

Guide to Performing Simulation in Xilinx 12.3 using ModelSim SE 6.6d

AUTHOR: LALITH NARASIMHAN
EMAIL: l3narasi@wmich.edu
DEPARTMENT OF ELECTRICAL ENGINEERING
WESTERN MICHIGAN UNIVERSITY
JANUARY 31, 2011

The Xilinx ISE Design Suite together with ModelSim Simulator are the primary tools used for synthesis and analysis of HDL designs here at Western Michigan University. Compatibility between these tools has often been excellent meaning that things worked out of the box with little or no effort. However with recent versions, users are required to perform certain modifications to get things working as desired. The purpose of this document is serve as a guide to setup up Xilinx and ModelSim softwares for the purpose of simulation. Every effort has been made to document all the difficulties experienced by us in getting both these softwares working. This document is not comprehensive, but should give you enough information to perform both behavioral and post-place & route simulations. As with any other guide there is always more than one way of doing things. If you come across better solutions, typos or have any suggestions for improving this manual please email the author or contact Dr. Grantner.

Throughout this document we will be using Xilinx Design Suite 12.3 and ModelSim SE 6.6d. As of this writing Xilinx has been updated to version 12.4 and ModelSim has come out with version 10.0. We did however experience the same problems with the newer versions as we did with the older. Other versions may or may not have the same problems, so we suggest you read the document in entirety before proceeding to do modifications.

1 Installation

If you have already installed both Xilinx and ModelSim skip to the next section. If you haven't already installed Xilinx Design Suite 12.3 and ModelSim SE 6.6d on your computer, install them in that order with default set of options. By default options we mean default destination directories and tools. If you are installing only WebPack and student versions of the softwares, some of the tools described in this document may not be available.

2 Simulation Libraries

Recent versions of the Xilinx suite do not come with pre-compiled simulation libraries for the ModelSim SE,PE,DE versions. Libraries are available only for ModelSim XE version and the Xilinx ISim simulator. ModelSim requires these libraries for simulation, hence libraries must be compiled before performing simulation. There are two methods for compiling the libraries:

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

2.1 Setup Environment Variables

Before we discuss these methods along with their pros and cons, it is important to understand how ModelSim accesses these libraries. A list of the compiled

libraries and their locations is stored in “*modelsim.ini*” file found in the ModelSim 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 then the simulation stops and ModelSim issues an error. Both the methods discussed above cannot access the “*modelsim.ini*” file directly, so it is important that we setup environment variables that points 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 (c:\modeltech.6.6d by default), right-click on “*modelsim.ini*”, select Properties and uncheck Read-only.
2. The easiest way to view or modify the environment variables is by selecting System from the Control Panel, selecting Advanced system settings, and clicking Environment Variables.
3. Click on New and enter **modelsim** for the variable name, **c:\modeltech.6.6d\modelsim.ini** for value. If you have installed ModelSim in a different location, then change the value accordingly.
4. In the similar manner, add another variable named **path** with value the **c:\modeltech.6.6d\win32**. If a variable called path already exists then make sure you append the above value to the existing. Note: Multiple values can be separated by semi-colon.
5. Click OK to save these settings.

2.2 Simulation Library Compilation Wizard

If you were able to setup the environment variables then we recommended using this method. This is because library compilation can be performed for more than one device and/or language. If you have not setup the environment variables then you have to manually copy the “*modelsim.ini*” file. Instructions for doing this are provided at the end of this section.

1. Open the Simulation Library Compilation Wizard. This can be accessed from Start Menu→Programs→Xilinx ISE Design Suite 12.3→ISE Design Tools→Tools.
2. The Select Simulator window opens up. Select the appropriate simulator, enter **c:\modeltech.6.6d\win32** for executable location, **compplib.cfg** for Compplib Configuration File and **compplib.log** for Compplib 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:\modeltech_6.6d\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 follow these steps. First, browse to the output directory entered in step 6 (`c:\Xilinx\12.3\ISE_DS\ISE` by default) and open `modelsim.ini`, verify that this file contains the location of the libraries that were just compiled. This file should be copied to the project folder of every design that 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 in the previous section, then Xilinx modifies the `modelsim.ini` file in the ModelSim installation directory, otherwise Xilinx creates a `modelsim.ini` file in the project folder. It does not make sense to compile the libraries with the same options for every project. So this file should be copied to the project folder of every design that 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 pane, expand Design Utilities. Right-click on Compile HDL Simulation Libraries, and select Properties from the right-click menu.
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:\modeltech_6.6d\win32`. Then click OK
6. To Compile HDL Simulation Libraries, in the Processes pane, double-click Compile HDL Simulation Libraries.

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, it should however be error-free.

3 Running the Simulation

Another issue that has cropped up is failure in loading of the design files when ModelSim is executed from within Xilinx. In spite of Xilinx creating the necessary files and invoking ModelSim, we find that we get a “*vsim-19 error*” in ModelSim. Ideally when Xilinx executes ModelSim, it loads up the “**.fdo*” file. This file instructs ModelSim to create a work directory, load up design files and setup the simulation. But unfortunately this sequence of events fails to execute leading to failure in the simulation and hence the above mentioned error. We are still unsure as to which of the two softwares is the root cause for this problem. However a workaround has been found and will be discussed now.

3.1 Behavioral Simulation

To run Behavioral Simulation, select the Simulation radio button in the View pane, choose Behavioral from the drop-down list. Select the Testbench in the Hierarchy pane and run Simulate Behavioral Model process in the Processes pane. ModelSim should now launch and issue one of the following errors

```
# vsim -do {do {task1.fdo}}
# ** Error: (vsim-19) Failed to access library 'work' at "work".
# No such file or directory. (errno = ENOENT)
# Error loading design
```

```
# vsim -do {do {task1.fdo}}
# ** Note: (vsim-3812) Design is being optimized...
# ** Error: (vopt-19) Failed to access library 'work.{task1}' at "work.{task1}".
# No such file or directory. (errno = ENOENT)
# ** Error: Library work.{task1} not found.
# Error loading design
```

The file “*task1.fdo*” is basically a *do* file containing simulation instructions for ModelSim. It is created by Xilinx and carries the same name as your top-level source file. The reason for the error above is that the file was not successfully executed by ModelSim. To fix this, in the transcript window at the ModelSim prompt, enter `do task1.fdo` then hit Enter. Now ModelSim should load up and display the wave window. If your top-level source is different change the name of the *.fdo* file accordingly. Now go ahead and load the custom *do* file that you created using File→Load. The behavioral simulation should now be loaded.

3.2 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 pane, Right-click on Implement Design and select Process Properties. Select Place & Route Properties and check Generate Post-Place & Route Simulation Model. Click OK to save these settings. Now to implement the design, in the Processes pane double-click on Implement Design. Once Implementation is done the files necessary for doing post-route simulation should be created.

To run Post-route Simulation, Select the Simulation radio button in the View pane, choose Post-Route simulation from the drop-down list. Select the Testbench in the Hierarchy pane and run Simulate Post-Place & Route Model process in the Processes pane. ModelSim should now launch and issue the following error

```
# vsim -do {do {task1.tdo}}
# ** Note: (vsim-3812) Design is being optimized...
# ** Error: (vopt-19) Failed to access library 'work.{task1}' at "work.{task1}".
# No such file or directory. (errno = ENOENT)
# ** Error: Library work.{task1} not found.
# Error loading design
```

This error is caused by the same reason as before. To fix this, in the ModelSim prompt that follows enter `do task1.tdo`. Substitute “*task1*” with the name of your top-level design file. Now ModelSim should load up the design and the wave window should appear. Again as before load your *do* file to successfully perform the simulation.