SYNOPSYS®

CUSTOMER EDUCATION SERVICES

CustomSim Workshop

Lab Guide

Synopsys Customer Education Services
700 East Middlefield Road
Mountain View, California 94043

Workshop Registration: **1-800-793-3448**

www.synopsys.com

Copyright Notice and Proprietary Information

Right to Copy Documentation

The license agreement with Synopsys permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies. These copies shall contain the following legend on the cover page:
"This document is duplicated with the permission of Synopsys, Inc., for the exclusive use of _____ and its employees. This is copy number _____."

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Synopsys, AMPS, Cadabra, CATS, CRITIC, CSim, Design Compiler, DesignPower, DesignWare, EPIC, Formality, HSIM, HSPICE, iN-Phase, in-Sync, Leda, MAST, ModelTools, NanoSim, OpenVera, PathMill, Photolynx, Physical Compiler, PrimeTime, SiVL, SNUG, SolvNet, System Compiler, TetraMAX, VCS, Vera, and YIELDirector are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, Astro, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, Columbia,Columbia-CE, Cosmos, CosmosEnterprise, CosmosLE, CosmosScope, CosmosSE, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, Direct Silicon Access, Discovery, Encore, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, HSIMplus, HSPICE-Link, iN-Tandem, i-Virtual Stepper, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport,Library Compiler, Magellan, Mars, Mars-Rail, Milkyway, ModelSource, Module Compiler, Planet, Planet-PL, Polaris, Power Compiler, Raphael, Raphael-NES,Saturn, Scirocco, Scirocco-i, Star-RCXT, Star-SimXT, Taurus, TSUPREM-4, VCS Express, VCSi, VHDL Compiler, VirSim, and VMC are trademarks of Synopsys, Inc.

Service Marks (℠)

MAP-in, SVP Café, and TAP-in are service marks of Synopsys, Inc.

SystemC is a trademark of the Open SystemC Initiative and is used under license. ARM and AMBA are registered trademarks of ARM Limited. Saber is a registered trademark of SabreMark Limited Partnership and is used under license. All other product or company names may be trademarks of their respective owners.

Document Order Number: 60-I-031-BSG-008
HSPICE Essentials Student Guide

1 Resistor ladder containing 2 billion resistors

Learning Objectives

During this lab:

- You will simulate Resistor ladder circuit containing 2 Billion resistors.
- You will demostrate XA capacity and simulation speed in handling large hierarchical resistive ladder.
- Use the waveform viewer.

After completing this lab, you should be able to:

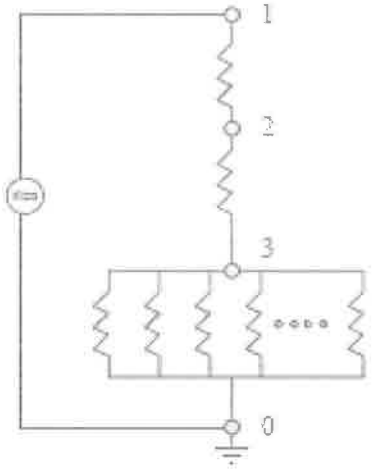
- Understand how XA is efficient in handling large circuits.
- Play with XA performace
- Analyze the output results using Waveform viewer.

Lab Duration: 20 minutes

Instructions:

The below circuit illustrates a resistor ladder that is constructed hierarchically with the following features:

The Resistor Ladder Test Case Circuit



About the circuit:
The lowest level subcircuit is called r10 and contains 20 resistors. Each resistor has a resistance of 1.0e12 ohms. The equivalent resistance between subcircuit port nodes A and B is 2.0e11ohm. The highest level subcircuit is called r2b. It contains 2 billion resistors with equivalent resistance of 2000 ohms.

The circuit in the above figure shows the top-level containing:

- Two 1000 ohm resistors
- One instance of subcircuit r2b forming a resistor ladder
- 2,000,000,002 total resistors
- The input is a PWL voltage source that ramps up then down from 0V to 3V to 0V

Task1 . Setup XA for Lab

1. Invoke XA from the **Lab1_2br** workshop lab Unix directory.

```
unix% module load xa
unix% which xa
unix% cd Lab1_2br
```

Task.2 Exploring Helpful XA commands

1. Just type "xa -help" in the command prompt to more comand prompt options.

```
% xa-
help
sureshv@cholacs9:/global/gts_tst_ams1/sureshv/XA/XA_CES_LABS/lab1_2hr % xa -help
XA 32-bit LINUX version D-2010.03-SP2 (built 19:36:39 Aug 19 2010) build id:
1854067.
Usage:
xa [[-hspice|-eldo|-spectre] infile] [-c scriptfile] [-D<x>] [-D<x=y>]
[-help] [-I[dir]] [-mt ncpu] [-out [outpath/]outfile] [-tcl] [-version]
[-top top_subckt] [-wavefmt fmt] [-webhelp] [-U<x>]

Following options are available:
-------------------------------------------------------------------------------
| -hspice infile  | Specify the name of the input netlist in Hspice format      |
| -eldo infile    | Specify the name of the input netlist in Eldo format        |
| -spectre infile | Specify the name of the input netlist in Spectre format     |
| -c scriptfile   | Specify the name of the command script file                 |
| -D<x>           | Defines string x and runs the C preprocessor.               |
|                 | This option is only valid in the Eldo and Spectre modes.    |
| -D<x=y>         | Defines string x as y, and runs the C preprocessor.         |
|                 | This option is only valid in the Eldo and Spectre modes.    |
| -h              | Same as -help                                               |
| -help           | Print usage information                                     |
| -I[dir]         | Search the directory for include files                      |
| -intr           | Keep IM database for invoking interactive mode during transient by Ctrl-C |
| -intr time      | Invoke the interactive mode at specified time               |
| -mt ncpu        | Set the number of CPU to be used                            |
| -o              | Same as -out                                                |
| -out outfile    | Set output file to outfile.log                              |
| -tcl            | Enable tcl programming for command script file              |
| -top subckt     | Specify the top subcircuit                                  |
| -U<x>           | Undefines string x and runs the C preprocessor.             |
|                 | This option is only valid in the Eldo and Spectre modes.    |
| -v              | Same as -version                                            |
| -version        | Print the version information                               |
| -wavefmt fmt    | Specify the format of output waveform file                  |
| -webhelp        | Invoke manual in a browser                                  |
-------------------------------------------------------------------------------

real    0m0.021s
user    0m0.006s
sys     0m0.002s
```

Task3. Running Resistor Ladder Lab

The following instructions provide step-by-step guidance for the resistor ladder test case.

1. Run the Resistor Ladder Simulation by using run script in the tutorial.

```
unix% run
```

or Run this netlist in the command prompt, by

```
unix% xa -hspice   top.spi -o   RESULT/xa
```

2. After the resistor ladder simulation is complete, use a waveform display tool to inspect the simulation results. To invoke waveform viewer,

```
unix% module load cx
unix%  wv &
```

3. This test case circuit shown on the above figure depicts the functionality of a voltage divider, where the voltages of node 2 and node 3 follow the voltage of input node 1.

4. Review the resistor ladder simulation statistics by inspecting the simulation statistics collected from the xa.log file. The simulation takes very few seconds to complete the simulation of 2 -BILLION resistors, which shows XA capacity in using for larger designs.

```
unix% gvim xa.log &
```

5. Most of the simulation time is used for Front End netlist processing and TR/SETUP of circuit elements. You can look at the statistics as given below from xa.log file,

=> FE SUMMARY: Time taken = 13 sec

Peak Virtual Memory = 74 M
Grand Total Time = 16 sec
Total Wall Time = 17 sec

Task4. Questions from the log file

- Which version of XA are you using? Is that 32-bit or 64-bit?
- How to run 64-bit XA simulation?
- What are the total number of elements/nodes in the netlist?
- What is the CPU time used for transient analysis?
- What is the memory usage for the run?

Task5. View the output results from the XA log file

1. In the XA log file, find the XA simulator version, total number of elements and nodes, CPU time and memory usage etc. You should see the following text in the log file:

   Circuit statistics:

```
Nodes statistics:
--------------------
| type  | count       |
|-------+-------------|
| TOTAL | 1000000004  |
|--------------------|

Element statistics:
-------------------------------------------------
| type     | format | count       | unique models |
|----------+--------+-------------+---------------|
| V        | HSPICE | 1           | 1             |
| SIMPLE R |        | 2000000002  |               |
| TOTAL    |        | 2000000003  | 1             |
|-------------------------------------------------|
```

2. Look at the partition summary and also waveform statistics,

```
Summary of Partition for Transient Analysis:
-----------------------------------
| number of | devices   | nodes     |
| regions   | in region | in region |
|-----------+-----------+-----------|
| 1         | 0         | 0         |
| 1         | 3         | 3         |
|-----------------------------------|

Summary of probed signals and waveforms:
---------------------------------------
| Total number of signals probed  | 3 |
| Voltage probes                  | 3 |
| Current probes                  | 0 |
| Other probes                    | 0 |
| Total waveforms written to file | 3 |
| Logic waveforms                 | 0 |
| Other waveforms                 | 3 |
| Waveform aliases                | 0 |
|---------------------------------------|
```

Task 6. Run WaveView

1. Run WaveView by entering the command "wv" at the command prompt.
2. Open the transient analysis results file, top.fsdb
3. Plot the signals and get the feel on waveview usage.

2 Interactive Mode

Learning Objectives

During this lab :

- You will work on XA interactive commands.
- Understand the usage of basic interactive commands
- You interact with the simulation on the fly.

After completing this lab, you should be able to:

1. Access interactive mode.
2. Use interactive mode help.
3. Use interactive mode commands.

Lab Duration: 20 minutes

Instructions:

Task1 . Setup XA for Lab Interactive

1. Invoke XA from the Lab2_interactive workshop lab Unix directory.

```
unix% cd Lab2_interactive
unix% which xa
```

2. The Table below lists the files needed for this tutorial.

File Name	Description
top.spi	Top-level SPICE netlist
circuit.sp	SPICE netlist
cells.inc	SPICE netlist
bsim3.mod	BSIM3 MOS model
run	Run script for this tutorial

Task.2 How to get into interactive mode?

From the UNIX command line, use the run script, which contains the -intr XA command line option to activate interactive mode.

Task3. Run XA Simulation

1. Run the top.spi netlist by using run script in the tutorial.

```
unix% run
```

or Run this netlist in the command prompt, by

```
unix% xa top.spi -o RESULT/xa -intr 0ns
```

2. At time zero, the simulation stops and switches to interactive mode. The XA> prompt is displayed.

```
XA>
```

3. You can try to enter interactive mode in different simulation times with the command

```
unix% xa top.spi -o RESULT/xa -intr 50ns
```

4. Type help to get the available XA interactive commands. Use "help <command> to get detailed usage of that <command>

```
XA>help
XA>help iprint_connectivity
```

5. You can practice the following recommend commands

iprint_connectivity	: print node connectivity
iprint_dcpath	: report dc path
iprint_elem_info	: print element information
iprint_exi	: print instances (name) which have excessive current
iprint_help	: print command syntax and description
iprint_node_info	: print node information
iprint_subckt	: print instances (name) which are instantiations of given subcircuit name
iprint_time	: print current simulation time
iprint_tree	: print instance tree information
imatch_elem	: print (eidx, name) for elems matching specified patterns
imatch_node	: print (nidx, name) for nodes matching specified patterns
iopen_log	: open log file
iclose_log	: close log file
icontinue_sim	: continue simulation
iquit_sim	: quit simulation

Task4. Answer the below Questions

1. How to find the current simulation time?

---

2. Go to 500ns. What is the command to find node index of node xinv1.1?
   What about node xinv1.2?

---

3. What is the command to find node connectivity of node xinv1.1? What about xinv1.2?

---

4. What is the voltage value of node xinv1.1 at 500ns? What about xinv1.2?

---

5. How to continue the run for another 50ns?

---

6. How to enter the interactive mode at 200ns?

---

3 Direct VCD Stimulus

Learning Objectives

During this lab :

- You will work VCD file simulation
- Explore the relevant files.

After completing this lab, you should be able to:

- Use a value change dump (VCD) file with XA
- Create Signal Information File
- Check the Log file for errors.

Lab Duration: 20 minutes

Instructions:

Task1 . Setup XA for Lab Direct VCD Stimulus

1. Invoke XA from the Lab3_vcd workshop lab Unix directory.

```
unix% cd Lab3_vcd
unix% which xa
```

2. The Table below lists the files needed for this tutorial.

File Name	Description
top.spi	Top-level SPICE netlist
adder.vcd	VCD file
adder.sig	Signal information file
cmd	XA command file
run	XA run script

Task.2 Exploring Helpful XA commands

```
load_vector_file -file adder.vcd -format VCD -ctl adder.sig
```

Task3. Run XA Simulation

1. Run the top.spi netlist by using run script in the tutorial. This script specifies the SPICE netlist, top.spi, and the command files to use a VCD file in XA. Use the UNIX cat command to display the content of the cmd command file:

```
unix% run
```

or Run this netlist in the command prompt, by

```
unix% xa top.spi -c cmd -o RESULT/xa
```

2. Open the adder.vcd and adder.sig files and understand signal directions.

3. When the simulation finishes, it contains the following directories and files:

./RESULT (directory)
    xa.err0
    xa.fsdb0
    xa.log
The .err0 file is the output checking result from the vector file

4. Use the UNIX cat command to display the xa.err0 file.

```
Expected Output Summary
-------------------------------------------------
| Node | Pass/ | Expected | Simulated | Time     |
| name | Fail  | state    | state     | (us)     |
|------+-------+----------+-----------+----------|
| cout | P     |          |           | no error |
| s[0] | P     |          |           | no error |
| s[1] | P     |          |           | no error |
| s[2] | P     |          |           | no error |
| s[3] | P     |          |           | no error |
|------+-------+----------+-----------+----------|
```

Lab 3

4 Probe/Power Analysis

Learning Objectives

During this lab :

- You will work on the various probing/power analysis commands
- Find the suitable commands to probe V/I
- Explore the relevant files.

After completing this lab, you should be able to:

- Probe the node voltages and also branch currents
- Plot the waveform of voltages and currents
- Report node power
- Check the Log file for errors/warnings.

Lab Duration: 30 minutes

Instructions:

Task1 . Setup XA for Lab Probe/Power Analysis

1. Invoke XA from the Lab4_probe workshop lab Unix directory.

```
unix% cd Lab4_probe
unix% which xa
```

2. The Table below lists the files needed for this tutorial.

File Name	Description
top.spi	Top level SPICE setup netlist
circuit.sp	Circuit netlist file
cells.inc	Leaf level SPICE netlist
bsim3.mod	MOS Models file
run	XA Run script
xa_voltage.cmd	Config Command file for voltage probing
xa_current.cmd	Config Command file for current probing
power_by_node_default	Config Command file to report power by node
power_by_port_default	Config Command file to report power by port

Task.2 Exploring Helpful XA commands

```
probe_waveform_voltage *
probe_waveform_voltage * -port 1
probe_waveform_voltage *.* -limit 4
probe_waveform_current –i vd* vss
probe_waveform_current –isub xinv1.a xinv1.z
report_power -port xinv1.* -label xinv1_block_power
report_power -by_node vdd1 -label xinv1_power_thru_vdd
```

Task3. Run XA Simulation

1. Run the top.spi netlist by using run script in the tutorial. This script specifies the SPICE netlist, top.spi, and the command files to probe node voltages or branch currents. Use the UNIX cat command to display the content of the xa.cmd command file.

```
unix% run
```

or The run script contains the following lines and you can run the simulation in the command prompt one by one as follows;

2. To run a simulation to probe node voltages use the command,

```
unix%  xa -hspice top.spi -c xa_voltage.cmd -o RESULT/xa_voltage
```

The command, "probe_waveform_voltage *" by default will print up to 3-level of hiearchy starting from the top-level (top-level is consider hierarhcy 0)

The command "probe_waveform_voltage * -port 1" will print the subcircuit port voltages till level 3.

The command "probe_waveform_voltage *.* -limit 4" will print all the node voltages till level 4.

3. To run a simulation to probe branch currents you can use the command,

```
unix%  xa top.spi -c xa_current.cmd        -o RESULT/xa_current
```

The command "probe_waveform_current -i vd* vss" will probe the current flowing through the Voltage Sources "vd*" and "vss", where * is a wildcard to match vd1, vd2, vd3 and vd4.

The command "probe_waveform_current -isub xinv1.a xinv1.z" will be the subcircuit port currents.

4. To report a power by ports, you can use the command,

```
unix%  xa top.spi -c power_by_port_default -o
RESULT/xa_power_report_by_port_default
```

5. To report a power by nodes, you can use the command,

```
unix%  xa top.spi -c power_by_node_default -o
RESULT/xa_power_report_by_node_default
```

6. Go to the ./RESULT folder and examine all the output files created. Load the FSDB files in WV to see the node voltages/branch currents. Check out the *.power file to find the power reports.

Lab 4

5 Back Annotation Flow

Learning Objectives

During this lab :

- You will work on post-layout BA simulation
- Run XA back-annotation flow
- Explore the relevant files.

After completing this lab, you should be able to:

- Set up back annotation simulation with XA
- Check the Log file for errors.

Lab Duration: 30 minutes

Instructions:

Task1 . Setup XA for Back-Annotation Simulation

1. Invoke XA from the Lab5_ba_flow workshop lab Unix directory.

```
unix% cd Lab5_ba_flow
unix% which xa
```

2. The Table below lists the files needed for this tutorial.

File Name	Description
top.spi	Top-level SPICE netlist
add4.sp	SPICE netlist
add4.spf	SPF file
mos.model	BSIM3 MOS model
cmd_ba	XA command file
cmd_ba_conly	XA command file
cmd_ba_min_res	XA command file
cmd_selective_extract	XA command file
cmd_ba_active_net	XA command file
run	Run script for this tutorial

Task.2 Exploring Helpful XA commands

```
load_ba_file -file add4.spf
load_ba_file -file add4.spf -cnet *
load_ba_file -file add4.spf -min_res 0.1
set_ba_option -active_net_file ./RESULT/active_net.rcxt
set_ba_active_file -file active_net -twindow 0ns 20ns
```

Task3. Run XA Simulation

1. Run the top.spi netlist by using run script in the tutorial. This script specifies the SPICE netlist, top.spi, and the command files to run the back-annotation flow. Use the UNIX cat command to display the content of the cmd command file.

```
unix% run
```

or The run script contains the following lines and you can run the simulation in the command prompt one by one as follows;

2. To run a simple back-annotation simulation use the command,

```
unix%  xa top.spi -c cmd_ba -o RESULT/XA_normal_ba_flow
```

3. To find a selective-nets based on a default threshold you can run the command,

```
unix%  xa top.spi -c cmd_selective_extract -o
RESULT/XA_selective_extract_ba_flow
```

4. To run a selective-nets based back-annotation you can use the command,

```
unix%  xa top.spi -c cmd_ba_active_net -o RESULT/XA_active_net_ba_flow
```

5. To perform Lump C only back-annotation you can use the command,

```
unix%  xa top.spi -c cmd_ba_conly -o RESULT/XA_ba_conly
```

6. To perform shorting of small resistors during the back-annotation you can use,

```
unix%  xa top.spi -c cmd_ba_min_res -o RESULT/XA_ba_remove_min_res
```

7. Go to the ./RESULT folder and examine all the output files created.