

## Introduction

The capacity and complexity of Altera® FPGAs continues to increase as the need for intellectual property (IP) becomes increasingly critical. Using IP megafunctions reduces the design and verification time, allowing you to focus on design customization. Altera and the Altera Megafunction Partners Program (AMPP<sup>SM</sup>) offer a broad portfolio of IP megafunctions optimized for Altera FPGAs. Through parameterization, these reusable blocks of IP can be customized to meet your design requirements.

Even when the IP source code is encrypted or otherwise restricted, Altera's Quartus® II software allows you to easily simulate designs that contain Altera IP. With the Quartus II software, you can custom configure IP designs, then generate a VHDL or Verilog HDL functional simulation model to use with your choice of simulation tools.

This chapter provides an overview of the process for instantiating the IP megafunctions in your design and simulating its functional simulation model in an Altera-supported, third-party simulation tool. In this chapter, IP megafunctions refer to Altera megafunctions, IP MegaCore® functions, and IP AMPP megafunctions.

All IP MegaCore functions come with IP functional simulation (IPFS) models to support functional simulation. Some Altera megafunctions and some AMPP megafunctions also require IPFS models for functional simulation.



An IPFS model is a cycle-accurate VHDL or Verilog HDL model produced by the Quartus II software. The model allows for fast functional simulation of IP using industry-standard VHDL and Verilog HDL simulators.



Use these models only for simulation. Do not use them for synthesis or any other purpose. Using these models for synthesis creates a nonfunctional design.

This chapter discusses the following topics:

- [“IP Functional Simulation Flow”](#)
- [“Instantiate the IP in Your Design” on page 6–3](#)
- [“Perform Simulation” on page 6–3](#)
- [“Design Language Examples” on page 6–9](#)

## IP Functional Simulation Flow

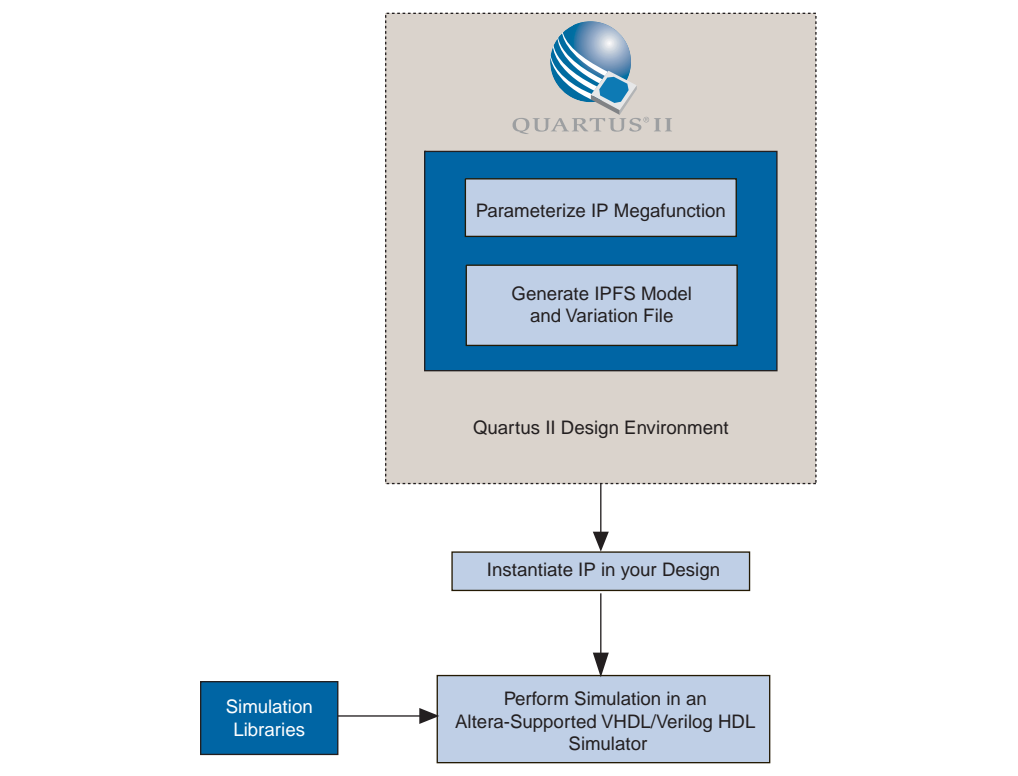
The IP megafunction's MegaWizard™ interface allows you to quickly and easily view documentation, specify parameters, generate an IPFS model, and output the files necessary to integrate a parameterized IP megafunction into your design. Within the Quartus II software, the MegaWizard Plug-In Manager can be used to select and parameterize your choice of IP megafunctions. The Quartus II software generates an

IP megafunction's variation file that is included in your Quartus II project. For IP megafunctions that require IPFS models, the Quartus II software can also generate a Verilog Output File (.vo) or VHDL Output File (.vho) that contains a Register Transfer Level (RTL) IPFS model after you have parameterized the megafunction. IPFS models are written to the Quartus II project directory.

Most Altera megafunctions and IP MegaCore functions support functional simulation in Verilog and VHDL for all Altera supported third-party simulators. Simulation libraries are required to simulate IP megafunctions. Refer to [Table 6-2 on page 6-8](#) for a subset of simulation libraries supplied with the Quartus II software.

[Figure 6-1](#) shows a typical simulation flow for Altera IP with third-party simulators.

**Figure 6-1.** IPFS Model Design Flow



## Verilog HDL and VHDL IPFS Models

Some IP megafunctions require IPFS models to support functional simulation. These IPFS models are generated in RTL HDL. These RTL models in Verilog HDL or VHDL format differ from the low-level synthesized netlist in Verilog HDL or VHDL format generated by the Quartus II software for post-synthesis or post place-and-route simulations. The IPFS models generated by the Quartus II software are much faster than the low-level post-synthesis or post place-and-route netlists of your design because they are mapped to higher-level primitives such as adders, multipliers, and multiplexers. These IPFS models can be simulated together with the rest of your design in any Altera-supported simulator. Altera recommends that you generate IPFS models in the same hardware language as the IP megafunction's variation file hardware language.



You can use an IPFS model for simulation only. Do not use it for synthesis or any other purpose. Attempting to synthesize an IPFS model will result in a nonfunctional design.



Generating an IPFS model for Altera MegaCore functions does not require a license. However, generating an IPFS model for AMPP megafunctions may require a license. For more information about licensing requirements, contact the IP megafunction vendor.

For details about how to parameterize and generate an IP, refer to the applicable IP user guide.

## Instantiate the IP in Your Design

For each IP megafunction in your design, you must instantiate the corresponding entity or module in your design. Each IP megafunction entity or module name is defined in its Quartus II generated megafunction variation file. After instantiating the IP megafunction in your design, you do not need to edit your design for synthesis or simulation.

To synthesize your design using the Quartus II software, add the Quartus II-generated Verilog HDL or VHDL variation file to your Quartus II project. When you create new variation files for a Quartus II project, they are added to the current open project when the megafunction is generated.

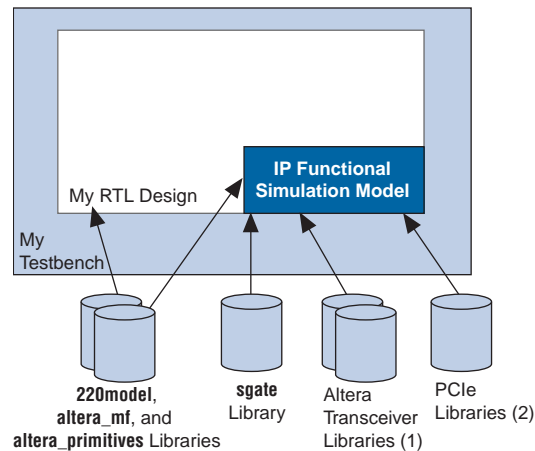
To synthesize your design using a third-party EDA tool, add the Quartus II-generated CMP file (*<megafunction variation>.cmp*) for your VHDL design or the Verilog HDL black box file (*<megafunction variation>\_bb.v*) for your Verilog HDL design to your third-party synthesis project.



For more information about synthesis and compilation with the Quartus II software, refer to the applicable chapters in *Volume 1: Design Synthesis* of the *Quartus II Handbook*.

## Perform Simulation

To perform simulation, in addition to adding your design files and testbench files, you also have to add the IP megafunction's variation file or IPFS model to your simulation project. If the IP megafunction does not require an IPFS model for simulation, add the megafunction's variation file to your simulation project. If the IP megafunction you are simulating requires an IPFS model, add the IPFS model to your simulation project. Your simulation project also requires Altera-supplied libraries for successful simulation. [Figure 6-2](#) shows how the Altera libraries are used in IP functional simulation.

**Figure 6-2.** IPFS Library Usage**Notes to Figure 6-2:**

- (1) The IP that uses the transceiver requires the Altera transceiver libraries for simulation. Refer to [Table 6-2](#) for all the Altera transceiver libraries.
- (2) An IP that uses the PCI Express (PCIe) hard core in a Stratix IV device requires the PCIe libraries. Refer to [Table 6-2](#) for all the PCIe transceiver libraries.

The Quartus II software contains all the libraries required for setting up and running a successful simulation of Altera IP. You can use the Quartus II NativeLink feature to set up your simulation if the IP megafunction you are using supports Quartus II NativeLink. Refer to the applicable IP megafunction user guide to determine if NativeLink is supported. Alternatively, you can simulate Altera IP with third-party simulators directly.

## Simulating Altera IP Using the Quartus II NativeLink Feature

The Quartus II NativeLink feature eases the task of setting up and running a simulation. The NativeLink feature lets you launch the third-party simulator to perform simulation from within the Quartus II software. The NativeLink feature automates the compilation and simulation of testbenches.

The following list briefly describes the steps to simulate IP megafunctions with third-party simulators using the Quartus II NativeLink feature. Each of these steps is described in more detail in the sections that follow.

1. [Select the Third-Party Simulation Tool.](#)
2. [Specify the Path for the Third-Party Simulator.](#)
3. [Specify the Testbench Settings.](#)
4. [Analyze and Elaborate the Quartus II Project.](#)
5. [Run RTL Functional Simulation.](#)

## Set up a Quartus II Project

To simulate IP megafunctions with the Quartus II NativeLink feature, you must open an existing project or create a new project in the Quartus II software. You can create and parameterize the IP you want to use in your design using the MegaWizard Plug-In Manager within the Quartus II software. Altera IP megafunction variation files are added to your Quartus II project when you create and parameterize the IP. You can also add any other required design files to your Quartus II project. If you are using the Quartus II NativeLink feature and your Quartus II project contains IP megafunctions that require IPFS models for simulation, you do not have to manually add the IPFS models to the Quartus II project for these IP megafunctions. When the Quartus II NativeLink feature launches the third-party simulator tool and starts the simulation, it automatically adds the IPFS model files required for simulation as long as they are present in the Quartus II project directory.

## Select the Third-Party Simulation Tool

You can select the third-party simulation tool on the **Simulation** page in the **Settings** dialog box.

Table 6-1 lists the third-party simulators supported by the Quartus II NativeLink feature.

**Table 6-1.** Third-Party Simulator Support with the Quartus II NativeLink Feature

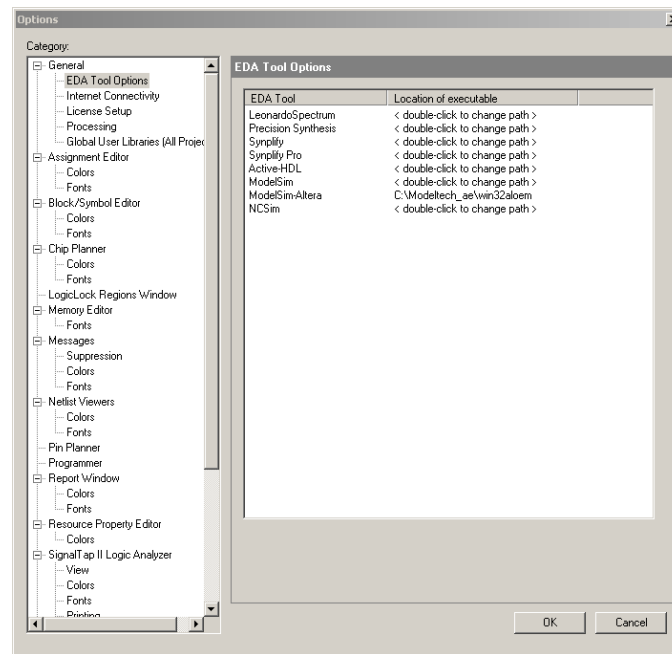
Third-Party Simulator	Can be Launched from Quartus II Software (1)	Testbench Support	Mixed Design (Verilog HDL and VHDL)
ModelSim® PE/SE	Yes	Yes	Yes
ModelSim-Altera Edition	Yes	Yes	No
Synopsys VCS	Yes	Yes	No
Synopsys VCS-MX	Yes	Yes	Yes
Cadence NC-Sim	Yes	Yes	Yes
Aldec Active-HDL	Yes	Yes	Yes

**Note to Table 6-1:**

(1) The Quartus II software must be running on the same platform as the third-party simulator to launch the simulator tool.

## Specify the Path for the Third-Party Simulator

To launch the third-party simulation tool, the absolute path for the selected simulator must be provided on the **Options** page under the Tools menu. See Figure 6-3. Double-click the **Location of executable** field to change or specify the absolute path.

**Figure 6–3.** Specifying the Simulator Path

### Specify the Testbench Settings

Specify the applicable testbench settings as follows:

1. Under **NativeLink settings** in the **Settings** dialog box (Figure 6–3), select the **Compile test bench** radio button and click **Test Benches** to display the **Test Benches** dialog box.
2. Click **New** to display the **New Test Bench Settings** dialog box.
3. In the **New Test Bench Settings** dialog box, set the appropriate fields with the names for the testbenches.



For specific instructions about specifying testbench settings for your MegaCore function, refer to your MegaCore function user guide.

4. After specifying the testbench files, close the **New Test Bench Settings**, **Test Benches**, and **Settings** dialog boxes.

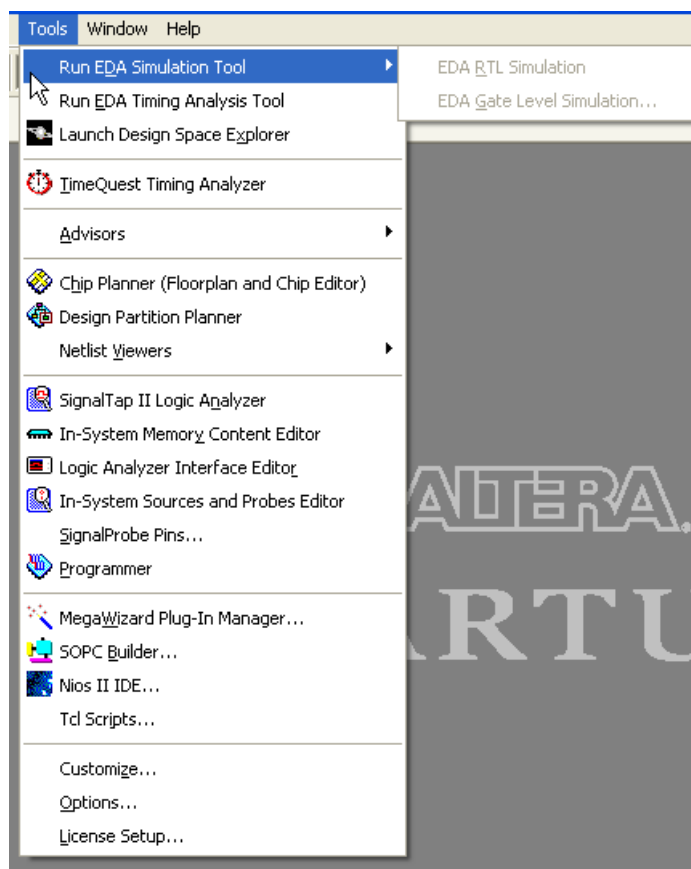
### Analyze and Elaborate the Quartus II Project

Before starting the simulation using the NativeLink feature, make sure that each IP megafunction's variation files are included in your design project. On the Quartus II Processing menu, point to **Start**, then click **Start Analysis & Elaboration**.

### Run RTL Functional Simulation

After the design is analyzed and elaborated, you can start the simulation by clicking **Run EDA Simulation Tool** from the Tools menu. See Figure 6–4. During RTL functional simulation, the IPFS models are compiled and used by the simulator.

**Figure 6-4.** Running Functional Simulation for IP using NativeLink



## Simulating Altera IP Without the Quartus II NativeLink Feature

You can also simulate Altera IP directly with third-party simulators. If your design instantiates an IP megafunction, add its variation file to your simulation project. If the IP megafunction requires IPFS model files, **do not** add the megafunction's variation file to your simulation project. Rather, add its IPFS model files (either Verilog HDL or VHDL) to your simulation project. The IPFS model generated by the Quartus II software instantiates high-level primitives such as adders, multipliers, and multiplexers, as well as the library of parameterized modules (LPM) functions and Altera megafunctions.

To properly compile, load, and simulate the IP megafunctions, you must first compile the following libraries in your simulation tool:

- **sgate**—includes the definition of the high-level primitives (needed for IPFS models)
- **altera\_mf**—includes the definition of Altera megafunctions
- **altera\_primitives**—includes the definition of Altera primitives
- **220model**—includes the definition of LPM functions

- **Altera transceiver**—includes the definition of all Altera transceiver megafunctions. If you use IP with the transceiver block, you must compile these libraries, which are device dependent.
- **PCIe**—includes the definition of PCIe hard core megafunctions for Stratix IV. If you use IP with the PCIe hard core for Stratix IV, you must compile these libraries.

You can use these library files with any Altera-supported simulation tool. If you are using the ModelSim-Altera software, the libraries are precompiled and mapped.



To simulate a design containing a Nios® processor or Avalon® peripherals, refer to [AN 189: Simulating Nios Embedded Processor Designs](#).

Table 6-2 lists the simulation library files, where *<path>* is the directory where the Quartus II software is installed.

**Table 6-2.** Simulation Library Files (Part 1 of 2)

Location	HDL Language	Description
<i>&lt;path&gt;/eda/sim_lib/sgate.v</i>	Verilog HDL	Libraries that contain simulation models for IP functional models
<i>&lt;path&gt;/eda/sim_lib/sgate.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/sgate_pack.vhd</i>	VHDL	Libraries that contain VHDL component declarations for the <b>sgate.vhd</b> library
<i>&lt;path&gt;/eda/sim_lib/220model.v</i>	Verilog HDL	Libraries that contain simulation models for the Altera LPM version 2.2.0
<i>&lt;path&gt;/eda/sim_lib/220model.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/220pack.vhd</i>	VHDL	Libraries that contain VHDL component declarations for the <b>220model.vhd</b> library
<i>&lt;path&gt;/eda/sim_lib/altera_mf.v</i>	Verilog HDL	Libraries that contain simulation models for Altera-specific megafunctions
<i>&lt;path&gt;/eda/sim_lib/altera_mf.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/altera_primitives.v</i>	Verilog HDL	Libraries that contain simulation models for Altera primitives
<i>&lt;path&gt;/eda/sim_lib/altera_primitives.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/altera_mf_components.vhd</i>	VHDL	Libraries that contain VHDL component declarations for the <b>altera_mf.vhd</b> library
<i>&lt;path&gt;/eda/sim_lib/stratixgx_mf.v</i>	Verilog HDL	Libraries that contain simulation models for Stratix® GX transceivers
<i>&lt;path&gt;/eda/sim_lib/stratixgx_mf.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/stratixgx_mf_components.vhd</i>	VHDL	Libraries that contain VHDL component declarations for the <b>stratixgx_mf.vhd</b> library
<i>&lt;path&gt;/eda/sim_lib/stratixiigx_hssi_atoms.v</i>	Verilog HDL	Libraries that contain simulation models for Stratix GX transceivers
<i>&lt;path&gt;/eda/sim_lib/stratixiigx_hssi_atoms.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/stratixiigx_hssi_components.vhd</i>	VHDL	Libraries that contain VHDL component declarations for the <b>stratixgx_mf.vhd</b> library
<i>&lt;path&gt;/eda/sim_lib/arriagx_hssi_atoms.v</i>	Verilog HDL	Libraries that contain simulation models for Arria® GX transceivers
<i>&lt;path&gt;/eda/sim_lib/arriagx_hssi_atoms.vhd</i>	VHDL	
<i>&lt;path&gt;/eda/sim_lib/arriagx_hssi_components.vhd</i>	VHDL	Libraries that contain VHDL component declarations for the <b>arriagx_mf.vhd</b> library
<i>&lt;path&gt;/eda/sim_lib/stratixiv_hssi_atoms.v</i>	Verilog HDL	Libraries that contain simulation models for Stratix IV transceivers
<i>&lt;path&gt;/eda/sim_lib/stratixiv_hssi_atoms.vhd</i>	VHDL	

**Table 6-2.** Simulation Library Files (Part 2 of 2)

Location	HDL Language	Description
<path>/eda/sim_lib/stratixiv_hssi_components.vhd	VHDL	Libraries that contain VHDL component declarations for the <b>stratixiv_mf.vhd</b> library
<path>/eda/sim_lib/stratixiv_pcie_hip_atoms.v	Verilog HDL	Libraries that contain simulation models for Stratix IV PCI Express
<path>/eda/sim_lib/stratixiv_pcie_hip_atoms.vhd	VHDL	
<path>/eda/sim_lib/stratixiv_pcie_hip_components.vhd	VHDL	Libraries that contain VHDL component declarations for the <b>stratix_iv_pcie_hip_atoms.vhd</b>

## Design Language Examples

The following design language examples explain how to simulate IP megafunctions directly with third-party simulator tools. These design examples describe simulation with:

- ModelSim Verilog
- ModelSim VHDL
- NC-VHDL
- VCS

### Verilog HDL Example: Simulating the IPFS Model in the ModelSim Software

The following example shows the process of simulating a Verilog HDL-based megafunction. The example assumes that the megafunction variation and the IPFS model are generated.

1. Create a ModelSim project by performing the following steps:
  - a. In the ModelSim software, on the File menu, point to **New** and click **Project**. The **Create Project** dialog box appears.
  - b. Specify the name of your simulation project.
  - c. Specify the desired location for your simulation project.
  - d. Specify the default library name and click **OK**.
  - e. Add relevant files to your simulation project:
    - Your design files
    - The IPFS model generated by the Quartus II software (if you are using the ModelSim-Altera software, skip to step 5)
    - The **sgate.v**, **220model.v**, and **altera\_mf.v** library files
    - The transceiver library files in Verilog HDL, if you use IP with transceivers. Transceiver libraries are family independent. Refer to [Table 6-2](#) for more information.
    - The PCIe library files in Verilog HDL, if you use IP with the PCIe hard core for Stratix IV devices. Refer to [Table 6-2](#) for more information.

2. Create the required simulation libraries by typing the following commands at the ModelSim prompt:

```
vlib sgate ←  
vlib lpm ←  
vlib altera_mf ←
```

3. Map to the required simulation libraries by typing the following commands at the ModelSim prompt:

```
vmap sgate sgate ←  
vmap lpm lpm ←  
vmap altera_mf altera_mf ←
```

4. Compile the HDL into libraries by typing the following commands at the ModelSim prompt:

```
vlog -work altera_mf altera_mf.v ←  
vlog -work sgate sgate.v ←  
vlog -work lpm 220model.v ←
```

5. Compile the IPFS model by typing the following command at the ModelSim prompt:

```
vlog -work work <my_IP>.vo ←
```

6. Compile your RTL by typing the following command at the ModelSim prompt:

```
vlog -work work <my_design>.v ←
```

7. Compile the testbench by typing the following command at the ModelSim prompt:

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

8. Load the testbench by typing the following command at the ModelSim prompt:

```
vsim -L <altera_mf_library_path> -L <lpm_library_path> \
-L <sgate_library_path> work.<my_testbench> ←
```

## VHDL Example: Simulating the IPFS Model in the ModelSim Software

The following example shows the process of performing a functional simulation of a VHDL-based, megafunction IPFS model. The example assumes that the megafunction's variation and the IPFS model are generated.

1. Create a ModelSim project by performing the following steps:
  - a. In the ModelSim software, on the File menu, point to **New** and click **Project**. The **Create Project** dialog box appears.
  - b. Specify the name for your simulation project.
  - c. Specify the desired location for your simulation project.
  - d. Specify the default library name and click **OK**.
  - e. Add the relevant files to your simulation project:
    - Add your design files
    - Add the IPFS model generated by the Quartus II software (if you are using the ModelSim-Altera software, skip to step 5)
    - Add the **sgate.vhd**, **sgate\_pack.vhd**, **220model.vhd**, **220pack.vhd**, **altera\_mf.vhd**, and **altera\_mf\_components.vhd** library files
    - The transceiver library files in VHDL, if you use IP with transceivers. Transceiver libraries are family independent. Refer to [Table 6-2](#) for more information.
    - The PCIe library files in VHDL, if you use IP with the PCIe hard core for Stratix IV. Refer to [Table 6-2](#) for more information.
2. Create the required simulation libraries by typing the following commands at the ModelSim prompt:

```
vlib sgate ←
vlib lpm ←
vlib altera_mf ←
```

3. Map to the required simulation libraries by typing the following commands at the ModelSim prompt:

```
vmap sgate sgate ←
vmap lpm lpm ←
vmap altera_mf altera_mf ←
```

4. Compile the HDL into libraries by typing the following commands at the ModelSim prompt:

```
vcom -work altera_mf -93 -explicit altera_mf_components.vhd ←
vcom -work altera_mf -93 -explicit altera_mf.vhd ←
vcom -work lpm -93 -explicit 220pack.vhd ←
vcom -work lpm -93 -explicit 220model.vhd ←
vcom -work sgate -93 -explicit sgate_pack.vhd ←
```

```
vcom -work sgate -93 -explicit sgate.vhd ←
```

5. Compile the IPFS model by typing the following command at the ModelSim prompt:

```
vcom -work work -93 -explicit <output_netlist>.vho ←
```

6. Compile the RTL by typing the following command at the ModelSim prompt:

```
vcom -work work -93 -explicit <RTL>.vhd ←
```

7. Compile the testbench by typing the following command at the ModelSim prompt:

```
vcom -work work -93 -explicit <my_testbench>.vhd ←
```

8. Load the testbench by typing the following command at the ModelSim prompt:

```
vsim work.my_testbench ←
```

## NC-VHDL Example: Simulating the IPFS Model in the NC-VHDL Software

The following example shows the process of performing a functional simulation of an NC-VHDL-based, megafunction IP functional-simulation model. The example assumes that the megafunction's variation and the IPFS model are generated.

1. Create a **cds.lib** file by typing the following entries:

```
DEFINE worklib ./worklib
DEFINE sgate ./sgate
DEFINE altera_mf ./altera_mf
DEFINE lpm ./lpm
```

2. Compile library files into appropriate libraries by typing the following commands at the command prompt:

```
ncvhd1 -V93 -WORK lpm 220pack.vhd ←
ncvhd1 -V93 -WORK lpm 220model.vhd ←
ncvhd1 -V93 -WORK altera_mf altera_mf_components.vhd ←
rncvhd1 -V93 -WORK altera_mf altera_mf.vhd ←
ncvhd1 -V93 -WORK sgate sgate_pack.vhd ←
ncvhd1 -V93 -WORK sgate sgate.vhd ←
```

3. Compile source code and testbench files by typing the following commands at the command prompt:

```
ncvhd1 -V93 -WORK worklib <my_design>.vhd ←
ncvhd1 -V93 -WORK worklib <my_testbench>.vhd ←
ncvhd1 -V93 -WORK worklib <my_IPtoolbench_output_netlist>.vho ←
```

4. Elaborate the design by typing the following command at the command prompt:

```
ncelab worklib.<my_testbench>:entity ←
```

## Verilog HDL Example: Simulating Your IPFS Model in VCS

The following example illustrates the process of performing a functional simulation of a design that contains a Verilog HDL-based, megafunction IPFS model. This example assumes that the megafunction variation and the IPFS model are generated.

### Single-Step Process

For the single-step process, type the following at the command prompt:

```
vcs <testbench>.v <RTL>.v <output_netlist>.v -v 220model.v \  
altera_mf.v sgate.v -R ↵
```

### Two-Step Process (Compilation and Simulation)

For compilation and simulation, perform the following steps:

1. Compile your design files by typing the following at the command prompt:

```
vcs <testbench>.v <RTL>.v <output_netlist>.v -v 220model.v \  
altera_mf.v sgate.v -o simulation_out ↵
```

2. Load your simulation by typing the following at a command prompt:

```
source simulation_out ↵
```



For more information about simulating a design in VCS, refer to the *Synopsys VCS and VCS-MX Support* chapter in volume 3 of the *Quartus II Handbook*.

## Conclusion

Altera Quartus II software provides full support for simulating IP megafunctions with third-party tools either directly or using its NativeLink feature. Using the Quartus II software, you can also generate IPFS models for supported megafunctions that enhance and simplify design verification. Using an IPFS model is transparent, requiring only the addition of different files in which to synthesize and simulate projects.

## Referenced Documents

This chapter references the following documents:

- *AN 189: Simulating Nios Embedded Processor Designs*
- *Synopsys VCS and VCS-MX Support* chapter in volume 3 of the *Quartus II Handbook*
- *Volume 1: Design Synthesis* of the *Quartus II Handbook*

## Document Revision History

Table 6-3 shows the revision history for this chapter.

**Table 6-3.** Document Revision History

Date and Document Version	Changes Made	Summary of Changes
March 2009 v9.0.0	Removed figures.	—
November 2008 v8.1.0	Changed to 8-1/2 x 11 page size. No change to content.	Updated for the Quartus II 8.1 software release
May 2008 v8.0.0	<ul style="list-style-type: none"> <li>■ Updated “Introduction.”</li> <li>■ Updated “Simulating Altera IP without the Quartus II NativeLink Feature.”</li> <li>■ Updated “Verilog HDL Example: Simulating the IPFS Model in the ModelSim Software.”</li> <li>■ Updated “VHDL Example: Simulating the IPFS Model in the ModelSim Software.”</li> <li>■ Updated Figure 6-2.</li> <li>■ Updated Table 6-1.</li> <li>■ Updated Table 6-2.</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](#).