

### **ModelSim\* - Intel<sup>®</sup> FPGA Edition Simulation Quick-Start**

### Intel<sup>®</sup> Quartus<sup>®</sup> Prime Standard Edition

Updated for Intel<sup>®</sup> Quartus<sup>®</sup> Prime Design Suite: **18.0** 



ID: 683248 Version: 2019.12.30

UG-01102

# intel.

### **Contents**

1. ModelSim* - Intel <sup>®</sup> FPGA Edition Simulation Quick-Start (Intel <sup>®</sup> Quartus <sup>®</sup> Prime Standard Edition)	3
1.1. Open the Example Design	
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	9
1.6. Rerun Simulation	10
1.7. Modify the Simulation Testbench	10
1.8. (Optional) Run Simulation at Command Line	11
2. ModelSim - Intel FPGA Edition Simulation Quick-Start Revision History	12

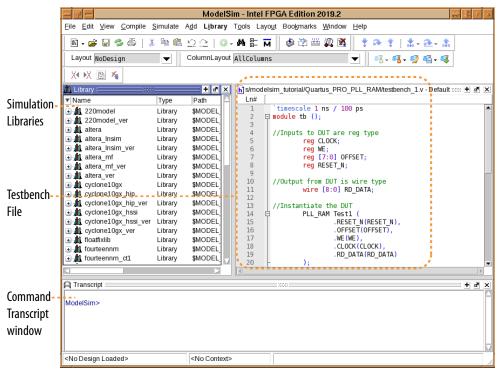


# intel

### **1.** ModelSim\* - Intel<sup>®</sup> FPGA Edition Simulation Quick-Start (Intel<sup>®</sup> Quartus<sup>®</sup> Prime Standard Edition)

This document demonstrates how to simulate an Intel<sup>®</sup> Quartus<sup>®</sup> 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 9

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in 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.



1. ModelSim\* - Intel® FPGA Edition Simulation Quick-Start (Intel® Quartus® Prime Standard Edition) 683248 | 2019.12.30

- 6. Rerun Simulation on page 10
- 7. Modify the Simulation Testbench on page 10
- 8. (Optional) Run Simulation at Command Line on page 11

### 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.
- 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

DA Tool Options				
Specify the directory	that contains the tool executable for each third-party	EDA tool:		
EDA Tool	Directory Containing Tool Executable			
Precision Synthesis				
Synplify				
Synplify Pro				
Active-HDL				
Riviera-PRO				
ModelSim				
QuestaSim		)		
ModelSim-Altera	/intelFPGA/17.0std/modelsim_ae/linuxaloem			
NCSim		[]		
vcs				
VCS MX				

- 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**

<u>r</u> ool name:	ModelSim-Alt	era				1
Ru <u>n</u> gate	-level simulatio	on automatically a	fter compila	ation		
EDA Netlis	t Writer setting	s				
Eormat for	r output netlist	Verilog HDL	\$	Time <u>s</u> cale:	1 ps	\$
Output dir	ectory: simula	tion/modelsim				
Map ille	egal <u>H</u> DL charac	ters	<u></u>	nable glitch filte	ring	
Options	for Power Estin	nation				
Gene						
	rate Value Cha	nge Dump (VCD) f	le script	Script Settings		
		nge Dump (VCD) f	ile script	Script Settings		
	rate Value Cha In instance nam		ile script	script Settings		
Desig	n instance nam	e:	ile script	cript Settings		
Desig		e:	ile script	script Settings	<u> </u>	
Desig	in instance nam Netlist Writer Se	e:	ile script	script Settings	<u> </u>	
Desig	in instance nam Netlist Writer Se	e:	le script	script Settings	<u>.</u>	
Desig More EDA N NativeLink O N <u>o</u> ne	in instance nam Netlist Writer Se	e:	le script	script Settings	¢	est <u>B</u> enches
Desig More EDA N NativeLink O N <u>o</u> ne O <u>C</u> ompil	n instance nam Netlist Writer Se settings Ne test bench: (	e: ettings) testbench_1	le script	script Settings	¢.	est Benches
Desig More EDA N NativeLink O N <u>o</u> ne O <u>C</u> ompil	in instance nam Netlist Writer Se settings	e: ettings) testbench_1	le script	script Settings	¢.	est <u>B</u> enches
More EDA N NativeLink O N <u>o</u> ne O Compil	n instance nam Netlist Writer Se settings Ne test bench: (	e:	le script	script Settings	¢]	est <u>B</u> enches

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

pecify setting	s for each test be	nch.			
xisting test be	ench settings:			[	<u>N</u> ew
Name	op Level Modu	Design Instance	Run For	st Bench File	Edit.
testbench_1	tb	NA		/data/jbr	- Frank
					Delete
			ОК	Cancel	Help

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**

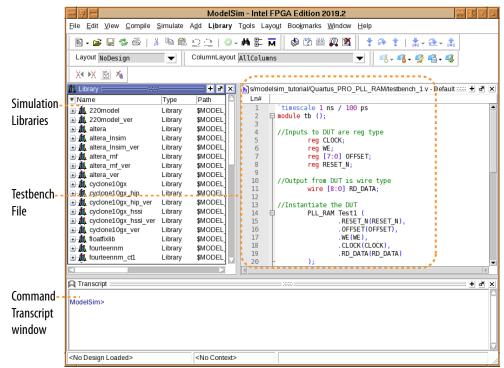
To generate and run the ModelSim-Intel FPGA Edition automation script from within the Intel Quartus Prime software, follow these steps:

- 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.



## intel

### Figure 5. ModelSim - Intel FPGA Edition GUI

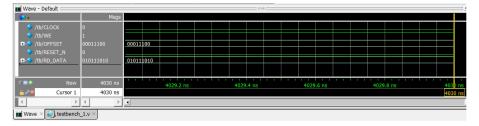


### **1.4. View Signal Waveforms**

Follow these steps to view signals in the testbench\_1.v simulation waveform:

 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 pll\_ram.v design, click the **Sim** tab. The **Sim** window synchronizes with the **Objects** window.





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

ModelSim - INTEL FPGA STAR	TER EDITION 10.5b
----------------------------	-------------------

	2 <u>0</u>   <b>0 - M</b>	▣- M  ] ♥ ☑	iiii 🛺 🛙	Ă   <u>6</u> 1 4 + →   II	100 ps 🔶 🖹 🖹	E# 187 🤐
I O T/O 🗊 AUL 🔗 🛛 🗙 🕅	🖹 🐐 🖹 🕅	<b>4</b> 11 11	0	******	Γ    <b>3• • →</b> € • <b>≩•</b>   Si	earch:
sim - Default ====================================			+ @ ×	斜 Objects		- + @ >
Instance	Design unit	Design unit type	Top C	▼ Name	Value 1 • 396566	2 ps 1/
– <mark>,</mark> ≢ tb	tb	Module	DU In:	🗉 🥠 data	Not Logged Net	
🖕 🗾 Test1	PLL_RAM	Module	DU Ins	🗉 🥏 rdaddress	10011101 Net	In
🔄 🗾 clock_module	ClockPLL	Module	DU In:	rdclock	Not Logged Net	In
🔄 🖬 UP_module	UP_COUNT	Module	DU In:	🗉 🤣 wraddress	00101000 Net	In
DOWN_module	DOWN_CO	Module	DU In:	🤣 wrclock	Not Logged Net	In
RAM_module	RAMhub	Module	DU Ins	🧳 wren	Not Logged Net	
ASSIGN#45	PLL_RAM	Process	-	🗷 🔶 q	010111011 Net	Out
	tb	Process	*	🗉 🗇 sub_wire0	Not Logged Net	Internal
	tb	Process	×			
- 🗾 std	std	VIPackage	Packa			
altera_Insim_functions	altera_Insi	VIPackage	Packa			
altera_generic_pll_functions	altera_gen	VIPackage	Packa			
😤 #vsim_capacity#		Capacity	Statis			

- 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.



## intel

🛔 Library 👘		ni 100 c
• Name	Type	Path
💽 🎎 work	Library	./libraries/work/
ClockPLL	Library	/libraries/ClockPLL/
DOWN_COUNTER_IP	Library	./libraries/DOWN_COUNTER_IP/
🗈 🎎 RAMhub	Library	./libraries/RAMhub/
. A UP_COUNTER_IP	Library	/libraries/UP_COUNTER_IP/
altera_iopil_170	Library	/libraries/altera_iopII_170/
Ipm_counter_170	Library	./libraries/lpm_counter_170/
. # ram_2port_170	Library	./libraries/ram_2port_170/
🗈 🎎 work_lib	Library	./libraries/work/
a libraries (empty)	Library	/data/dnielsen/17.0/PLL_RAM/mentor/
<ul> <li> <u>iii</u> 220model         </li> </ul>	Library	\$MODEL_TECH//altera/vhdl/220model
3 \$ 220model_ver	Library	SMODEL_TECH//altera/verilog/220m
🗉 🎎 altera	Library	\$MODEL_TECH//altera/vhdl/altera
🗉 🎎 altera Insim	Library	\$MODEL_TECH//altera/vhdl/altera_In
altera_Insim_ver	Library	\$MODEL_TECH//altera/verilog/altera
🗉 🎎 altera_mf	Library	SMODEL_TECH//altera/vhdl/altera_mf
Altera_mf_ver	Library	SMODEL_TECH//altera/verilog/altera
Altera_ver	Library	SMODEL_TECH//altera/verilog/altera
<ul> <li>A cyclone10gx</li> </ul>	Library	SMODEL_TECH//altera/vhdl/cyclone1
st cyclone10gx_ver	Library	SMODEL_TECH//altera/verilog/cyclon
🗈 🎎 fourteennm	Library	\$MODEL_TECH//altera/vhdl/fourteen
. A fourteennm_ct1	Library	SMODEL_TECH//altera/vhdl/fourteen
fourteennm_ct1_ver	Library	SMODEL_TECH//altera/verilog/fourte
. A fourteennm_ver	Library	SMODEL_TECH//altera/verilog/fourte
• A lom	Library	SMODEL_TECH//altera/vhdl/220model
Ipm_ver	Library	\$MODEL_TECH//altera/verilog/220m
🗉 🎎 sgate	Library	\$MODEL_TECH/_/altera/vhdl/sgate
sqate_ver	Library	\$MODEL_TECH//altera/verilog/sgate
🗉 🎎 sv_std	Library	\$MODEL_TECH//sv_std
🗉 🎎 twentynm	Library	\$MODEL_TECH//altera/vhdl/twentynm
🗉 🎎 twentynm_hip	Library	\$MODEL_TECH//altera/vhdl/twentyn
• A twentynm_hip_ver	Library	SMODEL_TECH//altera/verilog/twent
🗉 🎎 twentynm_hssi	Library	\$MODEL_TECH//altera/vhdl/twentyn
🗈 🎎 twentynm_hssi_ver	Library	\$MODEL_TECH//altera/verilog/twent
twentynm_ver	Library	\$MODEL_TECH//altera/verilog/twent
💽 🎎 vital2000	Library	\$MODEL_TECH//vital2000
🗉 🎎 ieee	Library	\$MODEL_TECH//ieee
🗉 🎎 modelsim_lib	Library	\$MODEL_TECH//modelsim_lib
🗉 🎎 std	Library	\$MODEL_TECH//std
🗉 🏦 std_developerskit	Library	SMODEL_TECH//std_developerskit
synopsys	Library	\$MODEL_TECH//synopsys
• A verilog	Library	\$MODEL_TECH//verilog

### Figure 8. ModelSim - Intel FPGA Edition Library Window

de a base

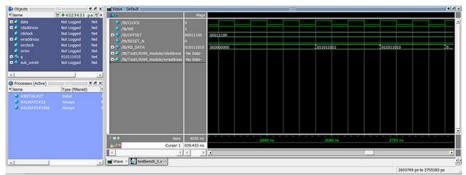
### 1.5. Add Signals to the Simulation

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.
- In the **Objects** window, select **RAM\_module**. The module's inputs and outputs display.



**Add Signals To Wave Window** Figure 9.



- 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.
- To generate the waveforms for a testbench that you modify, click Simulate ➤ Restart.
- 5. Click Simulate ➤ Run ➤ Run -all.



1. ModelSim\* - Intel<sup>®</sup> FPGA Edition Simulation Quick-Start (Intel<sup>®</sup> Quartus<sup>®</sup> Prime Standard Edition) 683248 | 2019.12.30



### **1.8. (Optional) Run Simulation at Command Line**

Follow these steps to generate a  $\,.\,{\rm do}$  file that runs the ModelSim - Intel FPGA Edition simulator from the command line.

- 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. 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. ModelSim - Intel FPGA Edition Simulation Quick-Start** Revision History

Document Version	Intel Quartus Prime Version	Changes
2019.12.30	18.0	Updated link to design example files.
2018.09.25	18.0	Corrected syntax errors in mentor_example.do Script.
2018.05.07	18.0	Removed unnecessary step from <i>Run Simulation at Command Line</i> procedure.
2017.07.15	17.1	Initial release.

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in 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.