Guide to Performing Simulation using Xilinx ISE 13.x and ModelSim 10.x

Lalith Narasimhan
Department of Electrical Engineering
Western Michigan University
l3narasi@wmich.edu

January 10, 2012

This document details the steps required to perform behavioral and post-route simulations using the Xilinx Integrated Software Environment (ISE) and Mentor Graphics ModelSim simulator. In the past, Xilinx bundled the ISE with a licensed edition of ModelSim, resulting in little or no issues between the two softwares. However, in recent years it appears as if the two companies are no longer on level terms. As a result, Mentor Graphics does not provide Xilinx with licenses for ModelSim and Xilinx for its part does not provide pre-compiled device libraries for ModelSim. This tutorial will show you how to resolve compatibility issues, so that you can continue to use the features that these softwares provide.

As with any software tutorial, there is always more than one way of doing things. If you come across better solutions or typos or have any suggestions for improving this document please email the author or contact Dr. Grantner.

1 Pre-requisite

Before reading any further you should have the following softwares installed on your computer

- Xilinx Design Suite 13.x or Xilinx WebPACK 13.x
- Modelsim SE 10.x or ModelSim PE Student Edition 10.x

If you have other versions installed we suggest you to first run simulations on a sample project. If you run into software issues then you can proceed to do modifications descibed in this document.

2 Simulation Libraries

As mentioned earlier, the Xilinx suite no longer comes with pre-compiled device libraries for ModelSim. These libraries contain device specific information like timing parameters required by ModelSim for performing simulation. Xilinx provides two methods for compiling device libraries:

- Simulation Library Compilation Wizard
- Compile HDL Libraries through Project Navigator

2.1 Setting up Environment Variables

Before we discuss the individual methods, let us first understand how ModelSim accesses these libraries. A list of the compiled libraries and their locations is stored in *modelsim.ini* file found in the installation directory. When ModelSim starts a simulation it searches for this *ini* file; first, in the local project folder, then in the installation directory of ModelSim. If a particular library is not found, the simulation stops with an error. Both the methods discussed above cannot access the *modelsim.ini* file directly, so it is important that we setup environment variables that point to the location of this file. Setting up environment variables requires administrative privileges on your computer. If you do not have these privileges skip to the next section.

1. The first step is to change the write permissions for the *modelsim.ini* file. Browse to the ModelSim install directory. In our setup this is *c:\modeltech64_10.0c* (this will be different for other versions). Right-click on *modelsim.ini*, select Properties and uncheck Read-only.
2. Make a backup of the *modelsim.ini* file for safety reasons.
3. The easiest way to view or modify the environment variables is by right-clicking on My Computer→Properties, click Advanced system settings→Environment Variables.
4. Under System variables, click New, enter *modelsim* for the variable name. For the Variable value enter*c:\modeltech64_10.0c\modelsim.ini*.
5. Similarly, add another variable named *Path* with the value *c:\modeltech64_10.0c\win64*. For our setup, *win64* is where the executable (*modelsim.exe*) is located. If a variable called path already exists, make sure you append to the existing values. Note that multiple values can be separated by semi-colon.
6. Click OK to save these settings.

2.2 Simulation Library Compilation Wizard

This method is highly recommended because library compilation can be performed for more than one device family and/or language.

1. Open the Simulation Library Compilation Wizard. This can be accessed from Start Menu→Programs→Xilinx ISE Design Suite 13.x→ISE Design Tools→32-bit Tools.

   Note: If you are using 64-bit version of ModelSim use the Simulation Library Compilation Wizard from 64-bit Tools. ModelSim PE Student Version 10.x is only available in the 32-bit version.
2. The Select Simulator window opens up. Select the appropriate simulator (For Student Edition select ModelSim PE), enter *c:\modeltech64_10.0c\win64* for executable location, *compxlib.cfg* for Compxlib Configuration File and *compxlib.log* for Compxlib Log File.

3. Next select the HDL used for simulation. If you are unsure select Both VHDL and Verilog. However, this will increase the compilation time and the disk space required.

4. Then select all the device families that you will be working with. Again the more number of devices, more the compilation time and the disk space required. Remember that you can always run the compilation wizard at a later time for additional devices.

5. The next window is for Selecting libraries for Functional and Timing Simulation. Different libraries are required for different types of simulation (behavioral, post-route, etc.). We suggest that you select All Libraries as the default option. Interested users can refer to Chapter 6 of the Xilinx Synthesis and Simulation Design Guide for additional information.

6. Finally the window for Output directory for compiled libraries is shown. We suggest to leave the default values that Xilinx picks. Then select Launch Compile Process.

7. Be patient as the compilation can take a long time depending on the options that you have chosen.

8. The compile process may have contain a lot of warnings but should be error-free. We have not explored the reasons behind these warnings, but they do not appear to affect the simulation of any of our designs.

9. Once the process is completed, open *c:\modeltech64_10.0c\modelsim.ini* and verify if there are libraries pointing to the output directory entered in step 6. This will happen only if you have set the environment variables.

Library compilation is now complete. If you have not set the environment variables then the wizard creates a *modelsim.ini* in the output directory entered in step 6. By default this location is *c:\Xilinx\13.x\ISE_DS\ISE*. Open this file and verify that it contains the location of the libraries that were just compiled. This file should be copied into every project you create.

2.3 Compile HDL Libraries through Project Navigator

In this method the libraries are compiled after creating a project through the Xilinx ISE Project Navigator. As you can see the major disadvantage of this method is that the libraries can be compiled for only one device at a time. The libraries are however compiled to the same output directory as before. If the environment variables were set up as discussed before, Xilinx modifies the *modelsim.ini* file in the ModelSim installation directory, otherwise Xilinx creates a *modelsim.ini* file in the current project folder. This file should be copied into every project you create.

1. Create a New project in Xilinx ISE Project Navigator.

2. In the View pane, select Implementation radio button, then select the project device entry (for example, xcv50-6bg256).

3. In the Processes panel, expand Design Utilities. Right-click on Compile HDL Simulation Libraries, and select Properties.

4. In the Simulation Library Compiler Properties dialog box, select the target simulator and the libraries you wish to compile.

5. In the simulator path enter *c:\modeltech64_10.0c\win64*. Then click OK

6. To Compile HDL Simulation Libraries, double-click Compile HDL Simulation Libraries in the Processes panel.

The compilation process should now start. The compilation should take less time as compared to the other method. Again the compile process may have quite a lot of warnings but should have no errors.

3 Running the Simulation

3.1 ModelSim SE 10.x

If you are using ModelSim SE version which is the one installed in the lab, running the simulation is straightforward. In the View pane, select Simulation, choose the type of simulation you want to run from the drop-down list and finally run ModelSim from the Processes panel. Xilinx automatically generates the files needed by ModelSim for compiliation and simulation. Once ModelSim loads, simply go to File→Load, to load the **.do* file.

3.2 ModelSim PE Student Edition 10.x

If you are using the PE Student Edition, the steps described above fails. The reason for this failure is quite trivial but one that Xilinx has ignored or failed to fix in ISE 13.x. Essentially when you run ModelSim from the Processes panel, Xilinx queries for the ModelSim *version* string and checks if this is compatible with ISE. With the PE Student Edition, Xilinx uses *mti_pe edition* as the string instead of *mti_pe*. Remember that when we compiled, we used ModelSim PE because ModelSim PE Student Edition wasn't available. This mismatch means that ModelSim cannot be invoked from with ISE. However a workaround has been found and will be discussed now.

3.2.1 Behavioral Simulation

1. Launch ModelSim from your operating system not from ISE. If you set up the *modelsim.ini* environment variable, ModelSim loads with a list of the compiled libraries.

2. Go to File→Change Directory. Browse to and select your Project folder. Click OK. If you haven't set the *modelsim.ini* environment variable, ModelSim reads the *.ini* file in your project folder and loads a list of the compiled libraries.

3. Go to Compile→Compile, select the design file you wish to run the simulation for and click Compile. If there is a pop-up asking you permission to create *work* library, click Yes. If the Transcript window shows that the compilation is done, click Done.

4. Go to Simulate→Start Simulation. In the Design tab, expand work and select the appropriate design unit (behavioral). Click OK.

5. When the design has been loaded you should see the VSIM>prompt. Now go to File→Load, and open your **.do* file. Behavioral Simulation is now complete.

3.3 Post-route Simulation

To run post-route simulation we first need to implement our design. Before we do this we need to change the implementation properties. In the View pane, select the Implementation radio button, then select your top level source file. In the Processes panel, right-click on Implement Design and select Process Properties. Under Place & Route Properties, check Generate Post-Place & Route Simulation Model. Click OK to save these settings. Now to implement the design, in the Processes panel double-click on Implement Design. Once Implementation is done the files necessary for doing post-route simulation should be created.

1. Launch ModelSim from your operating system not from ISE. If you set up the *modelsim.ini* environment variable, after ModelSim loads you should see a list of the compiled libraries.

2. Go to File→Change Directory. Browse to and select your Project folder. Click OK. If you haven't set the *modelsim.ini* environment variable, ModelSim reads the *.ini* file in your project folder and loads a list of the compiled libraries.

3. Go to Compile→Compile. Browse to *<project_folder>**netgen**par*. Here you should see the *timesim* version of your design file. Select this file and click Compile. If the Transcript window shows that the compilation is done, click Done.

4. Go to Simulate→Start Simulation. In the Design tab, expand work and select the appropriate design unit (structural). In the SDF tab, click Add and browse to *<project_folder>**netgen**par*. You should see a *timesim.sdf* version of your design file. Open this file. Click OK and OK again to start the simulation.

5. When the design has been loaded you should see the VSIM>prompt. Now go to File→Load, and open your **.do* file. Post-Place & Route Simulation is now complete.