Simulating Designs with Lower-Level Qsys Systems

As described in AN 351: Simulating Nios II Embedded Processor Designs, Qsys and Nios II Software Build Tools (SBT) for Eclipse can automatically generate an RTL simulation environment for Nios II designs in which the Qsys system is the top level entity. This guide shows to modify the automatically generated simulation files to accommodate designs in which the Qsys system is not the top level. This guide is presented using Verilog, but the process is similar for designs using VHDL.

Before You Begin

This guide assumes that you have prior experience using Qsys as well as a familiarity with Nios II SBT and the ModelSim simulator. In order to simulate the Nios II design using the instructions in this guide, 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

Design Description

As shown in Figure 1, the top level module of the example design used in this guide instantiates a Qsys system (**niosii_system**) and a simple counter module (**counter**). The output PIO of the Qsys system (**output_pio_export**) is a one-bit port that drives the enable input of the counter module.

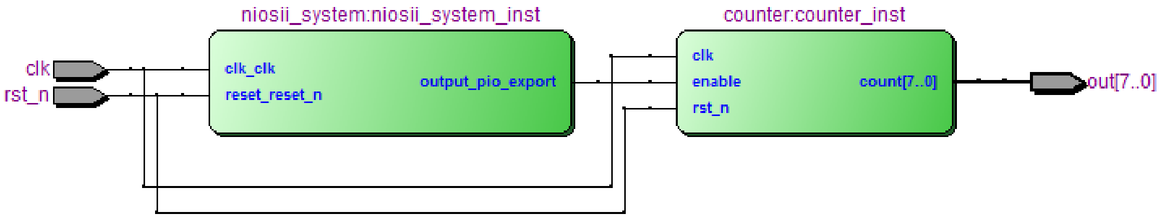


Figure 1 – The Qsys system and counter module instantiated by the top level module

The example software running on the Nios II system demonstrates interactions with hardware outside the top level module (the JTAG UART) and with a module below the top level (the counter module). As shown below, the example software enables the counter while a Hello World message is sent to the console via the JTAG UART.

```
int main()
{
        IOWR(OUTPUT_PIO_BASE, 0, 0x1);         // Enable the counter
        printf("Hello from Nios II!\n");       // Send Hello World to the JTAG UART
        IOWR(OUTPUT_PIO_BASE, 0, 0x0);         // Disable the counter

        return 0;
}
```

Generating the Simulation Files

This section summarizes the process of generating the simulation files. A more detailed explanation can be found in AN 351: Simulating Nios II Embedded Processor Designs. The necessary modifications to these files are described in the next section, Modifying the Simulation Files.

Setting Up and Generating the Simulation Environment in Qsys

1. Download the **lower_level_qsys.zip** design example listed in the Downloads section below. Extract the design example to a new directory. This directory will be referred to as ***<project directory>*** throughout the remainder of this guide.
2. Start the Quartus II software, and select **File > Open Project**. Browse to ***<project directory>***, select the **lower_level_qsys.qpf** file, and click **Open**.
3. Open Qsys by selecting **Tools > Qsys**. Open the **niosii_system.qsys** file.
    - The design example used for this guide is a complete Qsys system. If you are using your own Qsys design, ensure that your design is complete before you start to generate simulation models.
4. On the Generation tab, set the following parameters to these values.
    - **Create simulation model—None**
    - **Create testbench Qsys system—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**
5. Click **Generate**. Save the system if prompted.

Creating the Nios II Software

The software used for this guide will activate the counter module by writing to the output PIO and will send a Hello World message to the console via JTAG UART. To create and build the software project, perform the following steps:

1. Open Nios II SBT for Eclipse version 11.0 or later, and select **File > New > Nios II Application and BSP from Template**.
2. Select the SOPC Information File by browsing to ***<project directory>*** and selecting **niosii_system.sopcinfo**.
3. For Project Name, type "**lower_level_qsys**".
4. Select "**Blank Project**" from the Templates options.

5. Click **Finish**.
6. Next, import the **lower_level_qsys.c** source file into the software project by using the Eclipse Import dialog or by copying the source file directly to the software directory:

   ***<project directory>*/software/lower_level_qsys/**
7. To build the project, right-click on "**lower_level_qsys**" in Project Explorer and select "**Build Project**" (or select "**Clean Project**" if you copied the source file directly into the software directory).
8. When the project has finished building, right-click again on "**lower_level_qsys**" in Project Explorer, and select **Run As > Nios II ModelSim**. This command will start the ModelSim software and create the remaining simulation files.

Modifying the Simulation Files

At this point, the Qsys and Nios II SBT software have automatically generated simulation files necessary for simulating designs in which the Qsys system is the top level entity. In order to simulate other types of designs, two of these automatically generated files need to be modified: the testbench source file and the ModelSim simulation script.

Modifying the Testbench Source File

This file should be edited to correctly reflect the hierarchy of your design. The instantiation of the Qsys system should be removed and replaced by an instantiation of the correct top level module. Whether or not other modules instantiated in the testbench file, such as bus functional models (BFMs), should be removed depends on the usage of those modules. In general, you should remove BFM modules that represent components that should exist below the top level module – for example, a PIO from the Qsys system that connects to another module under the top level. You should not remove BFM modules that represent connections to the top level – for example, the clock and reset BFMs.

For this guide, the testbench file, **niosii_system_tb.v**, is located in:

***<project directory>*/niosii_system/testbench/niosii_system_tb/simulation**

In general, this file will be located in:

<project directory>/<Qsys system name>/testbench/<Qsys system name>_tb/simulation

For this example design, you can either make the modifications to the test bench file yourself, or you may copy the completed code into the testbench file from the **niosii_system_tb.v** file, which is located in **<project directory>/modified_files** directory.

To make the modifications, open the automatically generated testbench file in an editor of your choice and complete the following steps:

1. Remove or comment out the instantiation of the Qsys system as shown below:

```
// Removed instantiation of the Qsys system

// niosii_system_tb_niosii_system_inst niosii_system_inst (
//      .clk_clk           (niosii_system_inst_clk_bfm_clk_clk),
//      .reset_reset_n     (niosii_system_inst_reset_bfm_reset_reset),
//      .output_pio_export (niosii_system_inst_output_pio_export)
// );
```

2. Add the instantiation of the correct top level module with appropriate connections to the clock and reset. The **top_level_out** wire has been added in this case to connect to the output port of the top level module.

```
wire [7:0] top_level_out;

top_level top_level_inst (
        .clk            ( niosii_system_inst_clk_bfm_clk_clk ),
        .rst_n          ( niosii_system_inst_reset_bfm_reset_reset ),
        .out            ( top_level_out )
);
```

3. Remove or comment out the instantiation of the PIO BFM and the wire that connected the BFM to the Qsys system:

```
// wire  [7:0] niosii_system_inst_output_pio_export;
// niosii_system_tb_niosii_system_inst_output_pio_bfm niosii_system_inst_output_pio_bfm (
//      .sig_export (niosii_system_inst_output_pio_export)  // conduit.export
// );
```

4. Save the **niosii_system_tb.v** file.

Modifying the ModelSim Simulation Script

The ModelSim simulation script contains Tcl commands that set up the simulation environment and provide several command aliases for quickly performing tasks such as compiling source files and elaborating the top level design. The unmodified simulation script is automatically executed when Nios II SBT invokes ModelSim. The simulation script will need to be edited and re-executed as described below.

Important note: Two simulation scripts with identical names are generated in different directories while setting up the simulation environment. Make sure you are editing the correct file.

For this guide, the correct simulation script, **msim_setup.tcl**, is located in:

***<project directory>*/software/lower_level_qsys/obj/runtime/sim/mentor**

and in general will be located in:

<project directory>/software/<Nios II SBT project name>/obj/runtime/sim/mentor

For this example design, you can either make the modifications to the simulation script file yourself, or you may copy the completed code into the simulation script file from the **msim_setup.tcl** file, which is located in **<project directory>/modified_files** directory.

Important note: In the file paths throughout the completed simulation script, <project directory> is set as "c:/lower_level_qsys/". If you are using a different project directory, use "Find & Replace" to replace "c:/lower_level_qsys/" with your project directory.

To make the modifications yourself, open the automatically generated ModelSim simulation script file in an editor of your choice and complete the following steps. Again, make sure you are editing the correct simulation script.

1. In the "**Compile the design files in correct order**" section, add **vlog** commands to compile the additional source files, as shown below:

```
# ----------------------------------------
# Compile the design files in correct order
alias com {
.
.
.
   vlog -sv "C:/projects/lower_level_qsys/niosii_system/testbench/niosii_system_tb/simulation
        /submodules/verbosity_pkg.sv"
   vlog -sv "C:/projects/lower_level_qsys/niosii_system/testbench/niosii_system_tb/simulation
        /submodules/altera_avalon_clock_source.sv"
   vlog "C:/projects/lower_level_qsys/niosii_system/testbench/niosii_system_tb/simulation
        /submodules/niosii_system_tb_niosii_system_inst.v"

# CHANGE: compile additional design files after the Qsys system, before the test bench

   vlog "C:/projects/lower_level_qsys/niosii_system/testbench/niosii_system_tb/simulation
        /additional_modules/counter.v"
   vlog "C:/projects/lower_level_qsys/niosii_system/testbench/niosii_system_tb/simulation
        /additional_modules/top_level.v"

# END OF CHANGE

   vlog "C:/projects/lower_level_qsys/niosii_system/testbench/niosii_system_tb/simulation
        /niosii_system_tb.v"
}
```

When modifying the simulation script for your own design, make sure that your source files are compiled in the correct order. In ModelSim, a file containing the definition of a module must be compiled before a file containing instantiations of that module. In general, lower levels of your hierarchy should be compiled first, and your testbench should be compiled last.

2. In the "**Elaborate top level design**" and "**Elaborate the top level design with novopt option**" sections, add the hierarchy that is above the Qsys system before "**$SYSTEM_INSTANCE_NAME**", as shown below.

Important note: Do not put a slash between the added hierarchy and the $SYSTEM_INSTANCE_NAME variable.

```
# CHANGE: add the hierarchy that is above the Qsys system before $SYSTEM_INSTANCE_NAME.
# DO NOT put a slash between the added hierarchy and the $SYSTEM_INSTANCE_NAME variable.

# ----------------------------------------
# Elaborate top level design
alias elab "
vsim -t ps \
     -
G/$TOP_LEVEL_NAME/top_level_inst$SYSTEM_INSTANCE_NAME/niosii_system_inst/ram/INIT_FILE=\"C:/proje
cts/qsys_in_lower_hierarchical_level_example/software/hello_world_qsys_in_lower_hierachical_level
_example/mem_init/ram.hex\" \
     -L work -L altera_ver -L lpm_ver -L sgate_ver -L altera_mf_ver -L altera_lnsim_ver -L
cycloneiii_ver $TOP_LEVEL_NAME
"

# ----------------------------------------
# Elaborate the top level design with novopt option
alias elab_debug "
vsim -novopt -t ps \
     -
G/$TOP_LEVEL_NAME/top_level_inst$SYSTEM_INSTANCE_NAME/niosii_system_inst/ram/INIT_FILE=\"C:/proje
cts/qsys_in_lower_hierarchical_level_example/software/hello_world_qsys_in_lower_hierachical_level
_example/mem_init/ram.hex\" \
     -L work -L altera_ver -L lpm_ver -L sgate_ver -L altera_mf_ver -L altera_lnsim_ver -L
cycloneiii_ver $TOP_LEVEL_NAME
"

# END OF CHANGE
```

You do not need to list the test bench in the hierarchical description. **TOP_LEVEL_NAME** is the testbench module.

3. Finally, to ensure that you have edited the correct simulation script, add an echo command at the end of the simulation script file to print a confirmation message in the transcript window, as shown below:

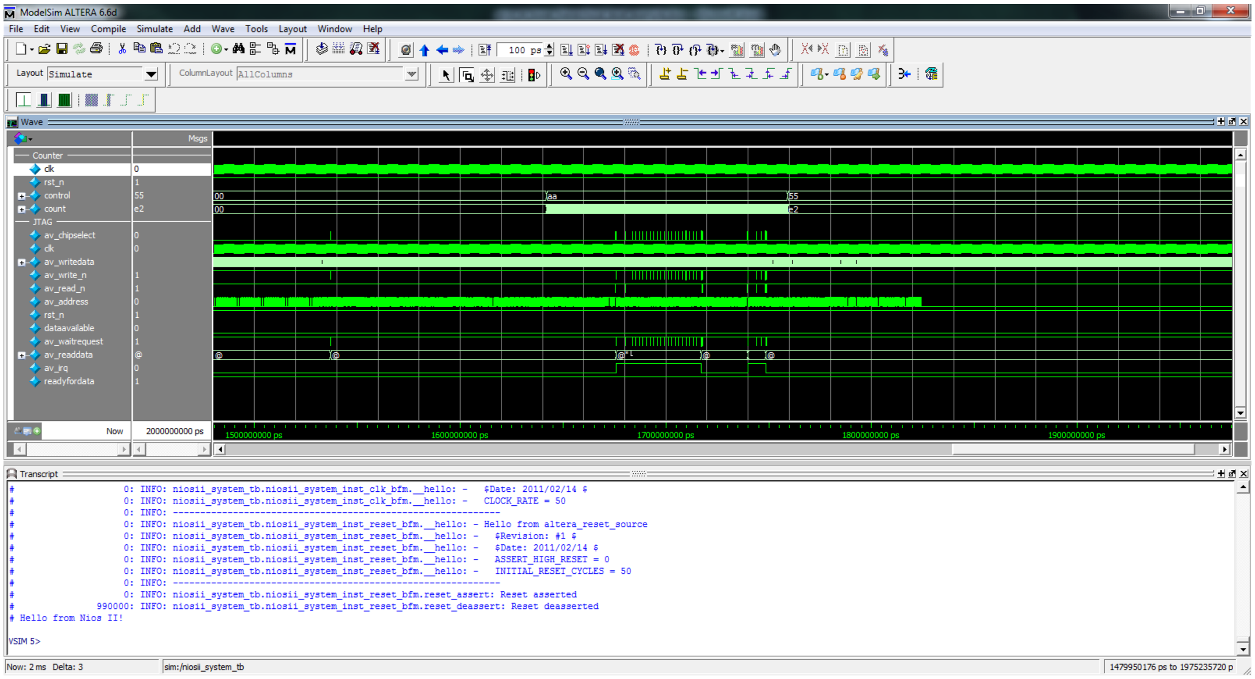
```
# ----------------------------------------
# Confirm that this is the correct simulation script
echo
echo “You edited the correct simulation script.”
```

4. If you are modifying the simulation script for your own design, you may need to add commands to compile libraries. These commands should be added to the “**Create compilation libraries**” and “**Compile device library files**” sections. For this example, no modifications need to be made to these sections. For information on compiling libraries, see the Mentor Graphics ModelSim and QuestaSim Support section of the Quartus II Handbook.
5. Save changes to the **msim_setup.tcl** simulation script.
6. In the ModelSim Transcript window, re-execute the simulation script by typing “**do msim_setup.tcl**” and pressing enter.

7. Finally, recompile the design files and elaborate the top level design by typing “**ld**” and pressing enter. You may optionally use the “**ld_debug**” command instead to use the **-novopt** option for **vsim**.

Running the Simulation

Now that the testbench file and the ModelSim simulation script have both been edited, load signals into the waveform viewer by selecting **File > Load…** and browsing to the ***<project directory>*/wave.do** waveform script file. Begin the simulation by typing “**run 2 ms**”. As shown in Figure 2, the counter is enabled while the Hello World message is sent to the console.



```
#                    0: INFO: niosii_system_tb.niosii_system_inst_clk_bfm.__hello: -   $Date: 2011/02/14 $
#                    0: INFO: niosii_system_tb.niosii_system_inst_clk_bfm.__hello: -   CLOCK_RATE = 50
#                    0: INFO: ------------------------------------------------------------
#                    0: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.__hello: - Hello from altera_reset_source
#                    0: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.__hello: -   $Revision: #1 $
#                    0: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.__hello: -   $Date: 2011/02/14 $
#                    0: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.__hello: -   ASSERT_HIGH_RESET = 0
#                    0: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.__hello: -   INITIAL_RESET_CYCLES = 50
#                    0: INFO: ------------------------------------------------------------
#                    0: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.reset_assert: Reset asserted
#               990000: INFO: niosii_system_tb.niosii_system_inst_reset_bfm.reset_deassert: Reset deasserted
# Hello from Nios II!

VSIM 5>
```

Now: 2 ms Delta: 3   sim:/niosii_system_tb   1479950176 ps to 1975235720 p

Figure 2 - Simulating the lower_level_qsys design

Additional Resources

- AN 351: Simulating Nios II Embedded Processor Designs
- Mentor Graphics ModelSim and QuestaSim Support – Quartus II Handbook, Vol. 3, Ch. 2