim-Altera installation directory>\altera\vhdl\mercury\

Note to [Table 1–9](#):

- (1) The `stratixiigx_hssi` precompiled library is required for functional and timing simulations.

If you are using the Mentor Graphics ModelSim software version for your timing simulation, libraries are available in the Quartus II software in the <Quartus II installation directory>\eda\sim\_lib\ directory. Mentor Graphics ModelSim software users must use the files provided with the Quartus II software.

## Simulating VHDL Designs

The following section provides step-by-step instructions for performing gate-level timing simulation for VHDL designs.



The following steps assume you have already created a ModelSim project. For additional information, refer to “[Altera Design Flow with ModelSim-Altera Software](#)” on page 1–3.

### Create Simulation Libraries

If you are using the Mentor Graphics ModelSim software version, create the gate-level simulation libraries and correctly link them to your design.



This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software.

### Create Simulation Libraries Using the ModelSim GUI

Perform the following steps to create simulation libraries:

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



The name of the libraries should be **altera\_mf** (for Altera megafunctions) and **lpm** (for lpm and MegaWizard® Plug-in Manager-generated entities).

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



The library name must be one of the library names listed in [Table 1-9](#).

### **Create Simulation Libraries Using the ModelSim Command Prompt**

Type the following command lines at the ModelSim command prompt:

```
vlib <library name> ←  
vmap <library name> <device family name> ←
```

### *Compile Simulation Models into Simulation Libraries*

This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software.

### **Compile Simulation Models into Simulation Libraries Using the ModelSim GUI**

Perform the following steps to compile simulation models into simulation libraries:

1. On the File menu, point to Add to Project and click **Existing File**.
2. Browse to the <Quartus II installation directory>/eda/sim\_lib directory, and add the necessary gate-level simulation files to your project.
3. Select the simulation model file, and on the View menu, click **Properties**.
4. In the **Compile to Library** list, select the correct library.
5. Click **OK**.
6. On the Compile menu, click **Compile selected**.

### Compile Simulation Models into Simulation Libraries Using the ModelSim Command Prompt

Type the following command lines at the ModelSim command prompt:

```
vcom -work <device family name> <Quartus II installation directory>  
/eda/sim_lib/<device family name>_atoms.vhd ←
```

```
vcom -work <device family name> <Quartus II installation directory>  
/eda/sim_lib/<device family name>_components.vhd ←
```

### Compile Testbench & VHDL Output File into Work Library

Compile testbench and VHDL Output Files into a work library on the Compile menu by clicking **Compile All** or clicking the **Compile All** toolbar icon.

### Compile Testbench & VHDL Output File into Work Library Using the ModelSim Command Prompt

Type the following command line at the ModelSim command prompt:

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



Resolve any compilation errors before proceeding to [“Loading the Design” on page 1–27](#).

### Loading the Design

Perform the following steps to load a design:

1. On the Simulate menu, click **Start Simulation**.
2. Click the **SDF** tab, and click **Add**.
3. In the **Add SDF Entry** dialog box, click **Browse** and select the Standard Delay Format Output File (**.sdo**).
4. In the **Apply to Region** dialog box, type in the instance path to which the Standard Delay Format Output File should be applied to. For example, if you are using a testbench exported in the Quartus II software from a Vector Waveform File, then the instance path should be set to `/i1`.



You do not have to choose from the **Delay** list because the Quartus II EDA Netlist Writer generates the Standard Delay Format Output File using the same value in each set of minimum, typical, and maximum timing values. The value is derived from either the fast (minimum) timing model or worst case (maximum) timing model, depending on which timing model was used in the last Timing Analysis. In the standard compilation flow, the Quartus II software writes the Standard Delay Format Output File using timing values from the worst case (maximum) timing model.

If you use the `vsim` executable, add the following option:

```
vsim -sdftyp /il=<path to SDO> filtref_vhd.sdo
```

5. Click **OK**.
6. Click the **Design** tab. In the **Resolution** list, select **ps**.
7. In the **Library** list (**Design** tab), 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. Click **OK**.

#### **Loading a Design Using the ModelSim Command Prompt**

Type the following command line at the ModelSim command prompt:

```
vsim -sdftyp work.<my_testbench> -t ps ←
```

#### *Running the Simulation*

Perform the following steps to run a simulation:

1. In 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. At the ModelSim command prompt, type the following:

```
run <time period> ↵
```

### Running a Simulation Using the ModelSim Command Prompt

Type the following commands at the ModelSim command prompt:

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

## Simulating Verilog HDL Designs

The following provides step-by-step instructions on performing gate-level timing simulation for Verilog HDL designs in the ModelSim-Altera software.



The following steps assume you have already created a ModelSim project. For additional information, refer to “[Altera Design Flow with ModelSim-Altera Software](#)” on page 1–3.

### Create Simulation Libraries

If you are using the Mentor Graphics ModelSim software version, you must create the simulation libraries and correctly link them to your design.



This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software.

### Create Simulation Libraries Using the ModelSim GUI

Perform the following steps to create simulation libraries:

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



The name of the libraries should be **altera\_mf** (for Altera megafunctions) and **lpm** (for lpm and MegaWizard® Plug-in Manager-generated entities).

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

### Create Simulation Libraries Using the ModelSim Command Prompt

Type the following commands at the ModelSim command prompt:

```
vlib <library name> ←  
vmap <library name> <device family name> ←
```

### *Compile Simulation Models into Simulation Libraries*

This process is not required for the ModelSim-Altera version because a set of pre-compiled libraries is created when you install the software.

### Compile Simulation Models into Simulation Libraries Using the ModelSim GUI

Perform the following steps to compile simulation models into simulation libraries:

1. On the File menu, point to Add to Project and click **Existing File**.
2. Browse to the <Quartus II installation directory>/eda/sim\_lib, and add the necessary simulation model files to your project.
3. Select the simulation model file, and on the View menu, click **Properties**.
4. In the **Compile to Library** list, select the correct library.
5. Click **OK**.
6. On the Compile menu, **Compile selected**.

### Compile Simulation Models into Simulation Libraries Using the ModelSim Command Prompt

Type the following command at the ModelSim command prompt:

```
vlog -work <device family name> <Quartus II installation directory> /eda/sim_lib/<device family name>_atoms.v ←
```

### *Compile Testbench & Verilog Output File into Work Library*

Compile a testbench and Verilog Output File into a work library on the Compile menu by clicking **Compile All** or clicking the **Compile All** toolbar icon.

### **Compile Testbench & Verilog Output File into Work Libraries Using the ModelSim Command Prompt**

Type the following command at the ModelSim command prompt:

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



Resolve any compilation errors before proceeding to “Loading the Design” below.

### *Loading the Design*

Perform the following steps to load a design:

1. On the Simulate menu, click **Start Simulation**. The **Start Simulation** dialog box is shown.
2. Click the **Libraries** tab.
3. In the **Search Libraries** box, click **Add**.
4. Specify the location to the **lpm** or **altera\_mf** simulation libraries.



If you are using the ModelSim-Altera version, refer to [Table 1–3](#) and [Table 1–4](#) for the location of the precompiled simulation libraries. If you are using the Mentor Graphics ModelSim software version, browse to the library that was created earlier.

5. In the **Load Design** dialog box, click the **Design** tab and expand the work library.
6. Select the top-level design unit (your testbench).
7. In the **Resolution** list, select **ps**.
8. Click **OK**.



When simulating in Verilog HDL, the Standard Delay Format Output File does not have to be manually specified because in the Quartus II generated Verilog Output File, there is a `$sdf_annotate` task that ModelSim uses to look into the current directory from which VSIM was run from and look for the Standard Delay Format Output File. If your Standard Delay Format Output File is not in the same directory from which you ran VSIM, you can either copy the Standard Delay Format Output File into your current directory or comment out the `$sdf_annotate` line in the Verilog Output File and manually specify the Standard Delay Format Output File.

### Loading the Design Using the ModelSim Command Prompt

Type the following command at the ModelSim command prompt:

```
vsim -L <location of the gate level simulation library> -work.<my_testbench> -t ps ↵
```

### Running the Simulation

Perform the following steps to run a simulation:

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. At the ModelSim command prompt, type the following:

```
run <time period> ↵
```

### Running the Simulation Using the ModelSim Command Prompt

Type the following commands at the ModelSim command prompt:

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

## Simulating Designs that Include Transceivers

If your design includes a Stratix GX or Stratix II GX transceiver, then you must compile additional library files to perform functional or timing simulations.

### Stratix GX Functional Simulation

To perform a functional simulation of your design that instantiates the `altgxb` megafunction, enabling the gigabit transceiver block (GXB) 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, so you must create these libraries to perform a simulation.

#### *Example of Compiling Library Files for Functional Stratix GX Simulation in Verilog HDL*

To compile the libraries necessary to functional simulation of a Verilog HDL design targeting a Stratix GX device, type the following commands at the ModelSim command prompt:

```
vlib work ←  
vlib lpm ←  
vlib sgate ←  
vlib altgxb ←  
vlog -work lpm 220model.v ←  
vlog -work sgate sgate.v ←  
vlog -work altgxb stratixgx_mf.v ←  
vsim -L lpm -L sgate -L altgxb work.<my design> ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, use the precompiled libraries instead of creating new libraries and compiling the library files.

Instead of creating new libraries, you can map the library names to work using the `vmap` command.

### *Example of Compiling Library Files for Functional Stratix GX Simulation in VHDL*

To compile the libraries necessary for functional simulation of a VHDL design targeting a Stratix GX device, type the following commands at the ModelSim command prompt:

```
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 ←  
vsim -L lpm -L sgate -L altgxb work.<my design> ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, you must use the precompiled libraries instead of creating new libraries and compiling the library files.

### **Stratix GX Post-Fit (Timing) Simulation**

Perform a post-fit 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, so you must create these libraries to perform a simulation.

### *Example of Compiling Library Files for Timing Stratix GX Simulation in Verilog HDL*

To compile the libraries necessary to timing simulation of a Verilog HDL design targeting a Stratix GX device, type the following commands at the ModelSim command prompt:

```
vlog -work lpm 220model.v ←  
vlog -work sgate sgate.v ←  
vlog -work stratixgx stratixgx_atoms.v ←  
vlog -work stratixgx_gxb stratixgx_hssi_atoms.v ←  
vsim -L lpm -L sgate -L stratixgx_gxb work.<my design> -t ps \  
+transport_int_delays +transport_path_delays ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, you must use the precompiled libraries instead of creating new libraries and compiling the library files.

### Example of Compiling Library Files for Timing Stratix GX Simulation in VHDL

To compile the libraries necessary to timing simulation of a VHDL design targeting a Stratix GX device, type the following commands at the ModelSim command prompt:

```
vcom -work lpm 220pack.vhd 220model.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 ←
vsim -L lpm -L sgate -L stratixgx -L stratixgx_gxb work.<my design> -t ps \
+transport_int_delays +transport_path_delays ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, you must use the precompiled libraries instead of creating new libraries and compiling the library files.

## Stratix II GX Functional Simulation

To perform a functional simulation of your design that instantiates the `alt2gxb` megafunction, enabling the gigabit transceiver block (GXB) on Stratix II GX devices, compile the `stratixiigx_hssi` model file into the `stratixiigx_hssi` library.



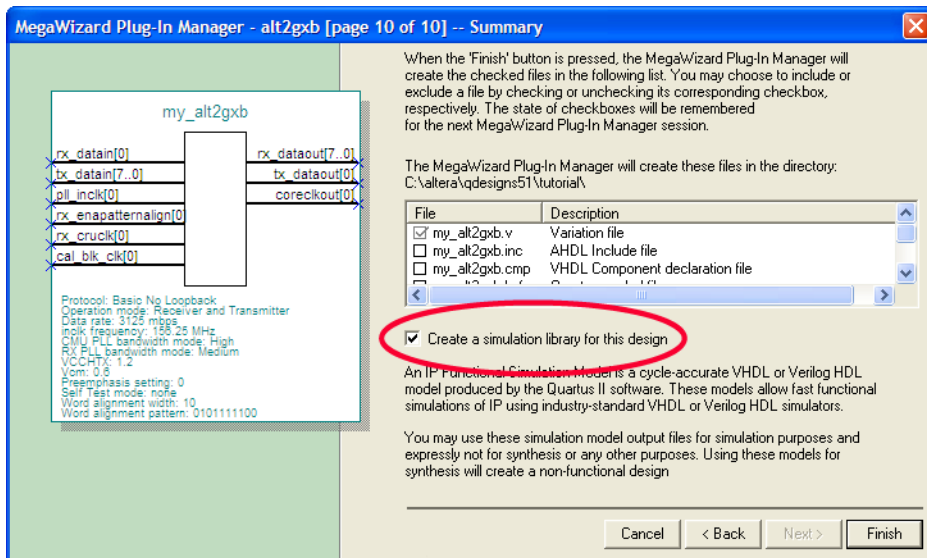
The `stratixiigx_hssi_atoms` model file references the `lpm` and `sgate` libraries, so you must create these libraries to perform a simulation.

Generate a functional simulation netlist by turning on **Create a simulation library for this design** in the last page of the `alt2gxb` Megawizard (Figure 1-2).



The Quartus II-generated `alt2gxb` functional simulation library file references `stratixiigx_hssi_wysiwyg_atoms`.

Figure 1–2. alt2gxb Megawizard



### Example of Compiling Library Files for Functional Stratix II GX Simulation in Verilog HDL

To compile the libraries necessary to functional simulation of a Verilog HDL design targeting a Stratix II GX device, type the following commands at the ModelSim command prompt:

```
vlog -work lpm 220model.v ←
vlog -work sgate sgate.v ←
vlog -work stratixiigx_hssi stratixiigx_hssi_atoms.v ←
vlog -work work <alt2gxb_megafunction variable>.vo ←
vsim -L lpm -L sgate -L stratixgx_hssi work.<my design> ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, you must use the precompiled libraries instead of creating new libraries and compiling the library files.

Instead of creating new libraries, you can map the library names to work using the vmap command.

### *Example of Compiling Library Files for Functional Stratix II GX Simulation in VHDL*

To compile the libraries necessary to functional simulation of a VHDL design targeting a Stratix II GX device, type the following commands at the ModelSim command prompt:

```
vcom -work lpm 220pack.vhd 220model.vhd ←
vcom -work sgate sgate_pack.vhd sgate.vhd ←
vcom -work stratixiigx_hssi stratixiigx_hssi_components.vhd \
stratixiigx_hssi_atoms.vhd ←
vcom -work work <simulation_netlist_alt2gxb>.vho ←
vsim -L lpm -L sgate -L stratixg_hssi work.<my design> ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, you must use the precompiled libraries instead of creating new libraries and compiling the library files.

### **Stratix II GX Post-Fit (Timing) Simulation**

To perform a post-fit 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, so you must create these libraries to perform a simulation.

### *Example of Compiling Library Files for Timing Stratix II GX Simulation in Verilog HDL*

To compile the libraries necessary to timing simulation of a Verilog HDL design targeting a Stratix II GX device, type the following commands at the ModelSim command prompt:

```
vlog -work lpm 220model.v ←
vlog -work sgate sgate.v ←
vlog -work stratixiigx stratixiigx_atoms.vhd ←
vlog -work stratixiigx_hssi stratixiigx_hssi_atoms.v ←
vsim -L lpm -L sgate -L stratixiigx -L stratixiigx_hssi work.<my design> \
-t ps +transport_int_delays +transport_path_delays ←
```



This example assumes you are using ModelSim PE/SE, if you are using ModelSim-Altera, use the precompiled libraries instead of creating new libraries and compiling the library files.

### Example of Compiling Library Files for Timing Stratix II GX Simulation in VHDL

To compile the libraries necessary to timing simulation of a VHDL design targeting a Stratix II GX device, type the following commands at the ModelSim command prompt:

```
vcom -work lpm 220pack.vhd 220model.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 work <alt2gxb megafunction variable>.vho ␣
vsim -L lpm -L sgate -L stratixiigx -L stratixiigx_hssi work.<my design> \
-t ps +transport_int_delays +transport_path_delays ␣
```



This example assumes you are using ModelSim PE/SE. If you are using ModelSim-Altera, use the precompiled libraries instead of creating new libraries and compiling the library files.

## Using the NativeLink Feature with ModelSim

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.

### Setting Up NativeLink

To run ModelSim automatically from the Quartus II software using the NativeLink feature on a PC, you must add the path to the ModelSim executables to your environment variable PATH.

For ModelSim-Altera and ModelSim SE/PE, executable files are stored in either the **win32aloem** or **win32** directories, respectively.

```
c:\<installation path>\win32aloem
```

```
c:\<installation path>\win32
```

To run ModelSim automatically from the Quartus II software using the NativeLink feature on a UNIX or Linux workstation, you must add the following environment variable to your **.cshrc** file:

```
QUARTUS_INI_PATH <path to ModelSim executables>
```

You might also be required to add the `QUARTUS_INIT_LIBPATH` environment variable. The `QUARTUS_INIT_LIBPATH` specifies the `LD_LIBRARY_PATH` variable needed by some EDA tools. Set this variable to the EDA tool's `LD_LIBRARY_PATH` if the EDA tool requires an `LD_LIBRARY_PATH` variable.



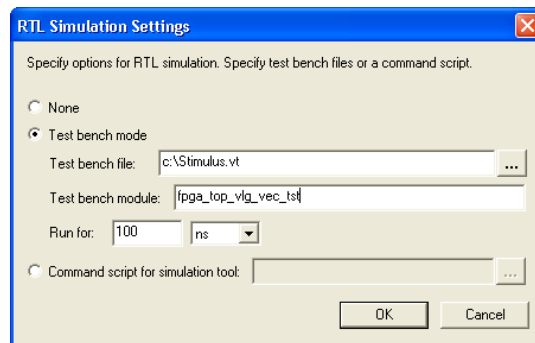
For more information on setting up your system for NativeLink, refer to the *Installation & Licensing for UNIX and Linux Workstations* manual.

## Performing an RTL Simulation Using Native Link

To run a functional RTL simulation with the ModelSim software automatically in the Quartus II software, perform the following steps:

1. On the Assignments menu, click **EDA Tool Settings**. The **Settings** dialog box is shown.
2. In the **Category** list, select **Simulation**. The **Simulation** page is shown.
3. In the **Tool Name** list, select one of the following choices:
  - ModelSim (VHDL)
  - ModelSim (Verilog HDL)
  - ModelSim-Altera (VHDL)
  - ModelSim-Altera (Verilog HDL)
4. If you have testbench files or macro scripts, enter the information into the **RTL Simulation Settings** dialog box (Figure 1–3).

**Figure 1–3. RTL Simulation Settings Dialog Box for Verilog HDL Simulation**



- a. In the **Simulation** page, under **NativeLink**, click **RTL Simulation Settings**.
- b. Select **None**, **Test bench mode** or **Command script for simulation tool** (Table 1–10).

**Table 1–10. RTL Simulation Settings**

Setting	Description
None	Compiles design files.
Test bench mode	<p>In the Test bench file box, click (...) to browse to your testbench file. In the <b>Test bench</b> module or <b>Test bench module</b> name box, type in the testbench entity/module name. In the <b>Run for</b> box, type in the end time and choose a time unit.</p> <p>Compiles all design and testbench files. Loads and simulates the testbench to the specified end time.</p>
Command script for simulation tool	In the <b>Command script for simulation</b> tool box, click (...) to browse to your macro script file. The script file can be either a TCL script file or a DO file. The script file entered passes to the ModelSim vsim executable using the <code>-do</code> option.

5. In the **RTL Simulation Settings** dialog box, click **OK**.
6. In the **Settings** dialog box, click **OK**.
7. On the Processing menu, point to Start and click **Start Analysis & Elaboration** to perform an analysis and elaboration. This command collects all your file name information and builds your design hierarchy in preparation for simulation.
8. On the Tools menu, point to **EDA Simulation Tool** and click **Run EDA RTL Simulation** to automatically launch ModelSim, compile all necessary design files, and complete a simulation.



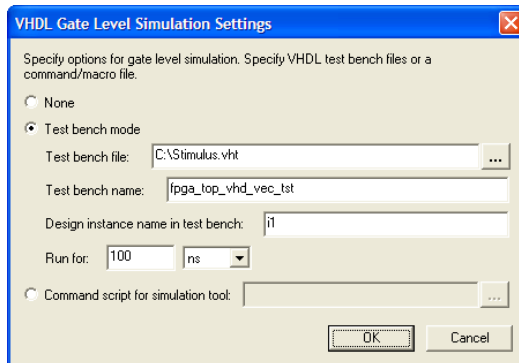
You cannot automatically perform an RTL simulation after a compilation.

## Performing a Gate Level Simulation Using NativeLink

To run a gate-level timing simulation with the ModelSim software automatically in the Quartus II software, perform the following steps:

1. On the Assignments menu, click **EDA Tool Settings**. The Settings dialog box is shown.
2. In the **Category** list, select **Simulation**. The **Simulation** page is shown.
3. In the **Tool name** list, select from one of the following choices:
  - ModelSim (VHDL)
  - ModelSim (Verilog HDL)
  - ModelSim-Altera (VHDL)
  - ModelSim-Altera (Verilog HDL)
4. To perform a gate level simulation after each full compilation, turn on **Run Gate Level Simulation automatically after compilation**.
5. If you have testbench files or macro scripts, enter the information into the **Gate Level Simulation Settings** dialog box. For example, [Figure 1-4](#) shows you what you must type in the **Gate Level Simulation Settings** dialog box if you use a Quartus II-generated testbench. For Verilog HDL, the file name should be *<file name>.vt* and in the **Test bench module** box, type *<top level design name>\_vlg\_vec\_tst*. For VHDL, the file name should be *<file name>.vht* and in the **Test bench name** box, type *<top level design name>\_vhd\_vec\_tst*. The Quartus II-generated testbenches design instance name is *i1*.

**Figure 1-4. Gate Level Simulation Settings Dialog Box**



- a. In the **Simulation** page, under **NativeLink**, click **Gate Level Simulation Settings**.
- b. Select **None**, **Test bench mode** or **Command script for simulation tool** (Table 1–11).

**Table 1–11. Gate Level Simulation Settings**

Settings	Description
None	Compiles design files.
Test bench mode	<p>In the Test bench file box, click (...) to browse to your testbench file. In the <b>Test bench</b> module or <b>Test bench module</b> name box, type in the testbench entity/module name. In the <b>Run for</b> box, type in the end time and choose a time unit.</p> <p>All design and testbench files compile. Testbench also loads and simulation I starts and ends after the specified end time.</p>
Command script for simulation tool	In the <b>Command script for simulation</b> tool box, click (...) to browse to your macro script file. The script file can be either a TCL script file or a DO file. The script file entered passes to the ModelSim vsim executable using the -do option.

6. In the **Gate Level Simulation Settings** dialog box, click **OK**.
7. In the **Settings** dialog box, click **OK**.
8. On the Processing menu, point to Start and click **Start EDA Netlist Writer** to generate a simulation netlist of your design.
9. On the Tools menu, point to EDA Simulation Tool and click **Run EDA Gate Level Simulation** to automatically launch ModelSim, compile all necessary design files, and complete a simulation.

## 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 **Qhelp** command line and Tcl API help browser.

Type this command to start the Qhelp 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 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_OUTPUT_DATA_FORMAT "VERILOG"  
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

You can 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 the following Tcl commands:

```
set_global_assignment-name EDA_OUTPUT_DATA_FORMAT "VERILOG"  
set_global_assignment-name EDA_SIMULATION_TOOL "Modelsim (Verilog)"
```

### *Command Line*

Generate a simulation output file for the ModelSim simulator by specifying VHDL or Verilog HDL for the format by typing the following:

```
quartus_eda<project name> --simulation=on --format=<format> - tool=Modelsim ↵
```

## Software Licensing & Licensing Set-Up

License the ModelSim-Altera software with a parallel port software guard (T-guard), USB guard, FIXEDPC license, or a network FLOATNET or FLOATPC license. Each Altera software subscription includes a license for either VHDL or Verilog HDL. Network licenses with multiple users may have their licenses split between VHDL and Verilog HDL in any ratio.



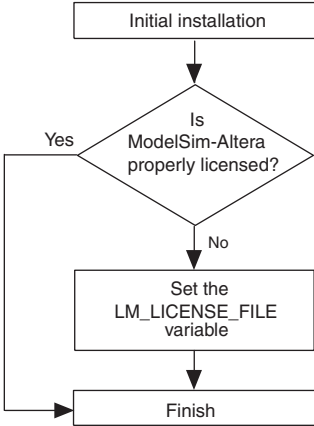
The USB software guard is not supported by versions earlier than Mentor Graphics ModelSim software 5.8d.

Obtain a license for the ModelSim-Altera 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. See [Figure 1-5](#) for the set-up process.



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

Figure 1-5. ModelSim-Altera Licensing Set Up Process



### LM\_LICENSE\_FILE Variable

Altera recommends setting the LM\_LICENSE\_FILE environment variable to the location of the license file.

## Conclusion

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

