SYNOPSYS®

VC Formal Lab
Formal Property Verification (FPV) App Formal Core Coverage Merge Setup and Standard Usage

Learning Objectives

In this VC Formal lab, you will use a traffic light controller example to learn to do the following:

- Review setup and run simulation using VCS to Generate Coverage Database
- Modify FPV VC Formal FPV Tcl script file to enable formal core coverage
- Add commands to run and report formal core result
- Add commands to run and compute formal core in coverage
- Review and run shell script to invoke VC Formal to generate formal core coverage
- Review and run shell script to merge formal core coverage with simulation coverage
- Review and run shell script to view simulation only, formal core only and merged database in Verdi GUI

Familiarity with the SystemVerilog Assertion (SVA) language and completion of FPV/FPV_General lab. Knowledge of VC Formal, VC Formal FPV and VC Formal FCA app are required for this lab.

Lab Duration: 20 minutes

Files Location

All files for this VC Formal lab are in directory:
$VC_STATIC_HOME/doc/vcst/examples/FPV/Formal_Core_Cov_Merge

Directory Structure	
Formal_Core_Cov_Merge	Lab main directory
README_VCFormal_Formal_Core_Cov_Merge.pdf	Lab instructions
design/	Verilog RTL code of the Device Under Test (DUT)
sva/	SVA properties to check functionality of the DUT
run/	Run directory
solution/	Solution directory

Resources

The following resources are available for in-depth guidance regarding VC Formal usage, commands, and variables.

VC Formal User Guide:
$VC_STATIC_HOME/doc/vcst/VC_Formal_Docs/VC_Formal_UG.pdf

VC Formal Apps Quick References Guides:
$VC_STATIC_HOME/doc/vcst/VC_Formal_Docs/Quick_Reference_Guides/

VC Formal Apps Tcl Templates:
$VC_STATIC_HOME/doc/vcst/VC_Formal_Docs/Quick_Reference_Guides/vcf_tcl_templates/

Prepare your Environment

1. Set environment variable pointing to your VC Formal and VCS installation directory:

```
%setenv VC_STATIC_HOME /tools/synopsys/vcstatic
%setenv VCS_HOME /tools/synopsys/vcs-mx
%setenv VERDI_HOME /tools/synopsys/verdi
```

2. Add path $VC_STATIC_HOME/bin, $VCS_HOME/bin and $VERDI_HOME/bin to the PATH environment variable.

3. Change your working directory to Formal_Core_Cov_Merge/run:

```
%cd Formal_Core_Cov_Merge/run
```

Now you are ready to begin the lab.

View Setup and Run Simulation to Generate Coverage Database

This lab is an extension to the FPV lab. It involves running simulation and generate coverage database, as well as running VC Formal to generate formal cove coverage data and save to coverage data base, and review the database generated by simulation alone, formal alone, and merged database. In this step, you will review the simulation testbench code and the shell script with commands used to run simulation and generate coverage database.

4. View files used in simulation setup.

```
%view ../design/sim_filelist
```

5. View testbench code used in simulation.

```
%view ../design/testbench.sv
```

6. View shell script to run simulation.

```
%view sim_cov.csh
```

7. Run simulation.

```
% sim_cov.csh
```

You should see text files in sim_alone directory. You can check the coverage information in those *.txt files. But don't dwell too much on this. We will show the same coverage later in the Verdi GUI. It will be much easier to get a more comprehensive report from the GUI view.

Modify FPV VC Formal FPV Tcl script file to enable formal core coverage

8. Open run.tcl with any text editor

```
%vi -f run.tcl
```

9. Add command below the comment to add variable enable reset property check. This will enable code under reset to be included in coverage

```
set_fml_var fml_reset_property_check true
```

10. Modify design compile command to add options to include line, condition and toggle coverage.

```
read_file -top $design -format verilog -cov line+cond+tgl \
-sva -vcs {-f ../design/filelist}
```

Note: For design compile using analyze/elaborate, the equivalent command is

```
analyze -format verilog -vcs {-f ../design/filelist}
elaborate $design -cov line+cond+tgl -sva
```

Add commands to run and report formal core result

11. Add command below the comment to compute formal core for proven property in block mode.

```
compute_formal_core -property [get_props -status proven] \
-block
```

12. Add command below the comment to report formal core run results.

```
report_formal_core
```

Add commands to run and compute formal core in coverage

13. Add command below the comment to compute formal core coverage

```
compute_formal_core_coverage -property [get_props -status
proven] -par_task FPV -block
```

14. Add command below the comment to save formal core results to coverage database

```
save_formal_core_results -db_name formal_core_cov
```

15. Add command below the comment to save uncoverable coverage goals in exclusion file

```
save_cov_exclusion -file uncoverables.el
```

Review and run shell script to invoke VC Formal to generate formal core coverage and merge simulation and formal

16. View shell script to run formal core, generate coverage database, and generate a merged database with simulation.

```
%view fc_cov.csh
```

17. Run shell scrip to invoke VC formal formal core to generate coverage and VCS URG to merge with simulation

```
% fc_cov.csh
```

Review and run shell script to view simulation only, formal core only and merged database in Verdi GUI

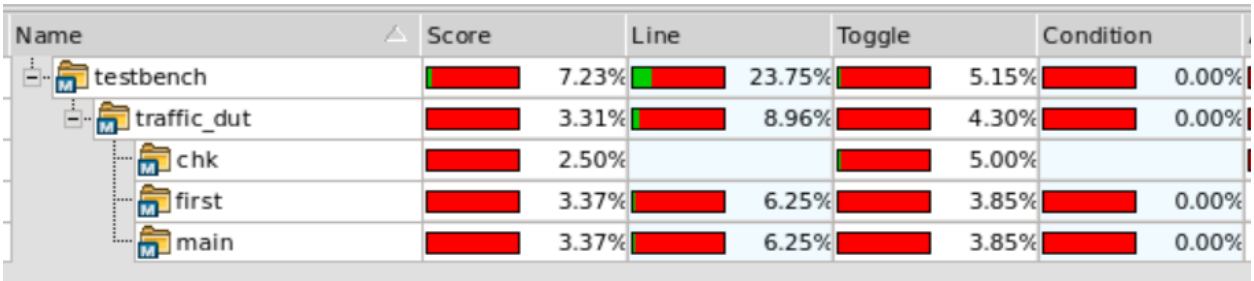
18. View shell script to run Verdi GUI to view simulation only, formal core only and merged database. This script will invoke three verdi sessions in parallel to view simulation coverage, formal core coverage and merged coverage results:

```
%view view_verdi_cov.csh
```

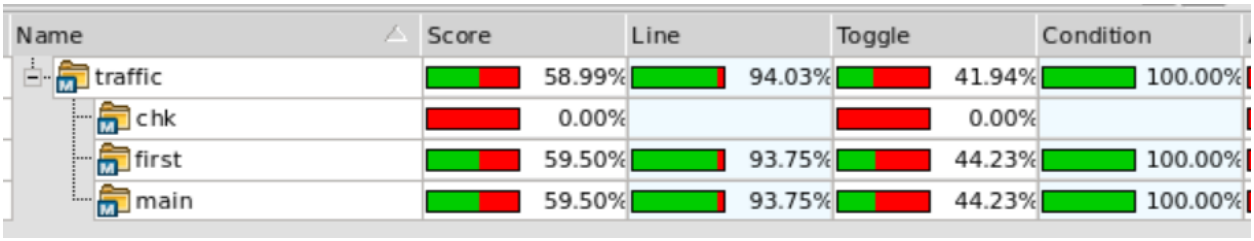
19. Run shell script to invoke Verdi GUI that show 3 GUI views for simulation only, formal core only and merged and observe the results.

```
% view_verdi_cov.csh
```

Simulation only (simv.vdb)

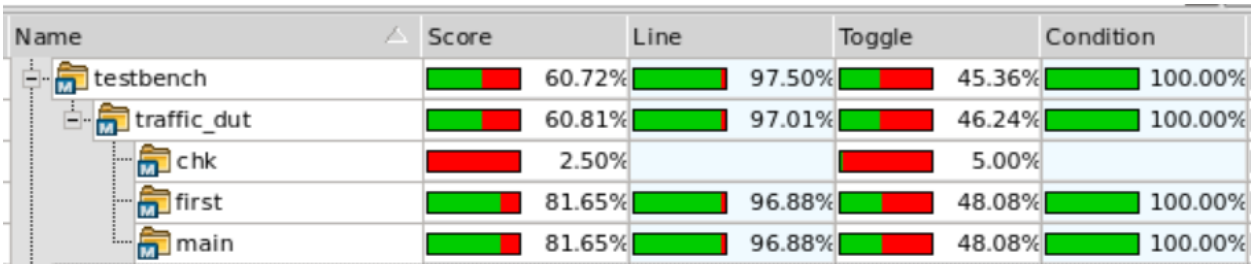


Formal core only (formal_core_cov.vdb)



Name	Score	Line	Toggle	Condition
traffic	58.99%	94.03%	41.94%	100.00%
chk	0.00%		0.00%	
first	59.50%	93.75%	44.23%	100.00%
main	59.50%	93.75%	44.23%	100.00%

Merged database (Merged_db.vdb)



Name	Score	Line	Toggle	Condition
testbench	60.72%	97.50%	45.36%	100.00%
traffic_dut	60.81%	97.01%	46.24%	100.00%
chk	2.50%		5.00%	
first	81.65%	96.88%	48.08%	100.00%
main	81.65%	96.88%	48.08%	100.00%