# Simulation Standard

Connecting TCAD To Tapeout

A Journal for CAD/CAE Engineers


## Role of Netlist Extraction in PDKs

### Introduction

Netlist extraction and the quality of netlist extraction is becoming of increasing concern for integrated circuit design flow. As circuits become more complicated with concomitant reductions in geometry, the design engineer faces the ever burgeoning demand of accurate of accurate netlist extraction. This simulation standard will detail this important area of netlist extraction and will provide an insight into Silvaco International's netlist extraction tools to aid the circuit design engineer.

### The Role of Netlist Extraction

#### What is NLE in a PDK?

A critical procedure in Custom/Analog Integrated Circuit Design Flow is Layout Netlist Extraction. Of particluar importance is the accuracy and completeness of the Layout Netlist Extraction for a productive design flow. Your design center may be staffed with the brightest circuit designers and use the most accurate models in the industry, however, your project will not be guaranteed first spin success with your silicon unless your PDK (Process Design Kit) contains a proper system for Netlist Extraction. First spin success is essential for meeting the demands of an ever diminishing required time to market, beating out your competition and producing a working product for your customer. The Extracted Netlist provides the means for physical verification. It's superior properties deliver the fastest, most cost effective way of debugging your circuit thus minimizing expenditure. Simply, make sure it is correct before you commit it to silicon, it is a direct descendent of the old phrase, "Measure twice, cut once".

#### What is an Extracted Layout Netlist?

One must have a general familiarity with the netlist before an Extracted Layout Netlist will have any meaning. A netlist is a file which describes electrical components and their interconnect. It is used primarily for simulation. A netlist can be created by a variety of methods. Most commonly, a netlist is created from a schematic entry tool. This is called a schematic netlist. A Layout Netlist is a netlist that describes an integrated circuit layout, while an Extracted Layout Netlist is a netlist that describes an integrated circuit layout, and is automatically created from the layout.

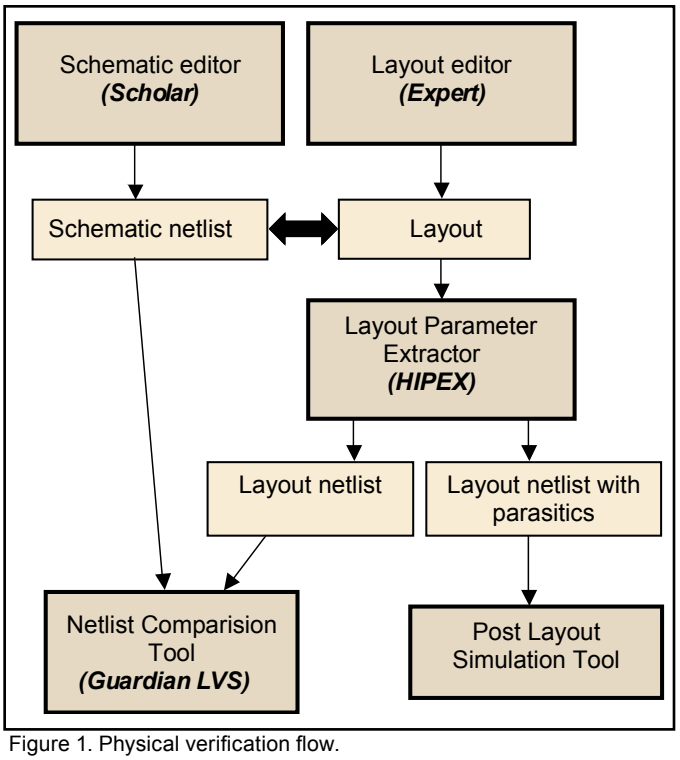


Figure 1. Physical verification flow.

*Continued on page 2 ...*

*INSIDE*





SILVACO INTERNATIONAL

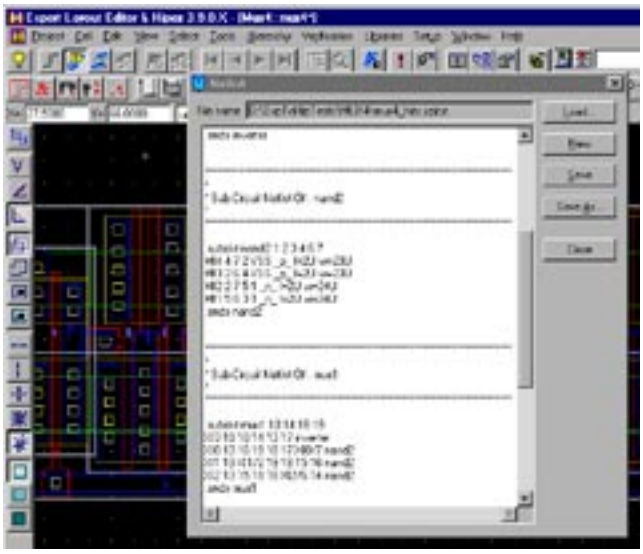

Figure 2. Example of Extracted Layout Netlist obtained by ***HIPEX-NET.***

### What is a layout netlist used for?

An extracted layout netlist is primarily used for physical verification (Figure 1). This physical verification can manifest itself in several ways. A common and efficient means is LVS (Layout vs. Schematic) verification. This is done to ensure that the semiconductor layout represents the circuit design. In many businesses, there are two separate staff: one that implements the mask layout and one that designs the circuit. LVS is the safety check that ensures that the mask designer has created a circuit layout that properly represents the circuit designer's intent. This is very comparable to a building inspector making sure the carpenters have followed the architect's plans. Another use for the Extracted Layout Netlist is Post Layout Simulation. The circuit designer can perform simulation testing on the physical mask layout. This can not only determine if the circuit, as it is laid out, works, but can also incorporate parasitic circuit elements that are now present in the physical circuit as a result of the layout style. These parasitic elements can affect the performance, and post layout simulation can be used as a final check to determine if the circuit design still falls within the targeted datasheet specifications.

### What creates a layout netlist?

A layout extraction engine processes the layout to create a layout netlist. The generic terminology for this tool is a NLE or a netlist extractor. Silvaco International offers two different NLEs. Our first product is ***Maverick***, which is a flat netlist extractor. It processes the layout by first flattening it, and then determining the devices and interconnect. This works well for smaller integrated circuits with less than 50,000 components, but as circuits get smaller and more complex, a flat NLE starts to suffer noticeable speed penalties. To overcome this problem, Silvaco International has developed ***Hipex-NET*** which utilizes a hierarchical approach to NLE. Through this novel hierarchical approach, the NLE can therefore take advantage of repetitive structures which is of paramount concern for large layouts that feature repetitive cells. For example ***HIPEX-NET*** successfully extracts layout hierarchical netlists with several millions devices on Win32 platform. The layout ***HIPEX-NET*** netlist is extracted in a fraction of the time and is fully compatible with ***Guardian LVS*** (Figure 2).

### What Data is Processed?

A NLE requires a number of elements in order to extract the netlist. The first, obviously, is a layout. With a layout, there comes an assignment of process-dependent GDSII layers. The skilled layout designer can create masks to create devices using these GDSII layers. The next element of importance is the technology file. Within the technology file, there are three important pieces of information the NLE uses to extract the netlist from the layout. It contains layer generation rules for creating pin layers and ID layers, as well as the device definitions that are comprised of the layers. The wiring interconnect information also resides within the technology file. An example of a technology file for ***HIPEX-NET*** is shown on Figure 3.

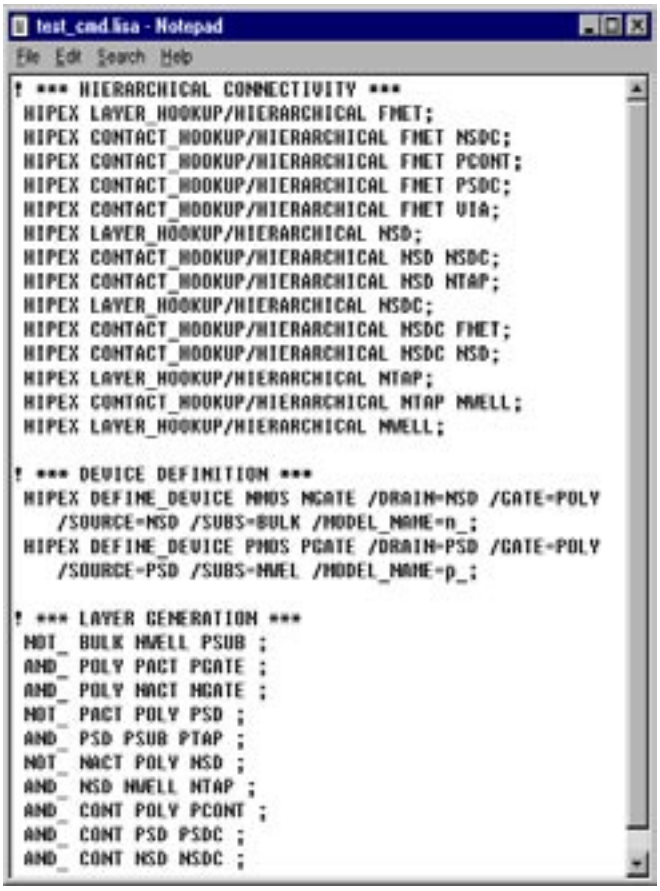


Figure 3. Fragment of technology file for Hipex-NET.

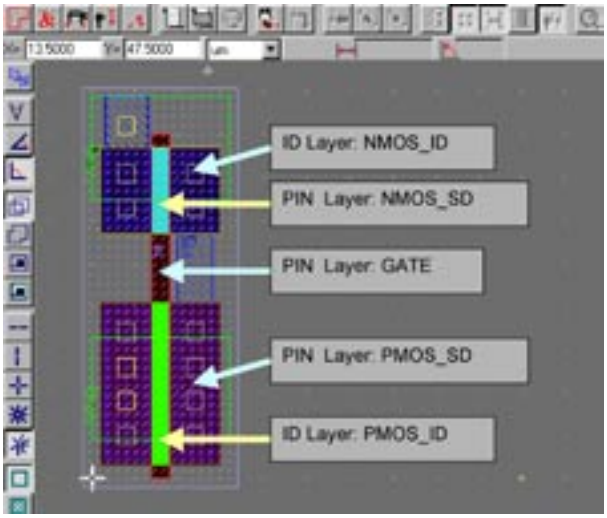

Figure 4. Example of ID and PIN layers for MOS transistors.

**How is a Device Recognized?**

The goal of the NLE is to locate and generate a netlist of devices that are located within the layout. A device is defined by an ID layer, a device type and its corresponding pin layers and is summarized in figure 4. For example, an n channel MOSFET can be defined by the ID layer NMOS_ID and its corresponding pin layers NMOS_S/D, Gate and P SUBSTRATE, where as a p channel MOSFET can be defined by the ID-Layer PMOS_ID with its corresponding PMOS_S/D, Gate and NWELL. These characteristics can be setup using the device setup form as shown in figure 5. This form is then used by the NLE running within the ***Expert*** environment. For example, the NLE scans the layout in order to identify ID layers as defined by the user. Each and every occurrence in the layout where it identified an ID layer, the NLE checks to see that all the necessary pins (those defined in the device definitions section of the technology file) are touching the ID layer. If this condition is met, we say that there is a recognized device, and the NLE will place this device in the netlist.

## Factors Affecting Netlist Extraction Performance

**How can we rate the quality of device definition?**

Quite Simply, in two ways: Speed and Accuracy.

The **accuracy** can be a limiting factor in the versatility of the PDK. Some PDK developers lack the knowledge of device physics, and produce code that restricts the layout styles of the intended devices, they consider only the device and not the process itself. This means that some NLE code will only extract a device if the layout is drawn in a certain fashion. This can severely restrict the freedom of a designer to create custom layouts, and specially tailor the device for its applications. This can prohibit the use of the most robust and compact layout, which can be a penalty in both cost and performance. This phenomenon is very prevalent in Bipolar Devices and special applications of devices such as power, high current or precision matching. This lack of device physics knowledge in the NLE code not only limits its ability to recognize a variation of a certain device, but can also cause failure to recognize *unintended* devices such as parasitic bipolar transistors or tub diodes. The accuracy of NLE code PDK can therefore affect the size of your layout, and also fail to recognize unintended devices which may prevent the circuit from properly operating.

The **speed**, or the time it takes to extract a netlist, is dependent on the size of the layout and the number of devices in the process. The speed can be optimized in a number of ways. The first way is to use a minimal amount of layer operations to define the pin and ID layers. This means it is good to use the same layer for several pins of different devices, i.e. an n-well is the body of a p-MOS, a collector of an NPN, and the base of a PNP device. Another way to optimize the speed is to pay attention for duplicate layer operations in different pin or ID layer definitions, and make them global. The time savings through these optimizations may seem trivial, but as the device count goes up, and designs go through the iterative process of verification, debugging and updating on their way to tapeout, the time savings become evident.

## 4.0 Conclusion

This simulation standard has described netlist extraction using advanced netlist extraction tools. Through sophisticated algorithms layouts can be analyzed with great speed and accuracy through the use of advanced netlist extraction tools provided by Silvaco International.

Figure 5. Example of ***Expert*** Device Setup form.

# Parasitic Resistor Extraction with *HIPEX-R*

## Introduction

Layout parasitic extraction (LPE) plays an important role in the post-layout verification process. Verification of the full chip layout is becoming a more common problem because the size of IC layouts continue to grow due to new IC technologies. The complexity of IC design requires consideration of the effects caused by parasitic capacitors and resistors. Parasitic devices are responsible for such effects such as time delay, voltage drop, and signal integrity violation, all of which impair chip performance.

***HIPEX-R*** is part of the ***HIPEX*** software package for physical verification of multimillion transistor designs. It is a hierarchical full chip parasitic resistance extraction tool.

The ***HIPEX-R*** flow is depicted in Figure 1.

***HIPEX-R*** can read either a GDSII or SLF input design.

The rules according to which ***HIPEX-R*** will extract the parasitic resistors are set in the technology file. Note that ***HIPEX-R*** also extracts the active device defined in ***HIPEX-NET*** technology file (which is also used by ***HIPEX-R***).

A set of options for customizing the extraction are defined in the option file.

Once the extraction is done, ***HIPEX-R*** generates a hierarchical and/or flat SPICE netlist, as well as a GWL/WLDS proprietary format hierarchical netlist (used by ***DistRC*** utility for RC distribution). All the information related to the extraction is output to a summary file.

***HIPEX-R*** can also optionally generate a "stripe" database (SDB), and a resistance database (RDB), containing geometrical and electrical parasitic informations.

The SDB can be used by ***HIPEX-C*** to extract parasitic capacitors, while the RDB will be used by the ***DistRC*** utility (together with the CDB produced by ***HIPEX-C***) for RC distribution.

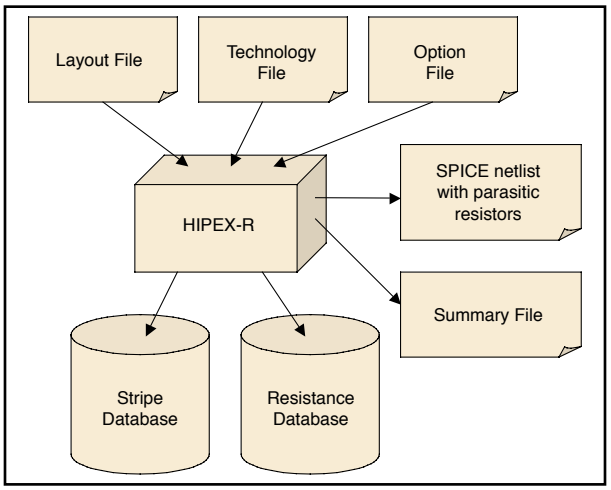

Figure 1. Hipex-R flow.

## Features

***HIPEX-R*** can extract two types of parasitic resistors: horizontal resistors and vertical (contact) resistors. It extracts these resistors on the layers specified by the user in the technology file.

The parasitic extraction is based mainly on a fragmentation algorithm that guarantees a decomposition of the parasitic layer into resistors (body and terminals) along an estimated current direction.

During the fragmentation, terminals may be created wherever there are bends (L-shapes, T-shapes, Cross-shapes) or wherever the layer to be extracted touches or overlaps another layer.

Figure 2 represents the main configurations handled by ***HIPEX-R*** fragmentation.

Once the fragmentation is done, the resistor values are calculated according to user-specified sheet and/or area

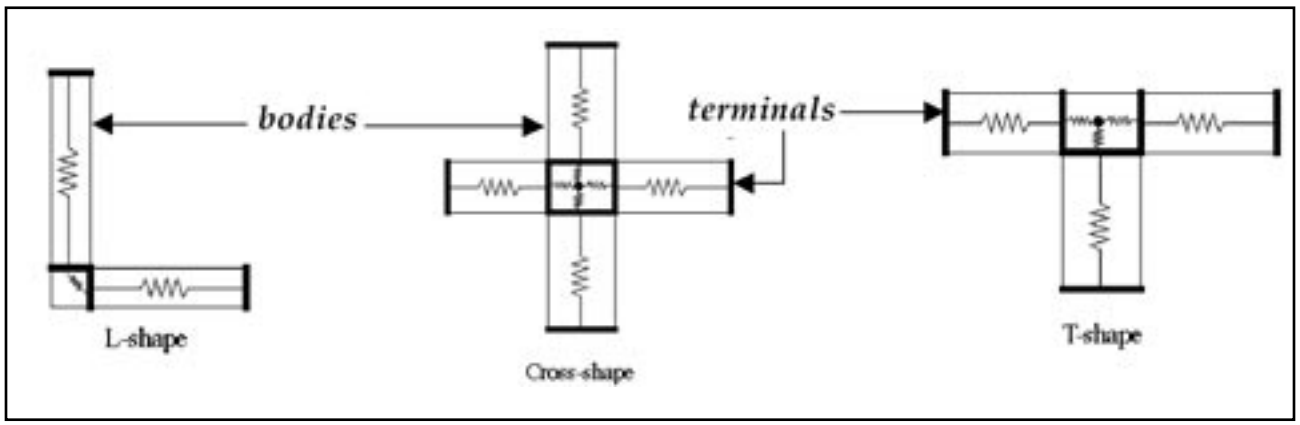

Figure 2. Hipex-R layer fragmentation configurations

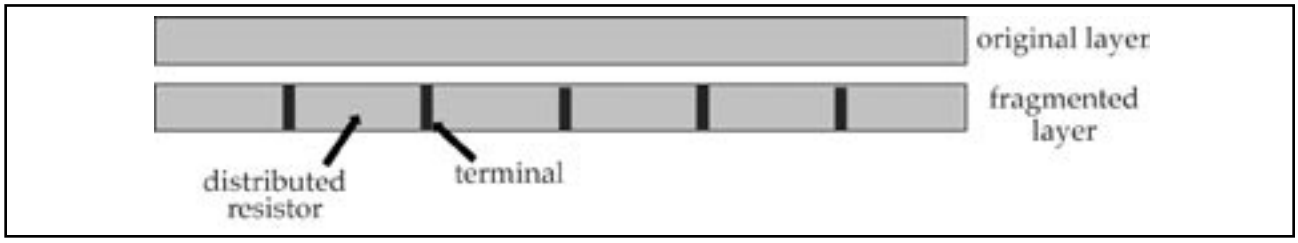

Figure 3. Long resistor fragmentation

resistivities, and the connectivity between the different resistors and the active device is then established or updated.

Finally, parasitic subnode names are assigned to the resistor terminals, and the active device terminal names are updated based on the connectivity information.

The extraction of parasitic resistors is done hierarchically, starting from the deepest substructure of the design. The hierarchical connectivity between instances of substructures in the design is updated throughout the extraction.

***HIPEX-R*** enables you to control the output of the extraction with multiple options.

The main option are described below.

*Fragmentation options*

- ***HIPEX-R*** enables you to specify the maximum length of a fragmented resistor body. If a resistor body is greater than this value, it will be cut into smaller pieces of length less than or equal to this value. (see Figure 3)
- Contact oversizing and contact clustering are two other options that can be very useful for reducing the number of extracted body resistors and simplify the fragmentation, see Figure 4

*Extraction options*

- ***HIPEX-R*** enables you to specify a resistance threshold value. All parasitic resistors found with a resistance less than this threshold will be ignored while connectivity is preserved
- ***HIPEX-R*** extraction can be performed either on the whole design or on a specified cell
- ***HIPEX-R*** extraction can be performed on a set of selected nets. It is also possible to ignore a set of specified nets as well as dangle nets

*Output options*

- ***HIPEX-R*** generates the parasitic netlist in hierarchical and/or flat SPICE format, as well as in hierarchical GWL/WLDS format. Geometrical and physical information can also be outputed to the netlist, such as the coordinates of the parasitic resistors, their size and their layer
- ***HIPEX-R*** can output a layout of all parasitic resistors' geometries (body and terminals) and parasitic nodes as text elements
- Finally, ***HIPEX-R*** can be used to generate the stripe database needed by the capacitor extractor ***HIPEX-C***, and the resistance database needed by ***DistRC*** utility for RC distribution

## Conclusion

An introduction to ***HIPEX-R*** features and benefits was presented in this article. Thanks to its hierarchical architecture, this full chip parasitic resistance extraction tool provides quick and efficient algorithms that are able to handle large IC designs. ***HIPEX-R*** can be used as a stand-alone product or as part of the ***HIPEX*** flow.

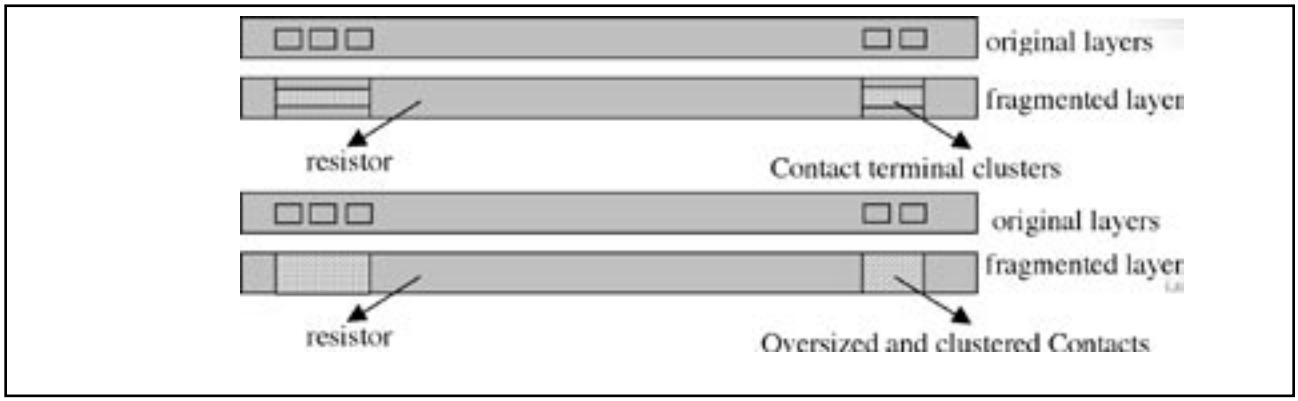

Figure 4. Contact oversizing and clustering

# HIPEX-NET:
# New SILVACO Full-Chip LPE Tool vs. Maverick

SILVACO is releasing its new layout parameter extractor: ***HIPEX-NET***. The new tool will replace Silvaco's ***Maverick***, which is a part of ***Guardian*** LVS/ERC. While being fully compliant with ***Maverick***, ***HIPEX-NET*** has many advantages over ***Maverick***. The comparison chart in Table 1 shows the critical features, which make ***HIPEX-NET*** much more powerful for layout verification than ***Maverick***.

***HIPEX-NET*** is a full-chip hierarchical netlist extractor. It can handle very complex layout hierarchy, this is in stark contrast to ***Maverick*** , where ***Maverick*** must flatten the layout. As a result, ***HIPEX-NET*** works much faster and needs less memory to process big hierarchical layouts. Today ***HIPEX-NET*** successfully extracts 10 million transistor layouts on Win32 platform, which provides only 2GB of virtual memory. The 64-bit version available for SunOS can handle much larger layouts.

Unlike many other hierarchical tools on the market, ***HIPEX-NET*** doesn't require additional efforts when preparing a mask layout. The user doesn't have to define cell ports. ***HIPEX-NET*** performs best with a true hierarchical design, however it also can deal with hierarchy violations by exploding individual cells. The user has means to make ***HIPEX-NET*** identify hierarchy violations and explode appropriate cells automatically during the extraction. Since this decreases performance, the better way is to indicate cells for explosion manually before starting extraction (pre-exploding). When using ***HIPEX-NET*** within the Expert framework, it is possible to run ***HIPEX-NET*** for hierarchy checking only. Then, the user can easily pre-explode cells from the list generated during the checking.

Besides greater performance, hierarchical extraction produces results that are easier to understand. A hierarchical netlist is much more compact and readable. The new Node Probing implementation in ***Expert***, which is now based on the layout annotation database produced by ***HIPEX-NET*** rather than ***Maverick***, allows one to trace a node path through nested instances of cells. Besides nets and devices, it is now possible to highlight separate instances by their name or by clicking on them with the mouse.

Unlike ***Maverick***, ***HIPEX-NET*** properly uses the design text to name nets. As a hierarchical extractor, it allows the designer to use text labels of the three types: global, local, and port. Global text labels have the highest priority and thus name nets in the whole layout. Local text labels nets in a given cell. Port text label have the lowest priority and are intended to name user-defined cell ports only (it is not necessary but possible to predefine cell ports in ***HIPEX-NET***).

***HIPEX-NET*** also uses the design text to perform Electric Rule Checking (ERC). ERC checks for open and short circuit nets. ***HIPEX-NET*** reports a global (local) open if two or more unconnected nets are named by the same global (local) text label. A short circuit is reported if one net is named by two or more different text labels of the same priority. In addition, ***HIPEX-NET*** can report dangles (nets that have no device connections), badly formed devices (e.g., with missing pin connections or with shorted pins), and perform an integrity check. ***HIPEX-NET*** writes in the summary file detailed information about every ERC error, including layout local coordinates, layer names, and text labels.

| Feature | HIPEX-NET | Maverick |
|---|---|---|
| SPICE netlist extraction | Hierarchical and flat | Flat only |
| Parameter extraction of transistors (MOSFET, MESFET, JFET, BJT) and design diodes, resistors, and capacitors | Yes | Yes |
| Integration with SILVACO layout editor, ***Expert*** | Yes | Yes |
| Hierarchical extraction | Yes | No |
| Advanced processing of net names | Yes | No |
| Schematic backannotation | Yes | No |
| Compatibility with ***HIPEX*** parasitic tools | Yes | No |

Table 1. HIPEX-NET vs. Maverick: features comparison chart

As part of the ***HIPEX*** family, Silvaco introduces a full-chip parasitic tool: ***HIPEX-RC.*** It divides parasitic extraction into three stages (see Figure 1). First, ***HIPEX-R*** performs parasitic resistance extraction. It produces the layout SPICE netlist annotated with parasitic resistances and the internal resistance database (RDB). ***HIPEX-R*** also creates an internal net database that stores geometrical and electrical information about all the nets, including parasitic subnets that correspond to parasitic resistor terminals. Second, the net database is used as input to ***HIPEX-C*** to extract parasitic capacitances between nets. ***HIPEX-C*** produces the SPICE netlist containing parasitic coupling capacitors and the internal capacitance database (CDB). Finally, RDB and CDB are combined into the distributed parasitic RC-network either in SPICE or DSPF format. This can be further processed by the RC-network reduction tool: ***HIPEX-CRC***.

Since both ***HIPEX-R*** and ***HIPEX-C*** produce ready-to-use SPICE netlists, they can be used separately. This way ***HIPEX-NET*** is used to create the net database (without parasitic subnet information) needed by ***HIPEX-C***.

If the layout netlist extracted by ***HIPEX-NET*** passes LVS verification, the designer can backannotate the schematic netlist with parasitic capacitances and resistances extracted by ***HIPEX-RC***.

***HIPEX-NET*** can be used from ***Expert*** or in standalone mode. The extractor is available for Win32, SunOS 32- and 64-bit, and Linux platforms.

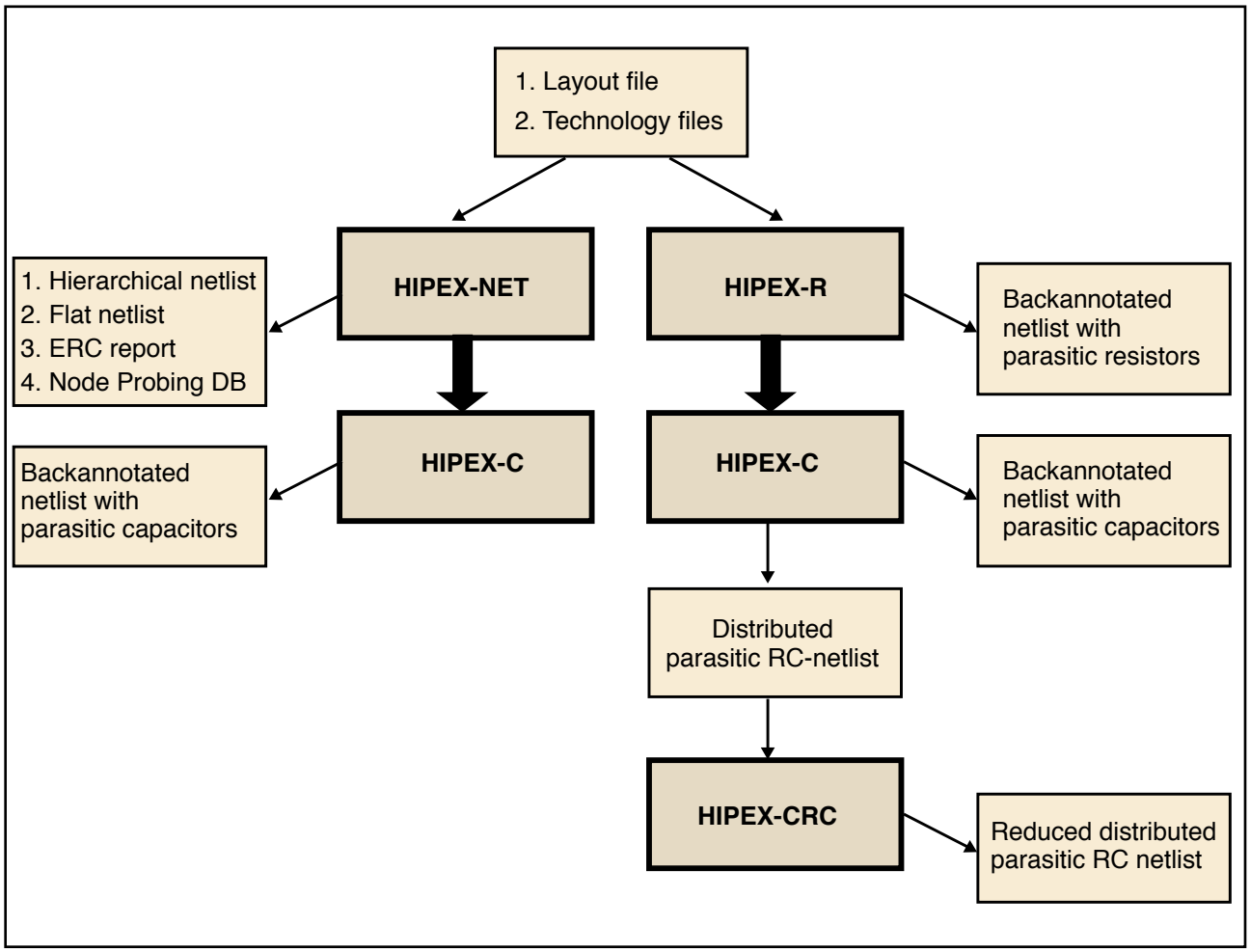


Figure 1. Hipex execution flow.

# Measurement of Spacing Checks

## 1. Introduction

In previous versions of ***Guardian*** DRC there were only two ways of measuring the distance between two segments: the ordinary, **Euclidean metric,** and the **square metric**, which behaves differently when measuring distances from a corner of a shape. With decreasing feature sizes Euclidean metric does not always provide the adequate measurement of tolerances required during the IC fabrication. Therefore various DRC systems introduced other types of measurement. This article describes how Guardian DRC system performs measurements required for the execution of DRC spacing checks, which are based on separations between line segments ("width", "indistance", "outdistance", "ovdistance", "distance", "compdistance").

## 2. Types of Measurement

Every spacing check for each layer(s) has to have a constraint that is evaluated for each particular edge of layout object of layer(s) that participate in this check. The full list of these constraints for all spacing checks is:

| Old syntax | New syntax | Meaning |
|---|---|---|
| LT a | limits < a | x < a |
| LE a | limits <=a | x <=a |
| EQ a | limits ==a | x ==a |
| RANGE a,b | limits >a <b | a < x < b |
| NE a | limits != a | x != a |
| GE a | limits >= a | x >= a |
| GT a | limits > a | x > a |
| GE_LE a,b | limits >= a <=b | a <= x <= b |
| GE_LT a,b | limits >= a < b | a <= x < b |
| GT_LE a,b | limits >a <=b | a < x <= b |
| GT_LT a,b | limits > a < b | a < x < b |

Graphically, these constraints can be depicted as shown in Figures 1 through 4.

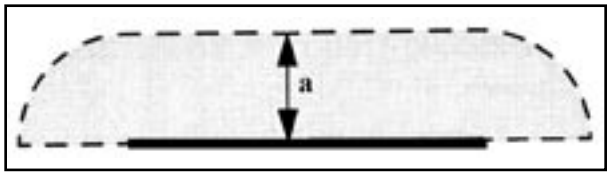

Figure 1. Graphical representation of constraints:LT, LE

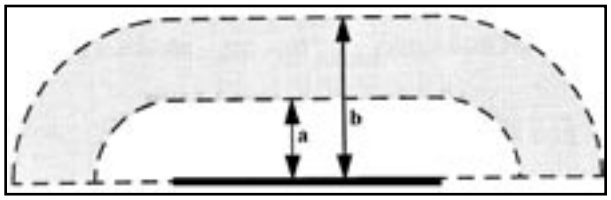

Figure 3. Range relation

If the edges (points) of another evaluated object fall into a violation ("gray") zone then these edges (points) are considered as errors that violate the checked rule. The side of an edge to construct the violation zone is chosen according to the particular check operation.

## 3. Other metrics

Figures 1 through 4 show the violation zones when the Euclidean metric is used.

There are several other metrics used by Guardian DRC. The easiest way to explain the differences between these metrics is by means of the violation zones shown on graphical examples in figures 5 through 7. For the Euclidean metrics, the violation zone may be constructed as the union of circles (or donuts) centered at each point of the checked segment and then cut by the appropriate half-plane. As a result, the violation zone is a rectangle along the checked edge augmented by two quarter-cycles (or quarter-donuts), see, e.g., Figures 1, 3 and 4. The violation zone for checks of type GT and GE are the complements of the violation zones of type LE and LT respectively.

**Square metric** is evaluated by replacing the circle (donut) by the square. As a result, the violation zone is a rectangle along the checked edge augmented by "quarter-squares", see Figure 5.

In the **opposite metric** the violation zone is not extended past the ends of the checked edge, see Figure 6.

The **extended opposite metric** is the generalization of both the square and the opposite metrics: the violation zone is extended past the ends of the checked edge by the specified value, see Figure 7. (In the square and Euclidean metrics, the extension value(s) is the same as the check limit value(s)).

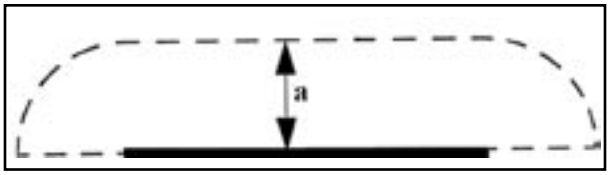

Figure 2. Graphical representation of constraints NE, EQ.

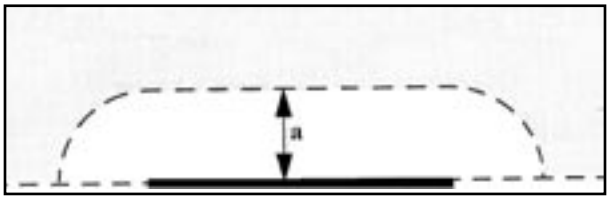

Figure 4. Graphical representation of constraints: GE. GT.

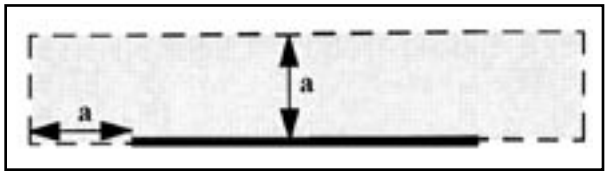


Figure 5. Graphical representation of Check operator with **Square** metric: LT a, Metric=Square

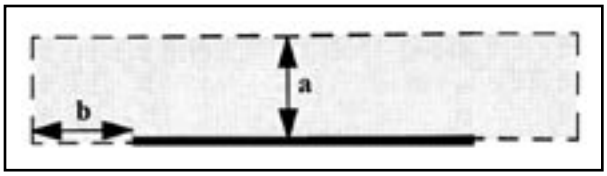


Figure 7. Check operator with **Extended Opposite** metric: LT a, Metric Extended=Opposite(b)

## 4. Reported Subsegments

Guardian DRC system may report DRC violations in several forms. They may be classified into two main groups: full segment report and subsegment report. For the first group, the report is based on the pair of complete edges for which a distance violation is detected. In the second group, only edge subsegments that fall into the violation zone of some edge are reported.

In some cases when a violation is reported for a pair of edges, it is not immediately clear how the subsegments on the two edges are related to each other. For example, consider the case of checking for limits 7 <x < 10, shown in Figure 8. It is not immediately clear why point B is not reported, since the distance from A to B is clearly between 7 and 10. The reason becomes evident if we draw the violation zones for this check, see Figure 9.

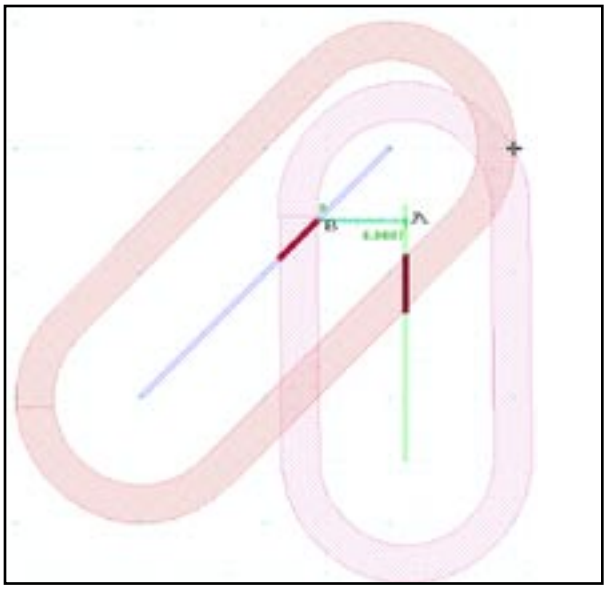

Figure 9. Violation zones for the example from Figure 8

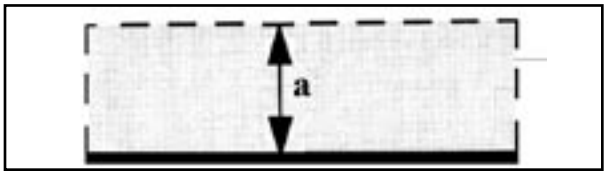


Figure 6. Graphical representation of Check operator with **Opposite** metric: LT a, Metric=Opposite

Figure 10 shows a more complex interaction of two segments with reported violations.

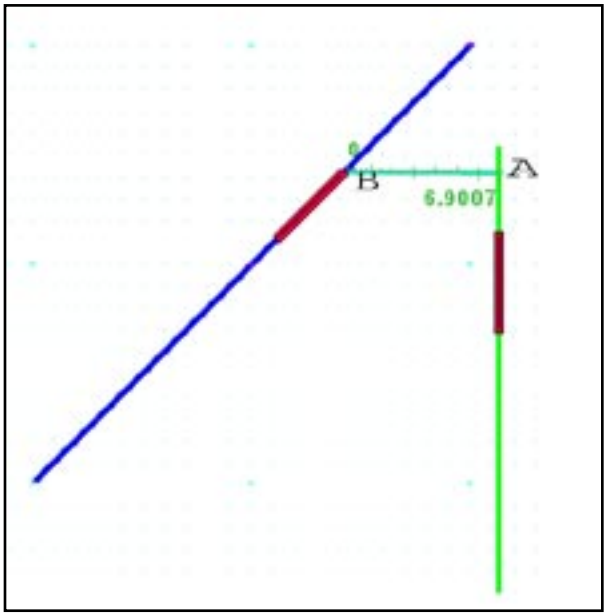


Figure 8. Example error subsegments for limits >=7 <= 10

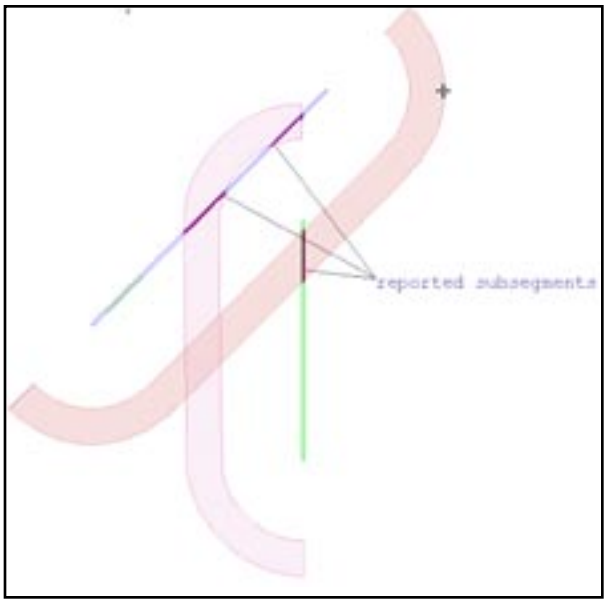


Figure 10. Violation zones for limits >=7 <=10

# Calendar of Events

## September

| | |
|---|---|
| 1 | |
| 2 | |
| 3 | SISPAD - Cambridge, MA |
| 4 | SISPAD - Cambridge, MA |
| 5 | SISPAD - Cambridge, MA |
| 6 | EGAOSAS - Munich, Germany |
| 7 | EGAOSAS - Munich, Germany |
| 8 | |
| 9 | GaSa IC Symposium-San Diego |
| 10 | GaSa IC Symposium-San Diego |
| 11 | GaSa IC Symposium-San Diego |
| 12 | GaSa IC Symposium-San Diego |
| 13 | |
| 14 | BCTM - Toulouse, France |
| 15 | BCTM - Toulouse, France<br>RADECS - Netherlands |
| 16 | BCTM - Toulouse, France<br>RADECS - Netherlands<br>SSDM - Tokyo, Japan<br>ESSDERC - Portugal |
| 17 | SSDM - Tokyo, Japan<br>ESSDERC - Portugal |
| 18 | SSDM - Tokyo, Japan<br>ESSDERC - Portugal |
| 19 | SSDM - Tokyo, Japan |
| 20 | |
| 21 | CICC - San Jose, CA |
| 22 | CICC - San Jose, CA |
| 23 | CICC - San Jose, CA |
| 24 | CICC - San Jose, CA |
| 25 | |
| 26 | |
| 27 | |
| 28 | |
| 29 | IEEE SOI-Newport Beach, CA |
| 30 | IEEE SOI-Newport Beach, CA |

## October

| | |
|---|---|
| 1 | CMRF - Toulouse, France<br>IEEE SOI-Newport Beach, CA |
| 2 | IEEE SOI-Newport Beach, CA |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | SAME - Sophia Antipolis, France |
| 9 | SAME - Sophia Antipolis, France<br>FSA Suppliers Expo - San Jose, |
| 10 | |
| 11 | |
| 12 | |
| 13 | |
| 14 | |
| 15 | |
| 16 | MSED - Barcelona, Spain |
| 17 | MSED - Barcelona, Spain |
| 18 | |
| 19 | |
| 20 | Polytronic - Switzerland |
| 21 | Polytronic - Switzerland<br>ASICON - Beijing, China |
| 22 | Polytronic - Switzerland<br>ASICON - Beijing, China |
| 23 | ASICON - Beijing, China |
| 24 | ASICON - Beijing, China |
| 25 | |
| 26 | |
| 27 | |
| 28 | |
| 29 | |
| 30 | |
| 31 | |

## Bulletin Board

### *CICC*

Visit Silvaco at the IEEE Custom Integrated Circuits Conference, the premier conference for information on leading-edge analog and digital circuits. CICC is dedicated to IC development, showcasing original, first published technical work and innovative circuit techniques that tackle practical problems. Silvaco will be demonstrating its analog custom IC design flow.

### *FSA Mixed Mode/RF PDK Committee*

Silvaco is chairing an FSA working group to:

1. Identify common issues shared by the foundry, fabless, EDA, IP and design service providers in the development and use of PDKs
2. Develop a format for PDK roadmap representation and common terminology for describing PDK contents and inputs
3. Establish Fabless customer requirements for RF/MS PDKs
4. Define PDK quality criteria at critical delivery milestones
5. Streamline the PDK development process

*If you would like more information or to register for one of our our workshops, please check our web site at* ***http://www.silvaco.com***

# Hints, Tips and Solutions

Galina Makovsky, Applications and Support Engineer

**Q: I would like to highlight two nodes at the same time in Expert. I currently use Verification->Node Probing->Pick Node to highlight a node, but I donít see how I can have 2 nets highlighted at the same time.**

A: If the <Shift> key is depressed while the Probe Node or Find Net by Name tools are activated, the picked net becomes highlighted and available for inspection, and the previously selected nets stay highlighted on the screen. When multiple nets are highlighted simultaneously, the active one is highlighted in yellow, all nets previously picked are highlighted in green.

**Q: When Probe Node or Find Net by Name tools are activated for power or ground nets, big highlighted objects from substrate layers often make other node objects hard to see. Is there a way to hide these substrate objects during Node Probing?**

A: Node objects from non-selectable layers are always displayed in a wireframe mode in Expert V.3.6.6.R and above. To see power and ground nets' objects clearly, set substrate layers non-selectable by pressing "Ctrl" when click on layer name, or using layer plan, where substrate layers not included or not selectable. Now all non-selectable layers' objects displayed in wireframe mode independently from filling mode of the probed node set by Verification>>Node Probing>>Node Filling submenu: wireframe, solid or stipple.

**Q: We were wondering if we have a layout with many P-cells, is there a way to simply export all the P-cell commands to either a single .xis file or multiple .xis files?**

A: Parameterized Cell Panel > PCell > Save saves P-cell as *.xis. The following code is automatically written at the beginning of the XI script:

```
DEFINE PCELL "MNL" /REPLACE
  PARAMETER gates /TYPE = (Integer) /DEFAULT = (1)
  PARAMETER W /TYPE = (Double) /DEFAULT = (0.7)
  PARAMETER L /TYPE = (Double) /DEFAULT = (0.25)
  BODY BEGIN
! TODO: Add code here
....
```

If you run this script from XI Script Panel, P-cell "MNL" will be created in current project.

**Q: The DRC command Angle Check with Type=non45 is for region boundary segments with slope not divisible by 45°. The Angle Check with Type=acute checks input layer regions for adjacent edges creating acute inner angles. How can I find segments with slope between say 0 and 10 degrees or regions with acute inner angles within specified limits?**

A: The SLOPE command selects edges with slopes within specified limits. The limits are specified in degrees, with values between 0 and 90.

Slope: Layer=<name>, LayerR=<name>, Limits <range>;

The output is an edge layer. The input layer may be shape or edge layer.

Example:

Slope: layer= M1, layerR=M1Acute, limits >0 <=35;

The SELECT_EDGES: Relation=CORNERS, command selects edges basing on the parameters of the adjacent corners: angles and side lengths. Convex specifies limits for the number of convex angles adjacent to the edge; length specifies limits for the length of the segment itself angle1,2 specify limits for the first/second adjacent angle, in range between 0 and 360 degrees. length1, 2 specify limits for length of the first/second adjacent side.

Select_Edges: Relation=Corners,
[convex=<012-limits>,] [length <limits>,]
[angle1 <0.0-360.0>,][Length1 <limits>,]
[angle2 <0.0-360.0>,][Length2 <limits>,]
layer=<shape-layer>, layerR=<edge-layer>;

Examples:

Select_Edges: Relation=Corners, convex <2, length >10, layer=m1, layerr = outbendM1;

Select_Edges: Relation=Corners, angle1 < 10, layer=m1, layerr = m1Loop;

**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, Bldg. 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*