



Simulation Quick-Start for ModelSim* - Intel® FPGA Edition

Intel® Quartus® Prime Standard Edition

UG-01102
2017.07.15

Last updated for Intel® Quartus® Prime Design Suite: 17.0



[Subscribe](#)

[Send Feedback](#)



Contents

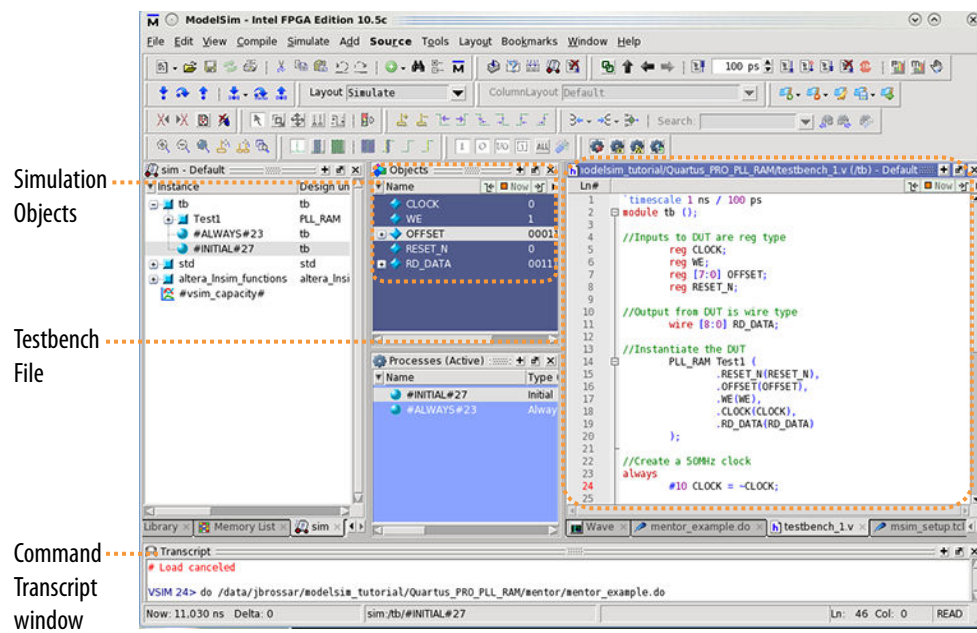
1 Simulation Quick-Start for ModelSim* - Intel® FPGA Edition (Intel® Quartus® Prime Standard Edition).....	3
1.1 Open the Example Design.....	4
1.2 Specify EDA Tool Settings.....	4
1.3 Launch Simulation from the Intel Quartus Prime Software.....	6
1.4 View Signal Waveforms.....	7
1.5 Add Signals to the Simulation.....	8
1.6 Rerun Simulation.....	9
1.7 Modify the Simulation Testbench.....	9
1.8 (Optional) Run Simulation at Command Line.....	10
2 Document Revision History.....	11



1 Simulation Quick-Start for ModelSim* - Intel® FPGA Edition (Intel® Quartus® Prime Standard Edition)

This document demonstrates how to simulate an Intel® Quartus® Prime Standard Edition design in the ModelSim*-Intel FPGA Edition simulator. Design simulation verifies your design before device programming. The Intel Quartus Prime software generates simulation files for supported EDA simulators during design compilation.

Figure 1. ModelSim-Intel FPGA Edition



Design simulation involves generating setup scripts for your simulator, compiling simulation models, running the simulation, and viewing the results. The following steps describe this flow in detail:

1. [Open the Example Design](#) on page 4
2. [Specify EDA Tool Settings](#) on page 4
3. [Launch Simulation from the Intel Quartus Prime Software](#) on page 6
4. [View Signal Waveforms](#) on page 7
5. [Add Signals to the Simulation](#) on page 8
6. [Rerun Simulation](#) on page 9
7. [Modify the Simulation Testbench](#) on page 9
8. (Optional) [Run Simulation at Command Line](#) on page 10



1.1 Open the Example Design

The PLL_RAM example design includes Intel FPGA IP cores to demonstrate the basic simulation flow. Download the example design files and open the project in the Intel Quartus Prime software.

Note: This Quick-Start requires a basic understanding of hardware description language syntax and the Intel Quartus Prime design flow, as the [Intel Quartus Prime Standard Edition Foundation Online Training](#) describes.

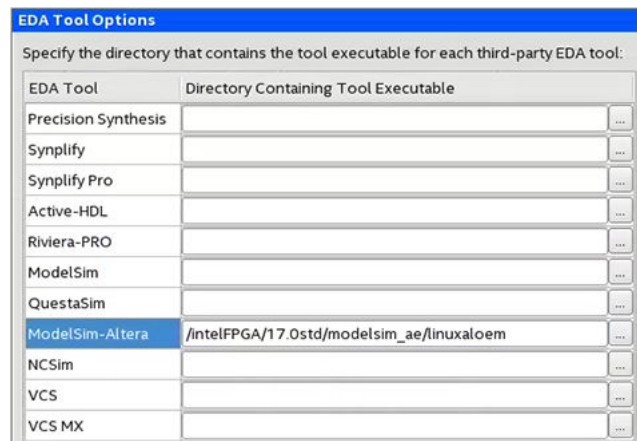
1. Download and unzip the [Quartus_STD_LITE_PLL_RAM.zip](#) design example from the [Altera wiki](#).
2. Launch the Intel Quartus Prime Standard Edition software.
3. To open the example design project, click **File > Open Project**, select the **pll_ram.qpf** project file, and then click **OK**.

1.2 Specify EDA Tool Settings

Specify EDA tool settings to generate simulation files for supported simulators.

1. In the Intel Quartus Prime software, click **Tools > Options > EDA Tool Options**.
2. To specify the location of your simulator for integration with the Intel Quartus Prime software, click **Tools > Options > EDA Tool Options**.

Figure 2. EDA Tool Options



3. In **ModelSim-Altera**, enter the ModelSim - Intel FPGA Edition executable path. Select the appropriate ModelSim - Intel FPGA Edition executable, rather than any other supported ModelSim software.
 - /intelFPGA_lite/<version>/modelsim_ase/win32aloem (Lite)
 - /intelFPGA/<version>/modelsim_ase/win32aloem (Standard)
4. Click **Assignments > Settings > EDA Tool Settings > Simulation**.
5. On the **Simulation** page, specify the following values for the options:



Option	Value
Tool name	ModelSim-Altera
Run gate-level simulation automatically after compilation	Disable checkbox
Format for output netlist	Verilog HDL
Map illegal VHDL characters	Disable checkbox
Enable glitch filtering	Disable checkbox
Generate Value Change Dump (VCD) file script	Disable checkbox

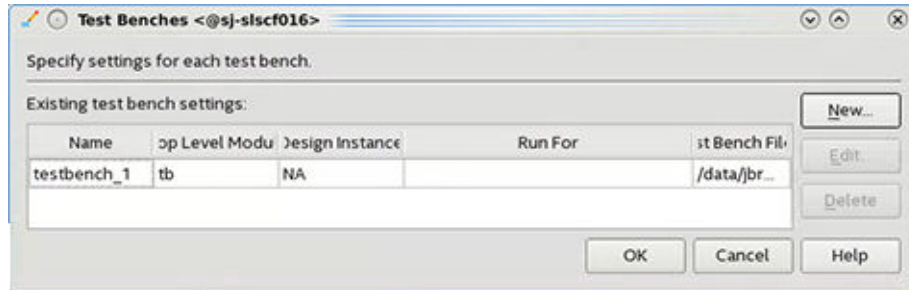
Figure 3. Simulation Options



6. Under **NativeLink settings**, select the **Compile test bench** option, and then click the **Test Benches** button.
7. Click **New**. Specify `testbench_1` as the **Test bench name**, and `tb` as the **Top level module in test bench**.
8. Under **Test bench and simulation files**, enter or select the `testbench_1.v` file, click **Add**, and then click **OK**. The **Test Benches** dialog box displays the properties of the testbenches in your project.



Figure 4. Test Benches Dialog Box



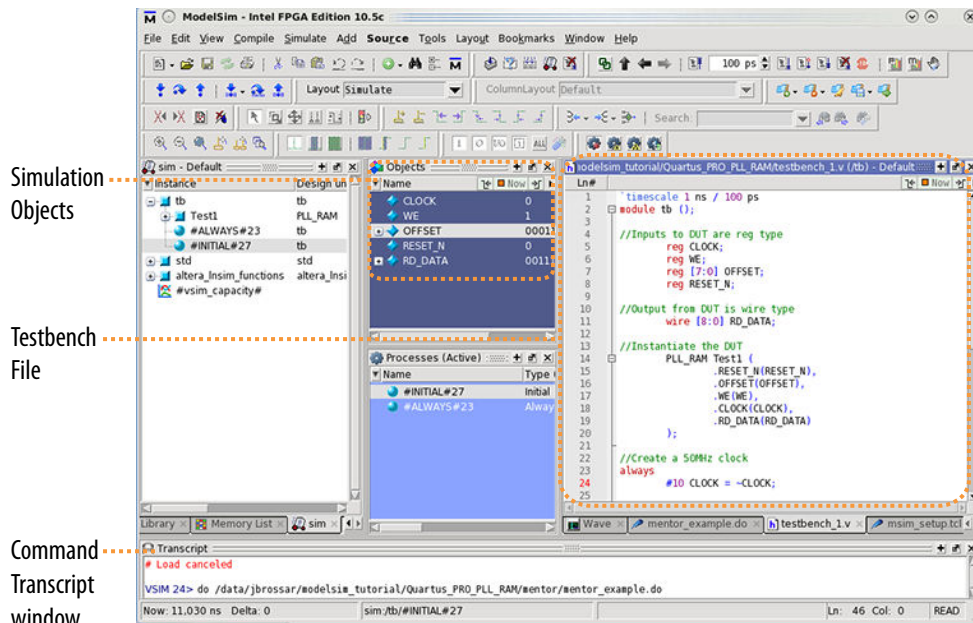
9. In the **Test Benches** dialog box, click **OK**. In the **Settings** dialog box, click **OK**.

1.3 Launch Simulation from the Intel Quartus Prime Software

To generate and run the ModelSim-Intel FPGA Edition automation script from within the Intel Quartus Prime software, follow these steps:

1. To compile the design and generate the .do file, click **Processing > Start Compilation**. The Messages window indicates when compilation is complete.
2. Click **Tools > Run Simulation Tool > RTL Simulation**. The Intel Quartus Prime software launches the ModelSim - Intel FPGA Edition simulator and simulates the testbench_1.v file, according to your specifications in the **Simulation** settings. The ModelSim - Intel FPGA Edition GUI organizes the elements of your simulation in separate windows. The right side of the GUI displays the testbench_1.v file that defines the simulation.

Figure 5. ModelSim - Intel FPGA Edition GUI



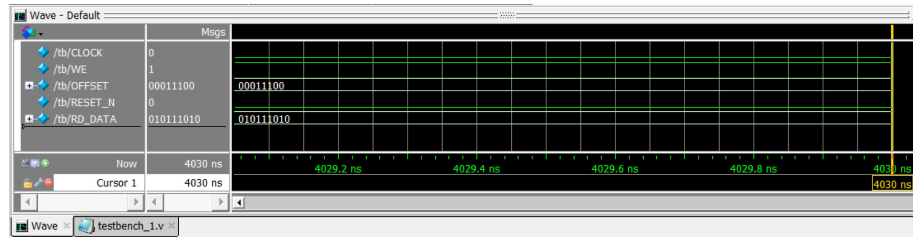


1.4 View Signal Waveforms

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

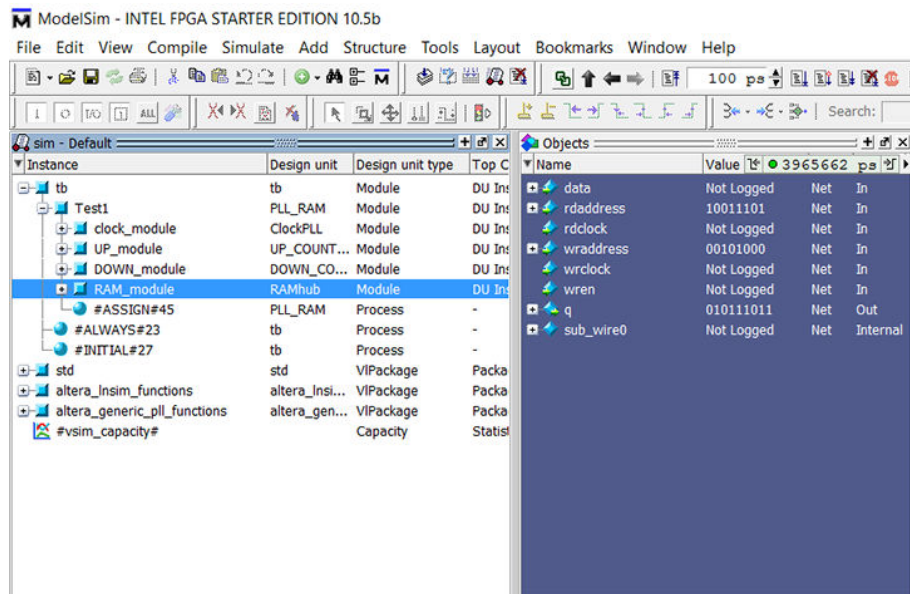
1. Click the **Wave** window. The simulation waveform ends at 4030 ns, as the testbench specifies. The **Wave** window lists the `CLOCK`, `WE`, `OFFSET`, `RESET_N`, and `RD_DATA` signals.

Figure 6. ModelSim - Intel FPGA Edition Wave Window



2. To view the signals in the top-level `p11_ram.v` design, click the **Sim** tab. The **Sim** window synchronizes with the **Objects** window.

Figure 7. ModelSim - Intel FPGA Edition Sim and Objects Windows



3. To view the top-level module signals, expand the `tb` folder in the **Objects** tab. Similarly, expand the `Test1` folder. The **Objects** window displays the `UP_module`, `DOWN_module`, `PLL_module`, and `RAM_module` signals.
4. In the **Sim** window, click a module under `Test1` to display the module's signals in the **Objects** window.
5. View the simulation library files in the **Library** window.



Figure 8. ModelSim - Intel FPGA Edition Library Window



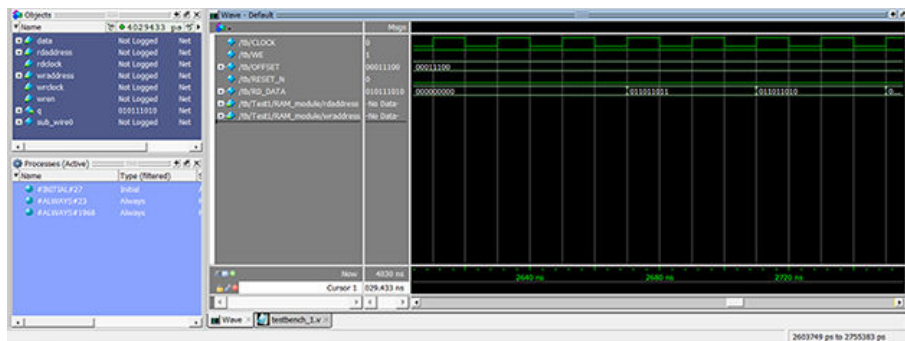
1.5 Add Signals to the Simulation

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.

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**. The module's inputs and outputs display.



Figure 9. Add Signals To Wave Window



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

1.6 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 ModelSim - Intel FPGA Edition simulator, click **Simulate** > **Restart**. Retain the default options and click **OK**. These options clear the waveforms and restart the simulation time, while retaining the necessary signals and settings.
2. 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.

1.7 Modify the Simulation Testbench

The `testbench_1.v` example testbench tests only a specific set of conditions and test cases. You can manually edit the `testbench_1.v` file in the ModelSim - Intel FPGA Edition simulator to test other cases and conditions:

1. Open the `testbench_1.v` file in the ModelSim - Intel FPGA Edition simulator.
2. Right-click in the `testbench_1.v` file to confirm that the file is not set to **Read Only**.
3. Enter and save any additional testbench parameters in the `testbench_1.v` file.
4. To generate the waveforms for a testbench that you modify, click **Simulate** > **Restart**.
5. Click **Simulate** > **Run** > **Run -all**.



1.8 (Optional) Run Simulation at Command Line

Follow these steps to generate a .do file that runs the ModelSim - Intel FPGA Edition simulator from the command line.

1. To open the example design project, click **File > Open Project**, select the **pll_ram.qpf** project file, and then click **OK**. The project opens in the Intel Quartus Prime GUI.
2. Click **Assignments > Settings > EDA Tool Settings > Simulation > More NativeLink Settings**.
3. Enable **Generate third-party EDA tool command scripts without running the EDA tool**, and then click **OK**. In the **Settings** dialog box, click **OK**.
4. To compile the design and generate the .do file, click **Processing > Start Compilation**.
5. Click **Tools > Run Simulation Tool > RTL Simulation**. The Intel Quartus Prime software generates the `PLL_RAM_run_msim_rtl_verilog.do` file that defines the compilation and simulation instructions for the ModelSim - Intel FPGA Edition simulator in the `/simulation/modelsim/` directory in your project.
6. In the ModelSim - Intel FPGA Edition software, open the `PLL_RAM_run_msim_rtl_verilog.do` file. This file contains all the commands and library inclusions the simulation requires.
7. If not already present, add the following commands at the end of the .do file. These commands set up the waveform interface and run the `testbench_1.v` testbench.

```
add wave *
view structure
view signals
run -all
```

8. In the Transcript window (**View > Transcript**), type the following command and press Enter:

```
do PLL_RAM_run_msim_rtl_verilog.do
```

The ModelSim - Intel FPGA Edition simulator runs compilation and simulation, as the .do file specifies.



2 Document Revision History

This document has the following revision history.

Table 1. Document Revision History

Date	Changes
2017.07.15	Initial release.