Simulation Standard

TCAD Driven CAD

A Journal for CAD/CAE Engineers

# *Guardian-Net* Netlist Extractor: An Example of Specific Device Extraction Using XI-Scripts

## Introduction

*Guardian LVS* from Silvaco is an efficient, reliable, and diverse package that brings the benefit of high-speed *verification to PC-based test environments. Guardian-Net* netlist extractor, included with the *Guardian* suite, is a dynamic module that supports the high-speed extraction of a variety of devices, including MOS transistors, resistors, capacitors, and other standard devices.

The complexity of device geometry has increased dramatically over time. Extremely detailed device makeup, especially in passive devices, is often too complex for accurate parsing by standard netlist extraction software. Silvaco has addressed these challenges with the *LISA* (Language for Interfacing Silvaco Applications) scripting language and its extension, xi. *LISA* adds unparalleled customization options and support to complex products. *LISA* is bundled with the Celebrity design and verification package available from Silvaco.

This article provides insights into the extraction of complex devices and offers methods of customizing *LISA*-based scripts. The first part discusses the definition of devices using the parameterized cells feature of Silvaco's *Expert*. The second part introduces the *LISA* script and presents illustrated instructions for using *LISA* and xi to extract a final SPICE netlist that is compatible with any SPICE simulator. The final section addresses optional improvements to the script.

## Part One: Using Parameterized Cells to Design and Define the Device

Parameterized cells, or P-cells, help increase designer productivity by adding enormous flexibility and efficiency to the design process. While standard cells help the designer to avoid repetitive drawing of identical pieces of layout, P-cells extend this functionality to the specific parameters that define the mask geometry. As a result, P-cells assist in the automation of layout design and help speed-up modification through the revision of P-cell parameters instead of wasting valuable resources by repeatedly redrawing the layout geometry.

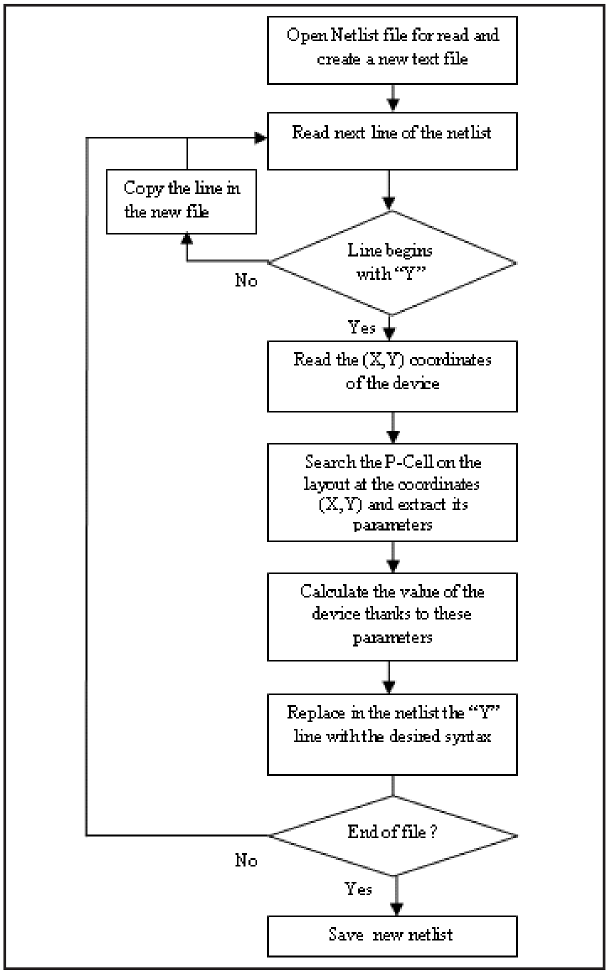

Open Netlist file for read and create a new text file
Read next line of the netlist
Copy the line in the new file
Line begins with "Y"
No
Yes
Read the (X,Y) coordinates of the device
Search the P-Cell on the layout at the coordinates (X,Y) and extract its parameters
Calculate the value of the device thanks to these parameters
Replace in the netlist the "Y" line with the desired syntax
End of file ?
No
Yes
Save new netlist

Figure 1. Flowchart of a LISA script that replaces a custom device with the correct device description.

*Continued on page 2....*

### INSIDE

Volume 13, Number 13, March 2003

*Guardian* makes extensive use of P-cells to extend custom functionality to non-standard or complex devices. The three steps necessary to successfully define a device are:

- **Create a P-CELL** that includes all relevant geometrical parameters (the number of turns in an inductance, for example), and any other parameters that prove useful when calculating a device's value.
- **Define a P-Cell layer.** This layer will serve as the device's "recognition layer" for later use in custom device definition. This layer is also definable as a Boolean operation that exists between the existing device layers.
- **Define a CUSTOM DEVICE**. A "custom device" is typically used in the C-Interpreter module of *SmartSpice*, but its unique "Y"-symbol syntax format makes it an ideal detector of specific devices in a netlist. Select the **Setup > Technology > Device > Setup > "Device Type"** menu, and provide three pieces of information:

  o Recognition layer

  o Pin layers

  o Spice model name

Once the new P-cell and its corresponding device are defined, they are ready to receive any combination of instance parameters. At this point, all the defined devices will appear to the *Guardian* Netlist Extractor as "CUSTOM." Connectivity should be good, and the syntax appears similar to:

```
Y4  2 0  "pin_name"  …  "model_name"

* Center of device: x = 1.9U y = 6.1U
```

This default syntax format is not supported by all SPICE simulators and does not contain any specific device parameters or values. A custom *LISA* script, when applied, detects all custom device instances, processes the information into useable syntax, and updates the netlist. The following section steps through this process.

## Part Two: An Analysis of the LISA Extractor Script

As previously mentioned, a custom device is extracted when indicated by a "Y" symbol in the netlist. The *LISA* script replaces these lines with correct device description parameters, including syntax, parameters and calculated value. This process flow is illustrated in Figure 1.

**The script performs the following tasks:**

- The LISA script detects each line that begins with a "Y" in the *Guardian* netlist.
- The script reads the (X,Y) coordinates of the device at each detected line and appends the new information after the device description.

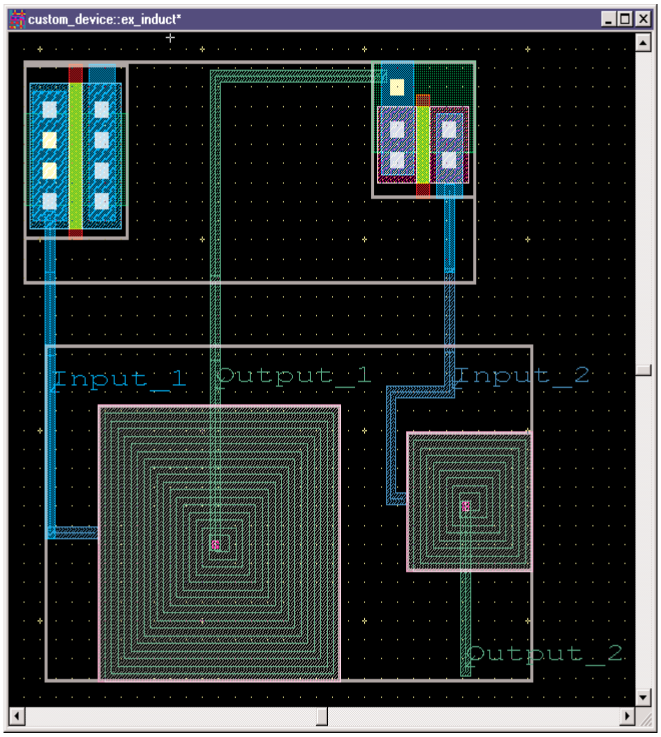


Figure 2. Layout cell that contains two MOS transistors connected to two square inductances.

- The script searches for the P-cell instance located at the detected coordinates within the layout.
- The script extracts the P-cell instance parameters and, with a suitable formula, calculates the device's value.
- The script then replaces each "Y" line with new, syntactically correct that includes the device's parameters and value. The connectivity information (node names) is retained.

These steps correspond to the following simplified flowchart shown in Figure 1.

The *LISA* script language includes all functions needed to implement this flowchart, including layout object search, string manipulation, and P-cell analysis. The script makes particular use of the following functions:

- These commands are used to read coordinates from the second line of the custom device description (e.g. "`Center of device: X = 161.960U  Y = 36.010U`"):

  o `STREAM_CREATE_INPUT` opens a file as input

  o `STREAM_GET_STRING` reads a line within the file

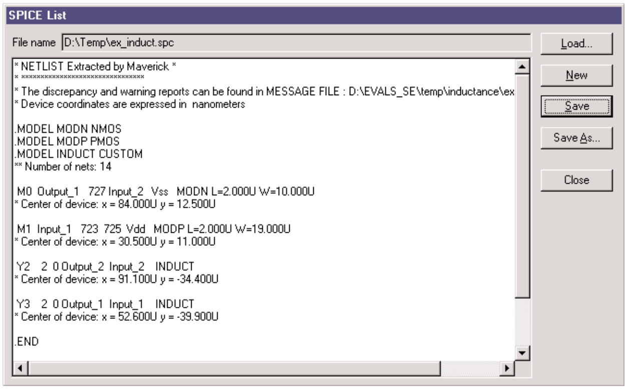
  o `STREAM_CLOSE` closes the file

- These commands are used to create a new netlist file:

  o `STREAM_CREATE_OUTPUT` opens a file as output

  o `STREAM_PUT_STRING` puts a line in the file

  o `STREAM_CLOSE` closes the file



```
* NETLIST Extracted by Maverick *
* ******************************
* The discrepancy and warning reports can be found in MESSAGE FILE : D:\EVALS_SE\temp\inductance\ex
* Device coordinates are expressed in  nanometers

.MODEL MODN NMOS
.MODEL MODP PMOS
.MODEL INDUCT CUSTOM
** Number of nets: 14

 M0  Output_1   727 Input_2   Vss   MODN L=2.000U W=10.000U
* Center of device: x = 84.000U y = 12.500U

 M1  Input_1   723  725  Vdd   MODP L=2.000U W=19.000U
* Center of device: x = 30.500U y = 11.000U

 Y2    2  0 Output_2  Input_2    INDUCT
* Center of device: x = 91.100U y = -34.400U

 Y3    2  0 Output_1  Input_1    INDUCT
* Center of device: x = 52.600U y = -39.900U

.END
```

Figure 3. ***Guardian*** Netlist extracted from the layout of Figure 2.

- The `SEARCH` command is used to locate all instances of the device by means of the coordinates. Since the search area is a region around the center of the device, the `search_area` is defined as a box around the stored X,Y coordinates. For example:
  - `search_area = RECT_CREATE (X-1, Y-1, 2, 2);`
  - `Pcells_sequence = SEARCH (SEARCH_INSTANCE, {}, search_area, CROSS_RECT, False, SEARCH_ALL);`
- The script places the P-cell parameter sequence for each found instance in the output sequence seq_params by using the `.PARAMS` layout object component:
  - `SEQ_ADD_LAST (seq_params, (Pcells_sequence[1].PARAMS));`
- The script uses these parameters to perform the calculations necessary in order to obtain the value of each device, as seen in this example of a square inductance calculation:
  - `L = POW (Nb_turns, 3) *` $\mu_0$`/ Radius`
- The script then writes the revised syntax to the netlist, as shown in this example of a custom inductance device:
  - `L4 In  Out INDUCT Radius=1e-06um Nb_turns=25   L=5e-008 nH`

Once the ***LISA*** script is correctly applied, the netlist is ready for analysis and simulation without the need for manual correction.

## Part Three: Examples of Cell Layout and Netlist

In order to illustrate this method of extraction, a layout cell was designed (Figure 2) that contains two typical devices (MOS transistors) connected to two specific devices (square inductances).

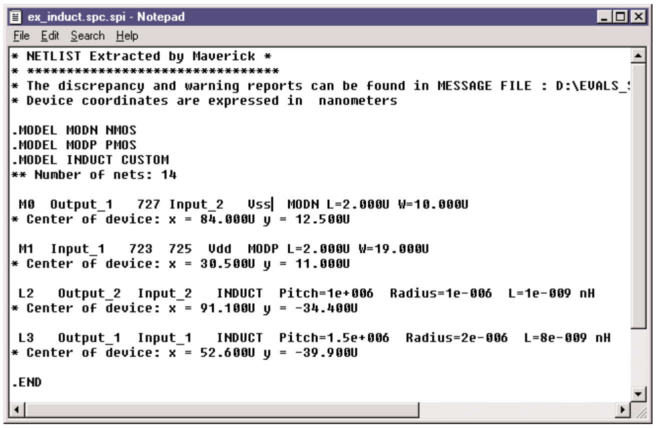
Figure 3 is the unmodified netlist extracted with ***Guardian*** Netlist Extractor, and shows the connectivity between the MOS transistors and "custom devices."

Finally, Figure 4 is the result of the successful application of the ***LISA*** script. All devices are corrected with the correct syntax, and include the necessary parameters ("`Pitch`" and "`Radius`") and value ("`L`") for each inductance.

## Part Four: Recommendations

The following suggestions may help to further automate this procedure:

- It is possible to integrate the ***Guardian-Net*** netlist extraction into the ***LISA*** script when using Guardian's batch mode command. While this allows the script to run without previously launching the ***Guardian*** application, it is necessary to create the ***Guardian*** Project File prior to execution.
  - **Command syntax:** `spawn "path_to_executable" "/e path_to_extraction_setup_file";`
  - **Example:** `spawn "C:\\Silvaco \\lib \\ ....\\guardian_extractor.exe"   "/e C:\\Silvaco \\lib \\ .... \\mux4.mpr";`
- Add this line to the ***LISA*** script in order to automatically display the revised file in Windows Notepad when finished:
  - `spawn "C: /WINNT /NOTEPAD.EXE " (file_extracted&".new")  /nowait;`
- These and other script commends can be integrated into the GUI of Silvaco's ***Expert*** layout editor with the new "custom menus "feature. For more information see *Simulation Standard*, Volume 12, Number 3, March 2002.



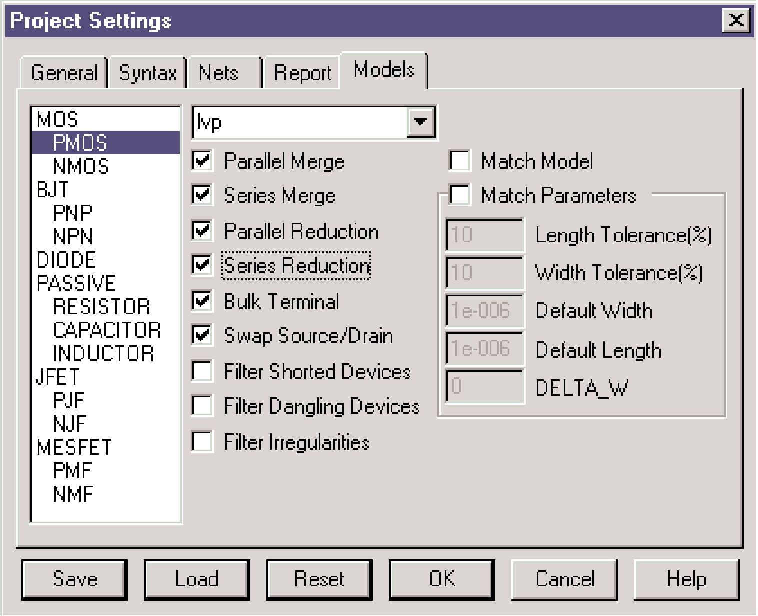
```
* NETLIST Extracted by Maverick *
* ********************************
* The discrepancy and warning reports can be found in MESSAGE FILE : D:\EVALS_
* Device coordinates are expressed in  nanometers

.MODEL MODN NMOS
.MODEL MODP PMOS
.MODEL INDUCT CUSTOM
** Number of nets: 14

 M0  Output_1   727 Input_2   Vss  MODN L=2.000U W=10.000U
* Center of device: x = 84.000U y = 12.500U

 M1  Input_1   723  725  Vdd  MODP L=2.000U W=19.000U
* Center of device: x = 30.500U y = 11.000U

 L2   Output_2  Input_2   INDUCT  Pitch=1e+006  Radius=1e-006  L=1e-009 nH
* Center of device: x = 91.100U y = -34.400U

 L3   Output_1  Input_1   INDUCT  Pitch=1.5e+006  Radius=2e-006  L=8e-009 nH
* Center of device: x = 52.600U y = -39.900U

.END
```

Figure 4. Final netlist generated using the LISA script.

# Complex Parallel-Series Reduction

## 1. Introduction

Silvaco's ***Guardian LVS*** tool compares two circuits that are defined by their netlists. The comparison is based strictly on the topological structure of these circuits. Topologically equivalent netlists are considered different, even if they are functionally equivalent. There are several techniques available for designing the same functionality by means of topologically different netlists. While it is impossible for the LVS tool to "know" about all these techniques, many are supported.



lvp

☑ Parallel Merge

☑ Series Merge

☑ Parallel Reduction

☑ Series Reduction

☑ Bulk Terminal

☑ Swap Source/Drain

☐ Filter Shorted Devices

☐ Filter Dangling Devices

☐ Filter Irregularities

☐ Match Model

☐ Match Parameters

10 Length Tolerance(%)

10 Width Tolerance(%)

1e-006 Default Width

1e-006 Default Length

0 DELTA_W

Figure 1. Model Settings Panel.

## 2. Reduction Varieties

A general method of ignoring strict topological equivalence during the layout design phase is the disregarding of the order of series-connected transistors in the reduction phase. **Series Reduction** is a process of reducing series-connected, same-type transistors that feature different gate terminals. In order for LVS to properly handle this process, a special "pseudo-node" device is introduced into the netlist graph. This pseudo-node represents the series of transistors as a single node. The resulting logical configuration is called a **cluster**. Users choose to recognize or disregard series transistor order by checking the box next to "Series Reduction" in the Model Settings Panel (Figure 1). The option is set independently for each device type and model.

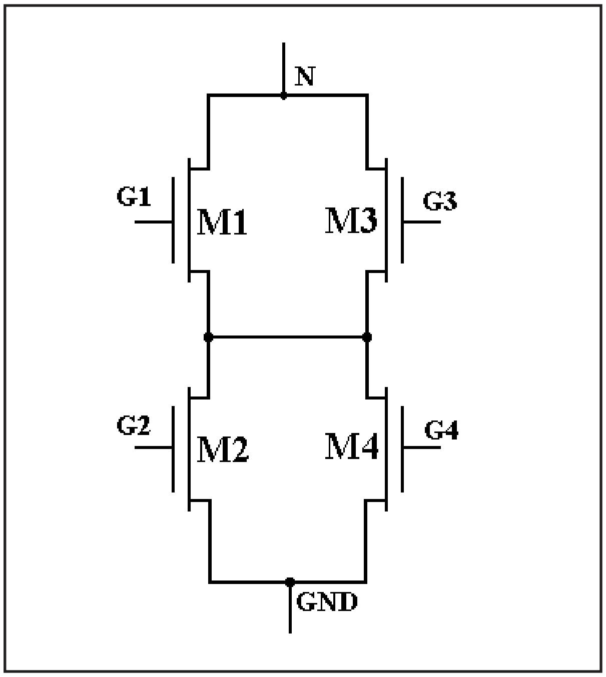


Figure 2. Series-parallel network of MOSFET transistors.

Another reduction process in the Guardian LVS tool is called **parallel reduction.** Parallel reduction is used to reduce parallel sets of transistors for later use in a separate series reduction. Parallel reduction also replaces sets of clusters that are connected in parallel and constructed previously from "pseudo-nodes." The logical configuration obtained as a result of parallel reduction is also called a cluster. Users choose to recognize or disregard parallel transistor order by checking the box next to "Parallel Reduction" in the Model Settings Panel (Figure 1). The option is set independently for each device type and model.

Since the LVS tool handles sets of parallel transistors without reducing to single "pseudo-node," parallel reduction must be activated in tandem with series reduction. In addition, the series reduction feature is also used to reduce either a series of parallel clusters or a series of both transistors, as well as parallel clusters that are connected in series form to a single cluster.

## 3. Simple Abstraction Model for Reduction

The section describes a simple reduction-stage abstraction model that helps to illustrate some problems that arise during the process. Figure 2 is an illustration of a

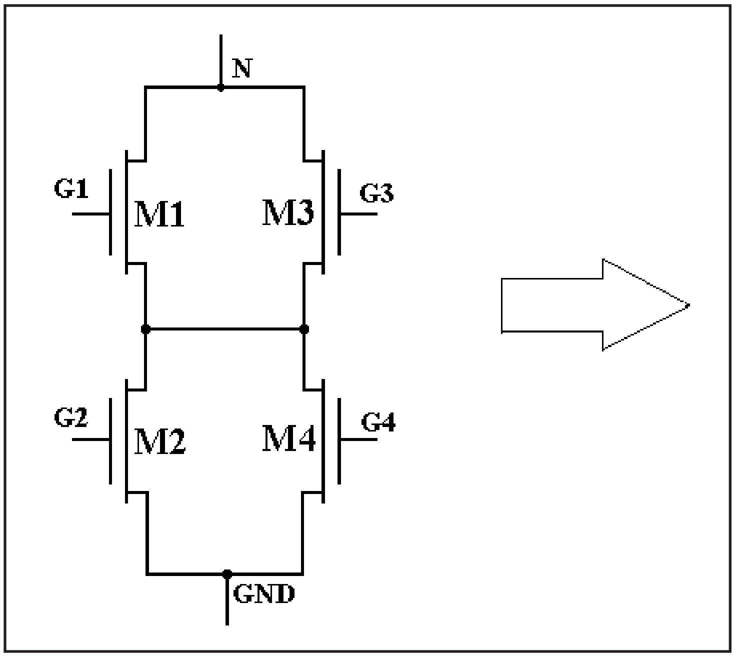


Figure 3. Reduction for series-parallel network using simple abstraction model.

series-parallel network of MOSFET transistors. Transistors **M1** and **M3** and gates **G1** and **G3** are connected in parallel, as are transistors **M2** and **M4** and gates **G2** and **G4**. Transistor pairs **M1-M3** and **M2-M4** are connected in series. There is no way for series reduction to transform this schematic if parallel reduction is not previously executed.

Note: Gates **G1** and **G3** nets should not be identical. Identical gates result in a "transistor parallel merging" pattern. The transistors are merged if the box next to the "Parallel Merge" option of Guardian LVS's Model Settings Panel (see Figure 1) is checked. Necessary merging is automatic if the Series Reduction and Parallel Reduction options are turned on, regardless of the state of the Series Merge or Parallel Merge settings.

How are parallel and series reduction performed? A relatively simple abstraction model for handling different reduction cases is illustrated by Figure 3.

After a combination of parallel and series reductions, the original network is replaced by single cluster with interchangeable **G1**, **G2**, **G3** and **G4** gate terminals. Pseudo-nodes in Figure 3 are indistinguishable from other pseudo-nodes obtained from the network shown in Figure 4, because Figure 4 is also reduced to a single cluster of interchangeable **G1**, **G2**, **G3** and **G4** gate terminals:

It is necessary to improve the abstraction model to distinguish the two previous cases, but to do so is complicated. It is possible that in both netlists in some place we have the configuration from previous figure. Let's assume that net **G1** in Figure 3 is connected to a network with local characteristic **A**. Net **G2** is connected to the network with characteristic **B**, **G3** to network **C**, and **G4** to network **D**. Let's assume also that net **G1** in Figure 4 is connected to the network with characteristic **A**, **G2** to **C, G3** to **B,** and **G4** to **D**. There is no difference between these configurations because in this abstraction model, **G2** and **G3** are not interchangeable.

## 4. Guardian LVS Abstraction Model for Reduction

The ***Guardian LVS*** matching engine abstraction model accurately reflects conditions of gate terminal interchangeability, and to solve problems like the one found in section 3. For the previous example it is possible to interchange **G1** and **G2**, **G3** and **G4**, as well as pairs **G1-G2** and **G3-G4**. An idea for an abstraction model that met our requirements is illustrated in Figure 5.

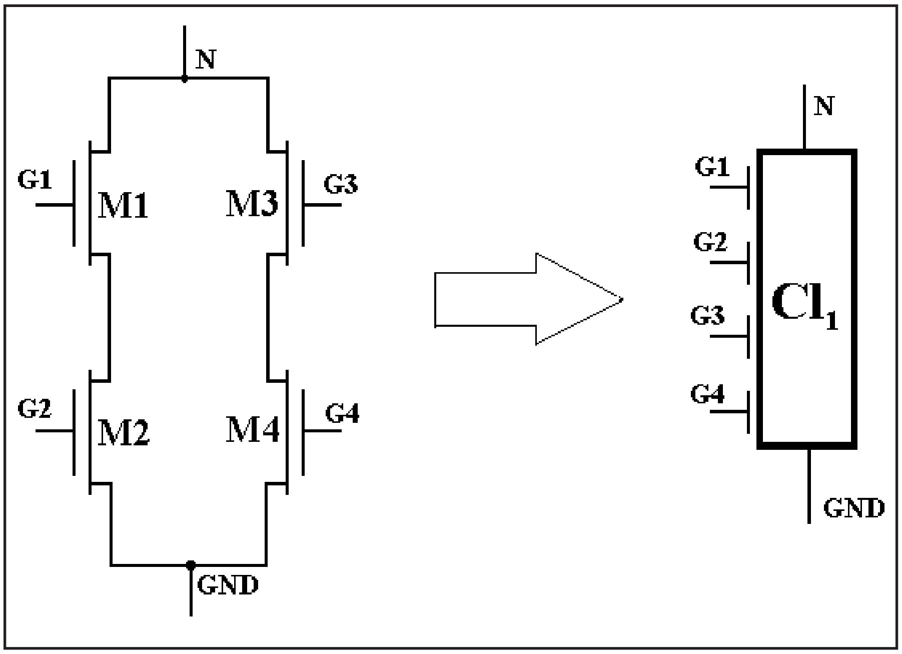


Figure 4. Reduction for parallel-series network using simple abstraction model.

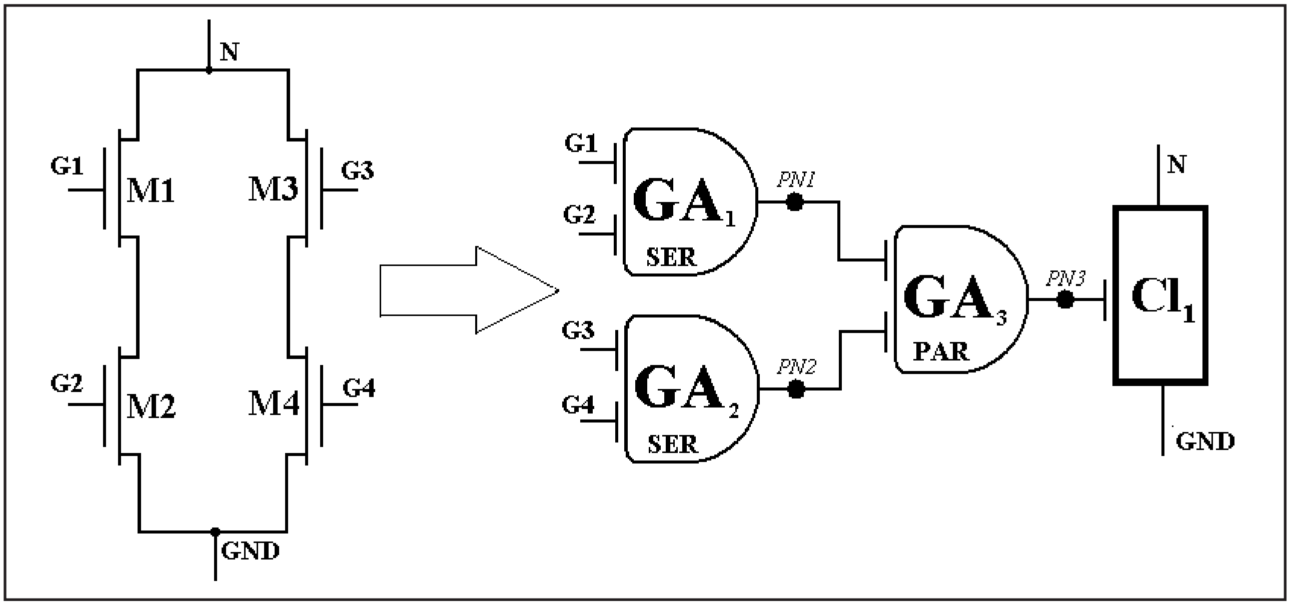


Figure 5. Reduction for parallel-series network realized in Guardian LVS tool.

In this model, several "pseudo-devices" and "pseudo-net" nodes represent the original network. Pseudo-nets are represented as **PN1**, **PN2** and **PN3**. Pseudo-devices are shown as "gate-arrays" **GA1**, **GA2**, **GA3** and cluster **CL1**. Each gate-array represents single interchangeability: **GA1** provides interchangeability of **G1** and **G2**, **GA2** makes **G3** and **G4** interchangeable, and **GA3** makes the pairs **G1-G2** and **G3-G4** interchangeable as well. If the matching engine finds an error in pseudo-node **GA1**, **GA2**, **GA3**, **PN1**, **PN2**, **PN3**, or **CL1**, it is reported to the user in transistor cluster **M1-M4**.

Reduction process for network of Figure 2 can be illustrated in Figure 6.

In Figure 6, cluster **CL1** is the result of a series reduction of "pseudo-devices" that was executed after the parallel reduction of transistor pairs **M1**, **M3** and **M2**, **M4.**

## Conclusion

*Guardian LVS* from Silvaco features several options for both parallel and series transistor reduction. These reductions improve interchangeability and help to reduce many of the problems associated with transistor reduction.

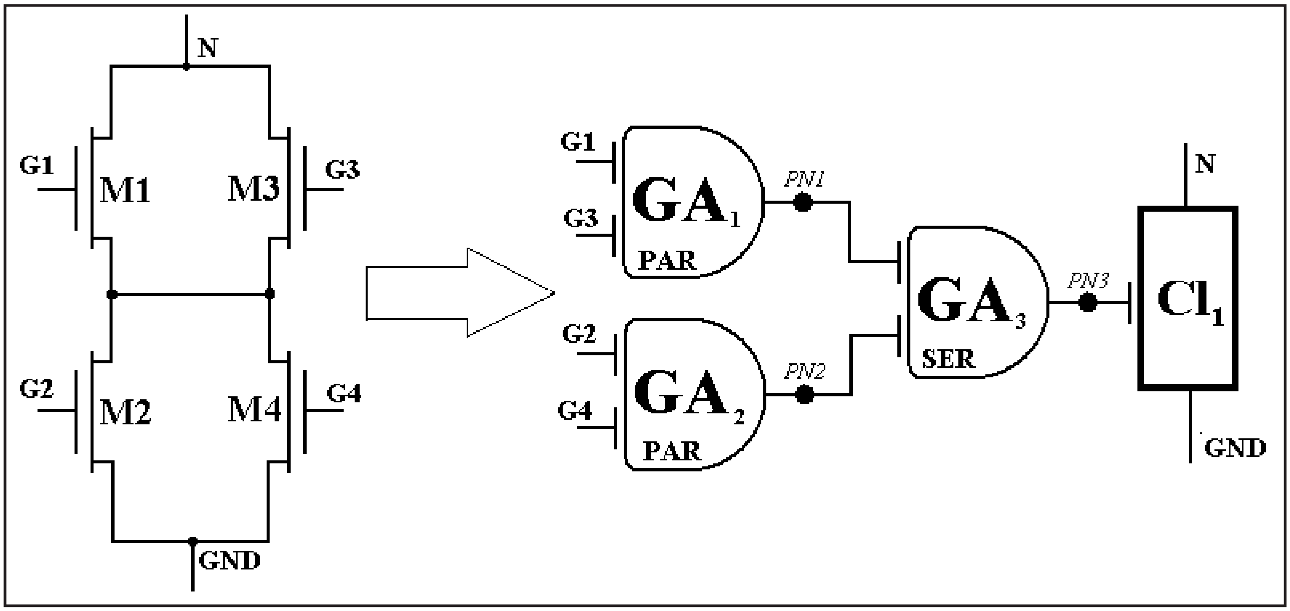


Figure 6. Reduction for series-parallel network realized in Guardian LVS tool.

# Calendar of Events

## March

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 ICMTS - Monterey, CA
18 ICMTS - Monterey, CA
19 ICMTS - Monterey, CA
20 ICMTS - Monterey, CA
21
22
23
24
25
26
27
28 Japan Applied Physics - Japan
29 Japan Applied Physics - Japan
30 Japan Applied Physics - Japan
31

## April

1 GOMACTech, Tampa, FL
2 GOMACTech, Tampa, FL
3
4
5
6 ISPD - Monterery, CA
7 ISPD - Monterery, CA
8 ISPD - Monterery, CA
9 ISPD - Monterery, CA
10
11
12
13
14 ISPSD - Cambridge UK
15 ISPSD - Cambridge UK
16 ISPSD - Cambridge UK
17 ISPSD - Cambridge UK
18 ISQED - San Jose, CA
19 ISQED - San Jose, CA
20 ISQED - San Jose, CA
21
22
23
24
25
26
27
28
29
30

## Bulletin Board

### *Spring 2003 Release of CDs and Manuals*

Silvaco has just completed the Spring 2003 software release that includes new baseline editions of the following 5 CDs—all with newly revised and reprinted manuals (21 volumes):

- ***Virtual Wafer Fab*** with ***ATHENA***™ Process Simulation, ***ATLAS***™ Device Simulation and ***MERCURY***™ FastFET Device Simulation.
- ***UTMOST III*** Data Acquisition and Parameter Extraction System
- ***SmartSpice***™ Analog Circuit Simulator and Scholar™ Schematic Editor
- ***CELEBRITY***™ Layout Design and Verification Tools
- ***DISCOVERY***™ Parasitic Extraction Tools

For latest release numbers see:
**http://www.silvaco.com/latest_releases.html**

For platform support see:
**http://www.silvaco.com/products/platforms.html**

These tool releases contain fixes for known problems and offers many new features and capabilities requested by our customers. They include performance improvements such as optimized execution speed and larger capacity handling. This release includes support for the most popular platforms including Solaris, Linux, HP-UX, Windows NT, Windows 2000, and Windows XP. The release has passed our quality assurance tests and was further field-tested by select customers. Software updates are available for ftp download by request, free of charge, to current maintenance customers. Printed manuals are available for purchase. Call your Silvaco representative for details.

*For more information on any of our workshops, please check our web site at **http://www.silvaco.com***

The Simulation Standard, circulation 17,000 Vol. 13, No. 3, March 2003 is copyrighted by Silvaco International. If you, or someone you know wants a subscription to this free publication, please call (408) 567-1000 (USA), (44) (1483) 401-800 (UK), (81)(45) 341-7220 (Japan), or your nearest Silvaco distributor.

# Hints, Tips and Solutions

Mikalai Karneyenka, Applications and Support Engineer

**Q: When I load *Expert* settings from a file sometimes I cannot see cell names in dialogs, see Figure 1.**

A: Probably your settings were saved on a computer with significantly different screen resolution. In this case sometimes sizes of some user interface features are calculated incorrectly. In your case the width of the column with cell names is wrong. Fortunately, it can be fixed easily, see Figure 2, where some other bad cases are shown.

Do not forget to save the updated settings.

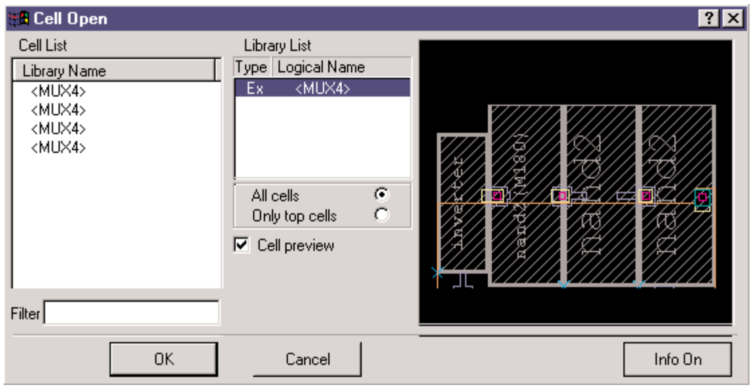
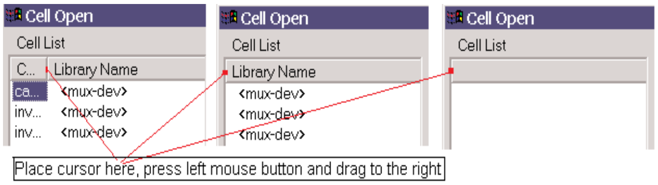


Figure 1. Problem with cell names.

**Q: I generated some layers in Expert by DRC script run, and after that, derived layers can not be cleaned by "Tools\Derived Layers\Clean Derived Layers\Whole Project", otherwise, derived layers generated by *Maverick* can be cleaned this way. How can I clean them all?**

A: There are two types of generated layers in ***Expert***: Derived and Scratch. You can see the type for each layer in Setup>>Technology>>Layer/Wire Setup. Derived layer requires derivation statement to be included in layer definition in technology. If a layer has flag Derived, click on 'Derivation' to see corresponding DRC command(s) for this layer. All layers, that marked Derived, can be cleaned by "Tools\Derived Layers\Clean Derived Layers\Whole Project".

NEW layer generated by DRC script has flag Scratch. All layers, that have checkbox Scratch checked, can be cleaned by "Tools\Scratch layers\Clean Scratch layers\Whole project". Scratch layers also can be completely deleted from technology by "Delete Scratch Layers from Technology" command. If DRC script used to generate shapes in layers, already listed in technology (e.g., input layers), they will not get this Scratch mark, so input layers can't be accidentally deleted from the project by this one command. You can use DRC script run for layer generation in ***Maverick*** as well. In fact, layer generation by script could be made more effective. In this case set "Use DRC Script File" instead of "Use Current Technology" for layer generation in ***Maverick*** Netlist Extractor Setup.



Place cursor here, press left mouse button and drag to the right

Fig.1a,b,c: Possible cases of wrong column sizes in cell lists

Figure 2. Fixing bad column width.

## Call for Questions

If you have hints, tips, solutions or questions to contribute, please contact our Applications and Support Department
Phone: (408) 567-1000 Fax: (408) 496-6080
e-mail: ***support@silvaco.com***

## Hints, Tips and Solutions Archive

Check our our Web Page to see more details of this example plus an archive of previous Hints, Tips, and Solutions
***www.silvaco.com***

SILVACO
INTERNATIONAL

### USA Headquarters

Silvaco International
4701 Patrick Henry Drive
Building 2
Santa Clara, CA 95054
USA

| | |
|---|---|
| Phone: | 408-567-1000 |
| Fax: | 408-496-6080 |

sales@silvaco.com
www.silvaco.com

### Contacts:

**Silvaco Japan**
jpsales@silvaco.com

**Silvaco Korea**
krsales@silvaco.com

**Silvaco Taiwan**
twsales@silvaco.com

**Silvaco Singapore**
sgsales@silvaco.com

**Silvaco UK**
uksales@silvaco.com

**Silvaco France**
frsales@silvaco.com

**Silvaco Germany**
desales@silvaco.com

*Products Licensed through Silvaco or e*ECAD*

E*ECAD PARTNER