Using Mentor Questa at PSU

Mark Faust
March 28, 2012
Revised June 26, 2014

Remember when logging in to MCECS systems to use your MCECS username and password, not your ODIN account. Mentor Questa (ModelSim) is available on MCECS Windows and Linux systems. You have the option of using these systems on campus (e.g. Intel PC Lab in basement of Fourth Avenue Building), or remotely (e.g. from home). This brief document is intended to help you with the latter.

You have a choice of using either a command line interface or the graphical user interface (GUI). This document describes how to access both from a remote Windows or Linux (including MacOS) system.

Windows

You can follow these instructions to remotely access Questa via your own Windows system.

Command line access

For command line access, I assume you'll remote login to the MCECS Linux systems.

You'll need the following tools:

- VPN – Provides secure remote access if you're not on campus
  You can download from CAT at http://cat.pdx.edu/windows/maseeh-college-vpn.html
- WinSCP – Used to transfer files from a home PC to MCEC. Note: Not needed if you use VPN and samba and map your U: drive.
  You can download from http://winscp.net/eng/index.php
- putty – needed to remote login to MCECS Linux systems
  You can download from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

You can move files from your PC to your remote directory in several ways. You can:

- Use WinSCP (using SFTP protocol) to copy them to an MCECS Linux system
- Map your MCECS Linux directory to a drive on your own PC by following the directions below

To map your Linux directory, you'll need to use VPN and (the first time) map the `U:` drive manually. See http://cat.pdx.edu/windows/mapping-a-drive-to-a-unix-account.html for details. See also: http://cat.pdx.edu/windows/accessing-your-unix-linux-files-from-windows-using-samba.html

Use putty to remotely login to `redhat.ece.pdx.edu` (`eve.ece.pdx.edu` or `walle.ece.pdx.edu`) using your MCECS username:

Before using Questa for the first time you'll need to do the following (only once):

```
% addpkg
```

```
Navigate with down-arrow key to find mentor-questa.
Select it with Enter key.

% logout

Login again

% listpkg

Confirm Mentor Questa installed.
```

Once you've got Questa installed, you begin by creating a library for your project. I recommend keeping every project in a separate directory. You only need to create the library once for a project:

```
% vlib work
```

Now you can compile one or more Verilog or SystemVerilog files (or VHDL, but that's another matter). There are several alternatives. If you use .sv as a suffix for your filename then Questa will assume it contains SystemVerilog. Alternatively, you can tell Questa to assume SystemVerilog regardless of the file suffix by using the –sv flag.

```
% vlog file1.sv file2.sv
```

or

```
% vlog –sv file1.v file2.v
```

Adding the –source flag will cause the compiler to print out the source line along with any error messages:

```
% vlog –source file1.sv file2.sv
```

If you're compiling multiple files you can put their names in a file:

```
% vlog –f file
```

After compilation you run the simulator, naming the top level module (not filename). The –c flag suppresses the GUI. After getting the VSIM prompt type "run –all":

```
% vsim –c modulename

  VSIM 1> run –all
```

GUI access

For GUI access, use VPN then run Microsoft's Remote Desktop Connection program and specify `ts.cecs.pdx.edu` as the computer to connect to then supply your username and password in the dialog box. Be sure to also specify domain "`cecs`".

Once you've got a display of your remote MCECS Windows desktop, click on the Start menu->All Programs and find Questa Sim.

Linux (and Mac)

You can follow these instructions to remotely access Questa via your own Linux or MacOS system.

Command line access

You can copy files using secure cp (scp). To copy a file from your local system to your MCECS Linux directory, execute the following command on your local system:

```
% scp localfile username@redhat.ece.pdx.edu:remotefile
```

If you use VPN, you can do a few things to make moving files easier. You can download the VPN tool you need from the CAT at

http://cat.pdx.edu/windows/maseeh-college-vpn.html

On the Mac, you can then use the Finder's Go->Connect to Server… pull down menu and create a connection for

```
smb://unix.cecs.pdx.edu/home
```

Then click on the redhat5 folder. This will allow you to use the Finder to drag/drop files between your local system and the remote Linux system.

On a Linux system you can do something similar by selecting the Places->Connect to Server… pull down menu to connect to `redhat.ece.pdx.edu`. Be sure to specify ssh for service type. This should create an icon on your local desktop which when opened shows your remote Linux directory.

You can use secure shell (ssh) to remotely login using your MCECS username:

```
% ssh username@redhat.ece.pdx.edu
```

Before using Questa for the first time you'll need to do the following (only once):

```
% addpkg

Navigate with down-arrow key to find mentor-questa.
Select it with Enter key.

% logout

% ssh username@redhat.ece.pdx.edu

% listpkg

Confirm Mentor Questa installed.
```

Once you've got Questa installed, you begin by creating a library for your project. I recommend keeping every project in a separate directory. You only need to create the library once for a project:

```
% vlib work
```

Now you can compile one or more Verilog or SystemVerilog files (or VHDL, but that's another matter). There are several alternatives. If you use .sv as a suffix for your filename then Questa will assume it contains SystemVerilog. Alternatively, you can tell Questa to assume SystemVerilog regardless of the file suffix by using the –sv flag.

```
% vlog file1.sv file2.sv
```

or

```
% vlog –sv file1.v file2.v
```

Adding the –source flag will cause the compiler to print out the source line along with any error messages:

```
% vlog –source file1.sv file2.sv
```

If you're compiling multiple files you can put their names in a file:

```
% vlog –f file
```

After compilation you run the simulator, naming the top level module (not filename). The –c flag suppresses the GUI. After getting the VSIM prompt type "run –all":

```
% vsim –c modulename
  VSIM 1> run –all
```

GUI Access

You can use the GUI by using X on your Linux or MacOS system:

```
% ssh -X username@redhat.ece.pdx.edu
```

Once the X window comes up, type:

```
% vsim
```

Be patient as this can take a little while initially.