Simulating Altera Designs

1

2014.06.30

QII53025 Subscribe Send Feedback

This document describes simulating designs that target Altera devices. Simulation verifies design behavior before device programming. The Quartus II software supports RTL- and gate-level design simulation in supported EDA simulators. Simulation involves setting up your simulator working environment, compiling simulation model libraries, and running your simulation.

Simulator Support

The Quartus II software supports specific EDA simulator versions for RTL and gate-level simulation.

Table 1-1: Supported Simulators

Vendor	Simulator	Version	Platform
Aldec	Active-HDL	9.3	Windows
Aldec	Riviera-PRO	2013.10	Windows, Linux
Cadence	Incisive Enterprise	13.1	Linux
Mentor Graphics	ModelSim-Altera (provided)	10.1e	Windows, Linux
Mentor Graphics	ModelSim PE	10.1e	Windows
Mentor Graphics	ModelSim SE	10.2c	Windows, Linux
Mentor Graphics	QuestaSim	10.2c	Windows, Linux
Synopsys	VCS/VCS MX	2013.06-sp1	Linux

Simulation Flows

The Quartus II software supports integration with EDA simulators.

Table 1-2: Simulation Flows

Simulation Flow	Description
NativeLink flow	The NativeLink automated flow supports a variety of design flows. Do not use NativeLink if you require direct control over every aspect of simulation. • Use NativeLink to generate simulation scripts to compile your design and simulation libraries, and to automatically launch your simulator. • Specify your own compilation, elaboration, and simulation scripts for testbench and simulation model files that have not been analyzed by the Quartus II software. • Use NativeLink to supplement your scripts by automatically compiling design files, IP simulation model files, and Altera simulation library models.
Custom flows	Custom flows support manual control of all aspects of simulation, including the following: • Manually compile and simulate testbench, design, IP, and simulation model libraries, or write scripts to automate compilation and simulation in your simulator. • Use the Simulation Library Compiler to compile simulation libraries for all Altera devices and supported third-party simulators and languages. Use the custom flow if you require any of the following: • Custom compilation commands for design, IP, or simulation library model files (for example, macros, debugging or optimization options, or other simulator-specific options) . • Multi-pass simulation flows. • Flows that use dynamically generated simulation scripts.
Specialized flows	Altera supports specialized flows for various design variations, including the following: • For simulation of Altera example designs, refer to the documentation for the example design or to the IP core user guide. • For simulation of Qsys designs, refer to *Creating a System with Qsys*. • For simulation of designs that include the Nios II embedded processor, refer to *Simulating a Nios II Embedded Processor*.

Related Information

- **IP User Guide Documentation**
- **Creating a System with Qsys**
- **Simulating a Nios II Embedded Processor**

HDL Support

The Quartus II software provides the following HDL support for EDA simulators.

Table 1-3: HDL Support

Language	Description
VHDL	• For VHDL RTL simulation, compile design files directly in your simulator. To use NativeLink automation, analyze and elaborate your design in the Quartus II software, and then use the NativeLink simulator scripts to compile the design files in your simulator. You must also compile simulation models from the Altera simulation libraries and simulation models for the IP cores in your design. Use the Simulation Library Compiler or NativeLink to compile simulation models. • For gate-level simulation, the EDA Netlist Writer generates a synthesized design netlist VHDL Output File (**.vho**). Compile the **.vho** in your simulator. You may also need to compile models from the Altera simulation libraries. • IEEE 1364-2005 encrypted Verilog HDL simulation models are encrypted separately for each Altera-supported simulation vendor. If you want to simulate the model in a VHDL design, you need either a simulator that is capable of VHDL/Verilog HDL co-simulation, or any Mentor Graphics single language VHDL simulator.
Verilog HDL SystemVerilog	• For RTL simulation in Verilog HDL or SystemVerilog, compile your design files in your simulator. To use NativeLink automation, analyze and elaborate your design in the Quartus II software, and then use the NativeLink simulator scripts to compile your design files in your simulator. You must also compile simulation models from the Altera simulation libraries and simulation models for the IP cores in your design. Use the Simulation Library Compiler or NativeLink to compile simulation models. • For gate-level simulation, the EDA Netlist Writer generates a synthesized design netlist Verilog Output File (**.vo**). Compile the **.vo** in your simulator.
Mixed HDL	• If your design is a mix of VHDL, Verilog HDL, and SystemVerilog files, you must use a mixed language simulator. Since Altera supports both languages, choose the most convenient language for any Altera IP core in your design. • Altera provides Arria V, Cyclone V, Stratix V, and newer simulation model libraries and IP simulation models in Verilog HDL and IEEE encrypted Verilog. Your simulator's co-simulation capabilities support VHDL simulation of these models using VHDL “wrapper” files. Altera provides the wrapper for Verilog models to instantiate these models directly from your VHDL design.
Schematic	You must convert schematics to HDL format before simulation. You can use the converted VHDL or Verilog HDL files for RTL simulation.

Simulation Levels

The Quartus II software supports various levels of simulation in supported EDA simulators.

Table 1-4: Supported Simulation Levels

Simulation Level	Description	Simulation Input
RTL	Cycle-accurate simulation using Verilog HDL, SystemVerilog, and VHDL design source code with simulation models provided by Altera and other IP providers.	• Design source/testbench • Altera simulation libraries • Altera IP plain text or IEEE encrypted RTL models • IP simulation models • Altera IP functional simulation models • Altera IP bus functional models • Qsys-generated models • Verification IP
Gate-level functional	Simulation using a post-synthesis or post-fit functional netlist testing the post-synthesis functional netlist, or post-fit functional netlist.	• Testbench • Altera simulation libraries • Post-synthesis or post-fit functional netlist • Altera IP bus functional models
Gate-level timing	Simulation using a post-fit timing netlist, testing functional and timing. Not supported for Arria V, Cyclone V, or Stratix V devices.	• Testbench • Altera simulation libraries • Post-fit timing netlist • Post-fit Standard Delay Output File (**.sdo**)

Note: Gate-level timing simulation of an entire design can be slow and should be avoided. Gate-level timing simulation is not supported for Arria® V, Cyclone® V, or Stratix® V devices. Use TimeQuest static timing analysis rather than gate-level timing simulation.

Files Generated for Altera IP Cores

The Quartus II software generates one or more of the following output file structures for your Altera IP core variation.

Figure 1-1: IP Core Generated Files

Note: To manually add an IP variation to a Quartus II project, click **Project** > **Add/Remove Files in Project** and add only the IP variation **.qip** or **.qsys** file to the project. Do not manually add the top-level HDL file to the project.

Preparing for Simulation

Preparing for RTL or gate-level simulation involves compiling the RTL or gate-level representation of your design and testbench. You must also compile IP simulation models, models from the Altera simulation libraries, and any other model libraries required for your design.

Compiling Simulation Models

The Quartus II software includes simulation models for Altera IP cores.

These models include IP functional simulation models, and device family-specific models in the ***<Quartus II installation path>*/eda/sim_lib** directory. These models include IEEE encrypted Verilog HDL models for both Verilog HDL and VHDL simulation. Before running simulation, you must compile the appropriate simulation models from the Altera simulation libraries.

Use any of the following methods to compile Altera simulation models:

- Use the NativeLink feature to automatically compile your design, Altera IP, simulation model libraries, and testbench.
- Run the Simulation Library Compiler to compile all RTL and gate-level simulation model libraries for your device, simulator, and design language.
- Compile Altera simulation models manually with your simulator.

After you compile the simulation model libraries, you can reuse these libraries in subsequent simulations to avoid having to compile them again.

Note: The specified timescale precision must be within 1ps when using Altera simulation models.

Related Information

Altera Simulation Models

Generating IP Simulation Files for RTL Simulation

The Quartus II software supports both Verilog HDL and VHDL simulation of encrypted and unencrypted Altera IP cores. If your design includes Altera IP cores, you must compile any corresponding IP simulation models in your simulator with the rest of your design and testbench. The Quartus II software generates and copies the simulation models for IP cores to your project directory.

You can use the following files to simulate your Altera IP variation.

Table 1-5: Altera IP Simulation Files

File Type	Description	File Name
Simulator setup script	Simulator-specific script to compile, elaborate, and simulate Altera IP models and simulation model library files. Copy the commands into your simulation script, or edit these files to compile, elaborate, and simulate your design and testbench.	Cadence • **cds.lib** • **ncsim_setup.sh** • **hdl.var** Mentor Graphics • **msim_setup.tcl** Synopsys • **synopsys_sim.setup** • **vcs_setup.sh** • **vcsmx_setup.sh** Aldec • **rivierapro_setup.tcl**
Quartus II Simulation IP File (**.sip**)	Contains IP core simulation library mapping information. The **.sip** files enable NativeLink simulation and the Quartus II Archiver for IP cores.	***<design name>*.sip**
IP functional simulation models	IP functional simulation models are cycle-accurate VHDL or Verilog HDL models generated by the Quartus II software for some Altera IP cores. IP functional simulation models support fast functional simulation of IP using industry-standard VHDL and Verilog HDL simulators.	***<my_ip>*.vho** ***<my_ip>*.vo**
IEEE encrypted models	Arria V, Cyclone V, Stratix V, and newer simulation model libraries and IP simulation models are provided in Verilog HDL and IEEE encrypted Verilog HDL. VHDL simulation of these models is supported using your simulator's co-simulation capabilities. IEEE encrypted Verilog HDL models are significantly faster than IP functional simulation models.	***<my_ip>*.v**

Generating IP Functional Simulation Models for RTL Simulation

Altera provides IP functional simulation models for some Altera IP cores. To generate IP functional simulation models, follow these steps:

- Turn on the **Generate Simulation Model** option when parameterizing the IP core.
- When you simulate your design, compile only the **.vo** or **.vho** for these IP cores in your simulator. In this case you should not compile the corresponding HDL file. The encrypted HDL file supports synthesis by only the Quartus II software.

Note: Altera IP cores that do not require IP functional simulation models for simulation, do not provide the **Generate Simulation Model** option in the IP core parameter editor.

Note: Many recently released Altera IP cores support RTL simulation using IEEE Verilog HDL encryption. IEEE encrypted models are significantly faster than IP functional simulation models. You can simulate the models in both Verilog HDL and VHDL designs.

Related Information

AN 343: OpenCore Evaluation of AMPP Megafunctions

Running a Simulation (NativeLink Flow)

The NativeLink feature integrates your EDA simulator with the Quartus II software and automates the following simulation steps:

- Set and reuse simulation settings
- Generate simulator-specific files and simulation scripts
- Compile Altera simulation libraries
- Launch your simulator automatically following Quartus II Analysis & Elaboration, Analysis & Synthesis, or after a full compilation.

Setting Up Simulation (NativeLink Flow)

Before running simulation using the NativeLink flow, you must specify settings for your simulator in the Quartus II software. To specify simulation settings in the Quartus II software, follow these steps:

1. Open a Quartus II project.
2. Click **Tools > Options** and specify the location of your simulator executable file .

Table 1-6: Execution Paths for EDA Simulators

Simulator	Path
Mentor Graphics ModelSim-Altera	<drive letter>**:**\<simulator install path>**\win32aloem** (Windows) **/**<simulator install path>**/bin** (Linux)
Mentor Graphics ModelSim Mentor Graphics QuestaSim	<drive letter>**:**\<simulator install path>**\win32** (Windows) <simulator install path>**/bin** (Linux)
Synopsys VCS/VCS MX	<simulator install path>**/bin** (Linux)
Cadence Incisive Enterprise	<simulator install path>**/tools/bin** (Linux)
Aldec Active-HDL Aldec Riviera-PRO	<drive letter>**:**\<simlulator install path>**\bin** (Windows) <simulator install path>**/bin** (Linux)

3. Click **Assignments > Settings** and specify options on the **Simulation** page and **More NativeLink Settings** dialog box. Specify default options for simulation library compilation, netlist and tool command script generation, and for launching RTL or gate-level simulation automatically following Quartus II processing.
4. If your design includes a testbench, turn on **Compile test bench** and then click **Test Benches** to specify options for each testbench. Alternatively, turn on **Use script to compile testbench** and specify the script file.
5. If you want to use a script to setup simulation, turn on **Use script to setup simulation**.

Running RTL Simulation (NativeLink Flow)

To run RTL simulation using the NativeLink flow, follow these steps:

1. Set up the simulation environment.
2. Click **Processing > Start > Analysis and Elaboration**.
3. Click **Tools > Run Simulation Tool > RTL Simulation**. NativeLink compiles simulation libraries and launches and runs your RTL simulator automatically according to the NativeLink settings.
4. Review and analyze the simulation results in your simulator. Correct any functional errors in your design. If necessary, re-simulate the design to verify correct behavior.

Running Gate-Level Simulation (NativeLink Flow)

To run gate-level simulation with the NativeLink flow, follow these steps:

1. Prepare for simulation.
2. Set up the simulation environment. To generate only a functional (rather than timing) gate-level netlist, click **More EDA Netlist Writer Settings**, and turn on **Generate netlist for functional simulation only**.
3. To synthesize the design, follow one of these steps:
   - To generate a post-fit functional or post-fit timing netlist and then automatically simulate your design according to your NativeLink settings, Click **Processing > Start Compilation**. Skip to step **step 6**.
   - To synthesize the design for post-synthesis functional simulation only, click **Processing > Start > Start Analysis and Synthesis**.
4. To generate the simulation netlist, click **Start EDA Netlist Writer**.
5. Click **Tools > Run Simulation Tool > Gate Level Simulation**.
6. Review and analyze the simulation results in your simulator. Correct any unexpected or incorrect conditions found in your design. Simulate the design again until you verify correct behavior.

Running a Simulation (Custom Flow)

Use a custom simulation flow to support any of the following more complex simulation scenarios:

- Custom compilation, elaboration, or run commands for your design, IP, or simulation library model files (for example, macros, debugging/optimization options, simulator-specific elaboration or run-time options)
- Multi-pass simulation flows
- Flows that use dynamically generated simulation scripts

Use these to compile libraries and generate simulation scripts for custom simulation flows:

- NativeLink-generated scripts—use NativeLink only to generate simulation script templates to develop your own custom scripts.
- Simulation Library Compiler—compile Altera simulation libraries for your device, HDL, and simulator. Generate scripts to compile simulation libraries as part of your custom simulation flow. This tool does not compile your design, IP, or testbench files.
- IP and Qsys simulation scripts—use the scripts generated for Altera IP cores and Qsys systems as templates to create simulation scripts. If your design includes multiple IP cores or Qsys systems, you can combine the simulation scripts into a single script, manually or by using the `ip-make-simscript` utility.

Use the following steps in a custom simulation flow:

1. Compile the design and testbench files in your simulator.
2. Run the simulation in your simulator.

Post-synthesis and post-fit gate-level simulations run significantly slower than RTL simulation. Altera recommends that you verify your design using RTL simulation for functionality and use the TimeQuest timing analyzer for timing. Timing simulation is not supported for Arria V, Cyclone V, Stratix V, and newer families.

Related Information

Running EDA Simulators

Generating Simulation Scripts

You can automatically generate simulation scripts to set up supported simulators. These scripts compile the required device libraries and system design files in the correct order, and then elaborate or load the top-level design for simulation. You can also use scripts to modify the top-level simulation environment, independent of IP simulation files that are replaced during regeneration. You can modify the scripts to set up supported simulators.

Use the NativeLink feature to generate simulation scripts to automate simulation steps. You can reuse these generated files and simulation scripts in a custom simulation flow. NativeLink optionally generates scripts for your simulator in the project subdirectory.

1. Click **Assignments > Settings**.
2. Under **EDA Tool Settings,** click **Simulation**.
3. Select the **Tool name** of your simulator.
4. Click **More NativeLink Settings**.
5. Turn on **Generate third-party EDA tool command scripts without running the EDA tool**.

Table 1-7: NativeLink Generated Scripts for RTL Simulation

Simulator(s)	Simulation File	Use
Mentor Graphics ModelSim QuestaSim	**/simulation/modelsim/<*my_ip*>.do**	Source directly with your simulator.
Aldec Riviera Pro	**/simulation/modelsim/<*my_ip*>.do**	Source directly with your simulator.
Synopsys VCS	**/simulation/modelsim/<*revision name*>_<*rtl or gate*>.vcs**	Add your testbench file name to this options file to pass the file to VCS using the `-file` option. If you specify a testbench file to NativeLink, NativeLink generates an **.sh** script that runs VCS.
Synopsys VCS MX	**/simulation/scsim/<*revision name*>_vcsmx_<*rtl or gate*>_<*verilog or vhdl*>.tcl**	Run this script at the command line using the command: `quartus_sh -t` <*script*> Any testbench you specify with NativeLink is included in this script.

Simulator(s)	Simulation File	Use
Cadence Incisive (NC SIM)	**/simulation/ncsim/<*revision name*>_ncsim_<*rtl or gate*>_<*verilog or vhdl*>.tcl**	Run this script at the command line using the command: `quartus_sh -t` <*script*>. Any testbench you specify with NativeLink is included in this script.

You can use the following script variables:

- `TOP_LEVEL_NAME`—The top-level entity of your simulation is often a testbench that instantiates your design, and then your design instantiates IP cores and/or Qsys systems. Set the value of `TOP_LEVEL_NAME` to the top-level entity.
- `QSYS_SIMDIR`—Specifies the top-level directory containing the simulation files.
- Other variables control the compilation, elaboration, and simulation process.

Generating Custom Simulation Scripts with ip-make-simscript

Use the `ip-make-simscript` utility to generate simulation command scripts for multiple IP cores or Qsys systems. Specify all Simulation Package Descriptor files (**.spd**), each of which lists the required simulation files for the corresponding IP core or Qsys system. The IP parameter editor generates the **.spd** files.

`ip-make-simscript` compiles IP simulation models into various simulation libraries. Use the `compile-to-work` option to compile all simulation files into a single work library. Use this option only if you require a simplified library structure.

When you specify multiple **.spd** files, the `ip-make-simscript` utility generates a single simulation script containing all required simulation information. The default value of `TOP_LEVEL_NAME` is the `TOP_LEVEL_NAME` defined in the IP core or Qsys **.spd** file.

Set appropriate variables in the script, or edit the variable assignment directly in the script. If the simulation script is a Tcl file that is sourced in the simulator, set the variables before sourcing the script. If the simulation script is a shell script, pass in the variables as command-line arguments to the shell script.

- To run `ip-make-simscript`, type the following at the command prompt:

```
<Quartus II installation path>\quartus\sopc_builder\bin\ip-make-simscript
```

Table 1-8: ip-make-simscript Examples

Option	Description	Status
`--spd=`<*file*>	Describes the list of compiled files and memory model hierarchy. If your design includes multiple IP cores or Qsys systems that include **.spd** files, use this option for each file. For example: `ip-make-simscript --spd=ip1.spd --spd=ip2.spd`	Required
`--output-directory=`<*directory*>	Specifies the location of output files. If unspecified, the default setting is the directory from which `ip-make-simscript` is run.	Optional

Option	Description	Status
`--compile-to-work`	Compiles all design files to the default work library. Use this option only if you encounter problems managing your simulation with multiple libraries.	Optional
`--use-relative-paths`	Uses relative paths whenever possible.	Optional

Related Information

- **Aldec Active-HDL and Riviera-PRO Support**
- **Synopsys VCS and VCS MX Support**
- **Mentor Graphics ModelSim and QuestaSim Support**

Document Revision History

Date	Version	Changes
2014.06.30	14.0.0	• Replaced MegaWizard Plug-In Manager information with IP Catalog.
May 2013	13.0.0	• Updated introductory section and system and IP file locations.
November 2012	12.1.0	• Revised chapter to reflect latest changes to other simulation documentation.
June 2012	12.0.0	• Reorganization of chapter to reflect various simulation flows. • Added NativeLink support for newer IP cores.
November 2011	11.1.0	• Added information about encrypted Altera simulation model files. • Added information about IP simulation and NativeLink.

Related Information

Quartus II Handbook Archive