


Introduction

Altera® intellectual property (IP) MegaCore® functions are developed and pre-tested by Altera, and are optimized for specific Altera device architectures. Altera provides PCI function behavioral models you can instantiate in your design and simulate in third-party VHDL or Verilog HDL simulation tools. This document describes how to use these models to perform pre-synthesis simulation of the Altera PCI MegaCore functions (`pci_mt64`, `pci_mt32`, `pci_t64`, `pci_t32`).

Altera provides the following PCI function behavioral models:

- Precompiled VHDL models for the ModelSim® PE and SE VHDL simulators
 -  The precompiled VHDL models exceed the number of instances allowed by the ModelSim-Altera simulator. Therefore, Altera does not recommend that you use this simulator with the precompiled VHDL models.
- Visual IP models for all other Altera-supported VHDL simulators (except the ModelSim simulators)
- Visual IP models for all Altera-supported Verilog HDL simulators

The precompiled VHDL models are supplied as a precompiled library for the Altera PCI MegaCore functions. You can instantiate an Altera PCI function in your design and simulate it in the ModelSim VHDL simulator by mapping the precompiled library to your design.

The Altera PCI Visual IP models are parameterizable, register transfer level (RTL) functional simulation models. When you test-drive an Altera PCI MegaCore function using the OpenCore® feature, you can instantiate a Visual IP model in your design and simulate it in your choice of simulation tool. Altera generates the Visual IP models using Visual IP software from Innoveda, Inc.

Before You Begin

The behavioral models are included with the PCI compiler software. If you are using the Visual IP models, you must install the Visual IP software and link it to your simulation tool by following the instructions given in the *Simulating Visual IP Models with the ModelSim Simulator for PCs* white paper or the *Simulating Visual IP Models with the NC-Verilog, Verilog-XL, VCS or ModelSim(Unix) Simulators* white paper. You can download the Visual IP software and user guide from the IP MegaStore area on the Altera web site at http://www.altera.com/ip/altera/visual_ip.html.

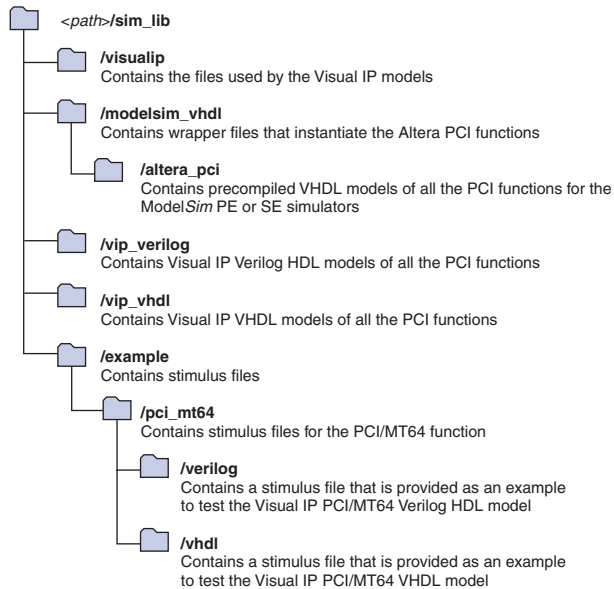


You do not need to download and install the Visual IP software if you are using the precompiled VHDL models.

Directory Structure

When you install the PCI compiler, the installation program creates the directory structure shown in [Figure 1](#) for the behavioral models.

Figure 1. Model Directory Structure *Note (1)*



Note:

(1) *<path>* is the directory in which you installed PCI compiler.


Simulator-Specific Procedures

Depending on the simulator you are using, consult the appropriate section of this application note as shown in [Table 1](#).

Simulator	Platform	Reference Section
ModelSim — VHDL	PC, UNIX	“Using the Precompiled VHDL Models”
LeapFrog or VSS	UNIX	“Using the VHDL Visual IP Models”
ModelSim — Verilog	PC	“Using the Verilog HDL Visual IP Models with Modelsim Simulator for PCs”
NC-Verilog, VCS, Verilog-XL or ModelSim-Verilog	UNIX	“Using the Verilog HDL Visual IP Models (UNIX)”

Using the Precompiled VHDL Models

Altera provides precompiled VHDL models generated using the ModelSim software version 5.5a with the `-nodebug` option. The `-nodebug` option hides the internal signals and the I/O ports of the PCI function. To use the models, you must use the wrapper files that instantiate the individual PCI MegaCore functions; only the ports used by the wrapper file are visible to the user. These wrapper files are provided in `<path>/sim_lib/modelsim_vhdl`.

 Install the ModelSim PE or SE software version 5.5a or later to use the precompiled models.

To use the precompiled VHDL models in your design, perform the following steps:

1. Run the ModelSim software.
2. Map the precompiled library into your design by executing the following command:

```
vmap altera_pci
<path>/pci_compiler_v<version_number>/sim_lib/modelsim
_vhdl/altera_pci ↵
```

3. Create a working directory for your design by executing the following command:

```
vlib work ↵
```

4. Refresh the precompiled library for the version of ModelSim software you are using by executing the following command:


```
vcom -work altera_pci -refresh ←
```
5. Compile the wrapper file, which instantiates the function you want to use, into the work library. There are four wrapper files—one for each MegaCore function—provided in `<path>/sim_lib/modelsim_vhdl`. You can pass the parameters needed for your application to the PCI function using the generic section of the wrapper file.
6. Instantiate the wrapper file in your design to simulate the PCI function.

To see an example, refer to the testbench provided in `<path>/testbench/vhdl/example/pci_mt64`. The testbench uses the precompiled library to instantiate the PCI function.

In addition to the wrapper files provided in `<path>/sim_lib/modelsim_vhdl`, the PCI compiler wizard outputs a wrapper file for your function variation after you have gone through the wizard. You can use the wizard-generated wrapper file with the precompiled library to simulate the function with custom parameters. To use the wrapper file generated by the PCI compiler wizard, add the following line to the beginning of the wizard-generated wrapper file:

```
library altera_pci;
```

Altera provides Visual IP VHDL models for use with Altera-supported VHDL simulators (except the ModelSim PE or SE).

Using the Models

Perform the following steps to use the Visual IP VHDL models:

1. Set up your simulator's C language interface to work with the Visual IP software as described in the *Visual IP User Guide For The End User*.
2. Ensure the `VIP_MODELS_DIR` environment variable includes `<path>/sim_lib/visualip`
3. Go to the `<path>/sim_lib/vip_vhdl/<simulator>` directory, where `<simulator>` is the VHDL simulation tool you are using.

Using the VHDL Visual IP Models

4. Compile the VHDL model that corresponds to the function you wish to simulate. The Visual IP model links to your VHDL simulator through the C language interface.
5. Instantiate the Visual IP model in your VHDL design.

The PCI compiler wizard outputs a wrapper file of your PCI function variation. You can use the wizard-generated wrapper file with the Visual IP model to simulate the function with custom parameters. Alternatively, you can modify the parameters of the function as needed for your application in the generic section of the Visual IP model.

Using the `pci_mt64` Stimulus File

This section describes how to use the example stimulus file provided with the Visual IP VHDL model for the Altera `pci_mt64` function.

1. Follow steps 1 and 2 from “Using the Models” on page 4.
2. Create a new project in `<path>/sim_lib/examples/pcimt64/VHDL`.
3. Compile `pci_mt64.vhd` from `<path>/sim_lib/vip_vhdl/<simulator>` directory, where `<simulator>` is the VHDL simulation tool you are using.
4. Compile `pci_top.vhd` into your work library from the current directory. `pci_top.vhd` instantiates `pci_mt64.vhd`.
5. Compile `test.vhd` into your work library from the current directory. `test.vhd` instantiates `pci_top.vhd` and performs PCI configuration writes and reads on the MegaCore function.
6. Simulate `work.test`.

Altera provides Verilog HDL Visual IP models to use with Modelsim Simulator for PCs.

Using the Model

Perform the following steps to use the Visual IP Verilog HDL model with ModelSim Simulator for PCs:

1. Obtain and Install the latest version of the PCI Compiler. (See the *PCI Compiler Data Sheet* for more information how to obtain and install the compiler.)

Using the Verilog HDL Visual IP Models with Modelsim Simulator for PCs

2. Set up your simulator to work with the Visual IP software by following instructions given in the *Simulating Visual IP Models with the ModelSim Simulator for PCs* white paper.
3. Go to the `<path>\sim_lib\vip_verilog` directory.
4. Compile the Visual IP Verilog HDL model that corresponds to the function you wish to simulate. The Visual IP model links to your Verilog HDL simulator through the PLI interface.
5. Instantiate the Visual IP model in your Verilog HDL design.

Common Issues

If, after following the instructions above, the simulator you are using returns either "Model Initialization Error" or "System task or function Vip_pli_sensi' is not defined" Ensure that

- You have set the Environment Variable VIP_MODELS_DIR to point to `d:\megacore\pci_compiler_v<version_number>\sim_lib\visualip`
- You have set the PATH variable to include `d:\VIP_EndUser_v<version>\bin`
- In your modelsim.ini file you have set `veriuser = d:\VIP_EndUser_v<version>\bin\libplimtivip`

Using the pci_mt64 Stimulus File

This section describes how to use the example stimulus file provided with the Visual IP Verilog HDL model for the Altera pci_mt64 function.

1. Follow steps 1 and 2 from "Using the Model" above.
2. Change to the `\<path>\pci_compiler_v<version_number>\sim_lib\examples\pci_mt64\verilog` directory.
3. Compile `pci_mt64.v` from the `\<path>\pci_compiler_v<version_number>\sim_lib\vip_verilog` directory.
4. Compile `pci_top.v` from the current directory to instantiate `pci_mt64.v`.
5. Compile `stimulus.v` from the current directory. `stimulus.v` performs configuration reads and writes on the pci_mt64 MegaCore function.

6. Compile `test.v` from the current directory to instantiate `stimulus.v` and `pci_top.v`.
7. Simulate `work.test`.

Using the Verilog HDL Visual IP Models (UNIX)

Altera provides Verilog HDL Visual IP models for use Verilog simulators.

Using the Model

Perform the following steps to use the Visual IP Verilog HDL models:

1. Obtain and Install the latest version of the PCI Compiler. (See the *PCI Compiler Data Sheet* for more information how to obtain and install the compiler.)
2. Set up your simulator to work with the Visual IP software by following instructions given in the *Simulating Visual IP Models with the NC-Verilog, Verilog-XL, VCS or ModelSim(Unix) Simulators* white paper.
3. Go to the `/<path>/sim_lib/vip_verilog` directory.
4. Compile the Visual IP Verilog HDL model that corresponds to the function you wish to simulate.
5. Instantiate the Visual IP model in your Verilog HDL design.

The PCI compiler wizard outputs a wrapper file of your PCI function variation. You can use the wizard-generated wrapper file with the Visual IP model to simulate the function with custom parameters. Alternatively, you can modify the parameters of the function as needed for your application in the parameters section of the Visual IP model.

6. Refer to "Simulate the Visual IP Models" on page 4 of *Simulating Visual IP Models with the NC-Verilog, Verilog-XL, VCS or ModelSim(Unix) Simulators* white paper for an example command to be used with the simulator you are using.

Common Issues

If, after following the instructions above, the software you are using returns either "Model Initialization Error" or "System task or function Vip_pli_sensi' is not defined" Ensure that

- You have set the Environment Variable `VIP_MODELS_DIR` to point to `/<path>/megacore/pci_compiler_v<version_number>/sim_lib/visualip`
- There was no error in running the Visual IP setup Script `<path>/VIP_EndUser_v<version>/<platform>/bin/vip.setup`

Using the pci_mt64 Stimulus File

This section describes how to use the example stimulus file provided with the Visual IP Verilog HDL model for the Altera pci_mt64 function.

1. Follow steps 1 and 2 from “Using the Model” above.
2. Change to the `/<path>/pci_compiler_v<version_number>/sim_lib/examples/pcimt64/verilog` directory.
3. Compile `pci_mt64.v` from the `/<path>/pci_compiler_v<version_number>/sim_lib/vip_verilog` directory.
4. Compile `pci_top.v` from the current directory to instantiate `pci_mt64.v`.
5. Compile `stimulus.v` from the current directory. `stimulus.v` performs configuration reads and writes on the pci_mt64 MegaCore function.
6. Compile `test.v` from the current directory to instantiate `stimulus.v` and `pci_top.v`.
7. Simulate `test.v`.

Known Issues

The Visual IP models do not support checkpoint/restart. Therefore, you must reload the simulation model to restart the simulation.

The behavioral models and the standard PCI core differ in the following ways:

- The PCI behavioral models do not support the `NUMBER_OF_BARS` parameter; therefore, the models instantiate all 6 BARs, regardless of the setting for `NUMBER_OF_BARS` in the top-level wrapper file. All unused BARs reserve 64 KBytes of system memory. Therefore, to avoid contention in claiming address space, do not use the memory range 0x00000000 to 0x0000FFFF.

For example, if the `NUMBER_OF_BARS` in the top-level wrapper file is 2, the unused BARs (BAR2, BAR3, BAR4, and BAR5) have default settings to reserve 64 KBytes of system memory and all four BARs can claim address range 0x00000000 to 0x0000FFFF. To avoid contention in claiming address space, do not use this memory range. If your design must use this address space, configure the unused BARs with known unused address space.

- The behavioral models do not support hardwired BARs.
- The behavioral models capabilities list is always enabled, even though you may have disabled it in the top-level wrapper file.

Revision History

The information contained this version supersedes information published in previous versions.

Version 1.2

This version contains updated information in [“Known Issues” on page 9](#).

Version 1.1

AN 169: Simulating the PCI MegaCore Function Behavioral Models version 1.1 version 1.1 contains the following changes:

- Updated path name in [Figure 1](#).
- Updated text throughout for 2.2.0 release of the PCI Compiler software.



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com

Copyright © 2003 Altera Corporation. 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. ModelSim is a registered trademark of Mentor Graphics Corporation. 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. All rights reserved.



I.S. EN ISO 9001