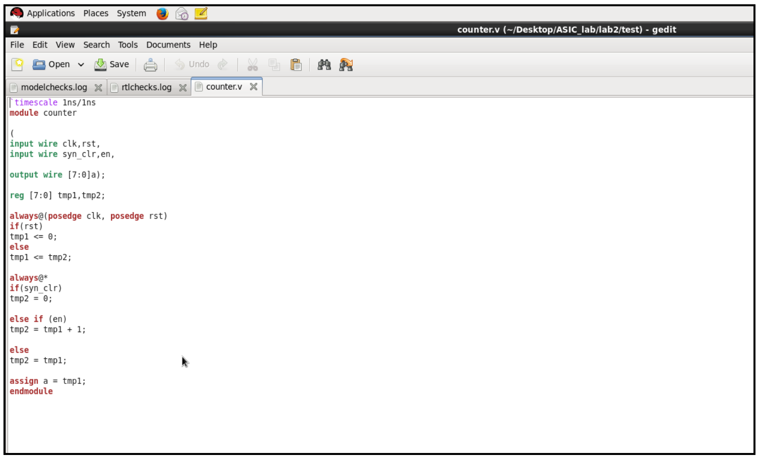
ASIC Lab - Linting
EC 6272, Spring 2019-2020
VLSI Laboratory, ECE Department, NIT Rourkela, India

This tutorial will discuss the various views that make-up a effective RTL design model and then illustrate how to use a Cadence ASIC linting tool to check an RTL design in conjunction with simulation, and before synthesizing RTL model.
This tutorial requires entering commands manually for each of the tools to enable students to gain a better understanding of the detailed steps involved in this process.

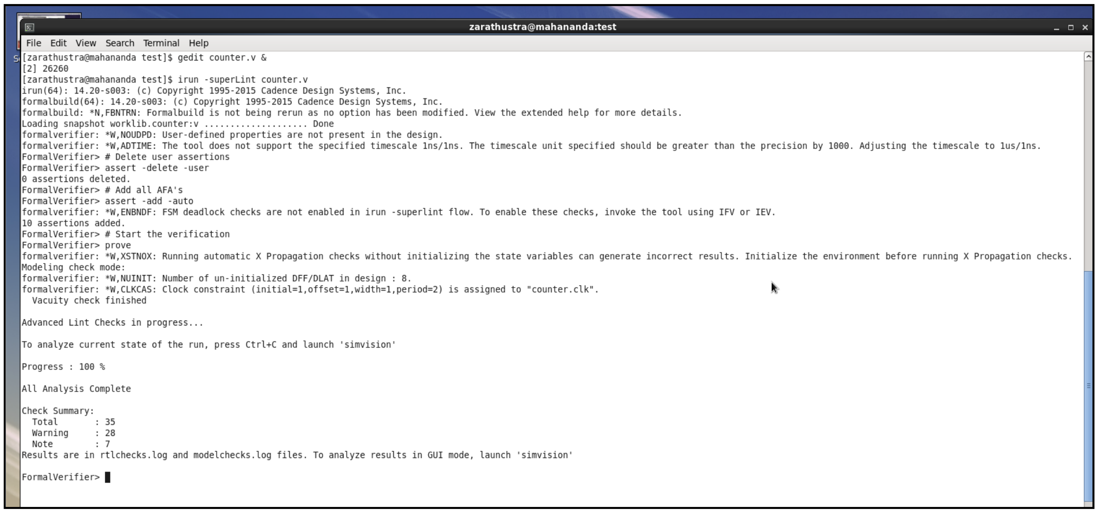
1. Open the terminal
2. Source the cadence.cshrc
3. In a new ASIC lab directory, write a Verilog coded design (in this example counter.v)



```
`timescale 1ns/1ns
module counter

(
input wire clk,rst,
input wire syn_clr,en,

output wire [7:0]a);

reg [7:0] tmp1,tmp2;

always@(posedge clk, posedge rst)
if(rst)
tmp1 <= 0;
else
tmp1 <= tmp2;

always@*
if(syn_clr)
tmp2 = 0;

else if (en)
tmp2 = tmp1 + 1;

else
tmp2 = tmp1;

assign a = tmp1;
endmodule
```

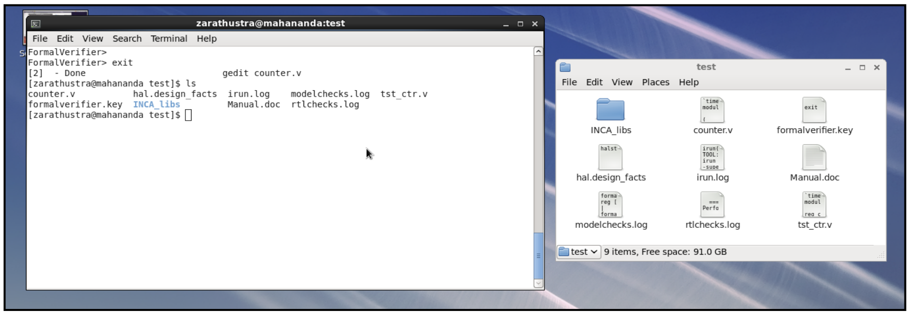
4. Write the linting test syntax as shown below with your design code.

test]$ irun –superLint counter.v



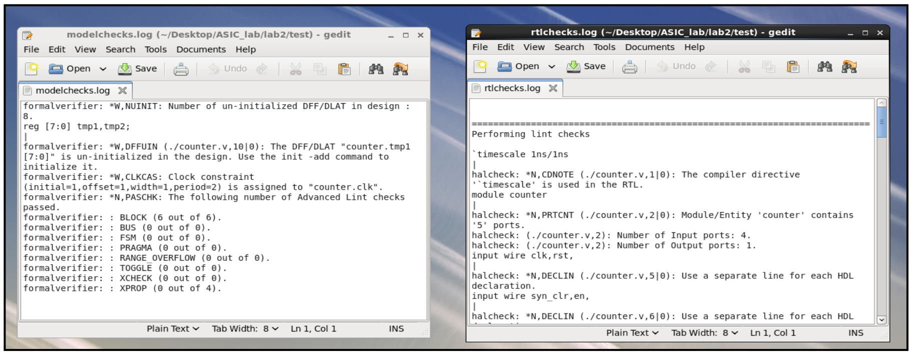
```
[zarathustra@mahananda test]$ gedit counter.v &
[2] 26260
[zarathustra@mahananda test]$ irun -superLint counter.v
irun(64): 14.20-s003: (c) Copyright 1995-2015 Cadence Design Systems, Inc.
formalbuild(64): 14.20-s003: (c) Copyright 1995-2015 Cadence Design Systems, Inc.
formalbuild: *N,FBNTRN: Formalbuild is not being rerun as no option has been modified. View the extended help for more details.
Loading snapshot worklib.counter:v .................... Done
formalverifier: *W,NOUDPD: User-defined properties are not present in the design.
formalverifier: *W,ADTIME: The tool does not support the specified timescale 1ns/1ns. The timescale unit specified should be greater than the precision by 1000. Adjusting the timescale to 1us/1ns.
FormalVerifier> # Delete user assertions
FormalVerifier> assert -delete -user
0 assertions deleted.
FormalVerifier> # Add all AFA's
FormalVerifier> assert -add -auto
formalverifier: *W,ENBNDF: FSM deadlock checks are not enabled in irun -superlint flow. To enable these checks, invoke the tool using IFV or IEV.
10 assertions added.
FormalVerifier> # Start the verification
FormalVerifier> prove
formalverifier: *W,XSTNOX: Running automatic X Propagation checks without initializing the state variables can generate incorrect results. Initialize the environment before running X Propagation checks.
Modeling check mode:
formalverifier: *W,NUINIT: Number of un-initialized DFF/DLAT in design : 8.
formalverifier: *W,CLKCAS: Clock constraint (initial=1,offset=1,width=1,period=2) is assigned to "counter.clk".
  Vacuity check finished

Advanced Lint Checks in progress...

To analyze current state of the run, press Ctrl+C and launch 'simvision'

Progress : 100 %

All Analysis Complete

Check Summary:
  Total       : 35
  Warning     : 28
  Note        : 7
Results are in rtlchecks.log and modelchecks.log files. To analyze results in GUI mode, launch 'simvision'

FormalVerifier> █
```

5. Exit from the FormalVerifier tool and Check the folder contents. It will have few log files, which we have to check for improving our coding style.


zarathustra@mahananda:test
File Edit View Search Terminal Help
test
File Edit View Places Help
INCA_libs
counter.v
formalverifier.key
hal.design_facts
irun.log
Manual.doc
modelchecks.log
rtlchecks.log
tst_ctr.v
test
9 items, Free space: 91.0 GB

```
FormalVerifier>
FormalVerifier> exit
[2]  - Done                       gedit counter.v
[zarathustra@mahananda test]$ ls
counter.v          hal.design_facts  irun.log    modelchecks.log  tst_ctr.v
formalverifier.key  INCA_libs         Manual.doc  rtlchecks.log
[zarathustra@mahananda test]$ 
```

6. Check the file rtlchecks.log and modelchecks.log. We have to figure out the issues in the design code.


modelchecks.log (~/Desktop/ASIC_lab/lab2/test) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
modelchecks.log
Plain Text
Tab Width: 8
Ln 1, Col 1
INS
rtlchecks.log (~/Desktop/ASIC_lab/lab2/test) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
rtlchecks.log
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

```
formalverifier: *W,NUINIT: Number of un-initialized DFF/DLAT in design :
8.
reg [7:0] tmp1,tmp2;
|
formalverifier: *W,DFFUIN (./counter.v,10|0): The DFF/DLAT "counter.tmp1
[7:0]" is un-initialized in the design. Use the init -add command to
initialize it.
formalverifier: *W,CLKCAS: Clock constraint
(initial=1,offset=1,width=1,period=2) is assigned to "counter.clk".
formalverifier: *N,PASCHK: The following number of Advanced Lint checks
passed.
formalverifier: : BLOCK (6 out of 6).
formalverifier: : BUS (0 out of 0).
formalverifier: : FSM (0 out of 0).
formalverifier: : PRAGMA (0 out of 0).
formalverifier: : RANGE_OVERFLOW (0 out of 0).
formalverifier: : TOGGLE (0 out of 0).
formalverifier: : XCHECK (0 out of 0).
formalverifier: : XPROP (0 out of 4).
```

```
==========================================================================
Performing lint checks

`timescale 1ns/1ns
|
halcheck: *N,CDNOTE (./counter.v,1|0): The compiler directive
'`timescale' is used in the RTL.
module counter
|
halcheck: *N,PRTCNT (./counter.v,2|0): Module/Entity 'counter' contains
'5' ports.
halcheck: (./counter.v,2): Number of Input ports: 4.
halcheck: (./counter.v,2): Number of Output ports: 1.
input wire clk,rst,
|
halcheck: *N,DECLIN (./counter.v,5|0): Use a separate line for each HDL
declaration.
input wire syn_clr,en,
|
halcheck: *N,DECLIN (./counter.v,6|0): Use a separate line for each HDL
```

7. Verify the reason behind the warnings and errors if any, and analyze the code accordingly.

8. Follow the first tutorial for post-linting synthesis and simulation