

This chapter describes the process for instantiating the IP megafunctions in your design and simulating their functional simulation models in Altera-supported, third-party simulation tools.

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 design and verification time, allowing you to focus on design customization. Altera and the Altera Megafunction Partners Program (AMPPSM) 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.

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 and 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 IPFS models for simulation only. Do not use them for synthesis or any other purpose. Using these models for synthesis results in a nonfunctional design.

This chapter discusses the following topics:

- [“IP Functional Simulation Flow” on page 7-2](#)
- [“Instantiate the IP in Your Design” on page 7-3](#)
- [“Perform RTL Functional Simulation” on page 7-3](#)
- [“Design Language Examples” on page 7-6](#)

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, you can use the MegaWizard Plug-In Manager 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 HDL 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 HDL and VHDL for all Altera-supported third-party simulators. Simulation libraries are required to simulate IP megafunctions.


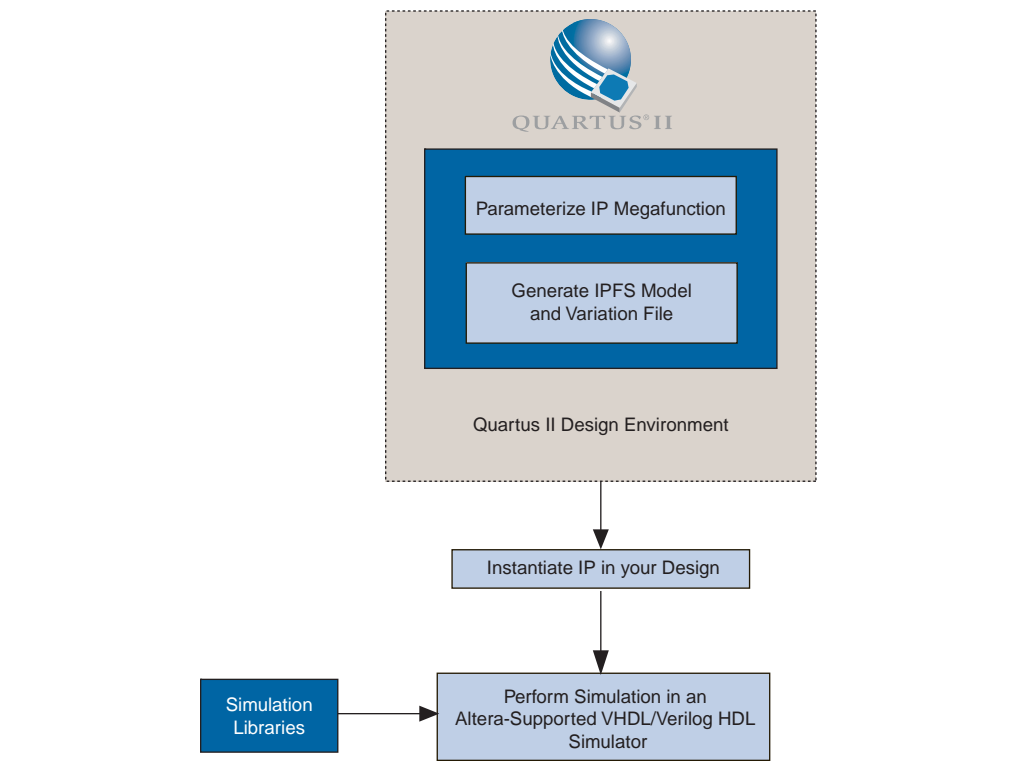
 For a list of simulation libraries supplied with the Quartus II software, refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

Figure 7-1 shows a typical simulation flow for Altera IP with third-party simulators.

Figure 7-1. 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 register transfer level (RTL) HDL. The 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.



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.

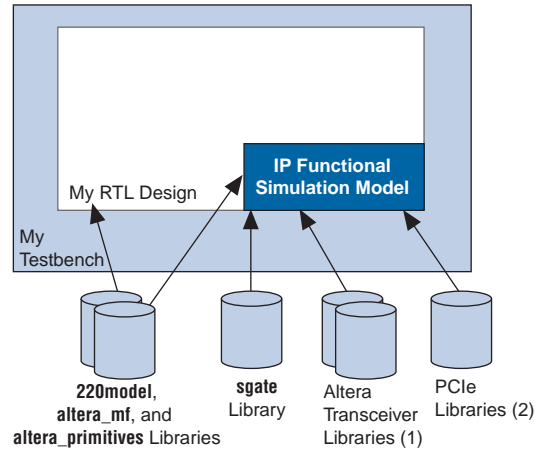
For information about instantiating IP megafunctions, refer to the *Megafunction Overview User Guide*.



For 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 RTL Functional Simulation

To perform RTL functional 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 7-2](#) shows how the Altera libraries are used in IP functional simulation.

Figure 7-2. IPFS Library Usage**Notes to Figure 7-2:**

- (1) The IP that uses the transceiver requires the Altera transceiver libraries for simulation. Refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook* for a list of all Altera transceiver libraries.
- (2) An IP that uses the PCI Express (PCIe) hard core in a Stratix IV device requires the PCIe libraries for simulation. Refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook* for a list of all PCIe transceiver libraries.

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

Simulating Altera IP Using the Quartus II NativeLink Feature

The Quartus II NativeLink feature eases the tasks 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.

Before running the simulation, you need to open your Quartus II project and perform Analysis and Elaboration, as described in the following section.

Perform Analysis and Elaboration on Your Design

To perform Analysis and Elaboration on your design, on the Quartus II Processing menu, point to **Start**, then click **Start Analysis & Elaboration**.

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 if they are present in the Quartus II project directory.

You can now continue the simulation process using the NativeLink feature.

Run Simulation Using the Quartus II NativeLink Feature

For detailed information about using the NativeLink feature to simulate Altera IP, refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

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. Instead, 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 PCI Express hard IP megafunctions for Stratix IV devices. If you use IP with the PCI Express hard IP for Stratix IV devices, 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, and no compilation is required.

Using the EDA Simulation Library Compiler

You can also compile these libraries with the EDA Simulation Library Compiler. If you use this tool, you do not need to know all of the required libraries; the compiler will retrieve them for you.

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

 To simulate a design containing a Nios® processor or Avalon® peripherals, refer to *AN 189: Simulating Nios Embedded Processor Designs*.

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 the following tools:

- ModelSim Verilog HDL
- 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. To create a ModelSim project, perform 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.
 - The PCIe library files in Verilog HDL, if you use IP with the PCIe hard core for Stratix IV devices.



For more information about simulation libraries, refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

2. To create the required simulation libraries, type the following commands at the ModelSim prompt:
3. Map to the required simulation libraries by typing the following commands at the ModelSim prompt:

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

```
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 how to perform 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:
 - 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.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.
 - The PCIe library files in VHDL, if you use IP with the PCI Express hard IP for Stratix IV.



For more information about simulation libraries, refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

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 how to perform 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 ←
```

```
ncvhdl -V93 -WORK sgate sgate_pack.vhd ←  
ncvhdl -V93 -WORK sgate sgate.vhd ←
```

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

```
ncvhdl -V93 -WORK worklib <my_design>.vhd ←  
ncvhdl -V93 -WORK worklib <my_testbench>.vhd ←  
ncvhdl -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 shows how to perform 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 the 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 enables you to simulate 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*
- *Megafunction Overview User Guide*
- *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*


- *Synopsys VCS and VCS-MX Support* chapter in volume 3 of the *Quartus II Handbook*
- *Volume 1: Design Synthesis* in the *Quartus II Handbook*

Document Revision History

Table 7-1 shows the revision history for this chapter.

Table 7-1. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v9.1.0	<ul style="list-style-type: none"> ■ Removed most NativeLink information ■ Added references to new chapter <i>Simulating Designs with EDA Tools</i> and <i>Megafunction Overview User Guide</i> ■ Minor text edits 	Updated for the Quartus II software version 9.1 release.
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 software version 8.1 release.
May 2008 v8.0.0	<ul style="list-style-type: none"> ■ Updated “Introduction.” ■ Updated “Simulating Altera IP without the Quartus II NativeLink Feature.” ■ Updated “Verilog HDL Example: Simulating the IPFS Model in the ModelSim Software.” ■ Updated “VHDL Example: Simulating the IPFS Model in the ModelSim Software.” ■ Updated Figure 6-2. ■ Updated Table 6-1. ■ Updated Table 6-2. 	Updated for the Quartus II software version 8.0 release.

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

