

Basic VCS practical

Renesas Design Vietnam Co., Ltd.
HW Design Engineering Division
Rev. 0.00

Outline

Exercise Description

Data Preparation

Set Local and LSF Environment

Exercise 1

Exercise 2

Exercise 3

Using VCS with LSF



Exercise Description

The purpose of this practice is guided user how to simulate a HW design in order to watch the result by using VCS.

User can verify the result by reading text in command line mode or by watching waveform in GUI mode.

Note: The design is a simple counter. It has two input signals: clock signal (`clk`), reset signal (`rst_n`) and one output signal counter value (`cnt_value`)

The operation: this counter will increase one value when clock signal (`clk`) is **1** and reset signal (`rst_n`) is **1**. Moreover, this counter will reset counter value (`cnt_value`) to **0** when reset signal (`rst_n`) is **0** regardless of clock signal.



Data Preparation

Open the terminal

At home directory, make a training directory:

```
$ mkdir vcs_training
```

Change to created directory

```
$ cd vcs_training
```

Copy “counter.v” from DMS location

```
Documents/1. General Documents/010_ENG/030_Training/EDA Tool Training/Basic Training/VCS Practice/counter.v
```



Set Local and VCS Environment

*Open connection in client machine (this step is used to display GUI later)
(This step must be done when user is in the local machine (user PC))*

```
$ xhost +
```

Set VCS environment

```
$ source /common/appl/Env/Synopsys/vcs-mx_vD-2009.12-4
```

```
$ setenv LM_LICENSE_FILE 27000@licedu1
```

Set the path for g++

```
$ setenv PATH /common/appl/gcc-3.3.6/bin/:$PATH
```

or (depending on your PC compatibility)

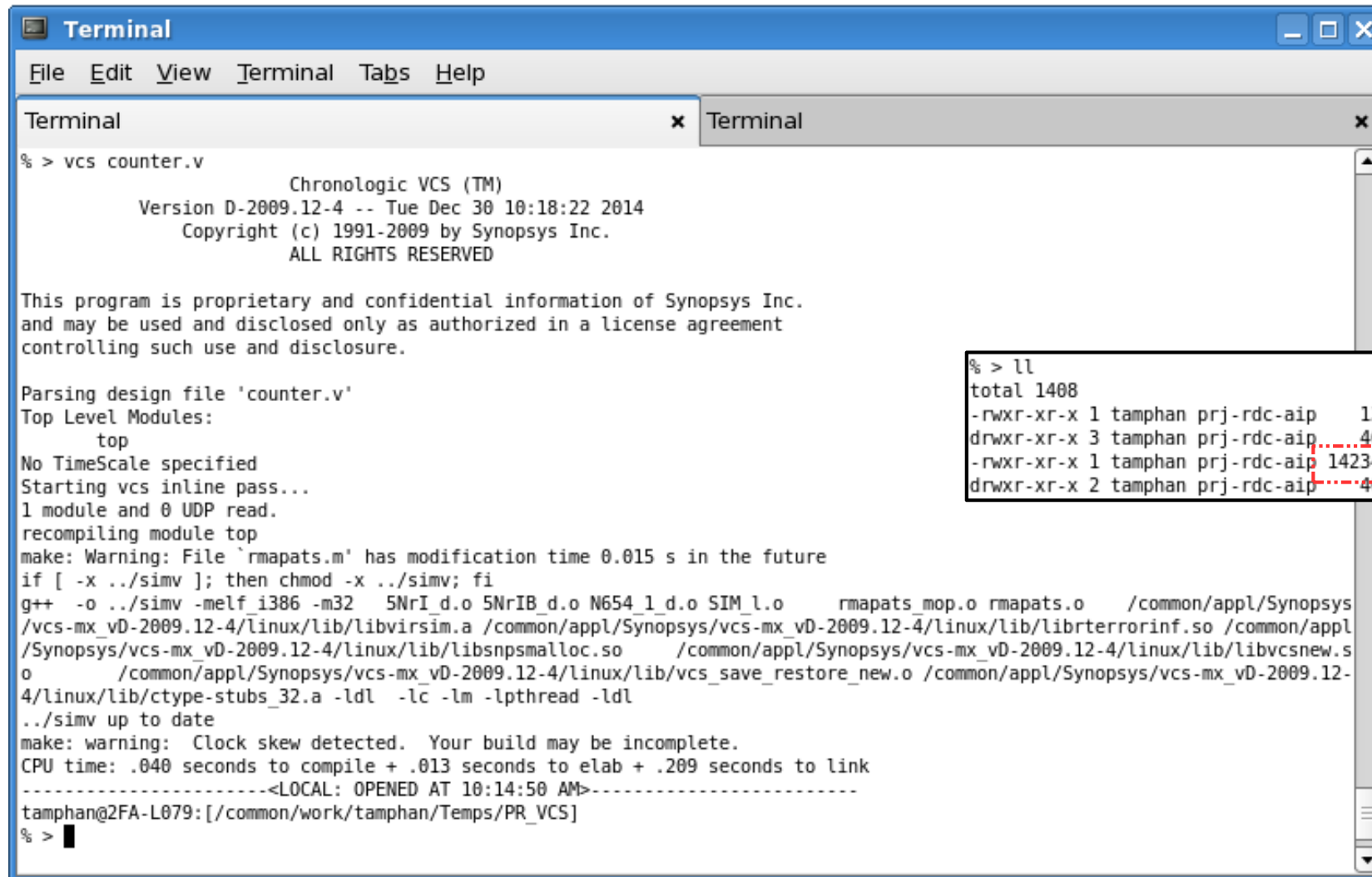
```
$ setenv PATH /common/appl/gcc-3.2.3/bin/:$PATH
```



Exercise 1. Compile and Simulate Design (1/2)

Compile Design

\$ vcs counter.v



```
Terminal
File Edit View Terminal Tabs Help

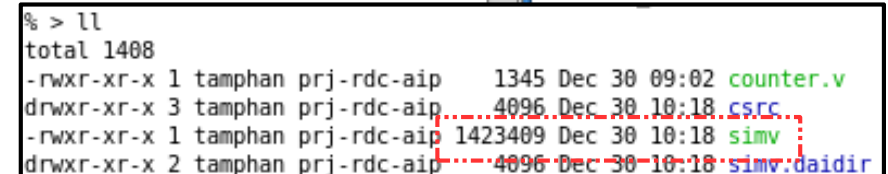
Terminal
% > vcs counter.v

Chronologic VCS (TM)
Version D-2009.12-4 -- Tue Dec 30 10:18:22 2014
Copyright (c) 1991-2009 by Synopsys Inc.
ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Parsing design file 'counter.v'
Top Level Modules:
    top
No TimeScale specified
Starting vcs inline pass...
1 module and 0 UDP read.
recompiling module top
make: Warning: File 'rmapats.m' has modification time 0.015 s in the future
if [ -x ../simv ]; then chmod -x ../simv; fi
g++ -o ../simv -melf_i386 -m32 5NrI_d.o 5NrIB_d.o N654_1_d.o SIM_l.o rmapats_mop.o rmapats.o /common/appl/Synopsys
/vcs-mx_vD-2009.12-4/linux/lib/libvirsim.a /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/librterrorinf.so /common/appl
/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/libsnpsmalloc.so /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/libvcsnew.s
o /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/vcs_save_restore_new.o /common/appl/Synopsys/vcs-mx_vD-2009.12-
4/linux/lib/ctype-stubs_32.a -ldl -lc -lm -lpthread -ldl
../simv up to date
make: warning: Clock skew detected. Your build may be incomplete.
CPU time: .040 seconds to compile + .013 seconds to elab + .209 seconds to link
-----<LOCAL: OPENED AT 10:14:50 AM>-----
tamphan@2FA-L079: [/common/work/tamphan/Temps/PR_VCS]
% >
```

If you compile the source file *successfully*, then you will receive the file **simv**. Please use the command “ls” to check the existence of this file.

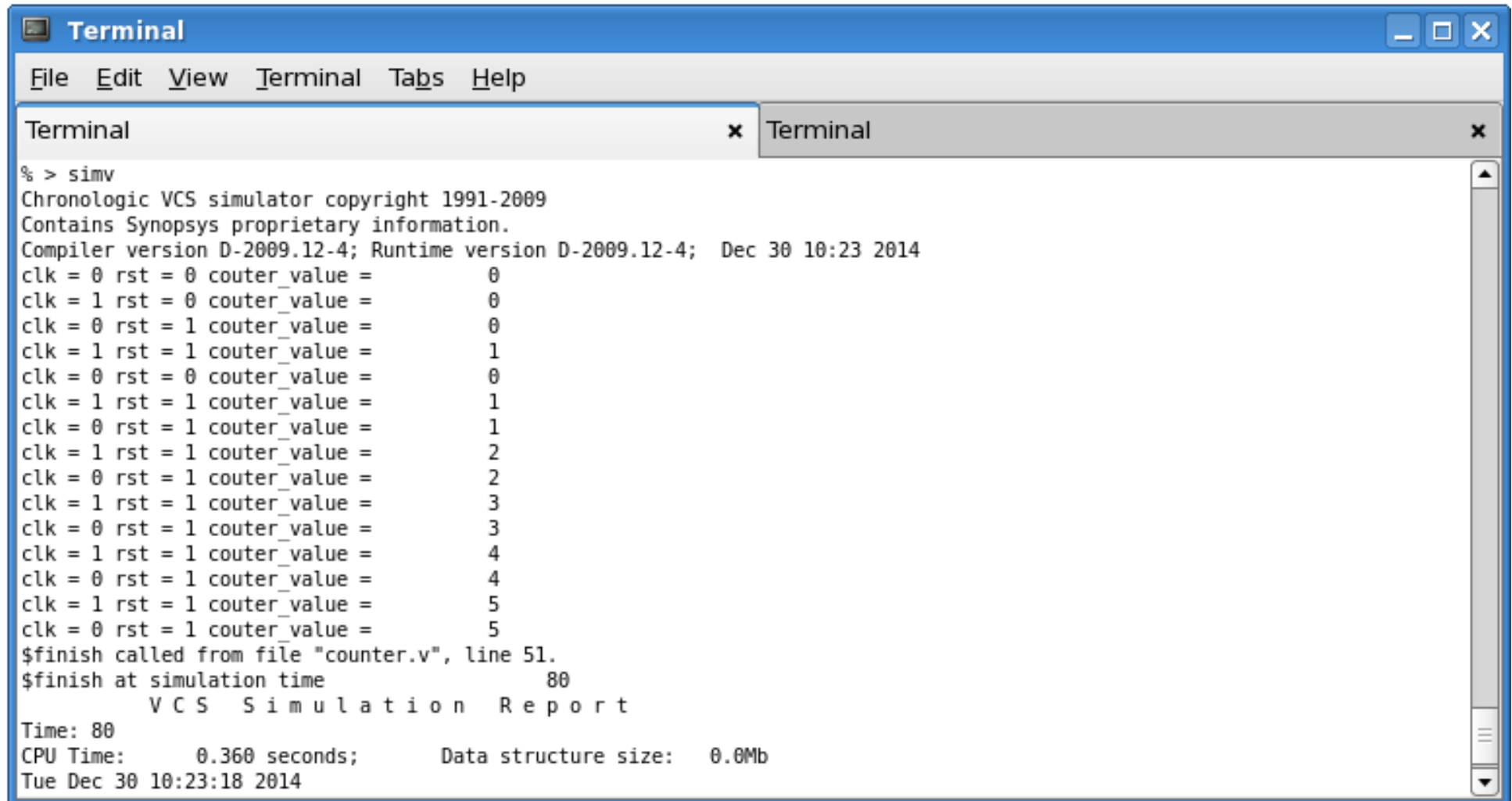


```
% > ll
total 1408
-rwxr-xr-x 1 tamphan prj-rdc-aip 1345 Dec 30 09:02 counter.v
drwxr-xr-x 3 tamphan prj-rdc-aip 4096 Dec 30 10:18 csrcc
-rwxr-xr-x 1 tamphan prj-rdc-aip 1423409 Dec 30 10:18 simv
drwxr-xr-x 2 tamphan prj-rdc-aip 4096 Dec 30 10:18 simv.daidir
```

Exercise 1. Compile and Simulate Design (2/2)

Simulate Design

\$ simv

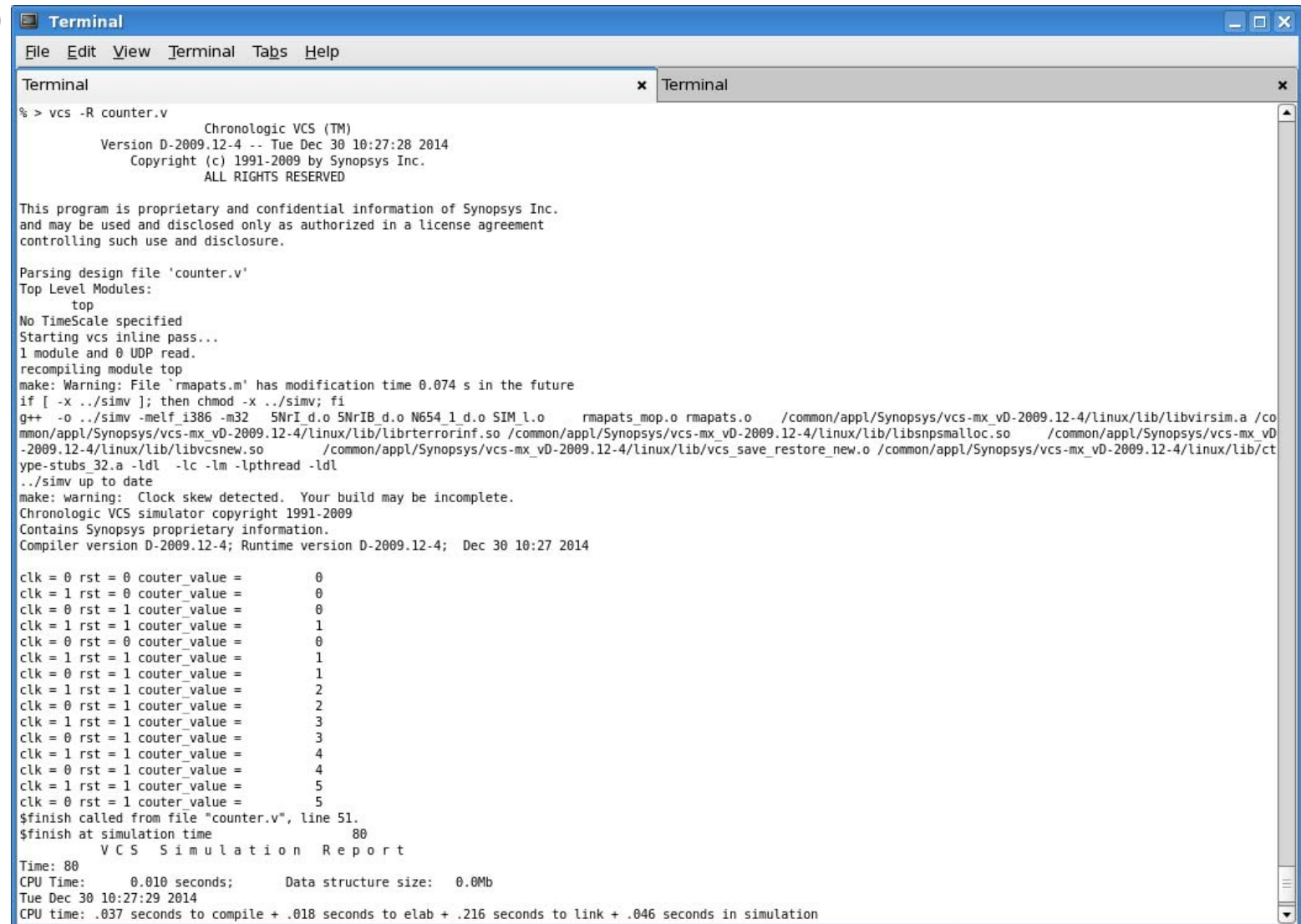


```
% > simv
Chronologic VCS simulator copyright 1991-2009
Contains Synopsys proprietary information.
Compiler version D-2009.12-4; Runtime version D-2009.12-4; Dec 30 10:23 2014
clk = 0 rst = 0 couter_value =      0
clk = 1 rst = 0 couter_value =      0
clk = 0 rst = 1 couter_value =      0
clk = 1 rst = 1 couter_value =      1
clk = 0 rst = 0 couter_value =      0
clk = 1 rst = 1 couter_value =      1
clk = 0 rst = 1 couter_value =      1
clk = 1 rst = 1 couter_value =      2
clk = 0 rst = 1 couter_value =      2
clk = 1 rst = 1 couter_value =      3
clk = 0 rst = 1 couter_value =      3
clk = 1 rst = 1 couter_value =      4
clk = 0 rst = 1 couter_value =      4
clk = 1 rst = 1 couter_value =      5
clk = 0 rst = 1 couter_value =      5
$finish called from file "counter.v", line 51.
$finish at simulation time          80
      V C S   S i m u l a t i o n   R e p o r t
Time: 80
CPU Time:      0.360 seconds;      Data structure size:  0.0Mb
Tue Dec 30 10:23:18 2014
```

Exercise 2. Compile and Simulate with options

Compile and Simulate Design with option -R (with this option user can compile and simulate design with one command)

\$ vcs -R counter.v



```
Terminal
File Edit View Terminal Tabs Help

Terminal
% > vcs -R counter.v
Chronologic VCS (TM)
Version D-2009.12-4 -- Tue Dec 30 10:27:28 2014
Copyright (c) 1991-2009 by Synopsys Inc.
ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Parsing design file 'counter.v'
Top Level Modules:
    top
No TimeScale specified
Starting vcs inline pass...
1 module and 0 UDP read.
recompiling module top
make: Warning: File 'rmapats.m' has modification time 0.074 s in the future
if [ -x ../simv ]; then chmod -x ../simv; fi
g++ -o ../simv -melf_i386 -m32 5NrI_d.o 5NrIB_d.o N654_1_d.o SIM_l.o rmapats_mop.o rmapats.o /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/libvirsim.a /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/librterrorinf.so /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/libsnpsmalloc.so /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/libvcsnew.so /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/vcs_save_restore_new.o /common/appl/Synopsys/vcs-mx_vD-2009.12-4/linux/lib/ctype-stubs_32.a -ldl -lc -lm -lpthread -ldl
../simv up to date
make: warning: Clock skew detected. Your build may be incomplete.
Chronologic VCS simulator copyright 1991-2009
Contains Synopsys proprietary information.
Compiler version D-2009.12-4; Runtime version D-2009.12-4; Dec 30 10:27 2014

clk = 0 rst = 0 couter_value =      0
clk = 1 rst = 0 couter_value =      0
clk = 0 rst = 1 couter_value =      0
clk = 1 rst = 1 couter_value =      1
clk = 0 rst = 0 couter_value =      0
clk = 1 rst = 1 couter_value =      1
clk = 0 rst = 1 couter_value =      1
clk = 1 rst = 1 couter_value =      2
clk = 0 rst = 1 couter_value =      2
clk = 1 rst = 1 couter_value =      3
clk = 0 rst = 1 couter_value =      3
clk = 1 rst = 1 couter_value =      4
clk = 0 rst = 1 couter_value =      4
clk = 1 rst = 1 couter_value =      5
clk = 0 rst = 1 couter_value =      5
$finish called from file "counter.v", line 51.
$finish at simulation time      80
VCS Simulation Report
Time: 80
CPU Time:      0.010 seconds;      Data structure size:  0.0Mb
Tue Dec 30 10:27:29 2014
CPU time: .037 seconds to compile + .018 seconds to elab + .216 seconds to link + .046 seconds in simulation
```


Exercise 3. Using DVE (1/5)

DVE is the integrated tool in VCS. This tool will help user to watch the waveform of design circuit.

This exercise will help user to watch the waveform of design circuit and explore some features in DVE.

Exercise 3. Using DVE (2/5)

User must open source code

\$ vi counter.v

Type following code in the correct location.
Please refer to the next picture for more detail:

```
initial begin  
    $vcdplusfile ("counter.vpd");  
    $vcdpluson ();  
end
```

Save file then run command :

\$ vcs -R -debug_pp counter.v

Check created wave file ("counter.vpd")

\$ ls

```
19 endmodule
20
21 module top;
22
23     parameter HALF_CYCLE = 5;
24
25     reg clk;
26     reg rst_n;
27
28     wire[31:0] cnt_value;
29
30     counter counter01 (.clk(clk), .rst_n(rst_n), .cnt_value(cnt_value));
31
32     always begin
33         #HALF_CYCLE clk = 1'b0;
34         #HALF_CYCLE clk = 1'b1;
35     end
36
37     //-----
38     // Type system task to dump waveform here
39     //-----
40     initial begin
41         $vcdplusfile ("counter.vpd");
42         $vcdpluson ();
43     end
44
45     //-----
46
47     initial begin
48         # ( 0*HALF_CYCLE) rst_n = 1'b0;
49         # ( 3*HALF_CYCLE) rst_n = 1'b1;
50         # ( 2*HALF_CYCLE) rst_n = 1'b0;
51         # ( 1*HALF_CYCLE) rst_n = 1'b1;
52         # (10*HALF_CYCLE)
53
54         $finish;
55     end
56
57     always @ (clk) begin
58         $strobe ("clk = %d rst = %d counter_value = %d",clk,rst_n,cnt_value);
59     end
60
61 endmodule
```

Exercise 3. Using DVE (3/5)

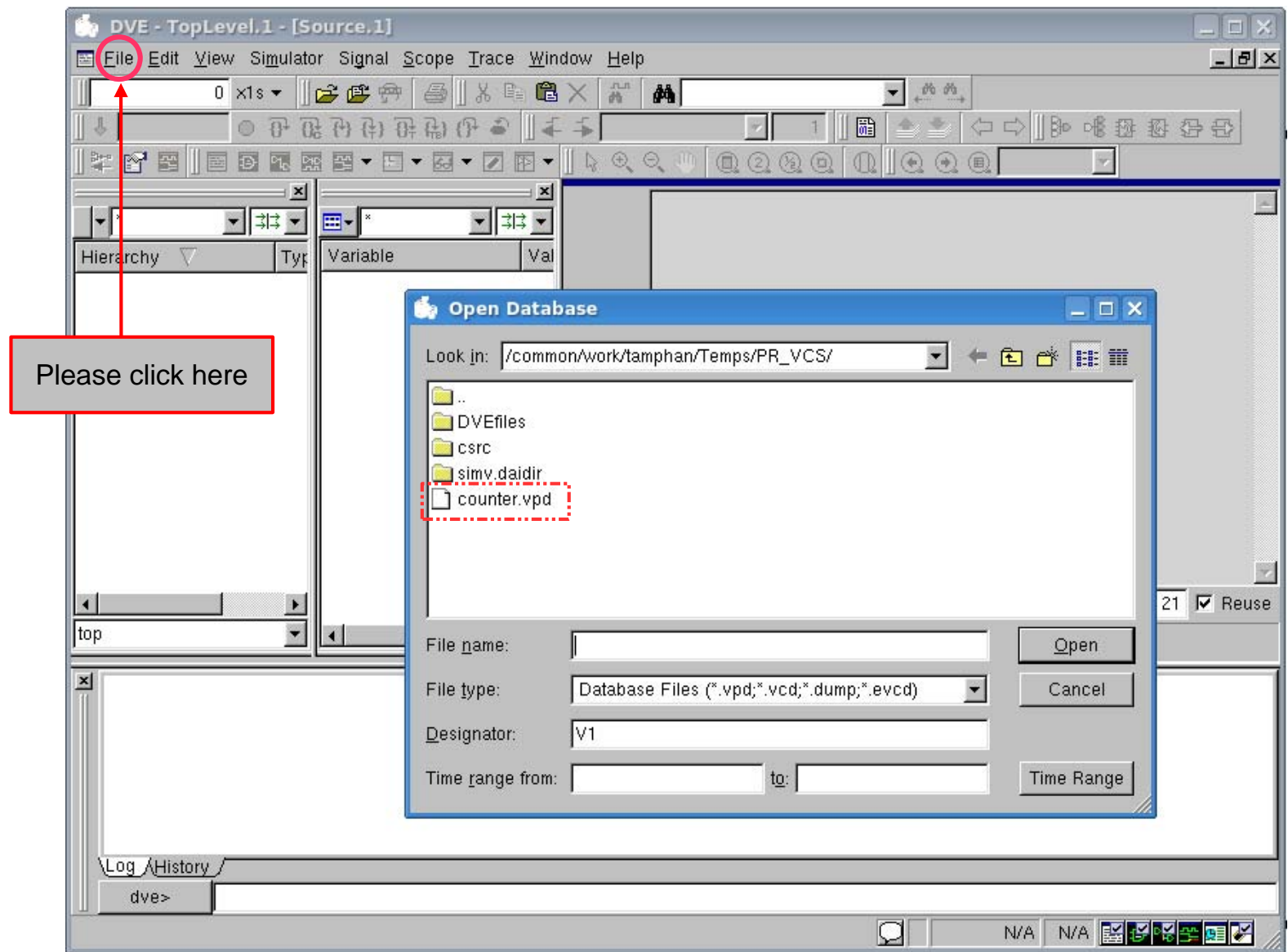
Open DVE:

\$ dve

Open wave form file:

File/Open Database ...

Or using Ctrl + O



Exercise 3. Using DVE (4/5)

Open wave form screen.

Step 1 : Open design :

- Left click on + sign
- Left click counter design

Step 2 : Select signals

- Use Ctrl + Left click to chose all signals

Step 3 : Open waveform screen

- Right click on selected signals
- Move mouse to *Add To Wave*
- Left click on *New Wave View*

| Variable | Value | Type |
|-----------------|-------|---------------|
| clk | | Wire(Port In) |
| rst_n | | Wire(Port In) |
| cnt_value[31:0] | | Reg(Port Out) |

Exercise 3. Using DVE (5/5)

Using waveform screen

The screenshot shows the DVE - TopLevel.2 - [Wave.1] window. The top toolbar contains icons for Zoom In, Zoom Full, and Zoom Out, which are highlighted with red boxes and labels. The main area displays a waveform for signal C1:0 REF, with a green oval highlighting a specific section of the waveform. A context menu is open over the waveform, showing options like Cut, Copy, Paste, and Set Radix. The Set Radix option is highlighted, and a red arrow points to it from a text box. The text box contains the following instructions:

Change radix of signal:

- Select signal cnt_value
- Right click
- Chose *Set Radix*
- You can chose binary, hex or decimal to display number

The waveform shows a clock signal (clk) and a reset signal (rst_n). The cnt_value signal is shown in hexadecimal, with values *0000, *01, *02, *03, *04, and *05. The waveform is titled 'The waveform of design' in a green box.

Using VCS with LSF (1/2)

*Open connection in client machine (this step is used to display GUI later)
(This step must be done when user is in the local machine (user PC))*

\$ xhost +

*Check to see your machine IP
(eth0 inet addr: 172.29.xxx.xx)*

\$ /sbin/ifconfig

Connect to LSF login server (Password here is Linux machine)

\$ ssh lsf-login11

Set LSF environment

\$ source /common/lsftool/RBS/dotfiles/lsf_cshrc

Set LSF configuration

\$ setenv LSF_PROJECT SV

*Set command so result will be returned from server to
client machine. (This step must be done when user is
in the server)*

\$ setenv DISPLAY [your machine IP]:0.0

Ex: \$ setenv DISPLAY 172.29.143.88:0.0



Using VCS with LSF (2/2)

Set environment for VCS

```
$ source /common/appl/Env/Synopsys/vcs-mx_vD-2009.12-4
```

Run VCS

```
$ bs vcs counter.v
```

or

```
$ bs simv
```

or

```
$ bs vcs -R counter.v
```

or

```
$ bs vcs -R -debug_pp counter.v
```

Run DVE

```
$ bs dve
```





Renesas Design Vietnam Co., Ltd.

© 2016 Renesas Design Vietnam Co., Ltd. All rights reserved.