# 1. Simulating Altera Designs

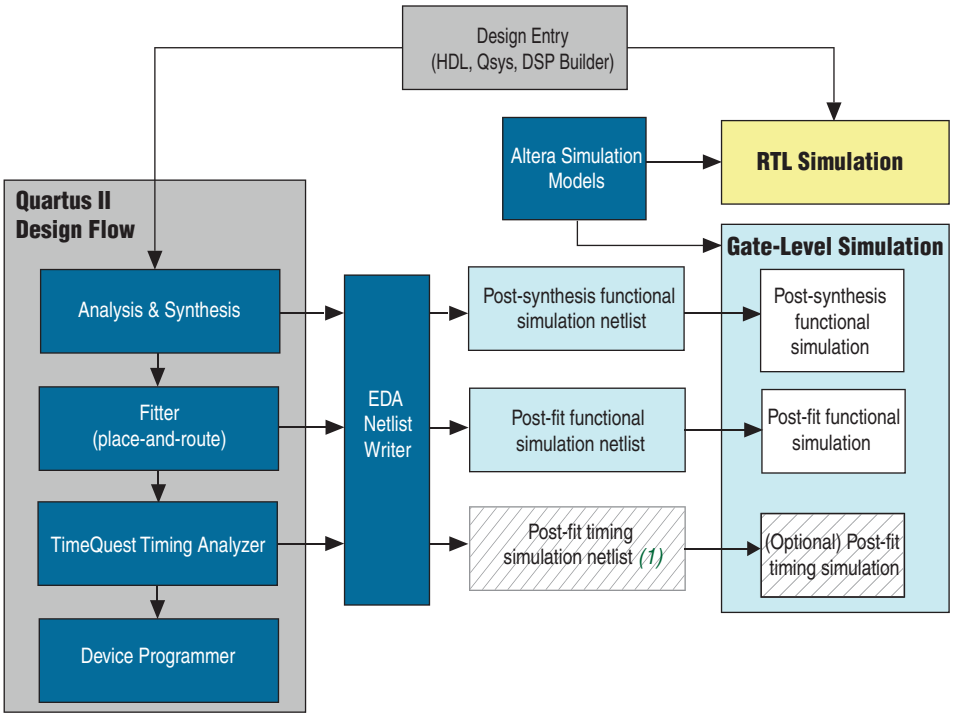
QII53025-13.0.0

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 third-party EDA simulators.

## Altera Simulation Overview

Simulation involves setting up your simulator working environment, compiling simulation model libraries, and running your simulation. Generate simulation files in an automated or custom flow. Refer to Figure 1–1 and Table 1–3.

**Figure 1–1. Simulation in Quartus II Design Flow**




(1) Timing simulation is not supported for Arria® V, Cyclone® V, Stratix® V, and newer families.

You can use the Quartus II NativeLink feature to automatically generate simulation files and scripts. NativeLink can launch your simulator a from within the Quartus II software. Use a custom flow for more control over all aspects of simulation file generation.

# Simulator Support

The Quartus II software supports specific versions of the following EDA simulators for RTL and gate-level simulation.

**Table 1–1. Supported Simulators**

| Vendor | Simulator | Platform |
|---|---|---|
| Aldec | Active-HDL | Windows |
| Aldec | Riviera-PRO | Windows, Linux |
| Cadence® | Incisive Enterprise | Linux |
| Mentor Graphics | ModelSim-Altera (provided) | Windows, Linux |
| Mentor Graphics | ModelSim PE | Windows |
| Mentor Graphics | ModelSim® SE | Windows, Linux |
| Mentor Graphics | QuestaSim | Windows, Linux |
| Synopsys | VCS/VCS MX | Linux |

## Simulation Levels

Table 1–2 describes the supported Quartus II simulation levels.

**Table 1–2. 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<br>■ Altera simulation libraries<br>■ Altera IP plain text or IEEE encrypted RTL models<br>■ IP simulation models<br>■ Altera IPFS models<br>■ Altera IP BFMs<br>■ Qsys-generated models<br>■ 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<br>■ Altera simulation libraries<br>■ Post-synthesis or post-fit functional netlist<br>■ Altera IP Bus BFMs |
| Gate-level timing | Simulation using a post-fit timing netlist, testing design's functional and timing correctness. Not supported for Arria V, Cyclone V, or Stratix V devices. | ■ Testbench<br>■ Altera simulation libraries<br>■ Post-fit timing netlist<br>■ Post-fit Standard Delay Output File (**.sdo**) |

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. Rely on TimeQuest static timing analysis rather than on gate-level timing simulation.

## Simulation Flows

Table 1–3 describes the supported Quartus II simulation flows.

**Table 1–3. Simulation Flows**

| Simulation Flow | Description |
|---|---|
| NativeLink flow | The NativeLink automated flow supports a variety of design flows. NativeLink is not recommended if you require direct control over every aspect of simulation.<br>■ Use NativeLink to generate simulation scripts to compile your design and simulation libraries, and to automatically launch your simulator, as described in "Setting Up Simulation (NativeLink Flow)" on page 1–8.<br>■ Specify your own compilation, elaboration, and simulation scripts for testbench and simulation model files that have not been analyzed by the Quartus II software.<br>■ Use NativeLink to supplement your scripts by automatically compiling:<br>  ■ Design files<br>  ■ IP simulation model files<br>  ■ Altera simulation library models |
| Custom flows | Custom flows support manual control of all aspects of simulation, including the following:<br>■ Manually compile and simulate testbench, design, IP, and simulation model libraries, or write scripts to automate compilation and simulation in your simulator.<br>■ Use the Simulation Library Compiler to compile simulation libraries for all Altera devices and supported third-party simulators and languages, as described in "Using IP and Qsys Simulation Setup Scripts (Custom Flow)" on page 1–12.<br>Use the custom flow if you require any of the following:<br>■ Custom compilation commands for design, IP, or simulation library model files (for example, macros, debugging or optimization options, or other simulator-specific options).<br>■ Multi-pass simulation flows.<br>■ Flow that use dynamically generated simulation scripts. |
| Specialized flows | Altera supports specialized flows for various design variations, including the following:<br>■ For simulation of Altera example designs, refer to the documentation for the example design or to the IP core user guide on the IP and Megafunctions Documentation section of the Altera website.<br>■ For simulation of Qsys designs, refer to *Creating a System with Qsys* chapter of the *Quartus II Handbook.*<br>■ For simulation of designs that include the Nios II embedded processor, refer to *AN 351: Simulating Nios II Embedded Processors Designs*. |

## HDL Support

Table 1–4 describes Quartus II simulation support for hardware description languages:

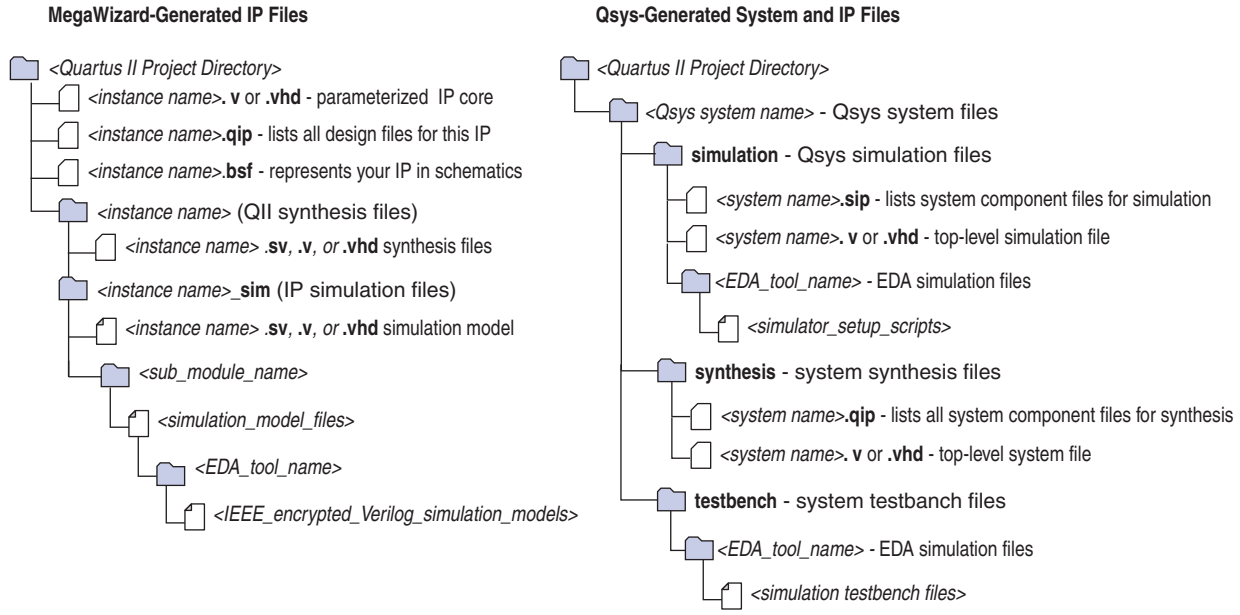
**Table 1–4. HDL Support**

| 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.<br>■ 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.<br>■ 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.<br>■ 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 and Verilog/SystemVerilog files, you must use a mixed language simulator. Since Altera supports both languages, choose the most convenient language for any Altera IP in your design.<br>■ Altera provides Stratix V, Arria V, Cyclone 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. |

## System and IP File Locations

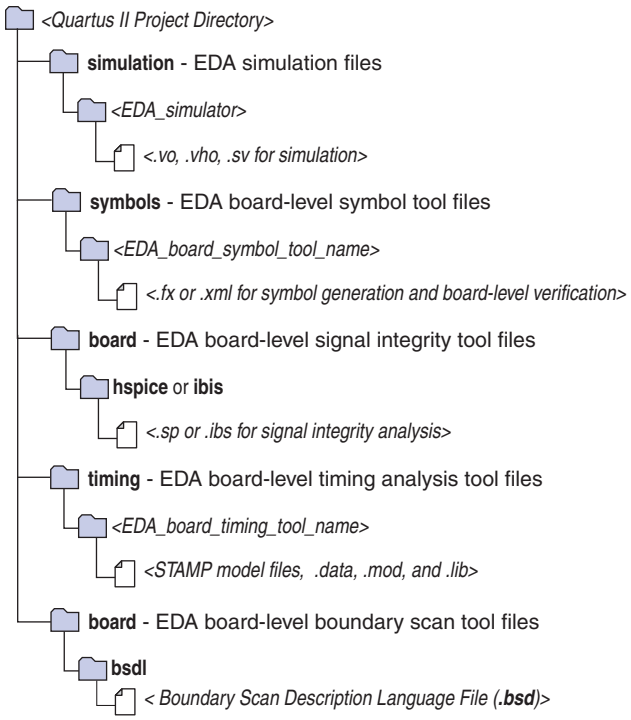
The Quartus II software generates the following files for Altera IP cores:.

**Figure 1–2. System and IP Files Generated by MegaWizard Plug-In Manager and Qsys**




The Quartus II software optionally generates the following files for other EDA tools:

**Figure 1–3. Quartus II Generated Files for Other EDA Tools**



# 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 megafunctions, primitives, library of parameterized modules (LPMs), IPFS 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 in the simulators listed in Table 1–1. 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, as described in "Running RTL Simulation (NativeLink Flow)" on page 1–9.
- Run the Simulation Library Compiler to compile all RTL and gate-level simulation model libraries for your device, simulator, and design language, as described in "Using Simulation Library Compiler (Custom Flow)" on page 1–10.
- Compile Altera simulation models manually with your simulator, as described in *Preparing for EDA Simulation* in Quartus II Help.

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

For a complete list of the Altera simulation models, refer to *Altera Simulation Models* in Quartus II Help.

## 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 along with the rest of your design and testbench. The Quartus II software generates and copies the simulation models for IP cores to your project design directory. For information about the location of IP simulation models for the IP cores in your design, refer to "Document Revision History" on page 1–13.

The Quartus II software can generate one or more of the files in Table 1–5 to support the IP core simulation. If generated, use these files to simulate your Altera IP core.

**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. Refer to "Using IP and Qsys Simulation Setup Scripts (Custom Flow)" on page 1–12. | Cadence<br>■ **cds.lib**<br>■ **ncsim_setup.sh**<br>■ **hdl.var**<br>Mentor Graphics<br>■ **msim_setup.tcl**<br>Synopsys<br>■ **synopsys_sim.setup**<br>■ **vcs_setup.sh**<br>■ **vcsmx_setup.sh**<br>Aldec<br>■ **rivierapro_setup.tcl** |
| Quartus II Simulation IP File (**.sip**) | Contains IP core simulation library mapping information..**sip** files enable NativeLink simulation and the Quartus II Archiver for IP cores. | *<design name>*.**sip** |
| IPFS models | IP Functional Simulation (IPFS) models are cycle-accurate VHDL or Verilog HDL models generated by the Quartus II software for some Altera IP cores. IPFS models support fast functional simulation of IP using industry-standard VHDL and Verilog HDL simulators. Refer to "Generating IP Functional Simulation Models for RTL Simulation" on page 1–7. | *<design name>*.**vho**<br>*<design name>*.**vo** |
| IEEE encrypted models | Stratix V, Arria V, Cyclone 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 IPFS models. | *<design name>*.**v** |

## Generating IP Functional Simulation Models for RTL Simulation

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

- Turn on the **Generate Simulation Model** option when parameterizing the IP core in the MegaWizard Plug-In Manager.
- When you simulate your design, only compile the **.vo** or **.vho** for these IP cores in your simulator, rather than the corresponding HDL file, which may be encrypted to support only synthesis by the Quartus II software.

Altera IP cores that do not require IPFS models for simulation lack the **Generate Simulation Model** option in the IP core parameter editor.

Many recently released Altera IP cores support RTL simulation using IEEE Verilog HDL encryption. IEEE encrypted models are significantly faster than IPFS models and can be simulated in both Verilog HDL and VHDL designs.

Generating an IPFS model for some AMPP megafunctions may require a license, refer to *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)<br>*/<simulator install path>*/**bin** (Linux) |
| Mentor Graphics ModelSim<br>Mentor Graphics QuestaSim | *<drive letter>*:*<simulator install path>***win32** (Windows)<br>*<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<br>Aldec Riviera-PRO | *<drive letter>*:*<simlulator install path>***bin** (Windows)<br>*<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, as described in "Setting Up Simulation (NativeLink Flow)" on page 1–8.
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, as described in "Preparing for Simulation" on page 1–6.
2. Set up the simulation environment, as described in "Setting Up Simulation (NativeLink Flow)" on page 1–8. 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 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, described in "Generating Custom Simulation Scripts with ip-make-simscript" on page 1–12.

Use the following steps in a custom simulation flow:

1. "Preparing for Simulation" on page 1–6.
2. "Using Simulation Library Compiler (Custom Flow)" on page 1–10
3. "Using NativeLink-Generated Scripts (Custom Flow)" on page 1–11.
4. "Using IP and Qsys Simulation Setup Scripts (Custom Flow)" on page 1–12.
5. Compile the design and testbench files in your simulator.
6. 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.

For more information about running EDA simulation, refer to *Running EDA Simulators* in Quartus II Help.

## Using Simulation Library Compiler (Custom Flow)

Simulation Library Compiler compiles all required Quartus II simulation library files for your HDL, device, and simulator. If your design includes IP cores generated with the classic IP file directory structure in Figure 1–2, you may need to compile additional library files.

If your design includes IP cores generated with the IP file directory structure illustrated in Figure 1–2, refer to "Generating Custom Simulation Scripts with ip-make-simscript" on page 1–12 to use the scripts in combination with the Simulation Library Compiler's generated simulation scripts.

For detailed steps on using Simulation Library Compiler, refer to *Preparing for EDA Simulation* in Quartus II Help. For a complete list of the Altera simulation models, refer to *Altera Simulation Models* in Quartus II Help.

## Using NativeLink-Generated Scripts (Custom Flow)

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 described in Table 1–7. To generate simulation scripts using the NativeLink feature, perform the following steps:

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/***<design>***.do** | Source directly with your simulator. |
| Aldec Riviera Pro | **/simulation/modelsim/***<design>***.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, and direct not to simulate, 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 `quartus_sh -t` *<script>*<br>Any testbench you specify with NativeLink is included in this script. |
| Cadence Incisive (NC SIM) | **/simulation/ncsim/***<revision name>***_ncsim_***<rtl or gate>_<verilog or vhdl>***.tcl** | Run this script at the command line using `quartus_sh -t` *<script>*.<br>Any testbench you specify with NativeLink is included in this script. |

# Using IP and Qsys Simulation Setup Scripts (Custom Flow)

Altera IP cores and Qsys systems generate simulation setup scripts. Modify these scripts to set up supported simulators. Use the scripts to compile the required device libraries and system design files in the correct order, and then elaborate or load the top-level design for simulation. Also use the script to modify the top-level simulation environment independent of the IP simulation files that are over-written during regeneration.

These simulation scripts variables set up your simulation environment:

- `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 simulation 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 MegaWizard Plug-In Manager and Qsys generate the **.spd** files.

This utility 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. If this is not the top-level instance in your design, specify the top-level instance of your testbench or design.

Setting appropriate variables in the script, or edit the variable assignment directly in the script. If the simulation script is a tcl file that can be 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 shell script.

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

*<ACDS installation path>*`\quartus\sopc_builder\bin\ip-make-simscript`

The following are examples of options you can use with the utility:

**Table 1–8.**

| 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:<br>`ip-make-simscript --spd=ip1.spd --spd=ip2.spd` | Required |
| `--output-directory=`*<directory>* | Directory path specifying the location of output files. If unspecified, the default setting is the directory from which `ip-make-simscript` is run. | Optional |
| `--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 |

Refer to *Aldec Active-HDL and Riviera-PRO Support*, *Synopsys VCS and VCS MX Support*, *Cadence Incisive Enterprise Simulator Support*, and *Mentor Graphics ModelSim and QuestaSim Support* for simulation script examples.

# Document Revision History

Table 1–9 shows the revision history for this chapter.

**Table 1–9. Document Revision History (Part 1 of 2)**

| Date | Version | Changes |
|---|---|---|
| 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.<br>■ Added NativeLink support for newer IP cores. |
| November 2011 | 11.1.0 | ■ Added information about encrypted Altera simulation model files.<br>■ Added information about IP simulation and NativeLink. |
| May 2011 | 11.0.0 | ■ Added note to Figure 1–1 on page 1–2<br>■ Added new section “Converting Block Design Files (.bdf) to HDL Format (.v/.vhd)” on page 1–4<br>■ Updated information in “Simulation Netlist Files”.<br>■ Updated information in “Generating Gate-Level Timing Simulation Netlist Files”.<br>■ Updated information in “Generating Post-Synthesis Simulation Netlist Files”.<br>■ Removed information from “Generating Timing Simulation Netlist Files with Different Timing Models”.<br>■ Removed information from “Running the Simulation Library Compiler Through the GUI”.<br>■ Updated Table 1–1.<br>■ Updated “Simulating Qsys and SOPC Builder System Designs” |

**Table 1–9. Document Revision History (Part 2 of 2)**

| Date | Version | Changes |
|---|---|---|
| December 2010 | 10.1.0 | ■ Title changed from "Simulating Designs with EDA Tools".<br>■ Merged content from "Simulating Altera IP in Third-Party Simulation Tools" chapter to "Simulating Altera IP Cores".<br>■ Added new section "IP Variant Directory Structure".<br>■ Added new section "Simulating Qsys and SOPC Builder System Designs".<br>■ Added information about simulating designs with Stratix V devices<br>■ Updated chapter to new template |
| July 2010 | 10.0.0 | ■ Linked to Quartus II Help where appropriate<br>■ Removed Referenced Documents section<br>■ Removed Creating Testbench Files<br>■ Added VCS and QuestaSim as third-party simulation tools<br>■ Updated "Running the EDA Simulation Library Compiler Through the GUI" on page 1–18<br>■ Updated "Setting Up the EDA Simulator Execution Path".<br>■ Updated "Configuring NativeLink Settings"<br>■ Updated "Setting Up Testbench Files Using the NativeLink Feature" |
| November 2009 | 9.1.0 | Initial release |

For previous versions of the *Quartus II Handbook*, refer to the Quartus II Handbook Archive.