

Department of Computer Systems

DESIGN VERIFICATION

IAF0620

LAB MANUAL

René Pihlak rene.pihlak@taltech.ee

Tallinn 2018

Contents

Ι	Connecting to lab computers	3
1	Using Linux	3
2	Using Windows	4
3	Using Mac OS X	5
II	Using Questa Sim for verification	6
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 lx9

where

- ${\tt ssh}$ name of the command,
- -X parameter requestion X forwarding (important!!), and
- **1x9** the ID of computer.



Figure 2. Connecting to lab computer



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/klassi-staatus/

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

Once you have successful 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

Notice that in order to be able to see GUI applications, you might have to installe 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.p	ihlak@	lx18:~			× 😣
File	Edit	View	Search	Terminal	Tabs	Help				
rei	ne.pihla	k@lx18				*				8
/usr /etc /etc rene The ECDS Are Warn sts. rene Crea Last /usr rene To e /hom	/bin/ /ksh. .pihl authe A key you s ing: .pihl ting logi /bin/ .pihl nter e/ren	xauth kshrc kshrc ak@pro ntici fingu ure yo Permal ak@lx: direc n: Tuo xauth ak@lx: CAD so e.pih	: file : line bksi:/h bksi:/h ty of h ty of h ty of h tory '/ e Oct 3 : file 18:~> c electio lak> ca	/home/rr 13: rest ome/rene ost 'lxli os SHA2: to cont. added 'l: ssword: home/rend 0 10:35:. /home/rr ad n menu ty d	ene.pi ricted .pihla B (192 56:A6D inue c x18,19 e.pihl 45 201 ene.pi	hlak/. : is r : is r k> -rk k> ssh :168.1 ucTaxe connect 2.168. .ak'. .8 from hlak/. gain 'c	Xauthority do read only read only (sh[1]: /dev/r h -X lx18 (6.103)' can'f mWdkthgkTgxNN ring (yes/no); 16.103' (ECDS n 192.168.19.1 Xauthority do	Hes not exist ull: restricted t be established. LuKy9tIItLPH+lCChe yes A) to the list of Hes not exist ↓	:Zr6E. f know	n ho

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).



Figure 5. Selecting Mentor Graphics 2018

					rene.p	oihlak@l	x18:~		8
File	Edit	View	Search	Terminal	Tabs	Help			
rer	e.pih	lak@lx18				*	vegan@vegan-HDD ~/Documents/TTU/LA		*
Load Load Load Load Load Load Load Load	ing ing ing ing ing ing ing ing ing ing	HDL-DE HL-3DE HL-SDE- LEONAR PRECIS PYXIS_ QUESTA QUESTA QUESTA QUESTA QUESTA VISUAL VISUAL VISUAL X-ENTP ent re	SIGNER_ M_15.3_ 6.5.U1 PI-TH_9 D0_2017 10N_201 10.5-7 -CCC-FM -CCRE-P -INFACT CER_201 T_2017.2 017.07 -ELITE_ IZER_10 -VX_2.2 ady for	2017.1 RH _RH .4.2_RH a 7.1 L_10.6c- T.1 [10.6c-1 _10.6c-1 _RH 4.7.0 .6c-1 _RH Mentor	1_RH 6c-1 Graphi	.cs 201	done. done.		
/hom	e/re	ne.pih	lak> vs	im					

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 of 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.¹ The following example is based on GCD design² (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

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

²Modified 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.

Version V		0	orta Sim 64 10	5 1								-	. 0
Implementation Imple		44	resca sini-o4 ro.	v_1									. 🗸
A set of	1eib												
Ansendon : Attendon Type: Annyounge Enable Pass Count Pass Cou	22 👾 🗶 🕅 🛛 🖄 👔 - 1 - 4- 1	· 월 10	10 pa 🗧 🕺 🕺	8 🕅 🚳	R 🖬 🖬 🕯	🕨 🕴 😵 🛛	🕴 🕺 🏦	余 🏻 😐	土 ∓ Threshold	100 🗧 👸			
Asserbios Asserbios <t< td=""><td>🔍 🥞 🥐 🐐 🤘 💷 🔤</td><td>30 🖭 🏧 🥓</td><td>一 画 原 承</td><td></td><td></td><td></td><td></td><td></td><td>Search:</td><td>V \$</td><td></td><td> 45 45 % </td><td></td></t<>	🔍 🥞 🥐 🐐 🤘 💷 🔤	30 🖭 🏧 🥓	一 画 原 承						Search:	V \$		 45 45 % 	
Assertions Operation <	, , , , , , , , , , , , , , , , , , , ,												
Austein:: Assettion::													
Nume Asterion Type Language Enable Pails Court Active Court Peak Alencop Peak Alencop The Cumulation Thread: ATV Asterion Type Language A wUltraserd State Cancurrent PSL on 0 1 0 08 03000 ps 1 on assign@ways (Tobe > etc) A wUltraserd State Tobo ps 1 0 08 08 0000 ps 1 on assign@ways (Tobe > etc) A wUltraserd Add LM Concurrent PSL on 0<	Assertions :												
A. wUUTasset dig. duft Concurrent PSL on 0 1 0 08 008 3000 ps 1 on assign(avery (tip)-4 eventy A. wUUTasset AssULTasset State Concurrent PSL on 0 1 0 08 208 7000 ps 1 on assign(avery (tip)-4 eventy A. wUUTasset AssULTasset Add Lag Concurrent PSL on 0 0 0 0 0 0 0 assign(avery (tip)-4 eventy A. wUUTasset Add Lag Concurrent PSL on 0 0 0 0 0 0 0 assign(avery (tip)-4 eventy A. wUUTasset Add Lag Concurrent PSL on 0 0 0 0 0 0 0 0 assign(avery (tip)-4 eventy assign(avery (tip)-	Name	Assertion Ty	pe Language	Enable	Failure Count	Pass Count	Active Count	Memory	Peak Memory Pea	k Memory Time C	umulative Threads	ATV Assertion Expr	ession
A cullmaser_ST005T1 Concurrent P5L on 0 6 6 0 68 3268 7000 ps 6 6 0 a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 Add Lut S0040 ps 1 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cullmaser_ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a assequavy (Sale - ST0 A cull s0000 ps 4 0 o a a	ie:UUT:assertdid_start	Concurrent	PSL	on	(ı 1	0	0B	80B	30000 ps	1	on assert(ahvays	rst)->(event
A eUUTastef_m6510 Cercurrent P3L on 0 1 0 08 08 08 0000 ps 1 1 on aster@away(07e)~5384 4 0 aster@away(07e)~5384 4 0 aster@away(07e)~5384 5 71)-inter 0 add Wax Add Log 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	:e:UUT:assertST0toST1	Concurrent	PSL	on	0	1 6	0	OB	320B	70000 ps	6	on assert(always i	State = STO
A cuUrraiset. A cuUr	A :e:UUT:assertrsttoST0	Concurrent	PSL	on	6	1	0	OB	80B	30000 ps	1	on assert(always	rst)->(State
A eUl aster. 2 A eUl aster. 2 A de List	Add Wave	Concernent Party	PSI Jactori Objecte	on		. 0	0	0B	80B	10000 ps	1	on assert((State =	ST1)->(next
A eUUTaaset 3 A eUUTaaset 4 A eUUTaaset 4 A eUUTaaset 5 A eUUT	Add Liet	30	nected Objects	on	6		. 0	OB	0B	0 ps	0	on assert(always i	State = ST2
A. oUUTaset Anulus Operation Grange 0 0 4 1 008 088 5000 ps 5 on asserd(avay) (State - ST) A. oUUTaset Analysis Very Survey 200 0 <t< td=""><td>A :e:UUT:assert_ST</td><td>긴면</td><td>olection Region</td><td>on</td><td></td><td>I 6</td><td>. 0</td><td>OB</td><td>80B</td><td>110000 ps</td><td>4</td><td>on assert(always (</td><td>(State = ST2</td></t<>	A :e:UUT:assert_ST	긴면	olection Region	on		I 6	. 0	OB	80B	110000 ps	4	on assert(always ((State = ST2
A eUUTaisef S	A recourtassent_St Add bog	10	bjects in Design	on		1 4	1	80B	80B	90000 ps	5	on assert(always {	State = ST1
A vr.dorr.ed.co. Problem FBL on 1 3 0 06 Disble AV FBL on asset@away(Sale - Sale AV X redUffiguret Falle FBL on 0 <td>A :e:UUT:assert_St</td> <td>1</td> <td>PSL</td> <td>on</td> <td>6</td> <td></td> <td>. 0</td> <td>OB</td> <td>08</td> <td>0 ps</td> <td>0</td> <td>on assert(always {</td> <td>State = ST1</td>	A :e:UUT:assert_St	1	PSL	on	6		. 0	OB	08	0 ps	0	on assert(always {	State = ST1
A. woldfrighter Pell on 0	A :e:UUT:st0_to_		PSL	on	1	. 3	0	OB	808	10000 ps	4	on assert(always (State - STO
A veUUTaiset ov View VIV. PSL on 0 8 0 08 8008 70000 ps 33 on asset@awaya (Sade - Still Configure. FSL on 0 5 0 08 3208 50000 ps 20 on asset@awaya (Sade - Still Configure. FBL on 0 5 0 08 3208 50000 ps 20 on asset@awaya (Sade - Still Configure. FBL on 0 5 0 08 3208 50000 ps 20 on asset@awaya (Sade - Still FBL on 0 5 0 000 ps 20 on asset@awaya (Sade - Still FBL on 0 5 0 000 ps 20 on asset@awaya (Sade - Still FBL on 0 5 0 000 ps 20 on asset@awaya (Sade - Still FBL on 0 5 0 000 ps 20 on asset@awaya (Sade - Still FBL on 0 5 0 000 ps 20 on asset@awaya (Sade - Still FBL on 0 5 0 000	A :e:UUT:Is_done	t	PSL	on	6		0	OB	08	0 ps	0	on assert(always i	ready_o)->(
Configure. C	A :e:UUT:assert_ou View ATV	1	PSL	on	6	1 6	. 0	OB	800B	70000 ps	33	on assert(always	State = ST0
Configure - Existence - Existence - Exist	A :e:UUT:assert_ou Report		PSL	on	(1 5	0	08	320B	90000 ps	20	on assert(always	State - ST1]
Compare Senade Failure Log Failure Log Failure Action F Start Action F Filter F Filter F Equard F Chipley Cellor F Comproves F Comproves F													
Prante Log Prais Log	Contigure												
Analyzis A ats Ditley Celement Vere	✓Enable												
Pass tog Falure Action P Falure Action P Start Action P Antalysis Attain P Falar P Expand P Analysis Attain P Expand P	✓ Failure Log												
Falue Addon F Stat Action F Stat Action F Stat Action F Analysis F Analysis F Cyand F Cyand F Cyand F Covergroups Wave \$\$ Browser Covergroups Wave \$\$ Browser	Pass Log												
Park Action Park Action Start Action Park Action Analysis Park Action Rar Park Action Equand Park Action Analysis Action action Analysis Action action Childley Caleboard Park Action	Failure Action	P											
Start Action Test Analysis Analysis Analysis Analysis Civiand	Pass Action	P											
Analysis Arts Ofbiely Options > Analysis Arts Ofbiely Options > Country of the country of th	Start Action												
Test Analysis Milliopat Hint Para Perand Perand Per	Antecedent Act	ion 🕨											
Analyzis A est Displey Collons Covering roups Covering roups Cover													
Analysis A ass Dipley point Coveragroups Wave Status Dipley point Coveragroups Wave Message Viewer Image	Test Analysis	P											
Filter F Expand F Analyzis Artalyzis Chipley Cyblone Expande Expande Expande Displey Cyblone Expande Expande Expande Expande Expande	XML Import Hin	t i											
Expand F Analysis A.Ass Display Options Conception Conceptions Conception Conceptions	Filter	•											
Analysis 🛦 Ass Display Options Name Covergroups 🔲 Wave 🗰 Browser 🌊 Message Viewer	Expand	4											
Evelution Selected	Analysis	; ►	2. Coverarouns	v wave	- 😫 Brouster		Viewer ×						
	Exclude Selecte	ed	C sousidioabs	- merinare	0104361	1 message							

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

		Que	sta Sim-64 10.6	<u>i</u> 1									-	· · 8
leip														
22 H 🗶 🕅 🕓	j 🖪 - 🛧 🤆 -	<u>k</u> 100	ps 🗧 😫 😫	8 🕅 🐵	🗶 🚺 🛄 🕯	👂 😵 🔞	🕴 🕺 😤	余 🔛	💶 😙 Threshold	100 🗧 🛱	\prec ? x *			
🔍 🐴 🤣 🐴	🥥 🛛 🖂 🛛	u 🖭 🚈 🥕	■際家						Search:	V 4			6 13 %	
,														
Acceptions -												_		
Name		Assertion Type	Language	Enable	Failure Count	Pass Count	Active Count	Memory	Peak Memory Pea	ak Memory Time	umulative Threads	ATV	Assertion Exp	ession
A :e:UUT:assertdid_:	start	Concurrent	PSL	on	(1	0	0B	80B	30000 ps	1	on	assert(ahvays	(rst)->(eventi
🔺 :e:UUT:assert_ST0	Add Wave		PSL	on	0	1 6	0	0B	320B	70000 ps	6	on	assert(always	(State = ST0)
⊢▲ :e:UUT:assertrsttc	to I bha		PSL	on	c	1 1	0	0B	80B	30000 ps	1	on	assert(ahvays	(rst)->(State
HA :e:UUT:assert_0	Add Lon		PSL	on	9	0	0	0B	60B	10000 ps	1	on	assert((State =	ST1)->(next
HA :e:UUT:assert_ST2	Add Log	· ·	PSL	on	0		0	OB	OB	0 ps	0	on	assert(always	(State = ST2
A :e:UUT:assert_ST2	View Source		PSL	on	L.	u u	U	UB	80B	110000 ps	4	on	assert(atways	{[State = S12
HA :e:UUT:assert_ST1	Enable ATV		PSL	on	0	4	1	80B	80B	90000 ps	5	on	assert(ahvays	{State = ST1
HA :e:UUT:assert_ST1	Vicus ATM		PSL	on			U	UB	UB	U ps	0	on	assert(always	{State = ST1
A :e:00T:st0_to_s	*16W G1 7		PSL	on			0	UB	OUB	ruuuu ps	4	on	assentarivays	(State = 510
HA :e:UUT:Is_done	нероп		PSL	on	L. L.			UB	UB	U ps	0	on	assert(arways	(ready_o)->(
A reductassen_out	Configura		PSL	on				00	000B	70000 ps	33	on	asserijanvays	(State = STU
FA :e:001:assen_out	Conligure		PSL	on		, a		0B	320B	annin ba	20	on	assen(arways	(State = 511)
	✓Enable													
	✓Failure Log													
	Failure Action	•												
		P												
	Start Action	P												
	Antecedent Act	ion 🕨												
	Test Analysis	•												
	XMI Import Hin													
	Filter	· .												
	Expand	P.												
	Display Option:	s 🍺												
	Evolude Select	94												1
Analysis 🛛 🛦 Asser	Evelude Select			× 🙀 Wave :	< 🚼 Browser	🛛 🚊 Message								K ()
	Exclude with C	ommente												
	ClearExclusion													

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



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

Ques	ta Sim-64 10.6_1						- * 8
	PP 🗧 🖺 🖥 🤷 🕴	: 🔟 🔟 🔶 🗍 💲	8 🖇 💲 🕭	🏦 🕹 🏛 🛨 🗇	ireshold 100 🗧 j	ā ≺ ? <i>x</i> +	
	■腹壁皿== 1			🛠 📴 Search:	V	89.90 0 0 0 8 4	
AssertionThreadView - :e:UUT:c_4-10000.0							
	cover((((State = ST0)*1	to inf];State = ST1};S	tate = ST2);State = STC				É
Cover Co		600 m					
Design Objects	STO	STO	STO	STO	ST1	ST2 ST0	2 2
Analysis 🛦 Assertions 🔺 Cover Directives	Covergroups to Wave	Browser	1essage Viewer	re:UUT-assert ST0toST	1 3 eutra o	73 eUUTc 4	
	The name of		and a second sec				

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

Eile Edit View Compile Simulate Add Structure Tools Layo	ut Bookmarks <u>W</u> indow <u>H</u> elp				
🖹 • 😅 🖬 🌣 🍜 3 🏁 🎕 💭 🖄 🕼 🔛 🗏 🖽	əlp 📃 🥻 🐐 🖏 🖉 🖉 🖉	🖥 🕄 - 🛊 🗰 🛉 🛛 💱 🛛 1000 no 🕇	: El El El 🕺 🤹 🗵 🔛 🖑	1 @ 1 L-@ 1	
표 ± ∓ Threshold 100 \$ 1 < ? X ⇒ Layo	It Coverage ColumnLayout Alle	1Columns 💌	- 🚱 🗳 🗳 🖓 🖞	न गा गर्न 🗈 👘 👎 म म्ह ज म्ह	2.5.5
3+ - +€ - ∰+ Search: 💌 🍂 🎄 🕸	R R A & 2 2 2 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
🔊 sim (Recursive Coverage Aggregation) - Default 🗆 🗰 🖬 🗙	📵 stails 🔤 🛨 🛋 🗙 📊 Wave - Default				+ # ×
▼ Instance		Msgs			
	A AUTOK				
UUT	A AUTOPESET				
🖲 📕 TB	A AULTOR				
🧮 standard		22222222			
📕 konstandid	• e-UUT-PD	1			
📕 textio		11			
std_logic_1164		110			
attributes	• • ellittas op	00			
sto_logic_misc	a de littria	77777777			
sid_logic_antri	:e:UUT:N 1	0			
minu shiff	:e:UUT:N_2	1			
	e:uut:n 3	0			
	RUUTN 4	0			
	:e:UUT:N 5	0			
	:e:UUT:N 6	1			
	:e:UUT:N_7	0			
	:e:UUT:N_8	0			
	:e:UUT:N_9	0			
	:e:UUT:N_10	1			
	:e:UUT:N_11	0			
	:e:UUT:N_12	0			
	:e:UUT:RD_DUM	1			
	:e:UUT:DATA_BL	. 0			
	🗉 🌧 :e:UUT:DATA_BL	110			
	:e:UUT:DATA_BL	. 1			
	A R C Now	925000 ps	200 ps 400 ps	600 ps	100 as
	🔓 🥕 🤤 🛛 Cursor 1	0 ps Dps			
Library × 🖹 Files × 💽 Instance × 🔛 Project × 🙀 sim × 🚺	S Analysis 🛛 🛆 Assertio	ons 🛛 🗚 Cover Directives 🗶 🔀 Cove	ergroups 🛛 📰 Wave 🛛		4.>
A Transcript					+ d' ×
Time: 925 ns Iteration: 0 Region: :e:TB					
add wave 'r sim/:e:UUT:*					
and wave 'r sim/:e:00T:"					
VSIM 6>					E
	Project : veri_lab Now: 925 ns Delta: 1	e:UUT:RD_DUMMY			

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