
Using IP Functional Simulation Models to Verify Your System Design

Introduction

The Quartus® II software can generate high-performance intellectual property (IP) functional simulation models for Altera® IP MegaCore® functions and Altera Megafunction Partners Program (AMPPSM) megafunctions. You can instantiate these models in your design, which can be simulated using an Altera-supported VHDL or Verilog HDL simulator. The process of generating an IP functional simulation model is completely contained within the Quartus II design environment. This white paper describes how to generate these models and instantiate them into your design.

Quartus II-Generated Simulation Models

The Quartus II software can generate an IP functional simulation model using a Verilog Output File (.vo) or VHDL Output File (.vho). This output file differs from the VO and VHO models generated by the Quartus II software for post-synthesis or post-place-and-route simulations.

Post-synthesis and post-place-and-route netlists are mapped to ATOMs. An ATOM is a parameterized, family-dependent representation of a WISIWYG primitive that corresponds to a device feature such as a logic element (LE), an I/O element (IOE), or memory. The VO or VHO models that are generated for post-synthesis or post-place-and-route are placed in the simulation folder under the Quartus II project folder.

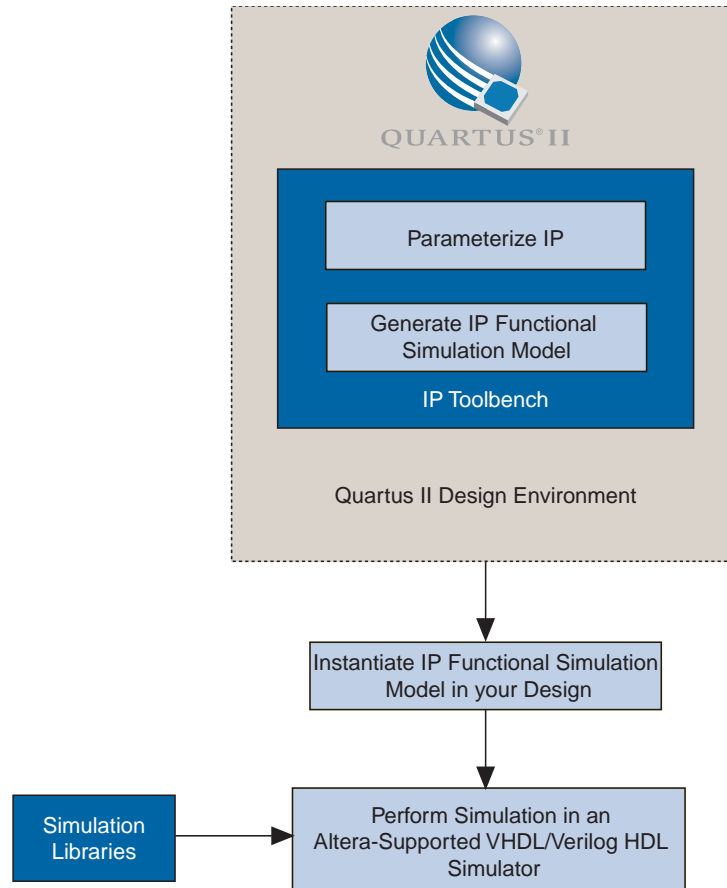
The IP functional simulation model is mapped to higher-level primitives such as adders, multipliers, and multiplexers. The higher level of abstraction results in much faster simulation times. Depending on the megafunction, the IP functional simulation model will be up to 100 times faster than a Quartus II post-synthesis or post-place-and-route simulation. The VO or VHO models that are generated for IP functional simulation purposes are output to the directory that you specify in the MegaWizard® Plug-In Manager, typically the same folder as your Quartus II project.

Generating an IP functional simulation model for Altera MegaCore functions does not require a license. However, generating an IP functional simulation model for AMPP megafunctions may require a license. For more information, contact the vendor.

You can only use the IP functional simulation model for simulation purposes, and not for synthesis or any other purpose. Using these models for synthesis will create a non-functional design.

Figure 1 outlines the process of generating an IP functional simulation model and simulating your design in an Altera-supported VHDL or Verilog HDL simulator.

Figure 1. Generating an IP Functional Simulation Model and Simulating Your Design



Generating & Simulating the IP Functional Simulation Model

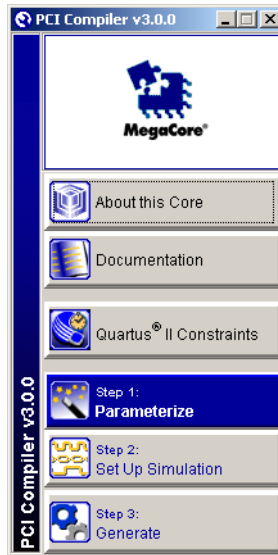
To launch IP Toolbench, start the MegaWizard Plug-In Manager in the Quartus II software, select the megafunction and HDL language you wish to use, specify the name and location of the file to be generated, and click **Next**. The following steps show how to generate and simulate the IP functional simulation model.

Step 1: Parameterize IP

After launching IP Toolbench, parameterize the megafunction by clicking **Parameterize**.

Figure 2 shows the IP Toolbench graphical user interface (GUI). The number of buttons in the IP Toolbench GUI and their names may vary for different megafunctions.

Figure 2. IP Toolbench GUI



Step 2: Set Up Simulation

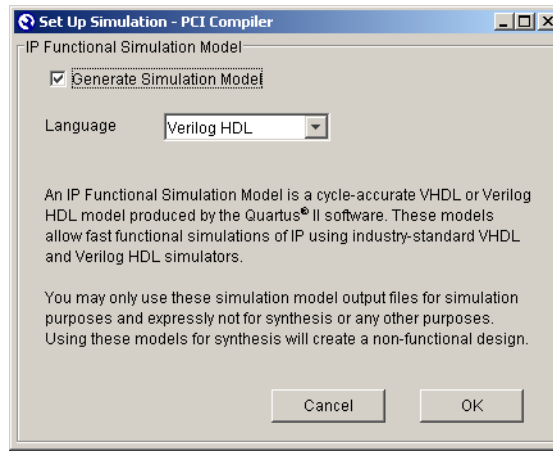
After the megafunction variation has been parameterized, click **Set Up Simulation** in the IP Toolbench GUI (Figure 3).

Figure 3. Set-Up Simulation Selection



The **Set Up Simulation** dialog box appears (Figure 4).

Figure 4. Set Up Simulation Dialog Box



In the **Set Up Simulation** dialog box, turn on **Generate Simulation Model**, choose the HDL language for the model, and click **OK**.

Step 3: Generate the IP Functional Simulation Model

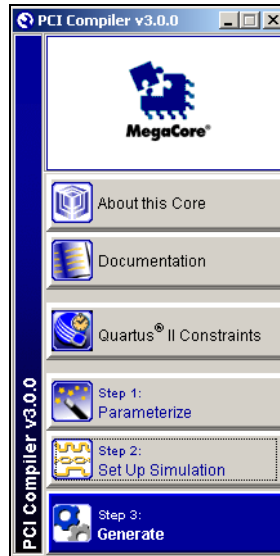
Next, click **Generate** to generate the custom megafunction variation and the IP functional simulation model (Figure 5).

When you click **Generate**, IP Toolbench creates several new files (described in Table 1) in the directory specified in the MegaWizard Plug-In Manager. These files vary based on the HDL language selected for your custom megafunction variation and IP functional simulation model.

Table 1. IP Toolbench-Generated Files

Extension	Description
.vhd, or .v	A megafunction variation file, which defines a VHDL or Verilog HDL top-level description of the custom megafunction. Instantiate the entity defined by this file inside of your design. Include this file when compiling your design in the Quartus II software.
.cmp	A VHDL component declaration file for the megafunction variation. Add the contents of this file to any VHDL architecture that instantiates the megafunction.
.inc	An AHDL include declaration file for the megafunction variation. Include this file with any AHDL architecture that instantiates the megafunction.
__bb.v	A Verilog HDL black-box file for the megafunction variation. Use this file when using a third-party EDA tool to synthesize your design.
.bsf	A Quartus II symbol file for the megafunction variation. You can use this file in the Quartus II block diagram editor.
.html	A megafunction report file.
.vo or .vho	A VHDL or Verilog HDL IP functional simulation model.
__inst.vhd or __inst.v	A VHDL or Verilog HDL sample instantiation file.

Figure 5. Generate Selection



Step 4: Instantiate the IP Functional Simulation Model in Your Design

Using the IP functional simulation model is transparent. The contents of your design files do not change. The only difference is which files you add to the simulation and synthesis projects. For simulation, add the IP functional simulation model VHO or VO file to your simulation project.

To synthesize your design using the Quartus II software, add the IP Toolbench-generated V or VHD megafunction wrapper file to your Quartus II project and add the megafunction's library files as user libraries. To synthesize your design using a third-party EDA tool, add the IP Toolbench-generated CMP file for your VHDL design or `<megafunction_variation>_bb.v` file for your Verilog HDL design to your third-party synthesis project, and add the megafunction's library files as user libraries in the Quartus II software.

Step 5: Perform Simulation

To compile and simulate the IP functional simulation model generated by the Quartus II software, you must first compile the `sgate`, `altera_mf`, and `220model` libraries in your simulation tool. You can use these library files with any Altera-supported simulation tool. If you are using the Modelsim-Altera software, these libraries have already been compiled and mapped and you do not need to compile these files.

To better illustrate the entire process of performing a functional simulation of a megafunction, refer to the VHDL example below and the Verilog HDL example on [page 7](#).

Table 2 lists and describes the library files needed used in simulation.

Table 2. Library Files

Library Location	Simulator	Library Description
<quartus_installation>/eda/sim_lib/sgate.v	Verilog HDL Simulator	Libraries that contain simulation models for IP functional models.
<quartus_installation>/eda/sim_lib/sgate.vhd	VHDL Simulator	
<quartus_installation>/eda/sim_lib/sgate_pack.vhd	VHDL Simulator	Libraries that contain VHDL component declarations for the sgate.vhd library.
<quartus_installation>/eda/sim_lib/220model.v	Verilog HDL Simulator	Libraries that contain simulation models for the Altera library of parameterized modules (LPM) version 2.2.0.
<quartus_installation>/eda/sim_lib/220model.vhd	VHDL Simulator	
<quartus_installation>/eda/sim_lib/220pack.vhd	VHDL Simulator	Libraries that contain VHDL component declarations for the 220model.vhd library.
<quartus_installation>/eda/sim_lib/altera_mf.v	Verilog HDL Simulator	Libraries that contain simulation models for Altera-specific megafunctions.
<quartus_installation>/eda/sim_lib/altera_mf.vhd	VHDL Simulator	
<quartus_installation>/eda/sim_lib/altera_mf_components.vhd	VHDL Simulator	Libraries that contains VHDL component declarations for the altera_mf.vhd library.

VHDL Example: Simulating the IP Functional Simulation Model in the ModelSim Software

The following procedure illustrates the process of performing a functional simulation of a design that contains a VHDL-based IP functional simulation model of a megafunction. This example assumes that the megafunction variation and the IP functional simulation model have been generated.

1. Create a ModelSim project:

- a. In the ModelSim software, select **New > Project** (File menu).
- b. In the **Create Project** dialog box, 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 relevant files to your simulation project:
 - i. Add your design files.
 - ii. Add the IP functional simulation model generated by IP Toolbench. (If you are using the ModelSim-Altera software, skip to step 5.)
 - iii. Add the **sgate.vhd**, **220model.vhd**, and **altera_mf.vhd** library files.

2. Create the required simulation libraries:

- a. At the ModelSim prompt, type: `vlib sgate` ←
- b. At the ModelSim prompt, type: `vlib lpm` ←
- c. At the ModelSim prompt, type: `vlib altera_mf` ←

3. Map to the required simulation libraries:

- a. At the ModelSim prompt, type: `vmap sgate sgate` ←
- b. At the ModelSim prompt, type: `vmap lpm lpm` ←
- c. At the ModelSim prompt, type: `vmap altera_mf altera_mf` ←

4. Compile the HDL into libraries:

- a. At the ModelSim prompt, type:
`vcom -work altera_mf -93 -explicit altera_mf_components.vhd` ←
- b. At the ModelSim prompt, type:
`vcom -work altera_mf -93 -explicit altera_mf.vhd` ←
- c. At the ModelSim prompt, type: `vcom -work lpm -93 -explicit 220pack.vhd` ←
- d. At the ModelSim prompt, type: `vcom -work lpm -93 -explicit 220model.vhd` ←
- e. At the ModelSim prompt, type: `vcom -work sgate -93 -explicit sgate_pack.vhd` ←

f. At the ModelSim prompt, type: `vcom -work sgate -93 -explicit sgate.vhd` ←

5. Compile the IP functional simulation model by typing the following at the ModelSim prompt:

```
vcom -work work -93 -explicit <output netlist>.vho ←
```

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

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

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

```
vcom -work work -93 -explicit <my testbench>.vhd ←
```

8. Load the testbench by typing the following at the Modelsim prompt: `vsim work.my_testbench` ←

Verilog HDL Example: Simulating Your IP Functional Simulation Model in VCS

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

Single-Step Process

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

```
vcs <testbench>.v <RTL>.v <output netlist>.v -v 220model.v altera_mf.v sgate.v -R ←
```

Two-Step Process (Compilation & 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 further information on simulating a design in VCS, see the *Synopsys VCS Support* chapter in Volume 3 of the *Quartus II Handbook*.

Conclusion

Using the Quartus II software and IP Toolbench, you can generate IP functional simulation models that enable you to efficiently simulate your design. The high level of abstraction of the IP functional simulation model, relative to post-synthesis or post place and route netlists, results in fast behavioral simulations of megafunctions. Using an IP functional simulation model is also transparent, requiring only adding different files to synthesis and simulation projects. These features enhance and simplify design verification.



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>

Copyright © 2004 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.