Simulating Nios II Embedded Processor Designs

AN-351-1.4 Application Note

This application note describes the process of generating an RTL simulation environment with Nios® II example designs, Qsys, and the Nios II Software Build Tools (SBT) for Eclipse. This application note also describes the process of running the Nios II RTL simulation in the ModelSim®-Altera® Edition simulator.

The increasing pressure to deliver robust products to market in a timely manner has amplified the importance of comprehensively verifying embedded processor designs. Therefore, consider the verification solution supplied with the processor when choosing an embedded processor. Nios II embedded processor designs support a broad range of verification solutions, including the following:

- **Board Level Verification**—Altera offers a number of development boards that provide a versatile platform for verifying both the hardware and software of a Nios II embedded processor system. You can use the Nios II SBT for Eclipse with its built-in debugger to verify designs running on either development or custom boards. You can further debug the hardware components that interact with the processor with the SignalTap™ II embedded logic analyzer.

  For more information about the Nios II SBT for Eclipse, refer to the *Nios II Software Developer's Handbook*.

  For more information about the SignalTap II embedded logic analyzer, refer to *AN 323: Using SignalTap II Embedded Logic Analyzers in SOPC Builder Systems* and *AN 446: Debugging Nios II Systems with the SignalTap II Embedded Logic Analyzer*.

- **Register Transfer Level (RTL) Simulation**—RTL simulation is a powerful means of debugging the interaction between a processor and its peripheral set. When debugging a target board, it is often difficult to view signals buried deep in the system. RTL simulation alleviates this problem as it enables you to functionally probe every register and signal in the design. You can easily simulate Nios II-based systems in the ModelSim simulator with an automatically generated simulation environment that Qsys and the Nios II SBT for Eclipse create.

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Before You Begin

This document assumes that you have prior experience using Qsys as well as a familiarity with the ModelSim simulator. In order to simulate the Nios II design using the instructions in this document, you must have the following software installed:

- The Quartus® II software version 11.0 or later
- ModelSim-Altera Edition version 6.6d or higher
- Nios II Embedded Design Suite version 11.0 or later

Setting Up and Generating Your Simulation Environment in Qsys

To open the example design, perform the following steps:

1. Download the **an351_design.zip** design example from the Simulating Nios II Embedded Processor Design page on the Altera website, and then extract the design example to your hard drive. The location to which you extract the file is referred to as <*your project directory*> throughout the remainder of this document.
2. Start the Quartus II software.
3. On the File menu, click **Open Project**.
4. Browse to <*your project directory*>/**an351_design**.
5. Select **an351_project.qpf**.
6. Click **Open**.
7. On the Tools menu, click **Qsys**.
8. Open the **niosii_system.qsys** file.

   The design example used for this application note is a complete Qsys system. Ensure that you have completed building your Qsys system before you start to generate the simulation models.

9. On the **Generation** tab, set the following parameters to these values:
   - **Create simulation model—None**
   - **Create testbench Qsys system—Simple, BFMs for clocks and resets**

     If your system has exported ports other than the clock and reset, choose **Standard, BFMs for standard Avalon interfaces**.

   - **Create testbench simulation model—Verilog**
   - **Create HDL design files for synthesis**—Turn off
   - **Create block symbol file (.bsf)**—Turn off
10. Click **Generate**. Save the system if prompted.

Qsys-Generated System Simulation Files

At this point in the design flow, Qsys has generated your system and created all of the files necessary for simulation listed in Table 1. These simulation files are located in the *<your project directory>*/**an351_design/niosii_system/testbench** directory.

Table 1. Qsys Files Generated for Nios II Simulation

File	Description
Qsys testbench system files	Qsys generates a testbench system when you enable the **Create testbench Qsys system** option. Qsys connects the corresponding Avalon Bus Functional Models to all exported interfaces of your system. For more information about Qsys, refer to the *System Design with Qsys* section in volume 1 of the *Quartus II Handbook*.
msim_setup.tcl	Sets up a ModelSim simulation environment and creates alias commands to compile the required device libraries and system design files in the correct order, and loads the top-level design for simulation.
Memory Initialization Files (**.mif**)	Creates Memory Initialization Files (**.mif**) to initialize memory components in your system. Use Nios II SBT for Eclipse to create Nios II processor program to populate the **.mif** files.

Memory Simulation Models

You can use two types of memory models for simulation purposes: generic and vendor-specific. For Altera-provided memory controllers, you are provided with generic simulation models. If you are using custom memory controllers, you should use the simulation models the memory controller vendor provides. This application note discusses the generic memory model.

Using IP and Qsys Simulation Setup Scripts

Altera IP cores and Qsys systems generate simulation setup scripts. Modify these scripts to set up supported simulators.

For more information on generating custom simulation scripts, please refer to the *Using IP and Qsys Simulation Setup Scripts (Custom Flow)* section of the Quartus II Handbook Version 13.1.

Creating the Nios II Software

This section describes how to finish setting up your simulation by using the Nios II SBT for Eclipse to create a software test project and to generate the necessary files for initializing the memories used in your simulation.

Creating a Nios II SBT for Eclipse Project

In this application note, you simulate a simple Hello World program with Qsys. The Hello World software prints a message to the console via JTAG UART. To create and build the software project, perform the following steps:

1. Open the Nios II SBT for Eclipse version 11.0 or later.
2. On the **File** menu, point to **New**, and click **Nios II Application and BSP from Template**.

3. Select the SOPC Information (**.sopcinfo**) file name by browsing to *<your project directory>*/**an351_design**, and then select **niosii_system**.**sopcinfo**.
4. For **Project Name**, type **hello_world_an351**.
5. Select **Hello World** from the **Templates** option.
6. Click **Finish**.
7. Right-click on **hello_world_an351** in Project Explorer and then click **Build Project**.

Now you have successfully built the Hello World project. In the next step, you will invoke ModelSim simulation from the Nios II SBT for Eclipse. This function populates the Memory Initialization File (**.mif**) with the Hello World program and starts the ModelSim software.

8. Right-click on **hello_world_an351** in Project Explorer. Point to **Run As**, and then click **Nios II ModelSim**.

Running Simulation in the ModelSim Simulator

After you have launched the ModelSim simulator from the Nios II SBT for Eclipse, ModelSim automatically compiles the required device libraries and system design files, and elaborates and loads the top-level design. The **msim_setup**.**tcl** script creates alias commands for each of the steps. These commands are listed in Table 2.

Table 2. Nios II Alias Commands

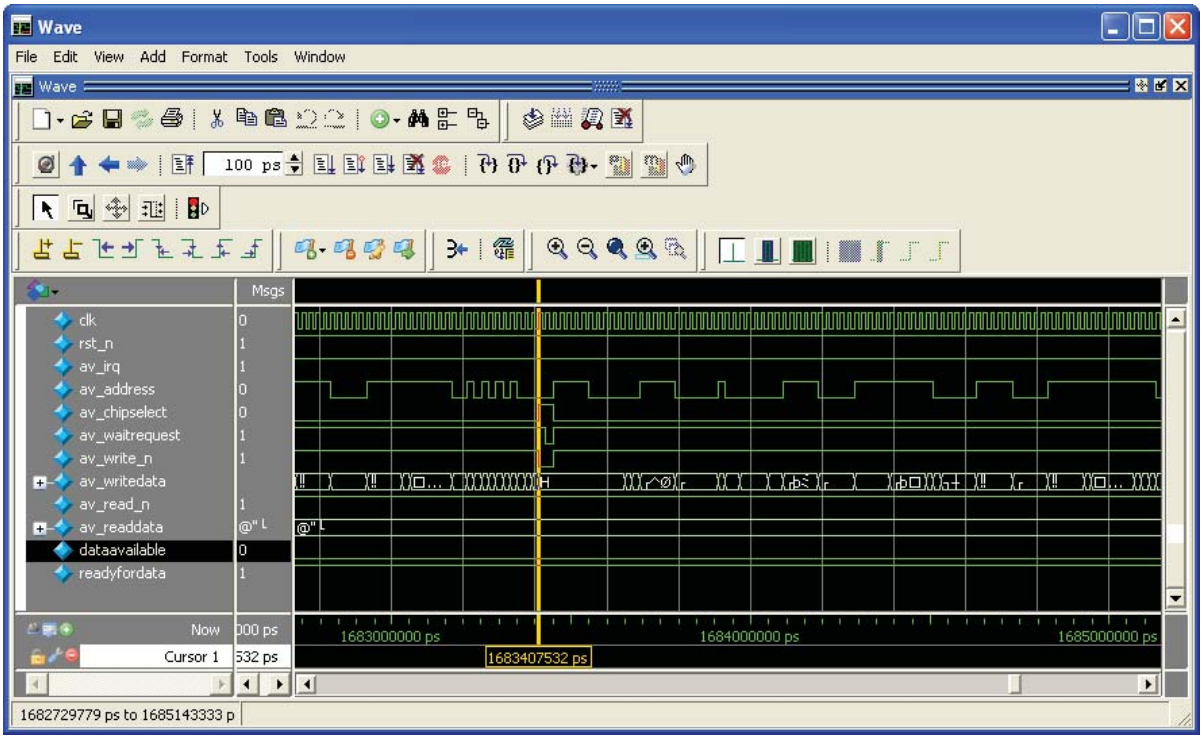
Macros	Description
`dev_com`	Compiles device library files.
`com`	Compiles the design files in correct order.
`elab`	Elaborates the top-level design.
`elab_debug`	Elaborates the top-level design with the **novopt** option.
`ld`	Compiles all the design files and elaborates the top-level design.
`ld_debug`	Compiles all the design files and elaborates the top-level design with the **novopt** option.

Run the simulation in the ModelSim simulator by performing the following steps:

1. In the ModelSim software, on the **File** menu, click **Load**. Browse to *<your project directory>*/**an351_design** and select **wave**.**do**. This step opens a waveform viewer with all the JTAG UART signals.
2. In the Transcript window, type `run 2 ms`. This step starts the simulation for two milliseconds.

At the end of the simulation, you should see a "Hello from Nios II!" message in the Transcript window. You can observe the simulation results from the waveform viewer as well. Figure 1 shows the simulation result. The waveform is zoomed in at a specific simulation time in which the Nios II processor writes the first `H` character to the JTAG UART component.

Figure 1. Simulation Results



Conclusion

Simulation and verification are vital parts of the design process. You can comprehensively verify the Nios II processors with board-level debugging, and RTL simulation with the ModelSim simulator. RTL simulation is an important part of the design process, particularly for configurable systems, because it enables you to probe deeply embedded signals in the processor and your peripheral set. RTL simulation also helps verify your system before you try out your design in the actual hardware.

Document Revision History

Table 3 shows the revision history for this document.

Table 3. Document Revision History

Date	Version	Changes
November 2013	1.4	Added the "Using IP and Qsys Simulation Setup Scripts" section.
June 2011	1.3	Updated for the Nios II processor 11.0 release
November 2008	1.2	Updated for the Nios II processor 8.1 release
November 2007	1.1	Updated for the Nios II processor 7.2 release
May 2001‘	1.0	Initial release.