intel Quartus Prime Pro Edition Software User Guide

Contents

intel Quartus Prime Pro Edition Software

Product Information

The Questa*-Intel FPGA Edition is a software designed for design simulation and verification before device programming. It involves generating simulation files, compiling simulation models, running the simulation, and viewing the results. The software is available in both paid and free versions, with the Questa-Intel FPGA Starter Edition being a free version. The software can be downloaded from the FPGA Software Download Center page and requires a valid software license that can be obtained from the Intel FPGA Self Service Licensing Center (SSLC).

Product Usage Instructions

Prerequisites

Before using the Questa*-Intel FPGA Edition software, ensure that you have a valid software license. Both the paid and free versions require licenses, but the Questa-Intel FPGA Starter Edition license is free. To download the software, visit the FPGA Software Download Center page, select the desired Intel Quartus Prime Pro software edition, operating system, and download the Questa-Intel FPGA Edition (Includes Starter Edition) software file(s).

Design Simulation Steps

1. **Open the Example Design:** Open the design example mentioned on page 6 of the user manual.
2. **Specify EDA Tool Settings:** Specify EDA tool settings on page 7.
3. **Generate a Simulator Setup Script Template:** Generate a simulator setup script template on page 8.
4. **Create the Simulation Script:** Create the simulation script on page 9.
5. **Compile and Simulate the Design:** Compile and simulate the design on page 12.
6. **View Signal Waveforms:** View signal waveforms on page 13.
7. **Add Signals to the Simulation:** Add signals to the simulation on page 15.
8. **Rerun Simulation:** Rerun simulation on page 16.

Note that this user manual provides basic features to simulate the design example mentioned on page 6. If you intend to use a different use case with advanced features and need more information to simulate your design, refer to the Questa*-Intel FPGA Edition documents from Siemens* available in the /questa_fe/docs/pdf_docs directory.

Questa*-Intel® FPGA Edition Quick-Start Intel® Quartus® Prime Pro Edition

This document demonstrates how to simulate an Intel® Quartus® Prime Pro Edition design in the Questa*-Intel FPGA Edition simulator.

Note:
This document is prepared specifically with basic features to accommodate requirements to simulate the design example mentioned in Open the Example Design on page 6. If you intend to use a different use case with advanced features and you need more information to simulate your design, then refer to the Questa*-Intel FPGA Edition documents from Siemens* available in the <installation directory>/questa_fe/docs/pdf_docs directory.
Design simulation verifies your design before device programming. Design simulation involves generating simulation files, compiling simulation models, running the simulation, and viewing the results. The following steps describe this flow:

Prerequisites

Both Questa*-Intel FPGA Edition and Questa-Intel FPGA Starter Edition software require valid software licenses. However, the Questa-Intel FPGA Starter Edition license is free.
Downloading the Questa-Intel FPGA Edition and Questa-Intel FPGA Starter Edition Software
To download the software with individual executable files:

1. Visit the FPGA Software Download Center page.
2. Using the left-hand filter pane, perform the following steps to refine the search results:
    - Select the Intel Quartus Prime Design Software option. This displays three Intel Quartus Prime software editions (Pro, Standard or Lite).
    - Select the Intel Quartus Prime Pro software edition. This displays a list of supported software versions.
    - Select the operating system (Linux or Microsoft Windows*).
3. In the refined list of pages, click on the desired page to download the software.
4. Under the Downloads section, click the Individual Files tab.
5. Download the Questa-Intel FPGA Edition (Includes Starter Edition) software file(s) by clicking the Download button below each file name.

Refer to Downloading and Installing Intel FPGA Software in the Intel FPGA Software Installation and Licensing for additional information.

Generating the License

You can obtain a license for the Questa-Intel FPGA Edition and Questa-Intel FPGA Starter Edition software from the Intel FPGA Self Service Licensing Center (SSLC). If you do not have access to SSLC, you must first complete registering to SSLC and create an account by visiting Register for Intel FPGA Self Service Licensing Center (SSLC).

Follow these steps to generate the license:

1. Go to the Intel FPGA Self-Service Licensing Center (SSLC).
2. Select the Sign up for Evaluation or Free Licenses option on the menu bar.
3. In the list of products displayed, select the Questa-Intel FPGA Starter Edition SW-QUESTA option.
4. Under the # of Seats column, enter the number of seats you require.
5. Read the license terms of use.
6. Select the “I have read and agree to the terms of use of this license as listed below” check box.
7. Click Get License. A pop-up window displays asking you to which computer should the license be assigned. You can use one of the following options:
    - Option 1: Click Create a New Computer if you want to assign the license to a new computer. You must provide information about the required hardware and license type. For information about the license type, refer to Intel FPGA Software License Types. For information about how to extract information about your computer hardware, refer to Hardware Information Required When You Request a License.
    - Option 2: Click Assign an Existing Computer and search for the computer name/NIC ID that you have created previously in your My Intel account. To view your list of computers, use of the following options:
        - Visit the License Assistant and select Regenerate License by Primary Computer ➤ View all computers and select
        - On the SSLC menu bar, click Computers and License Files and select the desired option.
8. Click Generate. You receive an email with the license attached to your registered email address.
9. Save the license.dat file on your computer (for example, ~/ intelFPGA_pro/LR-xxxxxx_License.dat).

Note:
Before using Questa-Intel FPGA Edition and Questa-Intel FPGA Starter Edition software, you must set an environment variable to point to the location of the license.
Setting Up the Questa-Intel FPGA Starter Edition Software License
After you receive and save the license.dat file on your computer, follow these instructions:

On Windows System

1. Go to This PC, right-click, and select Properties.
2. Click Advanced System Setting.
3. In the Advanced tab, select Environment Variable.
4. Under System variables, create a new variable with the name as LM_LICENSE_FILE and value as <license.dat file path>.
5. Click OK and restart the Questa software.
   Alternatively, open a command prompt and run the following command to set up the LM_LICENSE_FILE environment variable:
   setx LM_LICENSE_FILE <path_to_license_file>;%LM_LICENSE_FILE%
   For example: setx LM_LICENSE_FILE C:\intelFPGA
   \license.dat;%LM_LICENSE_FILE%

On Linux System
Run one of the following commands in a command prompt window:
export LM_LICENSE_FILE=<path to license>:$LM_LICENSE_FILE
setenv LM_LICENSE_FILE “<path_to_license_file>”

Renewing the License
The software license expires 12 months after the date of purchase. To renew an expired license file, revisit the SSLC. You can renew a license only for the version that you purchased.

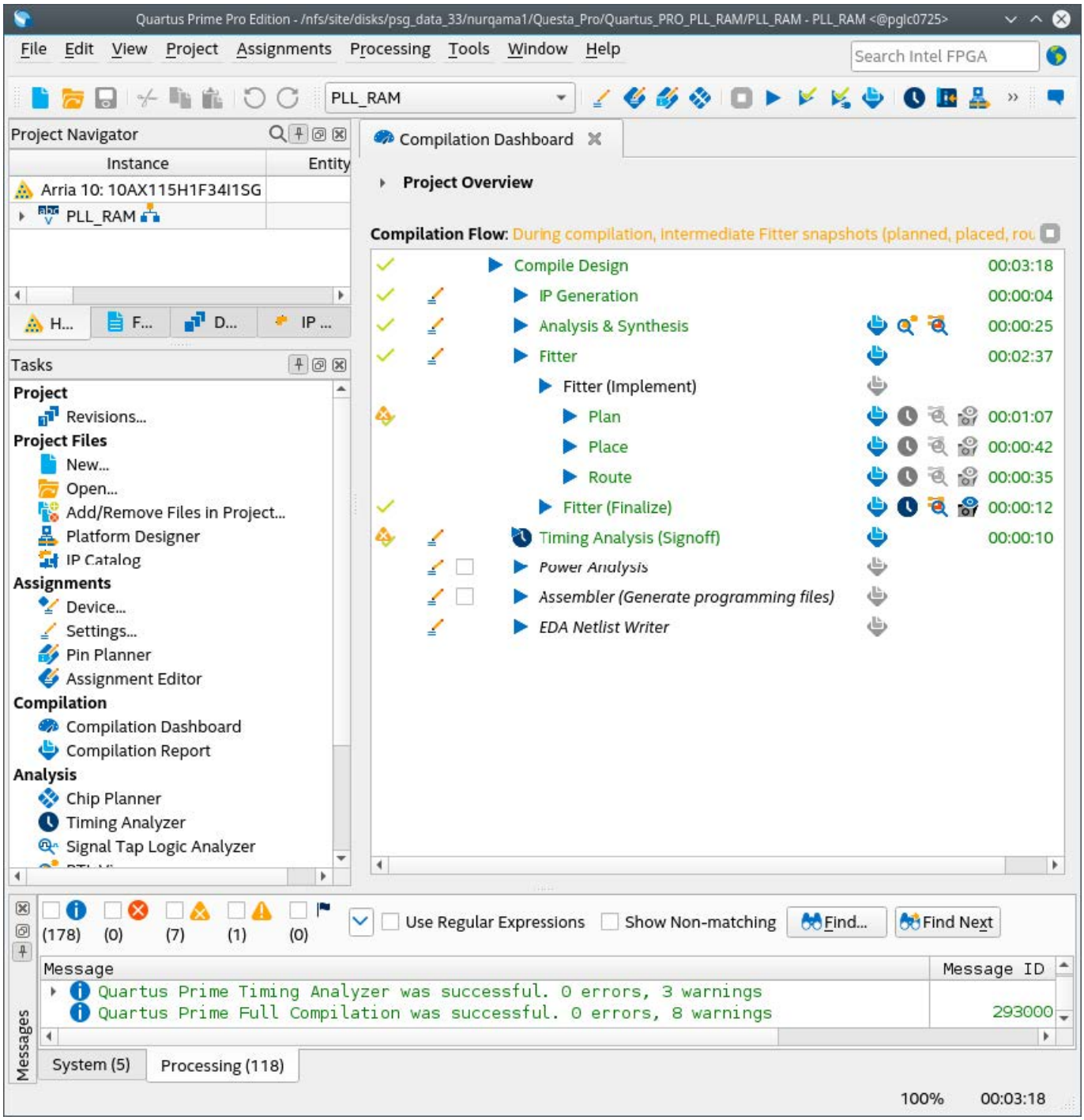
Related Information

- What's New in Questa-Intel FPGA Edition
- Intel FPGA Software Installation and Licensing
- Common Licensing Q & A
- How to Get and Manage License
- Where to Get the License Daemon

Open the Example Design
The PLL_RAM example design includes Intel FPGA IP cores to demonstrate the basic simulation flow. Perform the following steps to open the design example:

1. Download and unzip the Quartus_Pro_PLL_RAM.zip design example.
2. Launch the Intel Quartus Prime Pro Edition software version 21.3.
3. To open the example design project, click File ➤ Open Project, select the pll_ram.qpf project file, and then click OK.

Figure 1. pll_ram Project in the Intel Quartus Prime Pro Edition

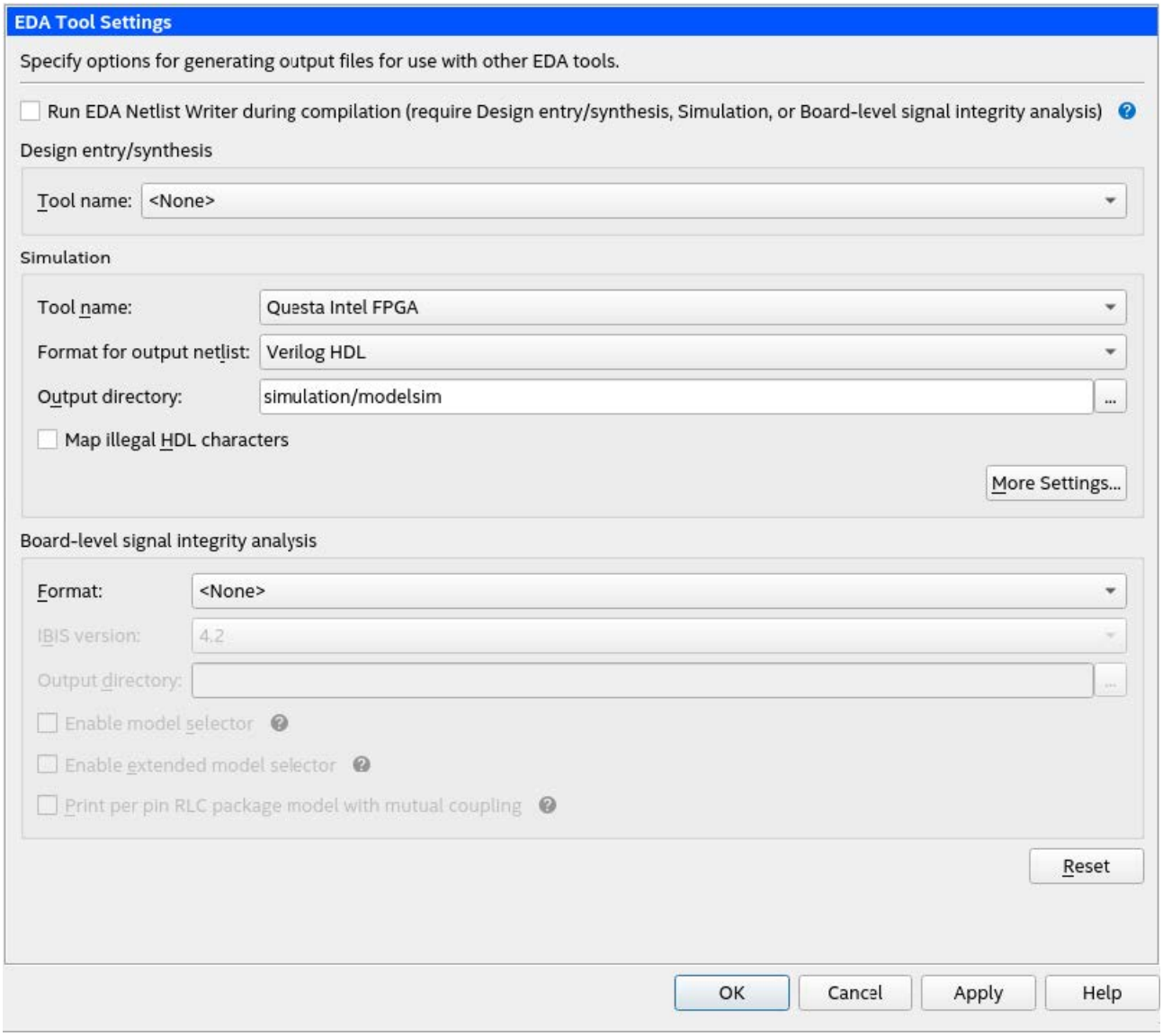


Specify EDA Tool Settings

Specify EDA tool settings to generate simulation files for supported simulators by performing the following steps:

1. In the Intel Quartus Prime software, click Assignments ➤ Settings ➤ EDA Tool Settings.
2. Under Simulation, select Questa Intel FPGA as the Tool name. Retain the default settings for Format for output netlist and Output directory.

Figure 2. EDA Tool Settings

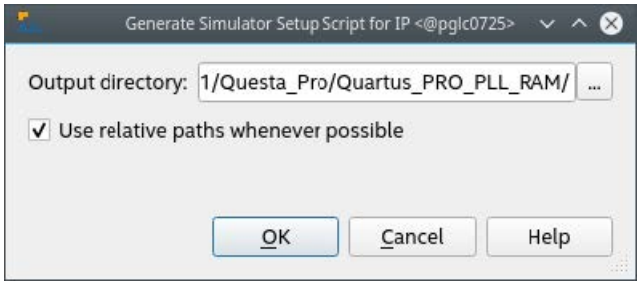


Generate a Simulator Setup Script Template

Simulator setup scripts help you to simulate IP cores in your design. Follow these steps to generate the vendor-specific simulator setup script template for the IP modules in the example design. You can then customize this template for your specific simulation goals.

1. To compile the design, click Processing ➤ Start Compilation. The Messages window indicates when compilation is complete.
2. Click Tools ➤ Generate Simulator Setup Script for IP. Retain the default Output directory and Use relative paths whenever possible setting for the setup script file. The setup script template and two sub-folders mentor/ and comman/ generates in the directory that you specify.

Figure 3. Generate Simulator Setup Scripts IP Dialog Box



Create the Simulation Script

Create the simulation script to enable specific commands that simulate the IP cores in the project.

1. In a text editor, open the /Quartus_PRO_PLL_RAM/mentor/msim_setup.tcl file.
2. Create a new text file with the name mentor_example.do and save it in the /PLL_RAM/mentor/ directory.
3. In the msim_setup.tcl file, copy the section of code enclosed within the TOP-LEVEL TEMPLATE – BEGIN and TOP-LEVEL TEMPLATE – END comments, and then paste this code into the new mentor_example.do file.
4. In the mentor_example.do file, delete single pound (#) characters preceding the following highlighted lines to enable compilation commands:

```
# ----------------------------------------
# # TOP-LEVEL TEMPLATE - BEGIN
# #
# # QSYS_SIMDIR is used in the Quartus-generated IP simulation script to
# # construct paths to the files required to simulate the IP in your Quartus
# # project. By default, the IP script assumes that you are launching the
# # simulator from the IP script location. If launching from another
# # location, set QSYS_SIMDIR to the output directory you specified when you
# # generated the IP script, relative to the directory from which you launch
# # the simulator.
# #
# set QSYS_SIMDIR <script generation output directory>
# #
# # Source the generated IP simulation script.
# source $QSYS_SIMDIR/mentor/msim_setup.tcl
# #
# # Set any compilation options you require (this is unusual).
# set USER_DEFINED_COMPILE_OPTIONS <compilation options>
# set USER_DEFINED_VHDL_COMPILE_OPTIONS <compilation options for VHDL>
# set USER_DEFINED_VERILOG_COMPILE_OPTIONS <compilation options for Verilog>
# #
# # Call command to compile the Quartus EDA simulation library.
# dev_com
# #
# # Call command to compile the Quartus-generated IP simulation files.
# com
# #
# # Add commands to compile all design files and testbench files, including
# # the top level. (These are all the files required for simulation other
# # than the files compiled by the Quartus-generated IP simulation script)
# #
# vlog <compilation options> <design and testbench files>
# #
# # Set the top-level simulation or testbench module/entity name, which is
# # used by the elab command to elaborate the top level.
# #
# set TOP_LEVEL_NAME <simulation top>
# #
# # Set any elaboration options you require.
# set USER_DEFINED_ELAB_OPTIONS <elaboration options>
# #
# # Call command to elaborate your design and testbench.
# elab
# #
# # Run the simulation.
# run -a
# #
# # Report success to the shell.
# exit -code 0
# #
# # TOP-LEVEL TEMPLATE - END
# ----------------------------------------
```

Figure 4. Uncomment Highlighted Simulation Commands in the Script

5. Replace the following lines in the mentor_example.do script:

   Table 1. Specify Values in the mentor_example.do Script

Replace this Line	With this Line
set QSYS_SIMDIR <script generation output directory>	
	set QSYS_SIMDIR ../
continued…	

	Replace this Line	With this Line
	vlog <compilation options> <design and testbench files>	
		vlog -vlog01compat -work work ../PLL_RAM.v vlog -vlog01compat -work work ../UP_COUNTER_IP/UP_COUNTER_IP.v vlog -vlog01compat -work work ../DOWN_COUNTER_IP/DOWN_COUNTER_IP.v vlog -vlog01compat -work work ../ClockPLL/ClockPLL.v vlog -vlog01compat -work work ../RAMhub/RAMhub.v vlog -vlog01compat -work work ../testbench_1.v
	set TOP_LEVEL_NAME <simulation top>	
		set TOP_LEVEL_NAME tb
	elab	
		elab_debug (elab_debug will evaluate vsim -voptargs=+acc which is used to preserve all signals in the waveform)
	run -a	
		add wave * view structure view signals run -all

6. Save the /Quartus_PRO_PLL_RAM/mentor/mentor_example.do file. The following figure shows the

mentor_example.do file after revisions are complete:

Figure 5. Completed Top-Level IP Simulation Setup Script

```
# ----------------------------------------
# # TOP-LEVEL TEMPLATE - BEGIN
# #
# # QSYS_SIMDIR is used in the Quartus-generated IP simulation script to
# # construct paths to the files required to simulate the IP in your Quartus
# # project. By default, the IP script assumes that you are launching the
# # simulator from the IP script location. If launching from another
# # location, set QSYS_SIMDIR to the output directory you specified when you
# # generated the IP script, relative to the directory from which you launch
# # the simulator.
# #
set QSYS_SIMDIR ../
# #
# # Source the generated IP simulation script.
source $QSYS_SIMDIR/mentor/msim_setup.tcl
# #
# # Set any compilation options you require (this is unusual).
# set USER_DEFINED_COMPILE_OPTIONS <compilation options>
# set USER_DEFINED_VHDL_COMPILE_OPTIONS <compilation options for VHDL>
# set USER_DEFINED_VERILOG_COMPILE_OPTIONS <compilation options for Verilog>
# #
# # Call command to compile the Quartus EDA simulation library.
dev_com
# #
# # Call command to compile the Quartus-generated IP simulation files.
com
# #
# # Add commands to compile all design files and testbench files, including
# # the top level. (These are all the files required for simulation other
# # than the files compiled by the Quartus-generated IP simulation script)
# #
vlog -vlog01compat -work work ../PLL_RAM.v
vlog -vlog01compat -work work ../UP_COUNTER_IP/UP_COUNTER_IP.v
vlog -vlog01compat -work work ../DOWN_COUNTER_IP/DOWN_COUNTER_IP.v
vlog -vlog01compat -work work ../ClockPLL/ClockPLL.v
vlog -vlog01compat -work work ../RAMhub/RAMhub.v
vlog -vlog01compat -work work ../testbench_1.v
# #
# # Set the top-level simulation or testbench module/entity name, which is
# # used by the elab command to elaborate the top level.
# #
set TOP_LEVEL_NAME tb
# #
# # Set any elaboration options you require.
# set USER_DEFINED_ELAB_OPTIONS <elaboration options>
# #
# # Call command to elaborate your design and testbench.
elab_debug
# #
# # Run the simulation.
add wave *
view structure
view signals
run -all
# #
# # Report success to the shell.
# #exit -code 0
# #
# # TOP-LEVEL TEMPLATE - END
# ----------------------------------------
```

Compile and Simulate the Design

Run the top-level mentor_example.do script in the Questa*-Intel FPGA Edition software to compile and simulate your design by performing the following steps:

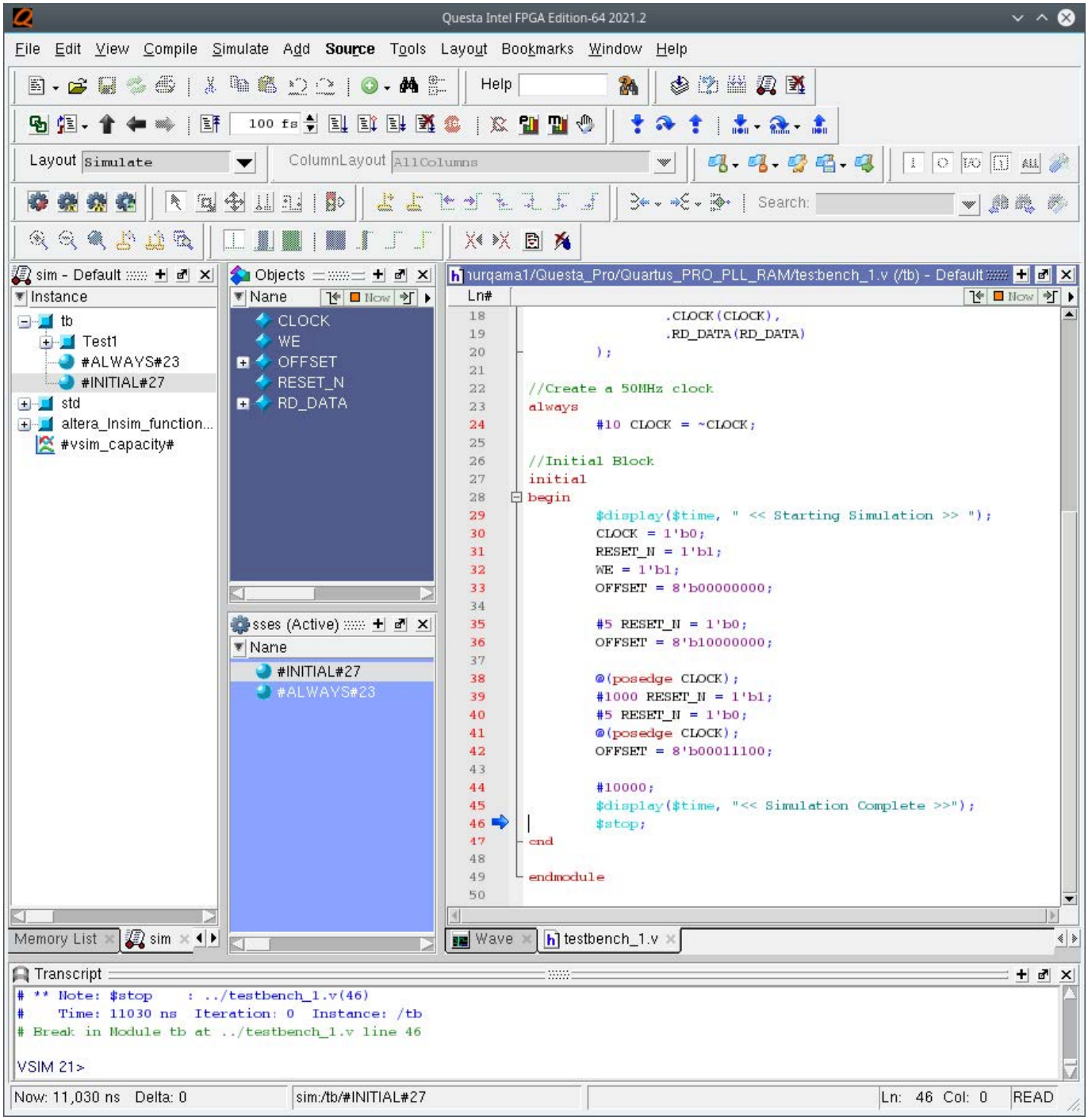
1. Launch the Questa*-Intel FPGA Edition software. The Questa*-Intel FPGA Edition GUI organizes the elements of your simulation onto separate windows and tabs.
2. From the PLL_RAM project directory, open testbench_1.v and mentor/mentor_example.do files.
3. To display the Transcript window, click View ➤ Transcript. Enter commands for Questa*-Intel FPGA Edition

directly in the Transcript window.

4. Type the following command in the Transcript window and then press Enter: do mentor_example.do

The design compiles and simulates according to your specifications in the mentor_example.do script. The following figure shows the Questa*-Intel FPGA Edition simulator:

Figure 6.Questa*-Intel FPGA Edition GUI

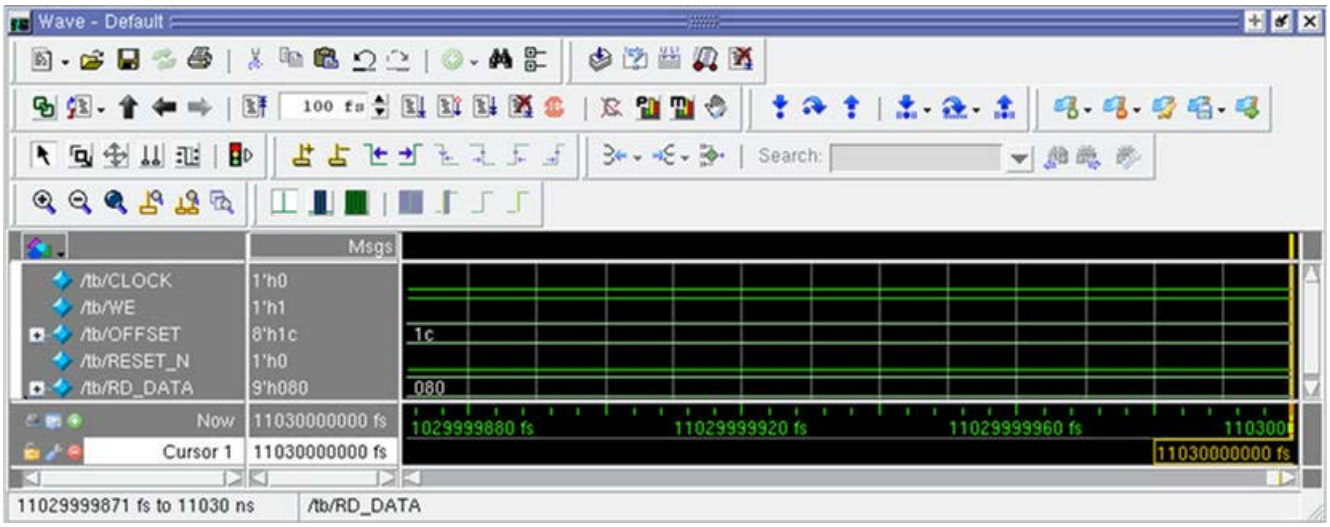


View Signal Waveforms

Follow these steps to view signals in the testbench_1.v simulation waveform:

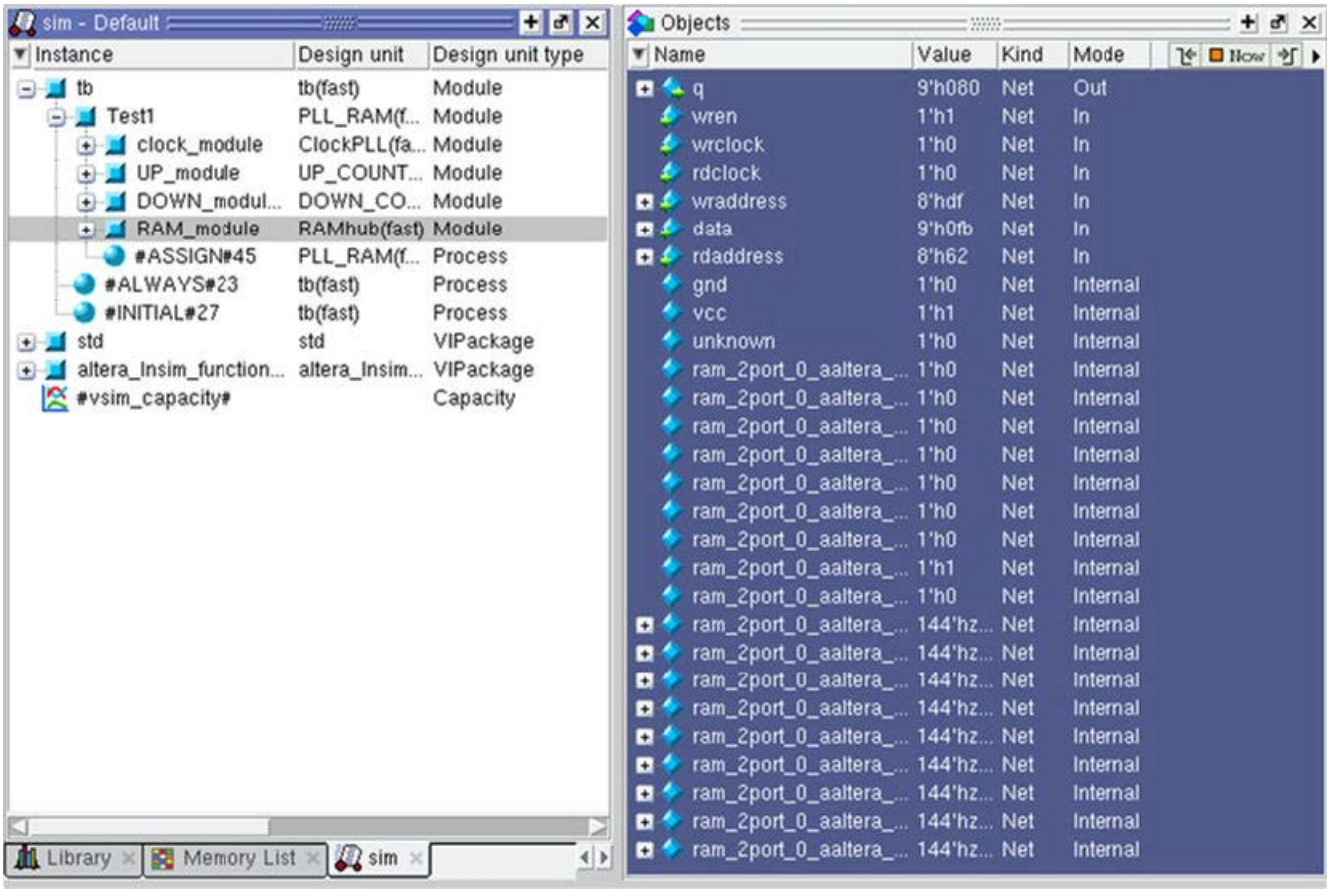
1. Click the Wave window. The simulation waveform ends at 11030 ns, as the testbench specifies. The Wave window lists the CLOCK, WE, OFFSET, RESET_N, and RD_DATA signals.

   Figure 7. Questa-Intel FPGA Edition Wave Window



2. Click the Sim tab to view the signals in the top-level pll_ram.v design. The Sim window synchronizes with the Objects window.
3. Expand the tb folder in the sim tab to view the top-level module signals.
4. Expand the Test1 folder. The Objects window displays UP_module, DOWN_module, PLL_module, and RAM_module signals.
5. In the Sim window, select a module under Test1 to display the module's signals in the Objects window.

Figure 8. Questa-Intel FPGA Edition Sim and Objects Windows



6. View the simulation library files in the Library window.

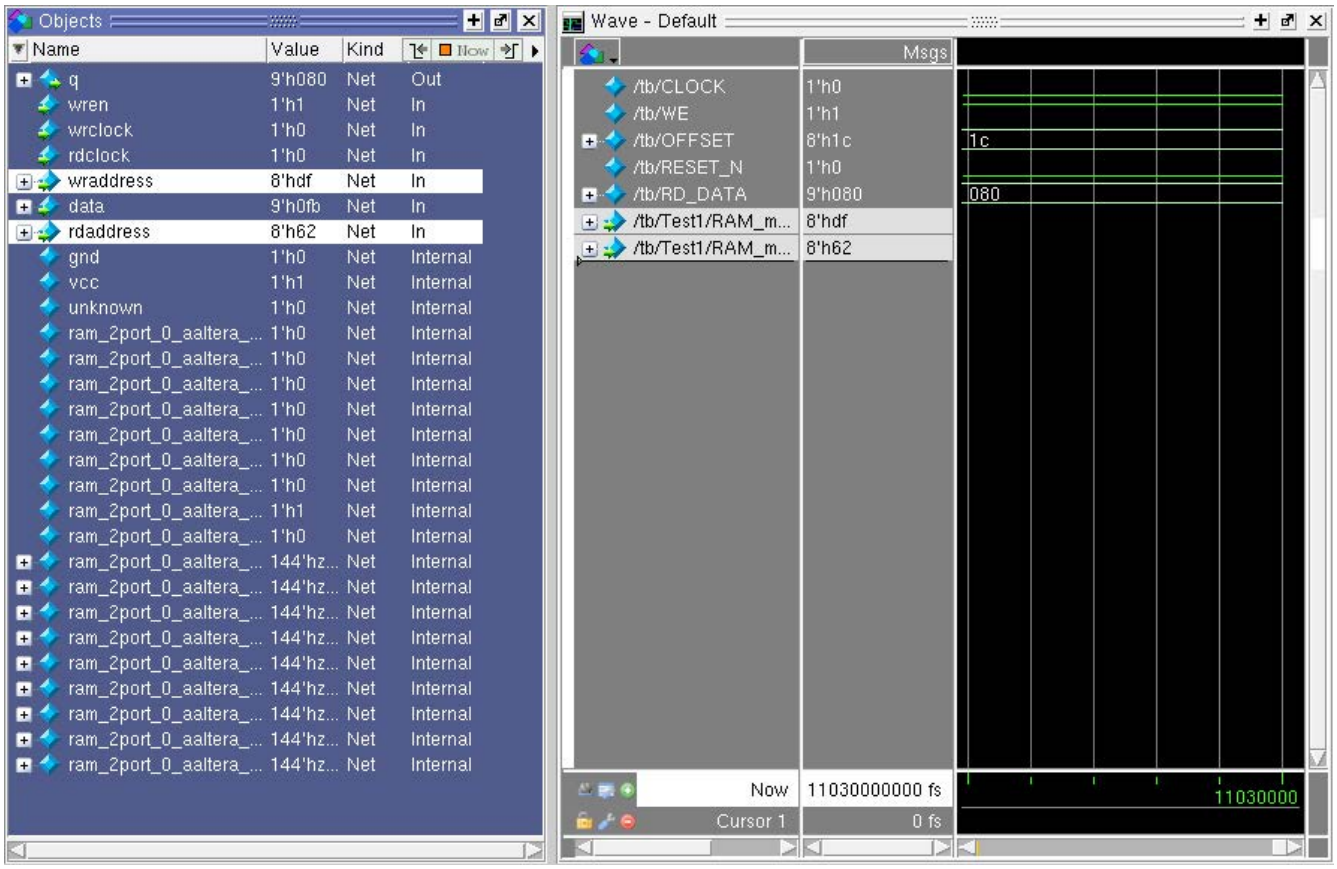
Figure 9. Questa-Intel FPGA Edition Library Window

Add Signals to the Simulation

The CLOCK, WE, OFFSET, RESET_N, and RD_DATA signals automatically appear in the Wave window because the top-level design defines these I/O. In addition, you can optionally add internal signals to the simulation. Perform the following steps to add signals to the simulation:

1. In the Objects window, locate the UP_module, DOWN_module, PLL_module, and RAM_module modules.
2. In the Objects window, select RAM_module to view the module's inputs and outputs.

Figure 10. Add Signals to the Wave Window



3. Right-click rdaddress and then click Add Wave to add the internal signals between the down-counter and dual-port RAM module.
4. Right-click wraddress and then click Add Wave to add the internal signals between the up-counter and dual-port RAM module. Alternatively, you can drag and drop these signals from the Objects window to the Wave window.
5. Click Simulate ➤ Run ➤ Continue to generate waveforms for the new signals you added.

Rerun Simulation

You must rerun the simulation if you make changes to the simulation setup, such as adding signals to the Wave window, or modifying the testbench_1.v file. Follow these steps to rerun simulation:

1. In the Questa-Intel FPGA Edition simulator, click Simulate ➤ Restart.
2. Retain the default options and click OK. These options clear the waveforms and restart the simulation time, while retaining the necessary signals and settings. Alternatively, you can re-run the /PLL_RAM/mentor/mentor_example.do script to re-run simulation at the command line.
3. Click Simulate ➤ Run ➤ Run -all. The testbench_1.v file simulates according to the testbench specifications. To continue simulation, click Simulate ➤ Run ➤ Continue. This command continues the simulation until you click the Stop button.

Known Differences Between the Questa-Intel FPGA Edition and ModelSim* – Intel FPGA Edition

The following table lists major differences between the Questa-Intel FPGA Edition and ModelSim* – Intel FPGA Edition:

Table 2. Known Differences and User Actions

Known Differences		Action for Questa-Intel FPGA Edition
Questa-Intel FPGA Edition	**ModelSim – Intel FPGA Edition**	
The simulator executable path is *not* auto-populated in **Tools ➤ Options ➤ EDA Tool Options**.	The simulator executable path is auto- populated in **Tools ➤ Options ➤ EDA Tool Options**.	Follow the instructions in **Specify EDA Tool Settings** on page 7 to specify the executable path.
If you open an existing project with ModelSim – Intel FPGA Edition EDA tool settings, Intel Quartus Prime software replaces ModelSim – Intel FPGA Edition with Questa-Intel FPGA Edition since ModelSim – Intel FPGA Edition is no longer valid.		No action is required.
The simulator may fail during elaboration with the following message: Error (suppressible): (vopt-14408) Intel FPGA Edition recommended capacity is 5000 non-OEM instances. There are … This error is issued when the design capacity for the simulator is exceeded.	The same message is issued as a warning message.	Suppress this error and continue with simulation. However, the simulation runs 30X slower.
By default, the simulator does not preserve signals for waveform viewing.	always preserves signals for waveform viewing.	Preserve signals explicitly by specifying vsim or vopt options, such as +acc. Refer to **Add Signals to the Simulation** on page 15 for more information.
The windows executable launches the simulator with the vsim.exe file.	The windows executable launches the simulator with the modelsim.exe file.	Launch the simulator with the vsim.exe file.
The Questa-Intel FPGA Starter Edition is free, but it requires a zero-cost license.	The ModelSim – Intel FPGA Starter Edition does not require a license.	Obtain the Questa-Intel FPGA Starter Edition license at no cost from Intel.

Questa*-Intel FPGA Edition Simulation Quick-Start Intel Quartus Prime Pro Edition Revision History

Document Version	Intel Quartus Prime Version	Changes
2022.03.28	21.3	Added the topic *Prerequisites*.
2021.10.04	21.3	Initial release.

accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel 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 Intel. Intel 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.
*Other names and brands may be claimed as the property of others.

Documents / Resources

	intel Quartus Prime Pro Edition Software [pdf] User Guide Quartus Prime Pro Edition, Software, Quartus Prime Pro Edition Software