ModelSim Installation & Tutorial

Greg Gibeling
UC Berkeley
gdgib@eecs.berkeley.edu

October 13, 2008

1 Introduction

In this document I will cover the basics of installing ModelSim (see section 2), compiling the Xilinx simulation libraries, and simulating a simple example project (see section 3). This document is targetted at both RDL users and, more generally, those new to the simulation of digital logic (EECS150 and CS61C students).

EECS150 and CS61C users needn't worry about installation, as the relevant Berkeley computers (those in 125 Cory for EECS150 and ilinux1, 2 & 3 for CS61C) already have ModelSim properly installed. Students may wish to read the installation section if they are interested in running ModelSim on their home computers. CS61C students, like all people not using Xilinx FPGAs, should ignore the instructions in this document which mention "Xilinx". In general, RDL users will need section 2, but should refer to the CounterExample, distributed with RDLC for use. Section 3 on the other hand is meant for those without ModelSim experience, or those who need a refresher, in particular U.C. Berkeley students in CS61C or EECS150.

Note that as of this writing the current versions of various pieces of software mentioned in this document are: RDLC 2.2007.6.7 and ModelSim SE 6.2g.

2 Installation & Setup

Make sure you have a license.

Step1. Download the installer from Mentor Graphs (http://www.model.com/).

i. You probably want SE (which is not the same as the student edition), assuming you, your company or school has purchased a license.
ii. Students should consider the http://www.model.com/resources/student_edition/download.asp, which has a limited time license. Otherwise you may wish to use the Xilinx Edition or Altera Edition of ModelSim which are both limited to simulating smaller designs.

Step2. Run the installer.

i. You will want the full, rather than the evaluation version, assuming you already have a license.
ii. The installer should create a directory `c:\Modeltech_version`.

Step3. Install the license by setting `MGLS_LICENSE_FILE` or running the licensing wizard.

i. The former is recommended, if you have a floating license on a server somewhere.
ii. The license for ModelSim should be of the form `1717@somehost.domain`.
iii. If you have an actual `license.dat` file, simply set `MGLS_LICENSE_FILE` to the absolute path to `license.dat`.

Step4. Build the simulation libraries

i. Who needs to do this?

   *. Not necessary unless you intend to use ModelSim to simulate designs for Xilinx FPGAs.
   †. RDLC2 as of v2.2007.3.26 will require this.

ii. Make `c:\Modeltech_`*version*`\ModelSim.ini` writable (it is marked read-only by the installer).

iii. Run `compxlib` to compile the Xilinx libraries. Note that it is best to run from `c:\Modeltech_`*version*.

   *. This will modify `ModelSim.ini` in the current directory to include paths to the Xilinx libraries.

Step5. Define the `MODELSIM` compiler flag for use with `Const.v`.

i. Who needs to do this?

   *. Students in CS61C and EECS150 may skip this step as ModelSim has been installed to do this by default.
   †. Anyone else not using Verilog files from RDLC2, CS61C or EECS150 may also skip this step.

ii. Open `c:\Modeltech_`*version*`\ModelSim.ini` in a text editor (make sure it is still writable).

iii. Add the line `OptionFile = C:\Modeltech_`*version*`\vlog.opt` under the `[vlog]` header.

iv. Create the file `C:\Modeltech_`*version*`\vlog.opt`.

v. Add the line `+define+MODELSIM` to `vlog.opt`.

vi. This will ensure that the preprocessor flags in `Const.v` are properly defined for simulation.

Step6. This should complete your installation of ModelSim.

Step7. You may test your installation by attempting to simulate, as described below, a simple project with a single Verilog file shown in program 1

Program 1 TestVerilog.v

```
1 module  TestVerilog;
2         initial begin
3                 $display(``Hello, World!'');
4         end
5 endmodule
```

3 Use

The purpose of an HDL simulator is to compile, and then simulate an HDL (hardware description language: Verilog and VHDL are examples) on a standard computer. While this is very slow compared to a real circuit implementation, it allows complete visibility and can be much less expensive, making it ideal for design and debugging. Note that as a circuit grows in complexity an FPGA will generally be a better platform, as the simulator will start to degrade in performance, and has no true IO connections.

ModelSim is a very powerful HDL simulation environment, and as such can be difficult to master. To correctly simulate many complex test benches, you will need to create and use a ModelSim project manually. Note that throughout this tutorial we assume you are attempting to simulate a purely Verilog based design. The steps are fairly simple:

Step1. Create a directory for your project (section 3.1).

Step2. Start ModelSim and create a new project (section 3.2).

Step3. Add all your Verilog files to the project (section 3.3).

Step4. Compile your Verilog files (section 3.4).

Step5. Start the simulation (section 3.5).

Step6. Add signals to the wave window (section 3.6).

Step7. Recompile changed verilog files (section 3.7).

Step8. Restart/Run the simulation (section 3.8).

3.1 Step1: Create a Directory for your Project

a. Because ModelSim creates rather large output files you should not save your ModelSim projects. It is a simple matter to recreate the project anyway.

b. Create a directory for your simulation.

   i. For CS61C Students: create a sub-directory somewhere in your home directory, perhaps something like `~/simulation/`.

   ii. For EECS150 students: create a sub-directory in `C:\Users\cs150-xxx\`, perhaps something like `C:\Users\cs150-xxx\Simulation`.

c. When you are done simulating delete this entire directory, this will remove the ModelSim project and all of its temporary files. Obviously your source code should be elsewhere, so that you do not delete it.

3.2 Step2: Start ModelSim and Create a Project

a. Start ModelSim.

   i. On windows there is often a shortcut on the desktop or start menu.

   ii. On UNIX or Linnux, simply run `vsim &`.

   iii. For CS61C students you will need to take the extra step of logging into ilinux1.eecs.berkeley.edu, ilinux2 or ilinux3 where ModelSim is installed. Be sure to login using `ssh -X` to enable X-Windows forwarding or the GUI will not be able to launch.

b. At the main ModelSim window go to `File` → `New` → `Project`.

   i. Enter a project name, this is for your reference only.

   ii. Set the `Project Location` to the directory you created in section 3.1 above.

   iii. You can leave the `Default Library Name` as `work`.

   iv. Click `OK`.

3.3 Step3: Add Your Verilog to the Project

a. Click `Add Existing File` to add your Verilog files to the project.

   i. Click `Browse` to locate the Verilog files you wish to add.

   ii. Note: you can add multiple files at a time by using `Shift-Click` or `Control-Click` to select them all at once.

   iii. Leave `Add File as Type` on default.

   iv. Leave `Folder` as `Top Level`.

   v. You will almost certainly want to select `reference from current location`.

      *. Otherwise you will end up with multiple copies of the same Verilog file floating around, a sure way to lose something.

      †. Note that you will want to select `copy to project directory` for `Const.v` and any other include files, otherwise ModelSim will not be able to find them. These files should rarely, if ever, change so making copies presents little danger.