

# Simulation Standard

TCAD Driven CAD

A Journal for CAD/CAE Engineers

## Cross-Sectional Viewer in *Expert*

### 1. Introduction

*Expert* is extremely powerful circuit layout editing tool with the integrated ability of geometric design rule checking and LVS design verification.

IC technologies continue to develop and become more and more complex. With increase of complexity in the contemporary design processes the designer must have knowledge not only of layout design, but in simulation and manufacturing as well. The more the design engineer knows about the process with respect to layout and technology, the higher the performance he can design.

Cross-Sectional Viewer is a tool within *Expert* to simulate the cross sectional view of ICs along an arbitrary drawn cut-line on the layout. Cross-sectional viewer is a link between the layout and the resulting device. It allows the designer to examine cross-sections of the device being designed. Cross-sectional drawings are useful for understanding design rules, parasitic coupling and other design and fabrication problems.

The current *Expert's* implementation of Cross-Sectional Viewer is based on the simplified simulation of the process steps using rough processing parameters from Expert technology file.

Future enhancements of the Cross-Sectional Viewer will include interfacing with SILVACO process and device simulation tools that will allow the users to obtain physically accurate representations of the physical structure of ICs in the problematic areas of the layout.

### 2. Cross-sectional viewer setup

Approximate parameters for the simulation of fabrication step must be specified in the technology file. For simplicity we assume that each fabrication step is

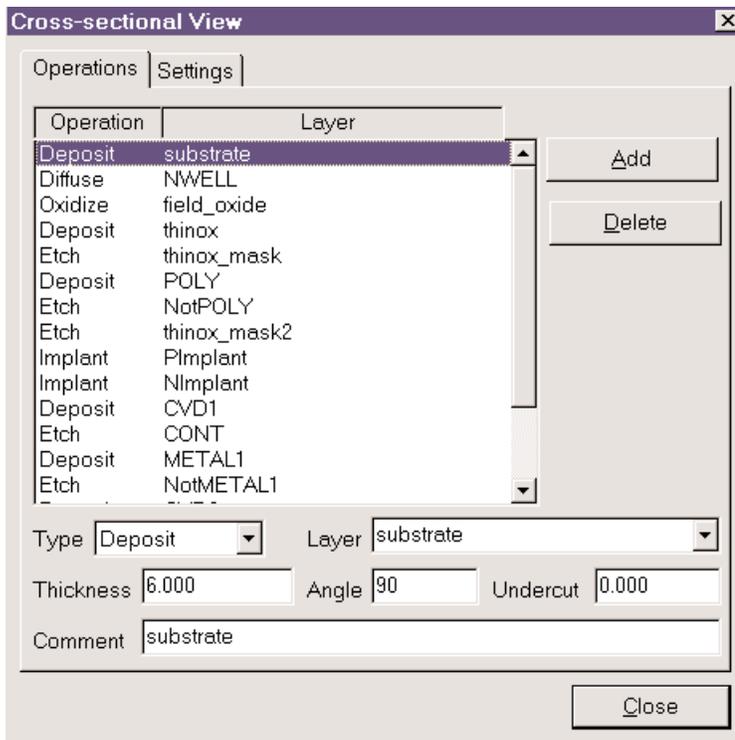


Figure 1. Cross-sectional viewer setup.

associated with a separate layer in the layout. This layer may be either a drawn layer or a derived one, i.e., produced by boolean, resizing or any other DRC operations on the drawn layers.

Continued on page 2...

### INSIDE

<i>Expert: Recent Improvements in Hierarchical Layout Inspection</i> .....	5
<i>Applicability of Distance Computation for Graphs to LVS Discrepancy Analysis</i> .....	7
<i>Calendar of Events</i> .....	9
<i>Hints, Tips, and Solutions</i> .....	10

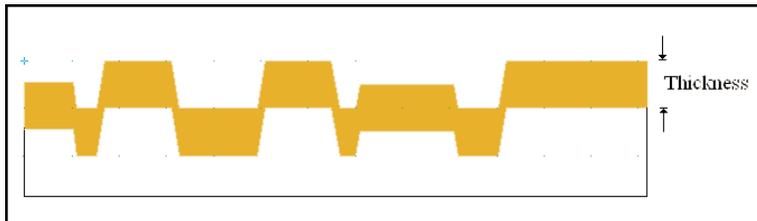


Figure 2. Thickness parameter for deposit step.

## DEPOSIT

This operation is used to simulate metal deposition or oxide growth above the whole surface of the structure, see Figure 2. Parameter 'Thickness' is used for the description of this fabrication step. Thickness must be specified in the units of the technology file for manually editing and in the measurement units of the current cell view if the dialog panel "Cross-sectional View" is used.

There are two ways to create and modify process simulation parameters:

- manually editing *Expert* technology file or
- interactively, using dialog panel named "Cross-sectional View" (Setup>>Cross-sectional Views), see Figure 1.

The Processing and Material sections of layer definition must be presented in the technology file for each processing step as follows:

```

Layer
{
  Name = "NWELL"
  ...
  Material
  {
    ...
    Thickness = 4
  }
  ...
  Processing
  {
    ProcessingStep = 1
    Operation = DIFFUSE
    Undercut = 0
    Angle = 100
  }
}

```

Each processing step is described by the following parameters: Layer Name, ProcessingStep, Operation, Thickness, Undercut and Angle. The 'Layer Name' is the name of layer in the technology. In some cases we must add special layers for the cross-sectional representation of layout. These layers may be without geometry or derived from other layers using derivation rules. 'ProcessingStep' is the number of the step in the fabrication sequence. 'Operation' must be one of the following fabrication steps

- DEPOSIT
- ETCH
- IMPLANT, DIFFUSE
- OXIDIZE.

'Thickness', 'Undercut' and 'Angle' fields are additional parameters for simulation operations described in the following sections.

## ETCH

Etch operation simulates the etching or patterning of a material layer. Parameter 'Thickness' defines the depth of the etching. Note, in the current implementation all layers are etched uniformly regardless. If you want to simulate "undercut" and slant effects that take place during etching, you can use 'Undercut' and 'Angle' parameters, see Figure 3.

## IMPLANT, DIFFUSION

Ion implantation and impurity diffusion processes are described using this step. These operations are used to create n+ and p+ doped regions as well as n- and p-wells. Parameter 'Thickness' indicates the depth of the p-n junction. 'Undercut' and 'Angle' parameters can be used for simulating lateral effects that take place during high-temperature diffusion. For describing implantation and diffusion, the same model as for etch is used, see Figure 3.

## OXIDIZE

This step is to simulate local oxidation of silicon. Parameter 'Thickness' defines thickness of the field oxide. It is assumed that field oxide both recesses into the surface (depth  $T1 = 0.45 * Thickness$ ) and grows above the surface ( $T2 = Thickness - T1$ ). Using additional parameters, "Undercut" and "Angle", you can simulate "bird's beak" effect, see Figure 4.

## 3. Using the Cross-sectional Viewer

Cross-sectional Viewer requests specific information to be presented in the technology of the opened project.

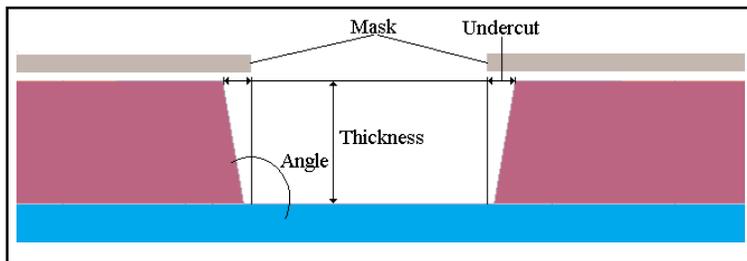


Figure 3. Etching step parameters.

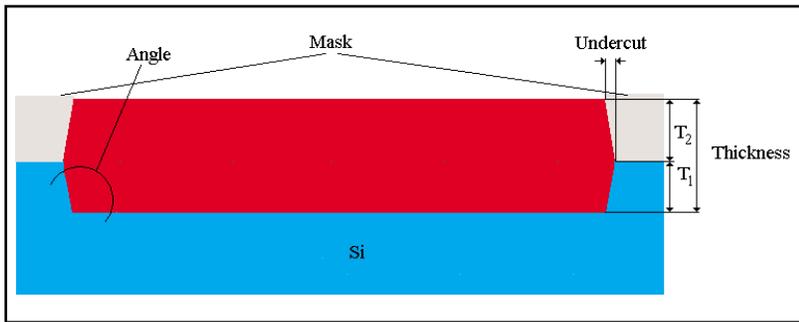


Figure 4. Local oxidation parameters.

The Setup>>Cross Sectional Views or Tools>>Cross Sectional Views>>Setup commands pop up the "Cross-sectional View" dialog panel for adding and editing parameters of simulation of fabrication steps, see Figure 1. The dialog parameters are as follows. 'Add' button is to add a new step into fabrication sequence. You must select the type of the process step from the drop-down list of the 'Type' field and specify the depth or thickness of the fabrication layer using 'Thickness' field. Thickness is measured in the technology units or units of the current cell view if the latter is opened. Additional fields 'Undercut' and 'Angle' can be specified for more realistic simulation of the described processes, see "Cross-sectional viewer setup" section. A fabrication sequence may be changed by dragging the selected operation in the Operation/Layer window. You may delete the selected fabrication step by pressing the 'Delete' button. The 'Settings' tab in the "Cross-sectional View" dialog panel is to select the representation of the opened cell view and the calculated cross-sectional view for a specified cut-line.

To generate a cross-sectional view for the opened cell, one must select "Tools>>Cross-sectional Views>>Generate" command. If no cross views for the current cell exist, the cross-sectional view tool will appear at your cursor. To specify a cut-line, one must move the cursor to a point on the layout view, press and release the left button, drag a line across layout in an arbitrary direction, and finally press the left button to finish the cut-line. After drawing the cut-line, the Cross-sectional Viewer Tool generates the corresponding cross view. The Cross View will appear according to the settings in the "Cross-sectional View" dialog panel, see Figure 5 ('Tile (Cross View and Cell View)',

'Show cross lines on Cell View' options are checked). The Cross View window is like an ordinary Cell View. It can be zoomed to a more close investigation of separate parts of the Cross View. You can also print Cross View as any Cell View to make hard copy using "Project>>Plot" command for documenting the layout.

The "Tools>>Cross-sectional Views>>Generate" pops up the "Cross Views" dialog if there are cross views for the current cell, see Figure 6. You can select the precalculated cross view by pressing the mouse left button or using the 'Down' or 'Up' keys. The cut-line corresponding to the Cross View will be selected on the Cell View and the fields 'First point' and 'Last point' will provide the cut-line coordinates on the layout. The 'New' button is to define a new cut-line and generate the correspond cross-sectional view. You can specify a new cross-sectional view by filling the 'First point' and 'Last point' fields at the 'Cross Views' dialog panel and pressing the 'Apply' button. All cross views are stored in a temporary file while the library is loaded. This allows the user to quickly review the cross views of interest. This feature is handy to get a three-dimensional feeling of critical pieces of the layout: one may precalculate several closely spaced cross-sections and then view them in rapid succession by clicking the 'Down' or 'Up' keys obtaining an animated view of the cross-sectioned "terrain".

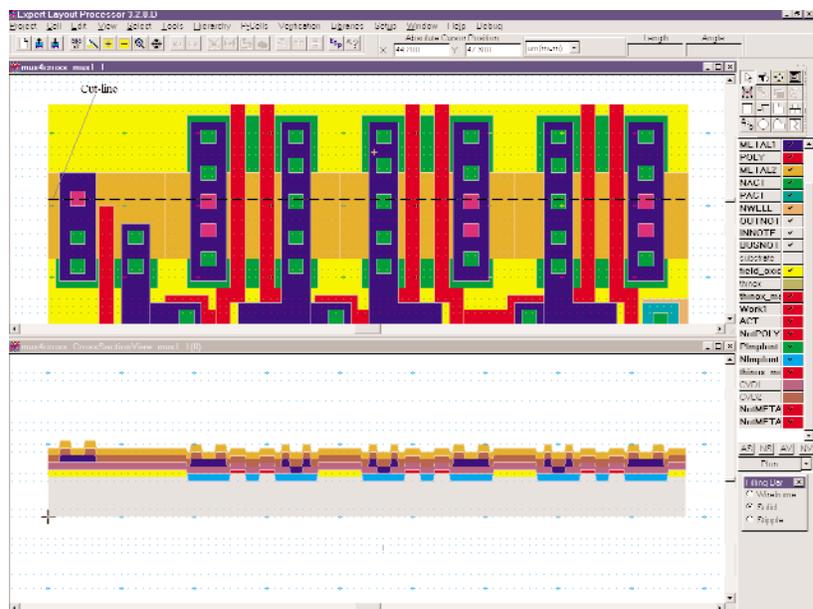


Figure 5. Cross-sectional line and the resulting cross-view.

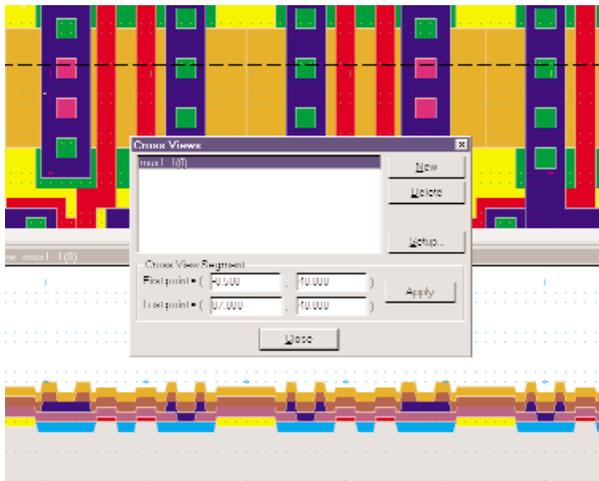


Figure 6. Selection from precalculated cross-views.

To remove an unnecessary cross view, one can use the 'Delete' button. The "Setup" button pops up the "Cross-sectional View" dialog panel to edit the fabrication sequence and process step parameters.

**Expert** Cross-sectional Viewer has a feature that allows the user to view the actual individual fabrication processes one step at a time and in the exact sequence that they occur. "Cross-sectional Views" menu command, see Figure 7, allows the user to choose the first, last, next or previous fabrication step for viewing. The cross view will be updated up to the specified process step. You can assign shortcuts for these menu commands to simplify the usage of this feature. "Fabrication Steps" dialog panel is a more powerful tool of Cross-sectional Viewer to show "step by step" simulation

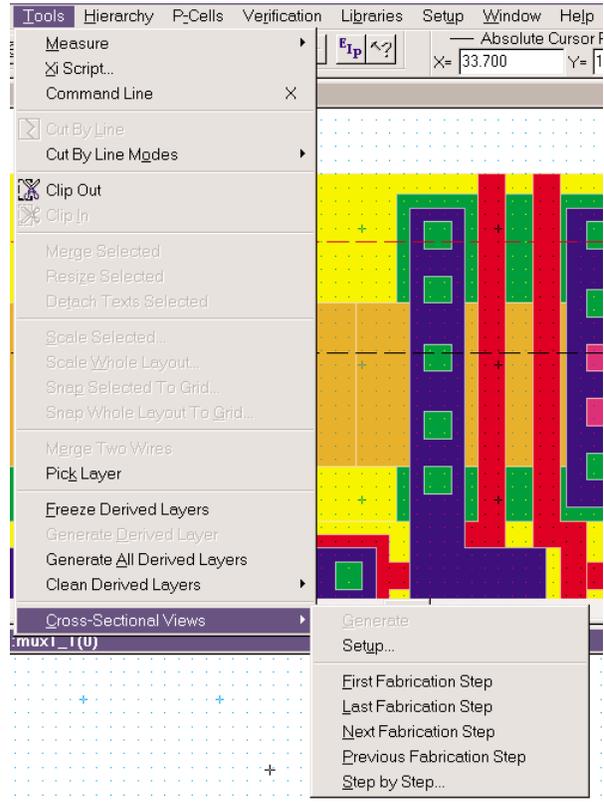


Figure 7. Step-by step viewing of fabrication sequence.

of fabrication sequence, see Figure 8. To invoke this dialog use "Tools>>Cross-sectional Views>>Step by Step" menu command. Press left mouse button or keys 'Down' or 'Up' to select the processing step you want to view. The Cross View will be updated to reflect the device structure after this process step.

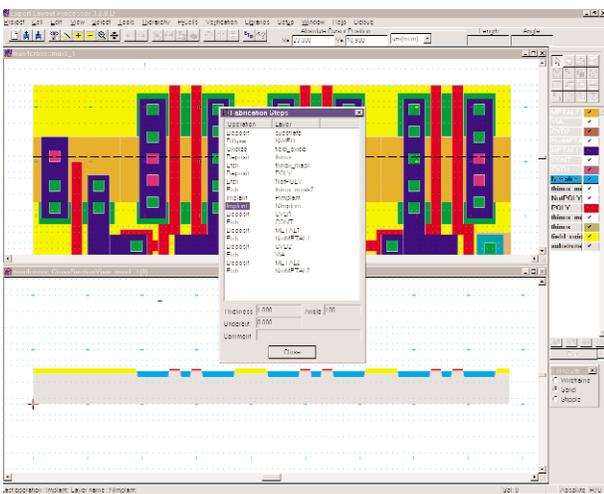


Figure 8. Random access for fabrication steps.

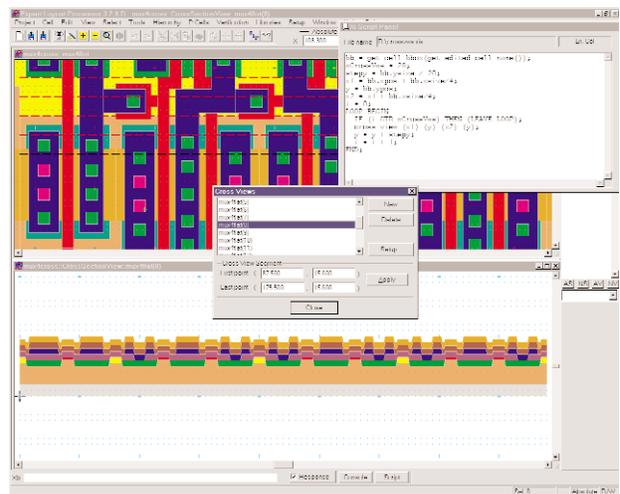


Figure 9. Cross-view animation example using xi-script.

# Expert: Recent Improvements in Hierarchical Layout Inspection

The functionality of several features of *Expert* layout editor enhanced to deal with hierarchical layouts in more convenient ways.

## 1. Edit in Place

The Hierarchy>>EIP>>Direct Pick command allows you to pick an object for editing in place anywhere in cell hierarchy. To facilitate the choosing of the required object, preselection highlighting is used.

The hierarchy depth at which an object can be picked is naturally defined by the viewing depth (as defined by commands from the View>>Cell view submenu). If a primitive object (shape or text) is picked, then you will edit the cell that contains this object. If a cell instance/array is picked, then you will edit in place the cell of this instance.

## 2. Pick Layer

The Tools>>Pick Layer command now allows you to pick the layer (i.e., make this layer active) from a shape anywhere in cell hierarchy. This tool uses preselection-type highlighting of objects to be picked. As a by-product, one may use this tool to inspect objects deep in hierarchy: the message line shows a brief description of the object, like in the case of an ordinary preselection.

## 3. Node Probing

The function Verification>>Node Probing of *Expert/Expert Views* is redesigned to trace electrically connected nodes throughout the hierarchy of a cell, unlike flat-mode operation of previous versions.

In addition, this feature provides an easy and convenient way to analyze the results of Connectivity/

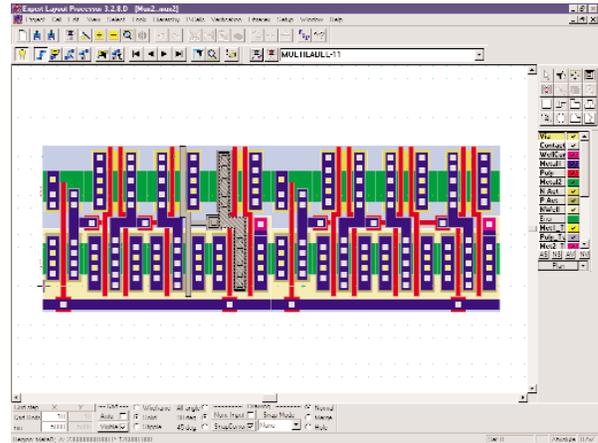


Figure 1. Electrical node highlighted.

Netlist extraction directly on the original hierarchical design and to make necessary corrections in the design. This tool makes it possible to search and inspect nodes listed in the extracted netlist file directly in the original layout.

When the menu command Verification>> Node Probing>>Pick Node is selected, the cursor assumes the node probing shape. It is necessary to point and click at the geometric object whose electric connections or device description are to be traced. As a result, the whole net electrically connected to the probed object or device shape will be highlighted (Figure 1), and you may inspect it using other commands from Verification>>Node Probing submenu. The corresponding commands can also be executed from the Node Probing bar. (This bar may be turned on/off by View>>Control Bars command.)

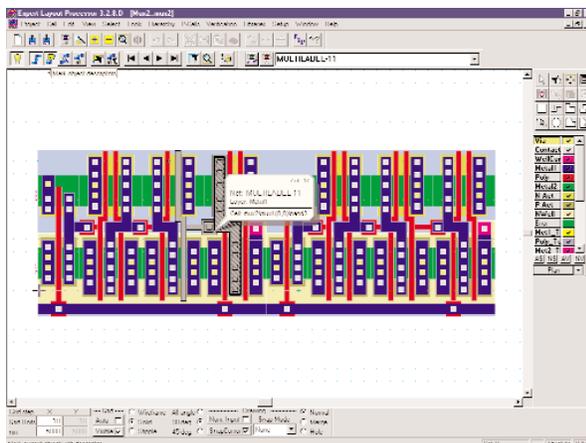


Figure 2. Object Descriptor for shape in probed node .

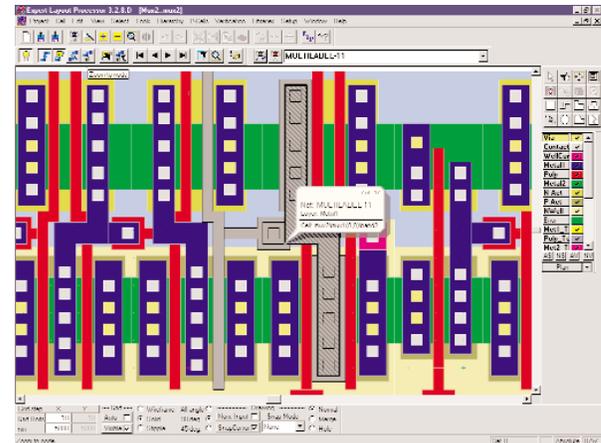


Figure 3. Zoom to Node.

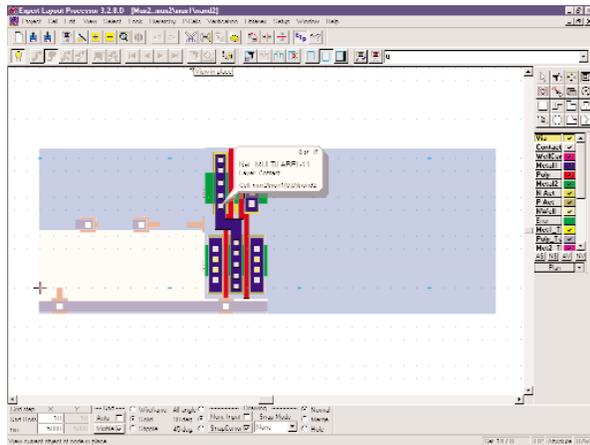


Figure 4. View in Place.

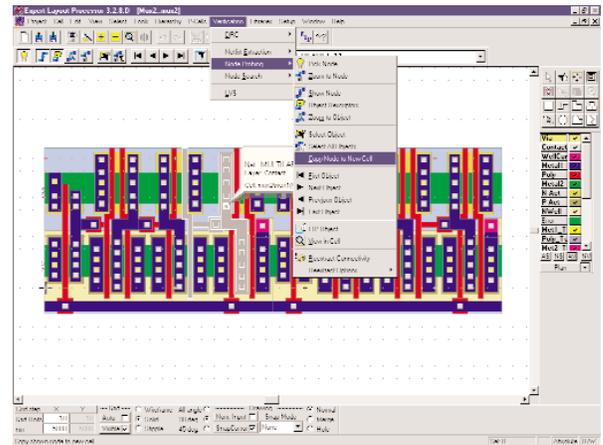


Figure 5. Copy Node to New Cell

To facilitate the picking of the required object in the layout, preselection highlighting is used. In addition, it is often convenient to hide some layers for better selectivity. For example, one may show only device definition layers to pick from. After an object is picked, the node probing function ignores layer visibility.

Verification>>Node Probing>>Object Descriptors menu command shows the Connectivity/Netlist Description Tag that covers the last probed node and contains information about Net/Device and the origin of the current shape in the original design (Figure 2).

There are commands to highlight the consecutive shapes of the node such as First Object, Previous Object, Next Object, Last Object They are useful for consecutive node browsing.

For a more detailed inspection, there are such functions as Zoom to Node (Figure 3), View in Place (Figure 4) and View in Cell (Figure 7).

It is possible to save the highlighted node into a new cell for later inspection or for historic references (Figure 5). The command Verification>>Node Probing>>Copy Node to New Cell copies all geometry in flat form to a newly created cell.

One more function Verification>>Node Search is added into menu and duplicated in the Search Net/Device Bar, used for searching the nets or devices by their name. There is an edit field on the Search Net/Device Bar for typing the name of the net/device to be found and highlighted in response to Find Device command (Figure 6). The full Spice statement for the found device will be shown in the Edit Net/Device Name field.

Another important function mentioned above is viewing the highlighted object in the cell where this object is allocated in design hierarchy. The result of this operation for a device shown in Figure 6 is represented in Figure 7.

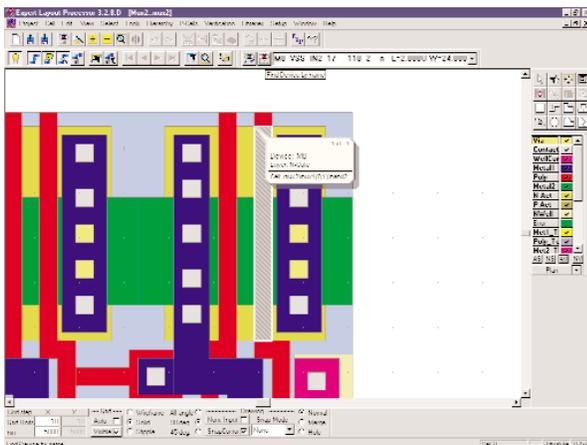


Figure 6. Find Device by Name

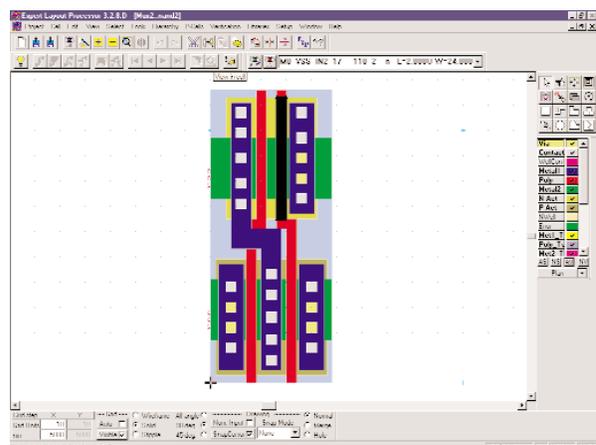


Figure 7. View in Cell.

# Applicability of Distance Computation for Graphs to LVS Discrepancy Analysis

## 1. Introduction

An important step in electronic circuit design is layout versus schematic verification (LVS). It consists of the comparison of a schematic circuit to its IC layout counterpart to determine whether they implement the same functionality. Typically, this comparison is performed in terms of spice netlists produced from the schematic and layout representation of the circuit. The ultimate goal of this verification is to ensure that these netlists completely match each other, possibly after some equivalence transformations of netlists, such as device filtering, merging, reduction. However at earlier iterations of the design cycle these netlists often fail to match. In some cases a single design error may lead to a long list of mismatched elements, and a common complaint is that LVS discrepancy reports are very difficult to analyze. Even with the aid of cross-probing and netlist browsing tools tracking the netlist discrepancies down to their source is often a tedious and time-consuming task.

Silvaco's *Guardian* LVS verification tool attempts to make this error-tracking task easier by postprocessing LVS discrepancy reports and pointing out potential matches of unmatched nodes, see Figure 1.

Graphs and hypergraphs are common mathematical models for netlists. *Guardian* used graph-theoretic techniques to process netlists. In graph-theoretical terms, netlist comparison is modeled by the well-known graph isomorphism problem. However graph isomorphism itself represents an ideal case of complete match between two circuits. When talking about discrepancies, a more adequate model is the notion of the distance between graphs.

A well-known type of distance between complex structured objects is the so-called editing distance. If  $S$  is a set of allowable editing operations on the set of all

```
GUARDIAN UNMATCHED NODES REPORT

first netlist:  mux2

number of unmatched nets      :  2
number of unmatched instances  :  0

second netlist:  mux2_flat

number of unmatched nets      :  4
number of unmatched instances  :  1

first netlist:  mux2          second netlist:  mux2_flat

DISCREPANCIES AND POTENTIAL MATCHES:

L0: N:  MUX2:VDD  . . . . . >>> MUX_2:1
L0: N:  MUX2:VSS  . . . . . >>> MUX_2:0
                                     MUX_2:10
L? N:  ????      . . . . . >>> MUX_2:NI2
L? N:  ????      . . . . . >>> MUX_2:MI13
```

Figure 1: *Guardian* potential match report.

objects in question, then the editing distance between two objects A and B is the minimal number of editing operations from set  $S$  required to transform object A into object B. In the case of graphs, the natural atomic editing operations are insertion and deletion of vertices and edges.

Moreover, these objects A and B are often identified with each other up to their isomorphism, so it is natural to talk about distances between isomorphic classes of objects. In the case of graphs this distinction of notions boils down to the distinction between the notion of distance for labeled graphs vs. distance between unlabeled graphs. In this article, a graph will mean an isomorphism class of labeled graphs. A number of papers discussed various types of instances between graphs, see e.g., references [M79], [MK79], [Z75]. This article considers the computational complexity for some definitions of distance between graphs. It proves that the problems of computing the distances between unlabeled graphs are NP-hard even in restricted classes of graphs. Please refer to [GJ79] for all notions related to computational complexity and NP-hardness.

Notice that the problem of computation of any kind of distance between (unlabeled) graphs is at least as computationally hard as the graph isomorphism problem [GJ79].

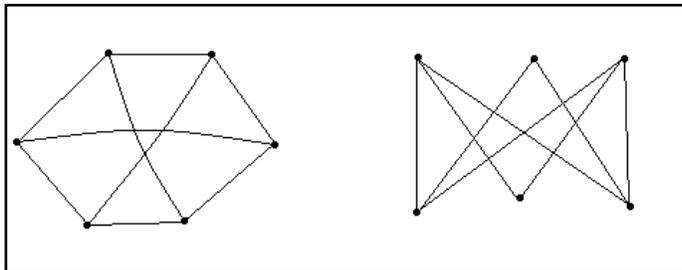


Figure 2: Graphs with edge-editing distance 1.

## 2. NP-Hardness of Edge-Based Graph Editing Metrics

Let  $G_1(V_1, E_1)$  and  $G_2(V_2, E_2)$  denote two graphs. One may consider the following natural metrics.

(a) For graphs with the same vertex count ( $|V_1| = |V_2| = n$ ):

$$d_1(G_1, G_2) = \min_{\{t\}} (|E_1 \setminus t(E_2)| + |t(E_2) \setminus E_1|),$$

$$d_2(G_1, G_2) = \min_{\{t\}} \max \{|E_1 \setminus t(E_2)|, |t(E_2) \setminus E_1|\},$$

where in both cases the minimum is over the set of all permutations  $t$  of the vertex set.

$$d_3 = n - k,$$

where  $k$  is the maximal vertex number for a graph isomorphic to generate subgraphs of  $G_1$  and  $G_2$ , [Z75].

(b) Over the set of  $(n, m)$ -graphs, i.e., graphs with  $n$  vertices and  $m$  edges.

$$d_4(G_1, G_2) = \min_{\{t\}} |t(E_2) \setminus E_1| =$$

$$q - 1/2 \max \sum (a_{ij} b_{t(i)t(j)}),$$

where  $(a_{ij})$  and  $(b_{ij})$  are the adjacency matrices of  $G_1$  and  $G_2$  respectively.

**THEOREM 1.** The problems of computation of distances  $d_1, d_2, d_3$  are NP- complete.

**PROOF:** If it  $d_1(G_1, G_2) = |E_2| - |E_1|$ , then it is easy to see that  $G_1$  is a subgraph of  $G_2$ . The same statement is true for  $d_2$ . But it is known that the subgraph isomorphism problem is NP-complete [GJ]. In the case of  $d_3$ , the problem of finding a maximum clique in a graph [GJ] is reduced to the computation of  $d_3$ , since its size for a graph  $G$  is equal to  $n - d_3(G, K_n)$ , where  $K_n$  is the complete  $n$ -vertex graph.

**THEOREM 2.** The problem of computation of  $d_4(G_1, G_2)$  is NP-complete even in the case when  $G_1$  and  $G_2$  are trees.

**PROOF:** Consider the following recognition problem:

**TREE-DISTANCE:**

**INPUT:** Trees  $G_1(V, E_1), G_2(V, E_2)$  and an integer  $B'$ .

**PROPERTY:** There exists a permutation  $t$  such that

$$|t(E_2) \setminus E_1| \text{ is at most } B'.$$

It is known [GJ] that the following **THREE-PARTITION** problem is NP-complete in strong sense:

**THREE-PARTITION:**

**INPUT:** A set  $A = \{a_i\}$  of  $3m$  elements, an integer  $B$  and positive integer sizes  $s(a_i)$  such that

$$B/4 \leq s(a_i) \leq B/2, i = 1, \dots, 3m, \quad (1)$$

$$\sum s(a_i) = Bm, \text{ where sum is over all } a_i \text{ in } A. \quad (2)$$

**PROPERTY:** It is possible to partition  $A$  into  $m$  disjoint subsets  $A_j$  such that

$$\sum s(a_i) = B \quad (3)$$

for all  $j = 1, \dots, m$ , where the sum is over all elements  $a_i$  in  $A_j$ .

Notice that conditions (1)-(3) imply that all  $A_j$  are 3-element sets.

Our goal is to reduce the **THREE-PARTITION** problem to **TREE-DISTANCE** problem.

Let  $TP_1 = \{A, B, s(a_1), \dots, s(a_m)\}$  be an instance of **THREE-PARTITION**. Consider a transformation  $T$  that maps  $TP_1$  into the following instance of **TREE-DISTANCE**:

$$TD_1 = T(TP_1) = \{B', G_1(V, E_1), G_2(V, E_2)\},$$

such that  $B' = 2m, E_1 = E'_{-1} + E''_{-1}, E_2 = E'_{-2} + E''_{-2}$ , where

$$E'_{-1} = \{(k, 0) \mid k \equiv 1 \pmod{B}\},$$

$$E''_{-1} = \{(k, k-1) \mid k \not\equiv 1 \pmod{B}\},$$

$$E'_{-2} = \{(k, 0) \mid k = 1 + \sum_{i=1, m'} s(a_i),$$

$$\text{for all } m' = 1, \dots, 3m-1\},$$

$$E''_{-2} = \{(k, k-1) \mid \text{for all remaining } k\}$$

Notice that since

$$\min_{\{v \in V\}} (\deg_{G_2}(v) - \deg_{G_1}(v)) = 2m, \quad (4)$$

$$\text{we have } d(G_1, G_2) \geq 2m. \quad (5)$$

Suppose that **THREE-PARTITION** has a solution  $\{A_i\}$ . Without loss of generality we may assume that

$$A_i = \{a_{3i-2}, a_{3i-1}, a_{3i}\}, i = 1, \dots, m.$$

If we take the identity permutation  $t$  ( $t(i) = i$  for all  $i$ ), we can readily verify that

$$|t(E_2) \setminus E_1| = 2m. \quad (6)$$

Conversely, if a permutation  $t$  satisfies (6) then (4) and (5) together imply that  $t(0) = 0$  and the edges from  $t(E_2) \setminus E_1$  are incident to 0. Then  $t$  isomorphically injects the connected components of  $G_2 \setminus \{0\}$  (which are chains) into graph  $G_1 \setminus \{0\}$  which is a union of  $m$  chains of length  $B$ . Clearly, this injection delivers a solution for the corresponding instance of **THREE-PARTITION**.

### Summarizing:

(a) an instance  $I$  of **TREE-DISTANCE** is accepted if and only if the instance  $T(I)$  of **THREE-PARTITION** is accepted;

(b) clearly, the computation of  $T$  requires time polynomial in  $m$  and  $B$ ;

*Continued on page 11...*

# Calendar of Events

## September

1
2
3 Japan Applied Physics - Hokkaido, Japan
4 Japan Applied Physics - Hokkaido, Japan
5 Japan Applied Physics - Hokkaido, Japan
6 Japan Applied Physics - Hokkaido, Japan SISPAD 2000-Seattle WS
7 Japan Applied Physics - Hokkaido, Japan SISPAD 2000-Seattle WS
8 SISPAD 2000-Seattle WS
9
10
11 ESSDERC, Cork, Ireland
12 ESSDERC, Cork, Ireland
13 ESSDERC, Cork, Ireland Int'l w/s on Power and Timing, Opt and Sim.-Germany
14 Int'l w/s on Power and Timing, Opt and Sim.-Germany
15 Int'l w/s on Power and Timing, Opt and Sim.-Germany
16
17
18
19
20
21
22
23
24
25 BCTM BiPOLAR- Minneapolis
26 BCTM BiPOLAR- Minneapolis
27 BCTM BiPOLAR- Minneapolis
28
29
30

## October

1
2 ISCS-Monterey, CA 2000 SOI-Wakefield, MA
3 ISCS-Monterey, CA 2000 SOI-Wakefield, MA EDA & Test Expo 2000- Hsinchu, Taiwan
4 ISCS-Monterey, CA 2000 SOI-Wakefield, MA EDA & Test Expo 2000- Hsinchu, Taiwan
5 ISCS-Monterey, CA 2000 SOI-Wakefield, MA
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 LCD/PDP Int'l 2000, Pacifco, Yokohama
26 LCD/PDP Int'l 2000, Pacifco, Yokohama
27 LCD/PDP Int'l 2000, Pacifco, Yokohama
28
29
30
31

## Bulletin Board



### SYSPAD 2000

Silvaco kicks off September with SISPAD 2000, happening September 6th through 8th. The International Conference on simulation of Semiconductor Processes and Devices will be held in Seattle, Washington. Engineers from Silvaco's Corporate Headquarters and Phoenix office will be at SISPAD to give demos of the newest releases to attendees.



### SSDRC 2000

Silvaco's UK office is headed to Cork, Ireland for this year's 30th European Solid-State Device Research Conference. Located at the University College of Cork this conference provides a forum for recent advances in solid-state devices, and with the help of Silvaco tools the room for advancement will continue to grow.



### BCTM 2000

Minneapolis is once again host to the 2000 Bipolar/BiCMOS Circuits and Technology Meeting September 24th through 26th. This year's conference emphasizes the strong future of Bipolar/BiCMOS circuits and technology. Silvaco has exhibited at BCTM for many years and looks forward to many more successful conferences with them.

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

The Simulation Standard, circulation 18,000 Vol. 11, No. 8, August 2000 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) 820-3000 (Japan), or your nearest Silvaco distributor.

Simulation Standard, TCAD Driven CAD, Virtual Wafer Fab, Analog Alliance, Legacy, ATHENA, ATLAS, MERCURY, VICTORY, VYPER, ANALOG EXPRESS, RESILIENCE, DISCOVERY, CELEBRITY, Manufacturing Tools, Automation Tools, Interactive Tools, TonyPlot, TonyPlot3D, DeckBuild, DevEdit, DevEdit3D, Interpreter, ATHENA Interpreter, ATLAS Interpreter, Circuit Optimizer, MaskViews, PSTATS, SSuprem3, SSuprem4, Elite, Optolith, Flash, Silicides, MC Depo/Etch, MC Implant, S-Pisces, Blaze/Blaze3D, Device3D, TFT2D/3D, Ferro, SiGe, SiC, Laser, VCSELS, Quantum2D/3D, Luminous2D/3D, Giga2D/3D, MixedMode2D/3D, FastBlaze, FastLargeSignal, FastMixedMode, FastGiga, FastNoise, Mocasim, Spirt, Beacon, Frontier, Clarity, Zenith, Vision, Radiant, TwinSim, .UTMOST, UTMOST II, UTMOST III, UTMOST IV, PROMOST, SPAYN, UTMOST IV Measure, UTMOST IV Fit, UTMOST IV Spice Modeling, SmartStats, SDDL, SmartSpice, FastSpice, Twister, Blast, MixSim, SmartLib, TestChip, Promost-Rel, RelStats, RelLib, Harm, Ranger, Ranger3D Nomad, QUEST, EXACT, CLEVER, STELLAR, HIPEX-net, HIPEX-r, HIPEX-c, HIPEX-rc, HIPEX-crc, EM, Power, IR, SI, Timing, SN, Clock, Scholar, Expert, Savage, Scout, Dragon, Maverick, Guardian, Envoy, LISA, ExpertViews and SFLM are trademarks of Silvaco International.

# Hints, Tips and Solutions

Mikalai Karneyenka, Applications and Support Engineer

**Q: It seems that `seq_find()` procedure does not work correctly: if you try the following code, you will see that the required object is not found (0 is displayed):**

```
it = {};
seq_add_last(it, "vss");
seq_add_last(it, "vdd");
node = "vdd";
location = seq_find(it, node, 0);
display(location);
```

**A: You must keep in mind the distinction between the LISA object and its value. You will readily find your 'node', if you modify the script as follows:**

```
it = {};
node = "vdd";
seq_add_last(it, node);
location = seq_find(it, node, 0);
display(location);
```

On the other hand, LISA object is not similar to the notion of pointer from, e.g., the C language, as the modified script shows this:

```
it = {};
node = "vdd";
seq_add_last(it, node);
loc = seq_find(it, node, 0);
display(loc);
display(it[loc]);
node="!vdd";
loc1 = seq_find(it, node, 0);
display(it[loc]);
display(loc1); !
```

However if the last example makes you think that LISA object is too 'volatile', please try the following script:

```
it = {};
node = "vdd";
seq_add_last(it, node);
loc = seq_find(it, node, 0);
display(loc);
display(it[loc]);
node1=node;
node="!vdd";
node2="vdd";
loc1 = seq_find(it, node, 0);
display(it[loc]);
```

```
display(loc1);
loc2 = seq_find(it, node1, 0);
display(loc2);
loc3 = seq_find(it, node2, 0);
display(loc3);
node3 = it[loc];
loc4= seq_find(it,node3, 0);
display(loc4);
loc5= seq_find(it, it[loc], 0);
display(loc5);
```

Returning back to your question, I guess that the real issue was to find the string "vdd" in the sequence 'it'.

This issue a little bit tricky due to the nature of a LISA sequence: unlike well-known arrays from other programming languages, it may contain objects of different types. For example, the following sequence is perfectly valid:

```
ss={0.01, "vdd" ,{17, 23, 42} ,{}, {}};
display(ss[3][2]);
display(ss[5].type);
```

Therefore generally your search for a string must be something like this:

```
FOUND = FALSE;
cur=0;
last=ss.size;
loop begin
    cur = cur + 1;
    if (cur GTR last) then (leave loop);
    if(ss[cur].type EQL string) then begin
        if (string_match(ss[cur], "vdd"))
            then (leave loop);
    end;
end;
if(cur LEQ last) then (FOUND = TRUE);
if (FOUND EQL TRUE) then
    (display("ss["&cur&"] = \"&ss[cur]&\""));
else
    (display("Power lost"));
```

Of course, if you are pretty sure that your sequence contains only strings, you may skip the type check. A good practice is to use `string_match_exact()` or other string procedures at will in this example and see what you find.

**Q: In the layer setup, I cannot change the color of the layer by mouse clicking in the palette if initially the layer's color is black, i.e.,**

RED=255, BLUE=255, GREEN=255.

A: To select color by mouse click in the palette, you must select two things:

- (a) color shade. It is selected on the bigger color palette;
- (b) color brightness. It is selected on the narrow vertical color strip.

The actual color you selected is shown in the box below the bigger palette. Please look at this box while clicking at different places of the VERTICAL color strip.

When you set (255, 255, 255) or (0 0 0) RGB values, e.g., using numeric fields, this means that you select either 100% brightness or 100% darkness. After this changing the color shade makes no difference: you will always get either black or white.

**Q: Often I need to measure big objects. It order to position the ruler precisely I must zoom in at one end, zoom out that zoom again in at the other end. Is it possible too make this operation more convenient? For example, can I start the measurement in one window and finish it in another one?**

A: Unfortunately, in the current versions of Expert all operations must start and end in the same window. A workaround is to combine the following two viewing commands:

- View>>Previous View (default shortcut: ",") and
- View>>Pan>>Pan to cursor (default shortcut: ".").

The latter one pans to cursor and zooms to grid resolution at the same time, so that it is easy to position a mouse precisely.

So, please try to execute the following commands:

- select a larger view, with both start and end points for the ruler seen;
- start the ruler tool and position the cursor near the start point;
- View>>Pan>>Pan to cursor (shortcut: ".") to click the exact start point;
- View>>Previous View (shortcut ",") for a coarser view to move cursor approximately into the ending area;
- View>>Pan>>Pan to cursor (shortcut: ".") to click the end point with fine resolution.

By the way, it could be useful for other operations as well, as a replacement to zoom-full-zoom sequence.

### 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](mailto: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](http://www.silvaco.com)

*...continued from page 8*

- (c) total number of vertices in G1 and G2 is  $6mB + 2$ , i.e., polynomial in m and B.
- (d)  $B'=2m$  as clearly bounded by a polynomial in terms of the length of the corresponding instance of THREE-PARTITION.

These four claims are the sufficient prerequisites for T to be a pseudo-polynomial transformation [GJ] between the two recognition problems. Since TREE-DISTANCE is clearly in class NP, it is strongly NP-complete, and the claim of Theorem 2 follows.

**COROLLARY.** The computation of d1 and d2 for trees is NP hard.

We may also mention that the computation of d1(G, Pn), where G is an arbitrary graph and Pn is a chain graph, is NP-hard, since this problem is equivalent to the NP-hard Hamiltonian completion problem [GH].

## Conclusion

It is proven that a number of natural distances introduced between unlabeled graphs in literature are NP-hard, i.e., computationally intractable. Finding an easily computable definition of distance between graphs or hypergraphs is a possible way for fast, automated discrepancy analysis in LVS systems.

## References

- [GJ] Garey, R.M. and D.S. Johnson. Computers and Intractability, Freeman and Co., 1979.
- [GH] Goodman, S., and S. Hedetniemi. On the Hamiltonian completion problem. Lecture Notes in Math., 406(1974), 262-272.
- [MK79] Metelski, N.N and N.M. Korneyenko, Metrization of a class of graphs.- Doklady Akademii Nauk BSSR, 23(1979), 5-7.
- [M79] Mironov, A.A., Some properties of numbers realized by a graph.- Trudy Moskovskogo Instituta Inzenerov, 1979, No. 640, 115-120.
- [Z75] Zelinka, B., On a certain distance between isomorphism classes of graphs.- Casopis pro Pestovanej Matematiky, 100(1975), 371-373



## Join the Winning Team!

### Silvaco Wants You!

- 1 Process and Device Application Engineers
- 1 SPICE Application Engineers
- 1 CAD Applications Engineers
- 1 Software Developers

fax your resume to:

408-496-6080, or

e-mail to:

jobs@silvaco.com

Opportunities worldwide for apps engineers: Santa Clara, Phoenix, Austin, Boston, Tokyo, Guildford, Munich, Grenoble, Seoul, Hsinchu. Opportunities for developers at our California headquarters.

# 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



Vendor Partner