



TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

Department of Computer Systems

## DESIGN VERIFICATION

IAF0620

LAB MANUAL

René Pihlak

[rene.pihlak@taltech.ee](mailto:rene.pihlak@taltech.ee)

Tallinn 2018

# Contents

<b>I</b>	<b>Connecting to lab computers</b>	<b>3</b>
1	Using Linux . . . . .	3
2	Using Windows . . . . .	4
3	Using Mac OS X . . . . .	5
<b>II</b>	<b>Using Questa Sim for verification</b>	<b>6</b>
4	Setting up Linux environment . . . . .	6
5	Setting up a project in Questa Sim . . . . .	7
5.1	Launching Questa Sim . . . . .	7
5.2	Create project and compile design files . . . . .	8
6	Code coverage in Questa Sim . . . . .	8
6.1	Compiling the design files . . . . .	8
6.2	Analysing coverage . . . . .	9
6.3	Save summary report . . . . .	11
7	PSL assertions in Questa Sim . . . . .	11
7.1	Writing PSL assertions in separate file . . . . .	11
7.2	Using PSL file in Questa Sim . . . . .	12
8	Waveforms and simulation . . . . .	15
8.1	Adding waveforms to simulation . . . . .	15
8.2	Save summary report . . . . .	16

## Part I

# Connecting to lab computers

## 1 Using Linux

Connecting to the lab computers requires connecting first to a proxy server. Connecting to the proxy server can be done using command line by entering the following command (see Figure 1):

```
ssh -X -l Heli.Kopter@intra.ttu.ee proksi.intra.ttu.ee
```

where

- ssh** – name of the command,
- X** – parameter requestion X forwarding (important!!), and
- Heli.Kopter** – your UNI-ID.

Follow the instructions (i.e. enter your password, etc).

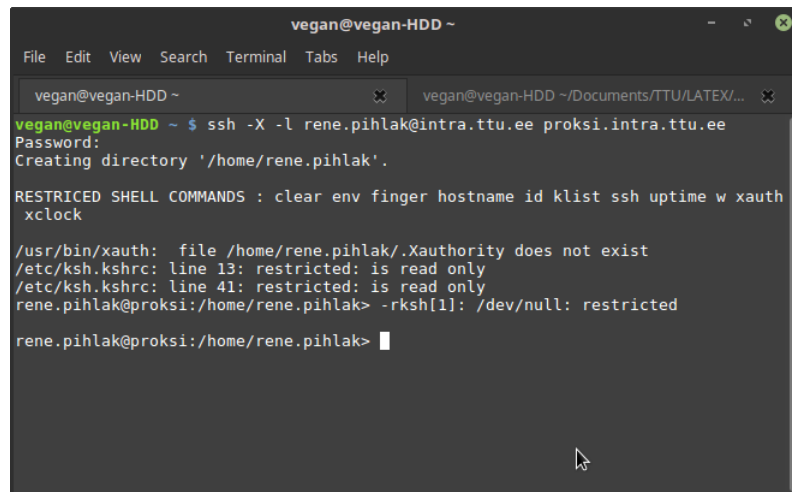


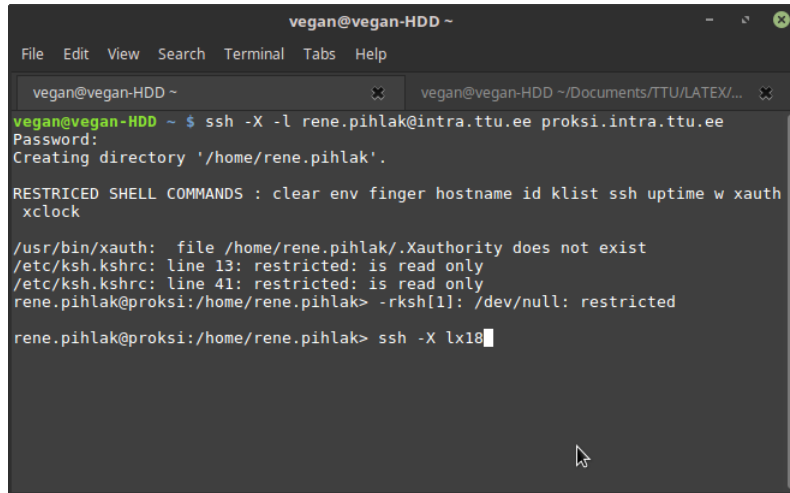
Figure 1. Connecting to proxy server

After you have logged in to the proxy server, connect to a specific lab computer to run your programs. In order to connect to a lab computer use the following command (see Figures 2 and 3):

```
ssh -X 1x9
```

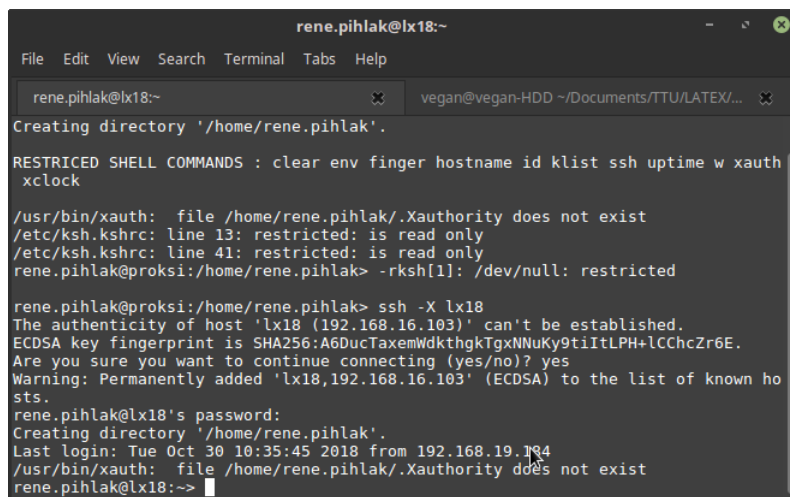
where

- ssh** – name of the command,
- X** – parameter requestion X forwarding (important!!), and
- 1x9** – the ID of computer.

A terminal window titled 'vegan@vegan-HDD ~' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help). It shows an SSH session initiated from 'vegan@vegan-HDD' to 'rene.pihlak@proksi.intra.ttu.ee'. The session includes password prompts, directory creation, and a restricted shell setup. The user then runs 'ssh -X lx18' to connect to the lab computer.

```
vegan@vegan-HDD ~  
File Edit View Search Terminal Tabs Help  
vegan@vegan-HDD ~ $ ssh -X -l rene.pihlak@proksi.intra.ttu.ee  
Password:  
Creating directory '/home/rene.pihlak'.  
  
RESTRICTED SHELL COMMANDS : clear env finger hostname id klist ssh uptime w xauth  
xclock  
  
/usr/bin/xauth: file /home/rene.pihlak/.Xauthority does not exist  
/etc/ksh.kshrc: line 13: restricted: is read only  
/etc/ksh.kshrc: line 41: restricted: is read only  
rene.pihlak@proksi:/home/rene.pihlak> -rksh[1]: /dev/null: restricted  
  
rene.pihlak@proksi:/home/rene.pihlak> ssh -X lx18
```

Figure 2. Connecting to lab computer

A terminal window titled 'rene.pihlak@lx18:~' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help). It shows the continuation of the SSH session from Figure 2. The user logs in as 'rene.pihlak' on host 'lx18'. The terminal displays the host's fingerprint, a warning about the new host, and the user's confirmation to proceed. It also shows the creation of the user's home directory and the setup of the restricted shell.

```
rene.pihlak@lx18:~  
File Edit View Search Terminal Tabs Help  
rene.pihlak@lx18:~  
Creating directory '/home/rene.pihlak'.  
  
RESTRICTED SHELL COMMANDS : clear env finger hostname id klist ssh uptime w xauth  
xclock  
  
/usr/bin/xauth: file /home/rene.pihlak/.Xauthority does not exist  
/etc/ksh.kshrc: line 13: restricted: is read only  
/etc/ksh.kshrc: line 41: restricted: is read only  
rene.pihlak@proksi:/home/rene.pihlak> -rksh[1]: /dev/null: restricted  
  
rene.pihlak@proksi:/home/rene.pihlak> ssh -X lx18  
The authenticity of host 'lx18 (192.168.16.103)' can't be established.  
ECDSA key fingerprint is SHA256:A6DucTaxemWdkthgkTgxNNuKy9tiItLPH+LCChcZr6E.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'lx18,192.168.16.103' (ECDSA) to the list of known ho  
sts.  
rene.pihlak@lx18's password:  
Creating directory '/home/rene.pihlak'.  
Last login: Tue Oct 30 10:35:45 2018 from 192.168.19.134  
/usr/bin/xauth: file /home/rene.pihlak/.Xauthority does not exist  
rene.pihlak@lx18:~>
```

Figure 3. After connecting to lab computer

Before connecting to a lab computer, please check if the computer is available (i.e. that it is running Linux) and what is the current load of the computer. You can get information about the current status of lab computers by visiting the following web page:

<https://ati.ttu.ee/klasi-staatust/>

This web page also provides the names/IDs of the computers.

Once you have successfully logged in to a lab computer you can set up the Linux system to start using Questa Sim (see Section 4).

## 2 Using Windows

If you are using Windows operating system, there are several ways to connect to the lab computers. Make sure you also read the instructions for Linux (see Section 1). One way

to connect to the proxy server (and lab computers) can be done using program called putty:

[http://mini.li.ttu.ee/~priit/IAY0340.2013/ssh\\_setup/putty\\_guide.html](http://mini.li.ttu.ee/~priit/IAY0340.2013/ssh_setup/putty_guide.html)

Notice that in order to be able to see GUI applications, you might have to install additional software such as Cygwin or Xming. Therefore the preferred and recommended way to connect is by using Linux.

### **3 Using Mac OS X**

This guide is not tested on latest Mac OS X. If the necessary X11 libraries are installed, then connecting should be identical to Linux (see Section 1).

## Part II

# Using Questa Sim for verification

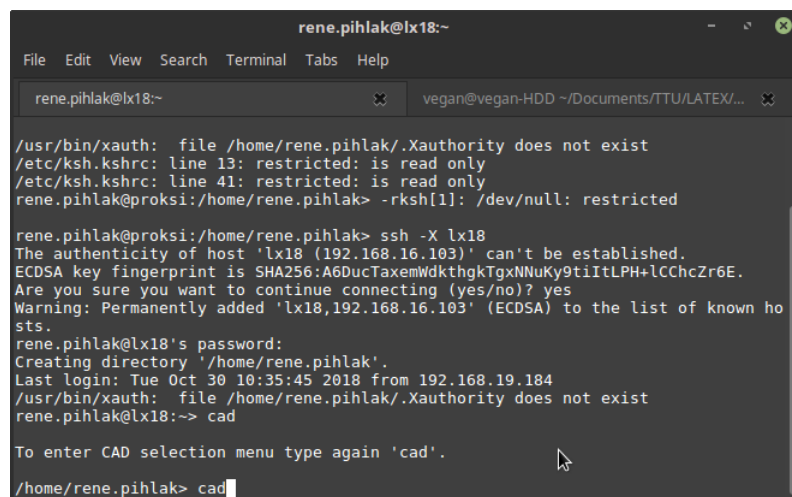
## 4 Setting up Linux environment

Before launching Questa Sim, the Linux environment has to be configured. This step is necessary both when using lab computers at school and when connecting to the lab computers remotely. The following example shows how to configure the system when using lab computers remotely.

After logging in to the lab computer running Linux operating system, run the following command(s) in the command line (see Figure 4):

**cad**

**cad**

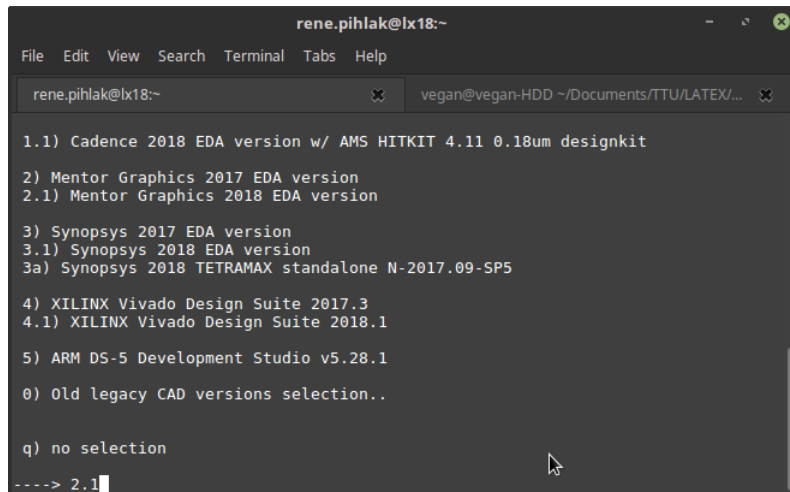


```
rene.pihlak@lx18:~  
File Edit View Search Terminal Tabs Help  
rene.pihlak@lx18:~  
vegane@vegan-HDD ~/Documents/TTU/LATEX/...  
/usr/bin/xauth: file /home/rene.pihlak/.Xauthority does not exist  
/etc/ksh.kshrc: line 13: restricted: is read only  
/etc/ksh.kshrc: line 41: restricted: is read only  
rene.pihlak@proksi:/home/rene.pihlak> -rksh[1]: /dev/null: restricted  
  
rene.pihlak@proksi:/home/rene.pihlak> ssh -X lx18  
The authenticity of host 'lx18 (192.168.16.103)' can't be established.  
ECDSA key fingerprint is SHA256:A6DucTaxemWdkthgkTgxNNuKy9tiItLPH+lCChcZr6E.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'lx18,192.168.16.103' (ECDSA) to the list of known ho  
sts.  
rene.pihlak@lx18's password:  
Creating directory '/home/rene.pihlak'.  
Last login: Tue Oct 30 10:35:45 2018 from 192.168.19.184  
/usr/bin/xauth: file /home/rene.pihlak/.Xauthority does not exist  
rene.pihlak@lx18:~> cad  
  
To enter CAD selection menu type again 'cad'.  
/home/rene.pihlak> cad
```

Figure 4. Run configuration selector

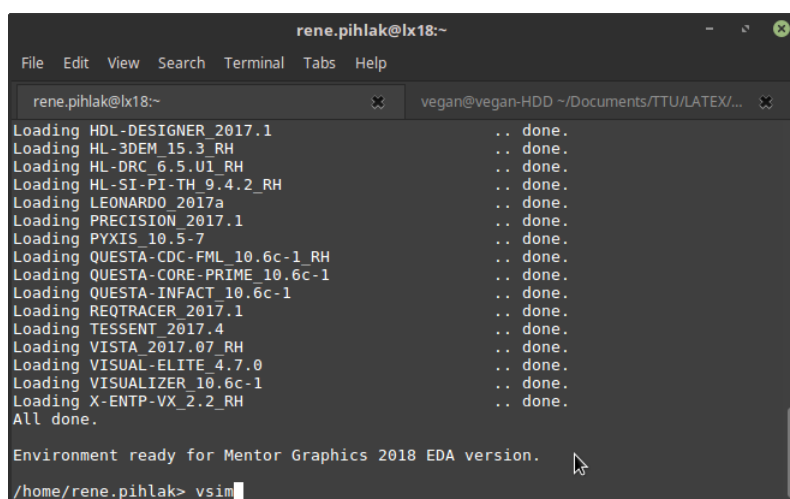
Once the menu appears (see Figure 5), type '2.1' to select Mentor Graphics 2018 and press ENTER.

Several paths, etc will be loaded (see Figure 6). You are now ready to launch Questa Sim (see Section 5).



```
rene.pihlak@lx18:~  
File Edit View Search Terminal Tabs Help  
rene.pihlak@lx18:~ x vegan@vegan-HDD ~/Documents/TTU/LATEX/... x  
1.1) Cadence 2018 EDA version w/ AMS HITKIT 4.11 0.18um designkit  
2) Mentor Graphics 2017 EDA version  
2.1) Mentor Graphics 2018 EDA version  
3) Synopsys 2017 EDA version  
3.1) Synopsys 2018 EDA version  
3a) Synopsys 2018 TETRAMAX standalone N-2017.09-SP5  
4) XILINX Vivado Design Suite 2017.3  
4.1) XILINX Vivado Design Suite 2018.1  
5) ARM DS-5 Development Studio v5.28.1  
0) Old legacy CAD versions selection..  
  
q) no selection  
----> 2.1
```

Figure 5. Selecting Mentor Graphics 2018



```
rene.pihlak@lx18:~  
File Edit View Search Terminal Tabs Help  
rene.pihlak@lx18:~ x vegan@vegan-HDD ~/Documents/TTU/LATEX/... x  
Loading HDL-DESIGNER_2017.1 .. done.  
Loading HL-3DEM_15.3_RH .. done.  
Loading HL-DRC_6.5.U1_RH .. done.  
Loading HL-SI-PI-TH_9.4.2_RH .. done.  
Loading LEONARDO_2017a .. done.  
Loading PRECISION_2017.1 .. done.  
Loading PYXIS_10.5-7 .. done.  
Loading QUESTA-CDC-FML_10.6c-1_RH .. done.  
Loading QUESTA-CORE-PRIME_10.6c-1 .. done.  
Loading QUESTA-INFACIT_10.6c-1 .. done.  
Loading REQTRACER_2017.1 .. done.  
Loading TESSENT_2017.4 .. done.  
Loading VISTA_2017.07_RH .. done.  
Loading VISUAL-ELITE_4.7.0 .. done.  
Loading VISUALIZER_10.6c-1 .. done.  
Loading X-ENTP-VX_2.2_RH .. done.  
All done.  
  
Environment ready for Mentor Graphics 2018 EDA version.  
/home/rene.pihlak> vsim
```

Figure 6. Loading the environment

## 5 Setting up a project in Questa Sim

### 5.1 Launching Questa Sim

Questa Sim is launched by the following command (see Figure 6):

**vsim**

It is advisable to launch Questa Sim in some project folder as it will generate several files and it is easier to keep track on all the necessary files if the working directory does not include other unrelated files and folders.

Also keep in mind that all files not saved to your P drive can (and will be) deleted after you log out.

## 5.2 Create project and compile design files

Create a new project (see Figure 7).

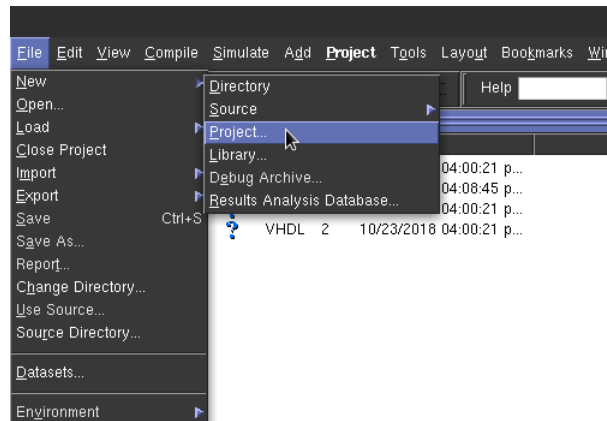


Figure 7. Creating Questa Sim project

Right click on the project window and add your design files to the project (see Figure 8).  
Select all the files in your design.

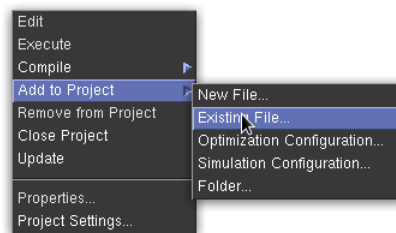


Figure 8. Add files to Questa Sim project

The example design contain the following files:

- data.vhd
- tb\_prose.vhd
- control.vhd
- prose.vhd

## 6 Code coverage in Questa Sim

### 6.1 Compiling the design files

Compile the design files with verification (coverage) options by entering the following command as one line (see Figure 9):



```
vcom -coveropt 3 +cover +acc data.vhd control.vhd prose.vhd tb_prose.vhd
```

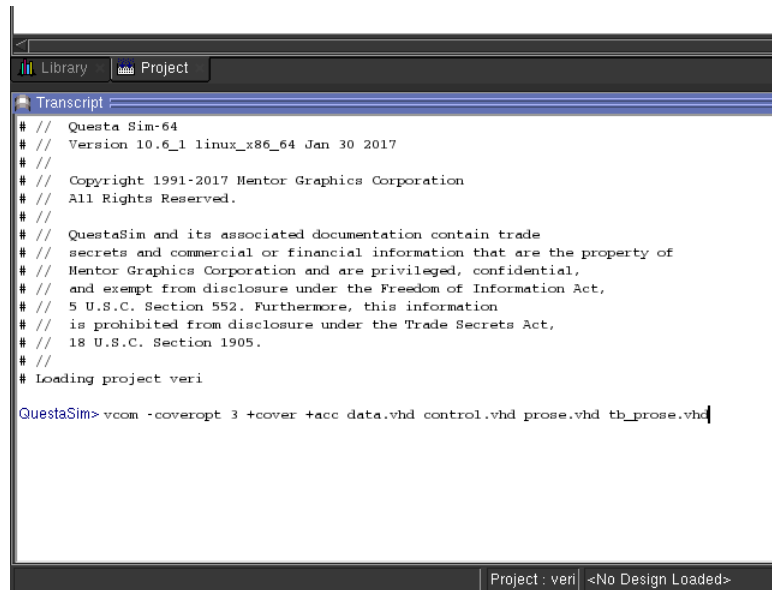


Figure 9. Compiling design files

Notice that the order of files is important. File `prose.vhd` depends on `data.vhd` and `control.vhd`, thus these files have to precede it. Similarly file `tb_prose.vhd` is the last file due to dependencies from the preceding files.

## 6.2 Analysing coverage

To analyse the coverage enter the following command as one line (see Figure 10):

```
vsim -coverage -vopt work.E -c -do "coverage save -onexit -directive
-codeAll ex_cov; run -all"
```

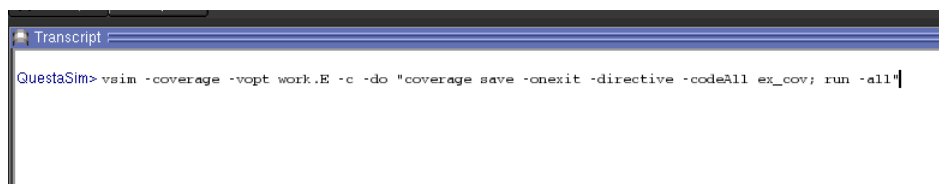


Figure 10. Run coverage analysis

Notice that the `E` in `work.E` is the name of the test bench entity.

The coverage results can be viewed either per file (see Figure 11) or by coverage type (see Figures 12 and 13). To choose view by coverage type click on tab called Analysis.

If you want to observe the waveforms from the test bench simulation follow the instructions in Figure 8.1. To exit the coverage/simulation mode, enter the following command:

```
quit -sim
```

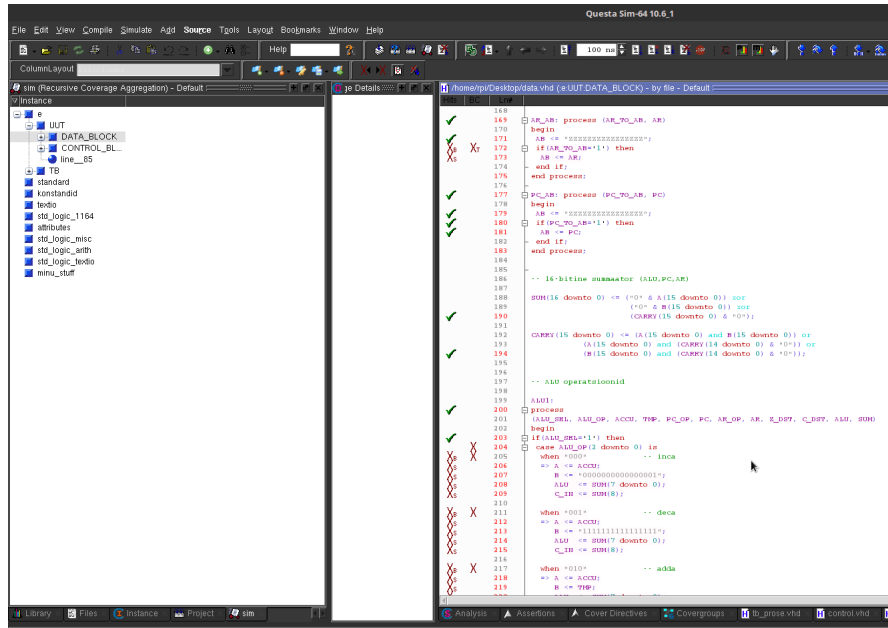


Figure 11. Coverage per file

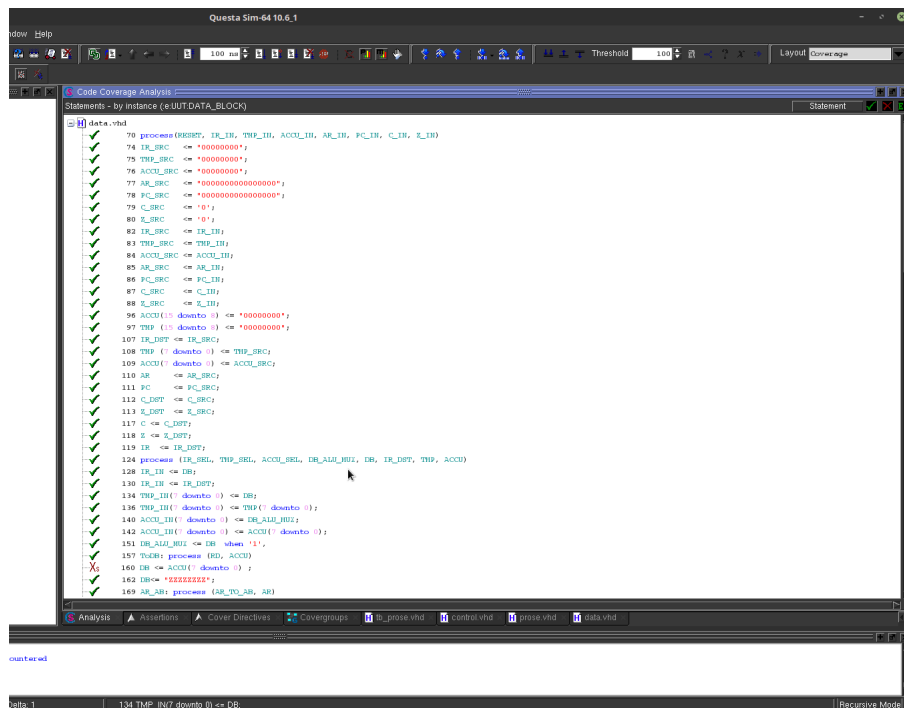


Figure 12. Coverage per coverage type

### 6.3 Save summary report

In order to save an aggregated report in html format, enter the following command:

**vcover report -html ex\_cov**

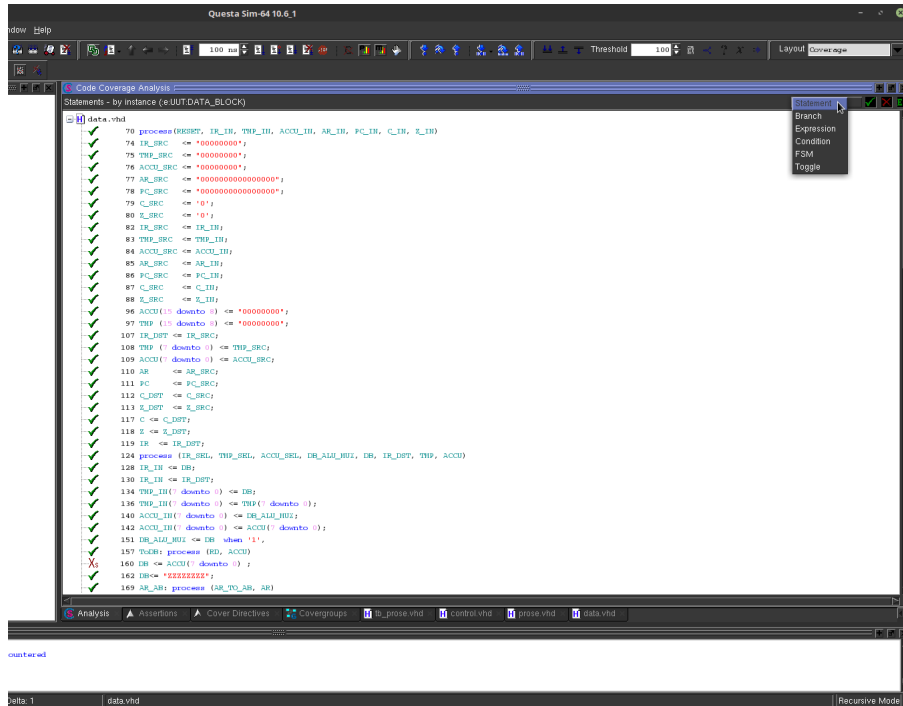


Figure 13. Choosing coverage type

## 7 PSL assertions in Questa Sim

### 7.1 Writing PSL assertions in separate file

In this subsection, the focus is on writing PSL assertions.<sup>1</sup> The following example is based on GCD design<sup>2</sup> (see Figure 14).

```
vunit check_gcd (gcd(FSMD)){
    default clock is rising_edge(clk);
    property did_start is always rst -> eventually! go_i;
    assert did_start;
    s_0: cover {State=ST0} report "Missing state?";
    s_1: cover {State=ST1};
    s_2: cover {State=ST2};
    sequence multiseq is {(State = ST0); (State = ST1)};
    cover {multiseq};
}
```

Figure 14. Sample PSL file

PSL assertions can be named (e.g. `s_0`) or not (see Figure 14). Using named assertions is highly recommended. Assertions can send messages (i.e. *'report'*) in case the assertion

<sup>1</sup>For detailed description on PSL refer to lecture notes and/or other materials.

<sup>2</sup>Modified GCD design is available at <http://ati.ttu.ee/~rpi>.

failed. This is also recommended.

The PSL file process always starts with a keyword `vunit` which is followed by custom name which is followed by the name of the entity (`gcd`) and the name of (optional) architecture (`FSMD`).

## 7.2 Using PSL file in Questa Sim

In order to use the PSL file to verify the design, Questa Sim has to be made aware of the PSL file (see Figure 15).

```
# include and compile design/testbench files
vcom -2008 -coveropt 3 +cover +acc "./gcd.vhdl" -pslfile gcd.psl
vcom -2008 -coveropt 3 +cover +acc "./tb_gcd.vhdl"
```

Figure 15. Compiling VHDL file with PSL file

Now the compiled file can be loaded to the simulation (see Figure 16). The argument `-coverage` is optional, but this allows to analyse also code coverage. However, before running the simulation, it is useful to enable so called ATV on some or all assertions and/or cover directives.

```
# start simulation
vsim -coverage -assertdebug work.E
atv log -enable :e:UUT:assert__did_start
atv log -covers -enable :e:UUT:s_0
```

Figure 16. Running simulation with PSL file

It might be useful to add the assertions and cover directives to the wave window (see Figure 17). This allows observing the changes in assertions (see Figure 18) after the simulation is restarted. See also Section 8.

The simulation can then be restated and ran for, say, 300 ns (see Figure 19).

In addition, the assertions can be analyzed in the 'ATV window'. For this first add the assertions/cover directives to the ATV window (see Figures 20 and 21). An example of ATV view is show in Figure 22.

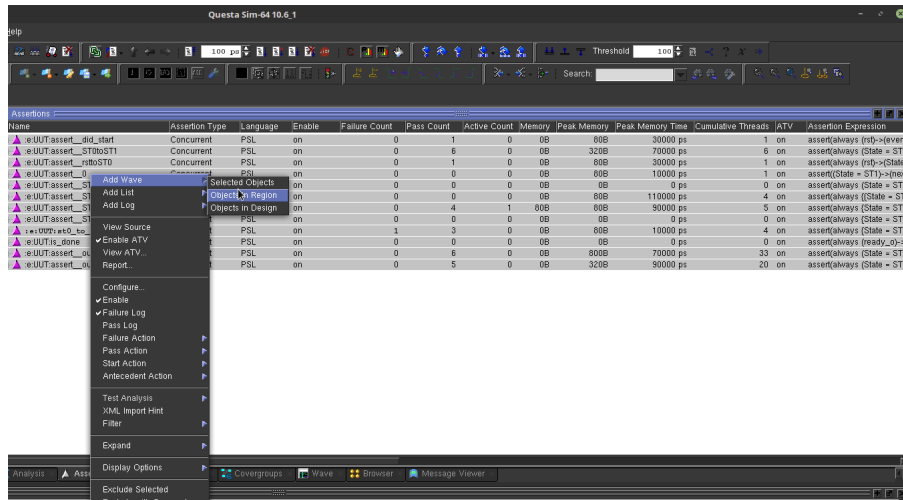


Figure 17. Adding assertions to the wave window

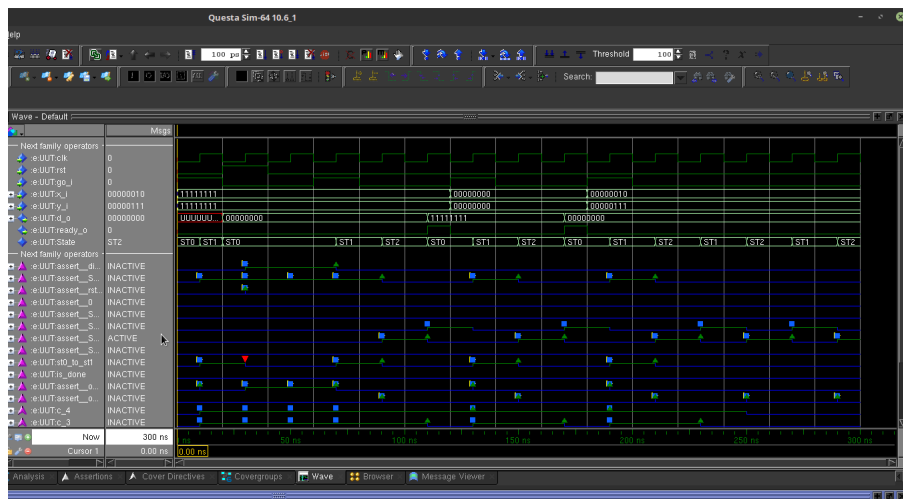


Figure 18. Observing assertions in wave window

```
# start simulation
restart; run 300ns
```

Figure 19. Running simulation with PSL file

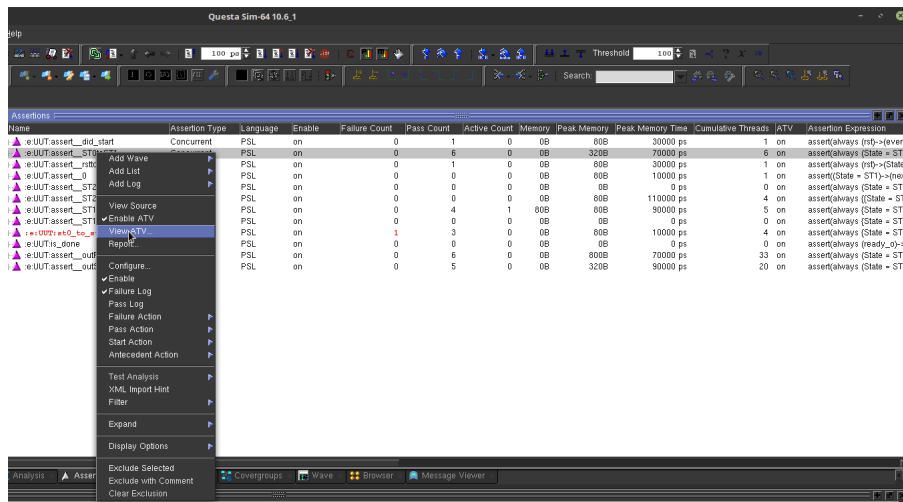


Figure 20. Adding PSL assertions to ATV window (1)

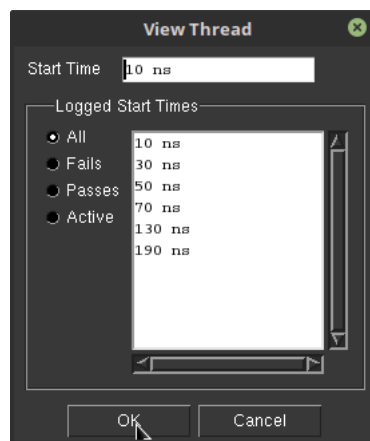


Figure 21. Adding PSL assertions to ATV window (2)

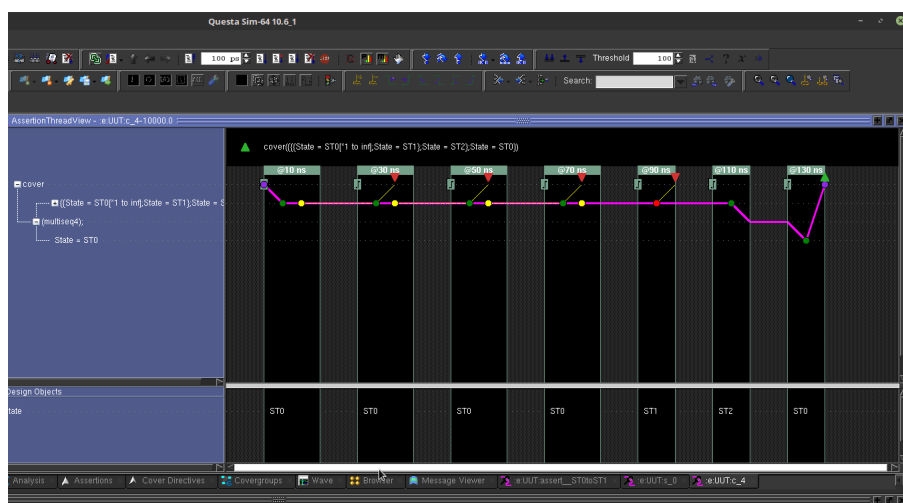


Figure 22. Observing PSL assertions in ATV window

## 8 Waveforms and simulation

### 8.1 Adding waveforms to simulation

Firstly, add signals to the waveform window (see Figure 23). By default no signals are added. When simulating more complex designs it might be a good idea to include only signals that you are investigating. In case of a smaller design, you can include all signal. Notice that you can change the order of the signals to group relevant signals together.

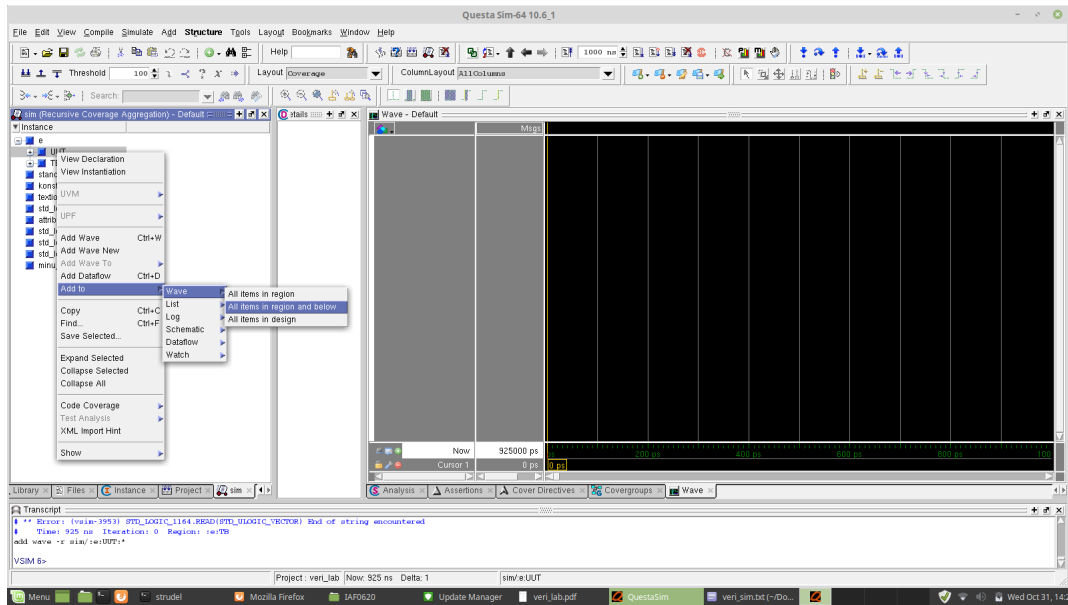


Figure 23. Add signals to the waveforms

Next restart simulation (see Figure 24). In order to see the newly added signals, the simulation has to be restarted.

After restarting the simulation, you have to run the simulation (see Figure 25). However, if you have not done it yet, you should change the simulation length to match your test vectors and clock period. In case there are 15 test vectors and each test vector is tested within one clock cycle, then the simulation should run at least 15 test cycles. If one test cycle is 20 ns, then the simulation should run at least  $(15 \times 20 =) 300$  ns.

Finally, enlarge the waveform window and zoom to fit the whole simulation (see Figure 26).

To exit the coverage/simulation mode, enter the following command:

```
quit -sim
```

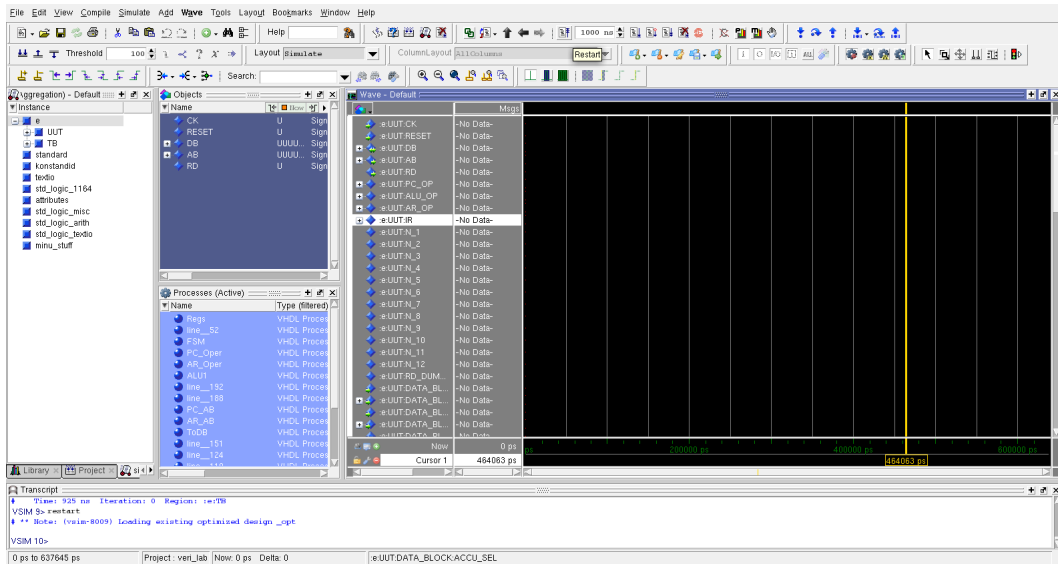


Figure 24. Restart simulation

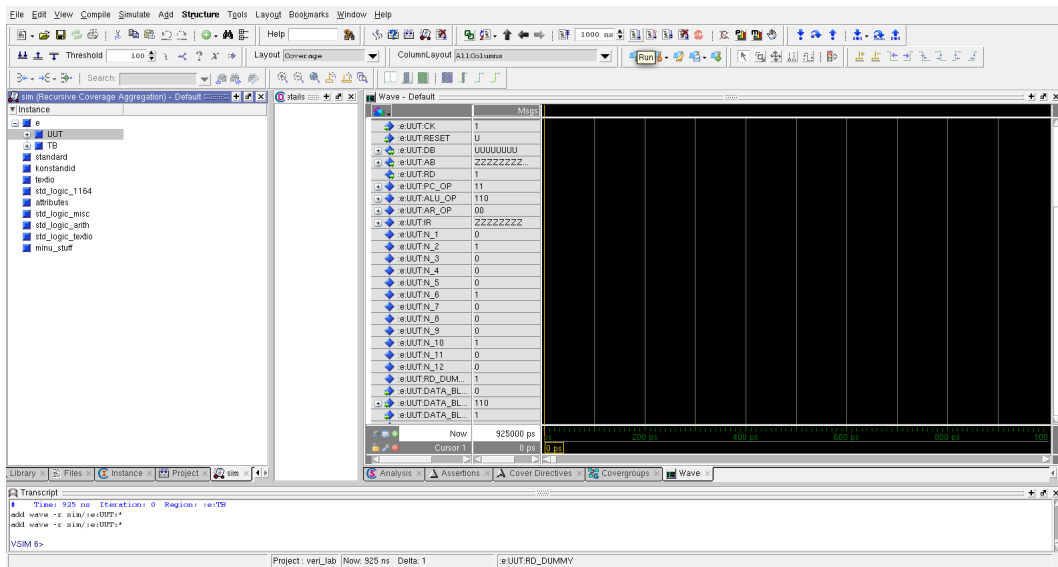


Figure 25. Run simulation

## 8.2 Save summary report

In order to save an aggregated report in html format, enter the following command:

```
vcover report -html ex_cov
```



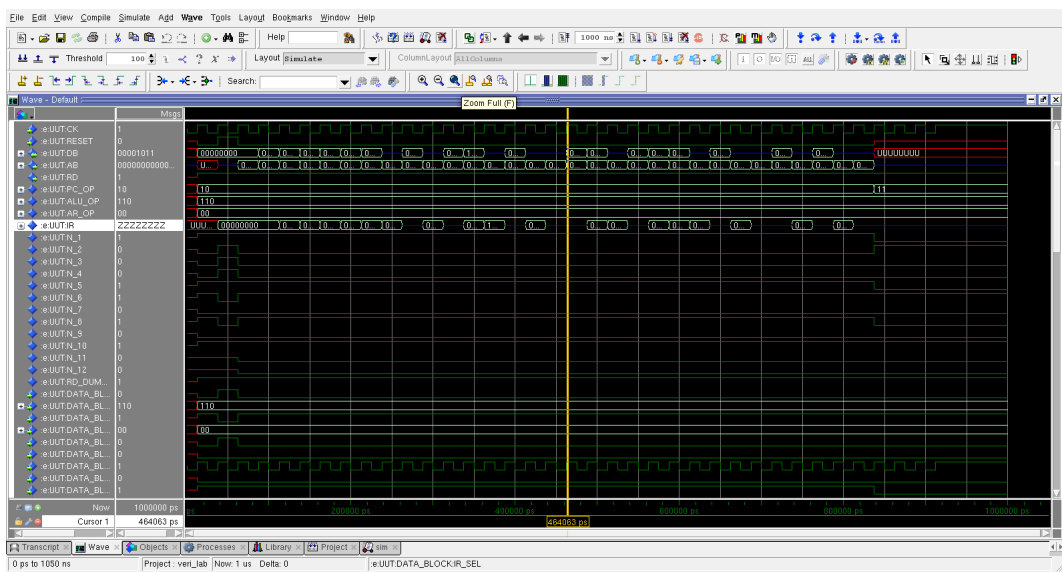


Figure 26. Zoom to include the whole simulation