

Simulation Standard

TCAD Driven CAD

A Journal for CAD/CAE Engineers

Parallel DRC by Functional Decomposition of DRC Decks

1. Introduction

A new version of *Savage* DRC system taps the full power of multiprocessor computers by introducing simultaneous execution of DRC commands from a DRC deck. A DRC deck is a set of DRC rules to be performed for a given layout. For typical DRC decks, the described approach alone delivers high processor utilization and speed-up. Further improvements are possible using parallelization by decomposition of IC layout data and exploiting its hierarchy.

This article describes the approach used in the current implementation. The description provides guidelines for writing efficiently parallelizable DRC decks.

2. Algorithm

The DRC deck parallelization has to address two issues:

- Every rule of deck is computationally independent from others (beneficial factor);
- There is input/output data dependency between rules of a DRC deck (adverse factor).

The main idea of DRC parallelization comes from the fact that a DRC command can start as soon as all input data for the command is ready, it does not have to wait for all preceding commands in deck to finish. By comparing the inputs and outputs of all DRC commands in deck we can determine the data dependency between the execution of the commands and hence build a data dependency graph. The graph consists of a set of nodes. A node may correspond to one command or several consecutive commands in deck. Each node may have parent nodes and child nodes. Node is ready to be processed if all parent nodes have been processed. Each node is assigned a "weight", which is the number of child nodes it has. If there are several nodes ready to be executed, nodes with higher weights are submitted for processing first.

The parallel algorithm is implemented by having one main thread of execution and N worker threads. The

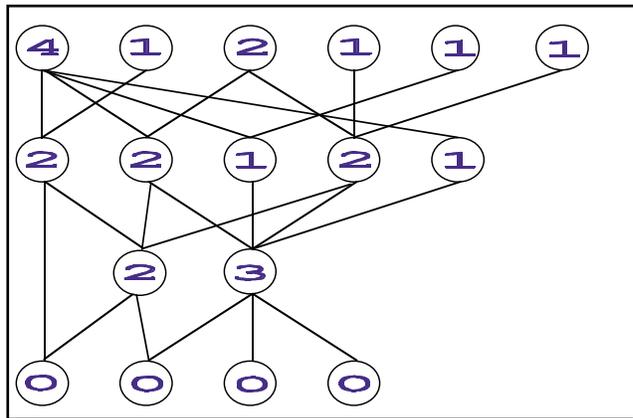


Figure 1. Data dependency graph of a DRC deck with weights of nodes.

main thread analyzes DRC deck to be executed and builds the dependency graph object. Then it starts N worker threads and waits for all of them to exit. Each worker thread takes 1 task from the graph, submits it for execution to a DRC executable module, notifies the graph when the task is completed and asks it for new task to execute...

Here is an outline of the parallel algorithm:

I. Main thread :

- Read design rules
- Build dependency graph
- Create N Worker threads
- Wait for all threads to exit
- Collect results of all performed commands (such as Error Database and Log file)

Continued on page 2....

INSIDE

CELEBRITY CAD on LINUX	4
Modification of BEM for Precise Capacitance Extraction	5
Advanced Layout Editing Using XI-scripts	8
Calendar of Events	10
Hints, Tips, and Solutions	11

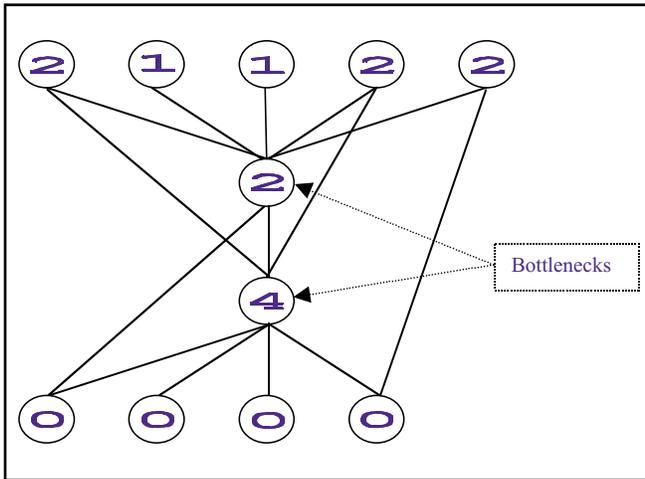


Figure 2. Bad parallelizable structure of graph (with node weights).

II. Worker thread :

While there are unprocessed tasks in graph DO

- ```
{
- Ask graph for a task to execute
- Call external DRC executable to perform corresponding command(s)
- Wait for executable to exit
- Notify graph of completion of the task
}
- Exit Thread
```

## 3. Results and Discussion

In Table 1 and Table 2 you can see the tables of runtime comparisons for 1-CPU vs. Multi-CPU DRC executions:

Note that performance of parallel DRC depends on the structure of the DRC deck. In above-mentioned examples, the test "S00" shows comparatively low speedup on 4-CPU computer. A reason of the low speedup of the test "S00" is "bad parallelizable" structure of DRC deck for this example. That means its graph of data dependency contains several "bottleneck" commands that have to wait completing of all (or almost all) previous ones. These "bottleneck" commands often generate a situation when several CPUs are not used.

Here is an example of a script (Script1) that is not parallelizable at all:

```
// Script name : Script1
/* 1 */ Substrate: LayerR=&SUB;
/* 2 */ Select: Relation=ENCLOSE, Layer1=&SUB, Layer2=M5, LayerR=&TMP;
/* 3 */ And: Layer1=M4, Layer2=$TMP, LayerR=PJCT;
/* 4 */ Select: Relation=ENCLOSE, Layer1=&SUB, Layer2=M4, LayerR=&TMP;
/* 5 */ Dif: Layer1=M1, Layer2=$TMP, LayerR=METLDIFF;
/* 6 */ Or: Layer1=M1, Layer2=M2, LayerR=&TMP;
/* 7 */ OverSize: Value=0.005um, Layer=&TMP, LayerR=OSIZEM1M2;
```

| Layout | GDSII size (flattened) | Execution time               |                            | Speedup    |
|--------|------------------------|------------------------------|----------------------------|------------|
|        |                        | Traditional DRC (uses 1 CPU) | Parallel DRC (uses 2 CPUs) |            |
| IZT    | 600                    | 2h 40min                     | 1h 55min                   | 1.4        |
| S00    | 390                    | 2h 15min                     | 1h 30min                   | 1.5        |
| SD0    | 757                    | 1h 15min                     | 45min                      | 1.66       |
| GRD    | 82M                    | 452                          | 37min                      | 25min 1.48 |
| TRN    | 60M                    | 270                          | 19min                      | 13min 1.46 |

Table 1. PC: Windows NT 4.0, 2-CPU, 800Mhz.

| Test name | # of commands in deck | Execution time               |                            | Speedup   |
|-----------|-----------------------|------------------------------|----------------------------|-----------|
|           |                       | Traditional DRC (uses 1 CPU) | Parallel DRC (uses 4 CPUs) |           |
| IZT       | 600                   | 5h 15min                     | 1h 28min                   | 3.6       |
| S00       | 390                   | 3h 35min                     | 2h 0min                    | 1.8       |
| SD0       | 757                   | 2h 15min                     | 39min                      | 3.5       |
| GRD       | 82M                   | 452                          | 34min                      | 12min 2.8 |
| TRN       | 60M                   | 270                          | 38min                      | 13min 2.9 |

Table 2. Sun: Solaris 5.7, 4-CPU, 400Mhz

Script1 has 3 independent parts that could be executed in parallel: (commands 1-3, 4-5, 6-7), but each independent part uses the same name for temporary layer (&TMP), which creates additional dependencies in graph. In order to use benefits of parallel DRC we can rewrite the same script using &TMP1, &TMP2, &TMP3 temporary layer names :

```
// Script name : Script2
/* 1 */ Substrate: LayerR=&SUB;
/* 2 */ Select: Relation=ENCLOSE,
Layer1=&SUB, Layer2=M5, LayerR=&TMP1;
/* 3 */ And: Layer1=M4, Layer2=$TMP1,
LayerR=PJCT;
/* 4 */ Select: Relation=ENCLOSE,
Layer1=&SUB, Layer2=M4, LayerR=&TMP2;
/* 5 */ Dif: Layer1=M1, Layer2=$TMP2,
LayerR=METLDIFF;
/* 6 */ Or: Layer1=M1, Layer2=M2,
LayerR=&TMP3;

/* 7 */ OverSize: Value=0.005um,
Layer=&TMP3, LayerR=OSIZEM1M2;
```

An additional observation is that small layouts demonstrate lower speed-ups, due to the increased influence of parallel thread management overhead.

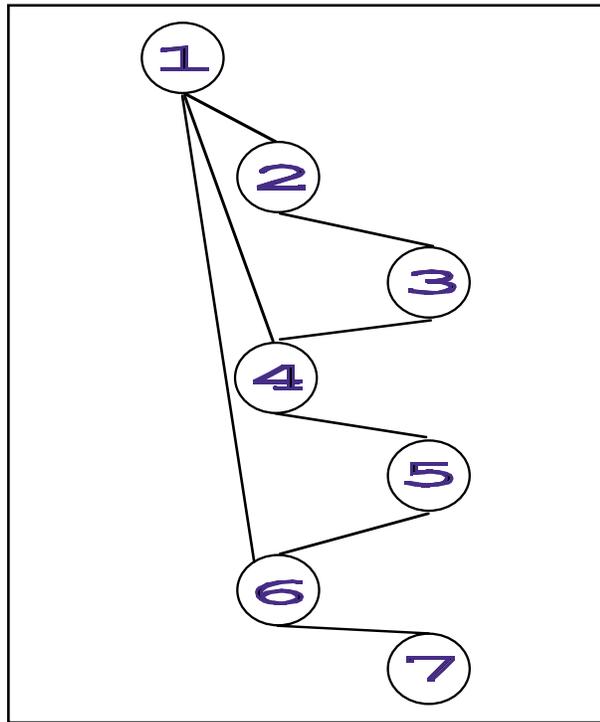


Figure 3. Graph of data dependency for Script1 (with command numbers).

#### 4. Implementation in Savage DRC System

Parallel, or multithreaded execution of a DRC script is possible both in on-line and off-line modes, if the parallel-DRC license is available and the computer has several processors. It is controlled by „parallel\_threads„ directive and command line parameter, and the corresponding settings in the

DRC Run Setup. The syntax is:

```
Parallel_threads: Auto;
Parallel_threads: <N>;
```

The keyword Auto means that the DRC system will select the optimal number of parallel threads itself. N is an integer to specify the number of parallel threads to be used for DRC run.

If N < 0, it is equivalent to Auto.

If N = 0, execution is non-parallel.

If N > 0, N is the maximal number of parallel threads.

NOTE: In many cases the optimal number of threads is number of parallel processors plus 1.

#### 5. Conclusion

The described approach to parallel execution of DRC decks is implemented in **Savage** DRC system of **CELEBRITY** suite of CAD tools. It delivers a significant gain in execution time of DRC checking. However this gain may depend on the structure of the DRC deck.

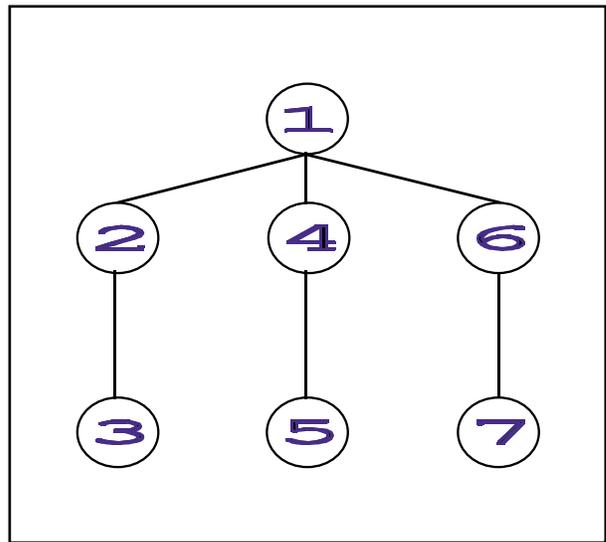


Figure 4. Graph of data dependency for Script2 (with command numbers).

# CELEBRITY CAD on LINUX

Silvaco Data Systems announces the release of **CELEBRITY** CAD suite: **Expert** layout editor, **Savage** DRC system and **Guardian** LVS for Linux platform.

Trends in the industry show that designers are using more and more Linux-based hardware and software for day-to-day applications due to nature of Linux functionality and availability. Linux is often considered an excellent, low-cost alternative to other more expensive operating systems. Linux supports IBM PC Architecture, Digital Alpha workstations, Sun Sparc processors, Motorola 680x0 based processors. There are a lot of Linux projects to support many others machines and architectures.

**Expert** originally designed for Windows platform was ported to the Linux using the Mainsoft's Visual MainWin application porting software (www.mainsoft.com). The new Linux version of **Expert** met at least the same performance and reliability standards as their established Windows solution. Figures 1 and 2 show two different flavors of **Expert's** user interface on Linux.

**Expert** is a powerful IC layout editor for Win NT platform. **Expert** is fast, robust, interactive, fully hierarchical and very easy to use. Key Features and Benefits of **Expert**:

- No limitations on size and complexity of ULSI designs
- Intelligent geometry database offering fast processing speed and low memory requirements
- Flexible tuning to different IC technologies
- Integrated DRC with real-time, on-line or batch modes of use

- Client-server design and library management
- Powerful script language and command line interface
- Parametric cell support
- C/C++ interface
- Efficient utilization of multithreading / multiprocessing
- Convenient hierarchical navigation and selection
- Cross sectional viewer
- Electrical node probing
- GDSII, Applicon and CIF import/export

The core DRC engine of **Savage** is ported to Linux directly, while its user interface is shared with **Expert**. As an additional benefit of the latest release of **Savage** is its parallel execution on multiprocessor computers, described in a separate article of this issue of "Simulation Standard".

The nearest plans for **CELEBRITY** suite are porting to Sun/Solaris.

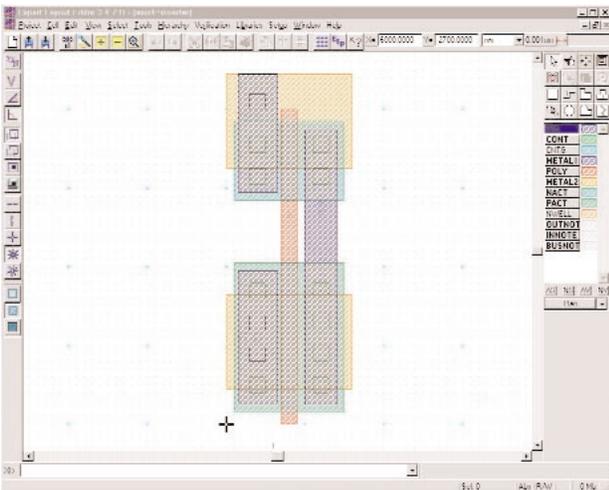


Figure1. **Expert** UI: Microsoft Windows Flavor .

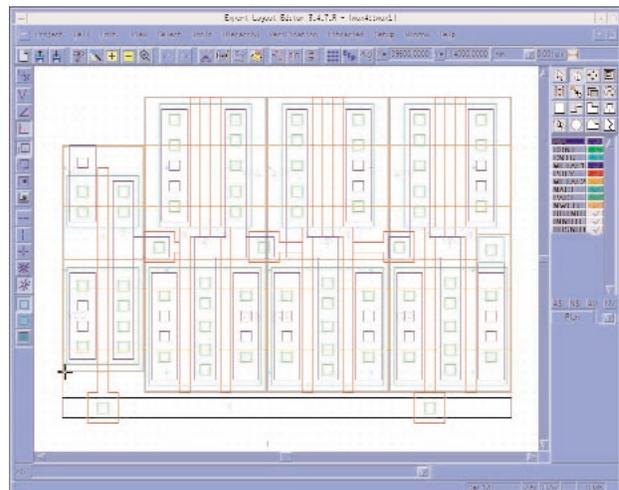


Figure 2. **Expert** UI: OS/Motif Flavor

# Modification of BEM for Precise Capacitance Extraction

## 1. Introduction

It is well-known that parasitic effects are no longer negligible for modern integrated circuit technology. Among the most important parasitic effects limiting the chip performance are interconnect related problems like extensive signal delays and crosstalk. The main numeric problem is extraction of parasitic capacitance, because it is a more complicated task than resistance extraction. The latest codes use directly 3D electrostatic solvers for this purpose. But the disadvantage of such approach is prohibitively long solution time for the problem. So the simplest model of 3D structures, when layout is splitted to 2D small pieces, can be used when it is necessary to handle complete IC's in a reasonable amount of time.

Usually 2D pieces are approximated with empirical formulas [1,2], which are valid only for the simplest shapes of conductors, usually rectangular (Figure 1). For more complicated shapes or for more accurate results, numerical methods are used. A standard approach is using Finite Element Method (FEM) for solving electrostatic problems. However for the 2D case, the Boundary Element Method (BEM) has a number of advantages. The first advantage is that a 2D problem reduces to a 1D problem in BEM, and it is not necessary to use the 2D mesh. The second advantage is that it calculates the surface charge directly, and hence the accuracy of output results is higher than for FEM.

## 2. Capacitance Problem and BEM

The Capacitance problem for N conductors is defined as calculation of capacitance matrix  $C_{ij}$  for these conductors. Symmetrical capacitance matrix  $C_{ij}$  is defined by the following simultaneous equations (Figure 2):

$$Q_i = \sum_{j=1}^N C_{ij} \cdot U_j,$$

where  $Q_i$  is the electric charge on i-th conductor,  $U_j$  is the potential on the j-th conductor.

So, in order to find  $C_{ij}$  we must know  $Q_i$  for a fixed distribution of  $U_j$ . Numerically, each coefficient  $C_{ij}$  is equal to  $Q_i$  for  $U_j=1$  and  $U_k=0$ , for all  $k \neq j$ .  $Q_i$  is calculated as  $\int \sigma \cdot dS$ , where  $\sigma$  is the surface charge density, S is the conductor surface.

The BEM is convenient for this purpose because the unknown function for BEM is  $\sigma$  and it is defined along the conductor contour only (problem reduction: 2D to 1D). It is in contrast with FEM, where the electrostatic problem is solved for potentials at first, then potentials

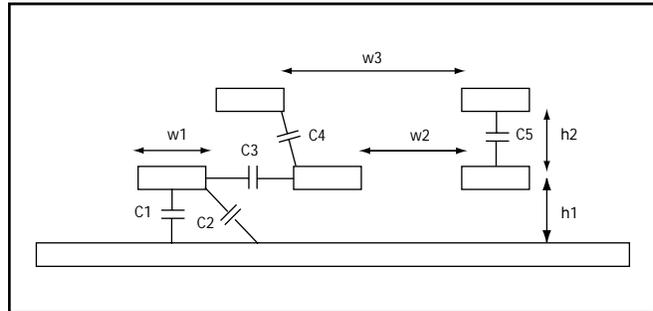


Figure 1. Capacitance C1-C5 can be approximated by empirical formulas for C1-C5. Formulas are parameterized only by geometrical constants like w1, w2, w3 and h1, h2.

must be numerically differentiated to obtain the electric field (the accuracy is reduced at this step) and after that the surface charge density is calculated from the electric field.

## 3. Improvements of BEM for the Precise Capacitance Extraction

The BEM uses the integral analog of the Laplas equation to solve the electrostatic problem.

If we write this equation for the conductor surface, we obtain the integral equation for surface electric charge density (2D case):

$$\int_S \sigma(\vec{r}) \ln \left( \frac{1}{|\vec{r}_c - \vec{r}|} \right) ds = U(\vec{r}_c), \quad (1)$$

or

$$\int_L \sigma(\eta) \cdot G(\xi, \eta) dl = U_c(\xi),$$

where  $\eta = (x,y)$  and  $\xi = (x_0,y_0)$  are observation and integration points,  $\xi \in L$ , L is the contour of conductors,  $G(\xi, \eta)$  is the kernel of the integral equation.

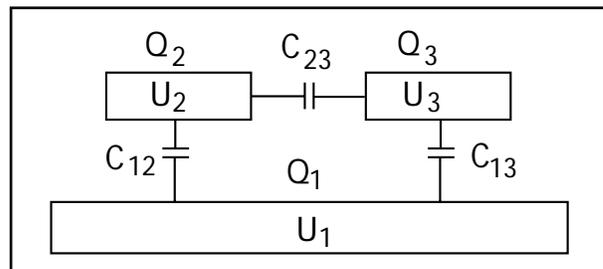


Figure 2. An example of a system that consists of three conductors. Capacitors C<sub>12</sub>, C<sub>13</sub>, C<sub>23</sub> are elements of the capacitance matrix.

### 3.1. Spline interpolation of surface charge density

In BEM, this equation is written at a finite number of nodes on the conductor surface and the values of electric charge density are found at the same nodes. One way to increase the accuracy of the solution is using a high order approximation of the charge density. The third order spline interpolation gives a good smooth solution [3] (Figure 3) :

$$\begin{aligned} \sigma(\eta(l)) = & (M_{i-1}f_{1i} \\ & + M_i f_{2i}) \cdot \frac{h_i^2}{6} + (\sigma_{i-1} - M_{i-1} \frac{h_i^2}{6}) \cdot f_{3i} \\ & + (\sigma_i - M_i \frac{h_i^2}{6}) \cdot f_{4i}, \end{aligned} \quad (2)$$

where  $l$  is contour parametric length,  $l_{i-1} \leq l \leq l_i$ ,  $h_i = l_i - l_{i-1}$ ,  $i=2, \dots, N$ ,  $N$  is number of nodes,

$M_i = \frac{\partial^2 \sigma}{\partial l^2}(l_i)$  is the second moment of the spline,  $f_{ki}$  are the weight functions of the spline:

$$\begin{aligned} f_{1i}(l) &= \frac{(l_i - l)^3}{h_i^3}, \\ f_{2i}(l) &= \frac{(l - l_{i-1})^3}{h_i^3}, \\ f_{3i}(l) &= \frac{(l_i - l)}{h_i}, \\ f_{4i}(l) &= \frac{(l - l_{i-1})}{h_i}, \end{aligned} \quad (3)$$

At the endpoint of an interpolation segment the following condition of free bounds is used:

$$M_1 = M_2, M_{N-1} = M_N.$$

Using spline features we can obtain the linear equation system for the second moment of the spline:  $AM=B$ , where

$$A = \begin{bmatrix} 2 & -2 & 0 & \dots & 0 \\ \mu_2 & 2 & \lambda_2 & 0 \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \mu_{N-1} & 2 & \lambda_{N-1} \\ 0 & \dots & 0 & -2 & 2 \end{bmatrix}, \quad (4)$$

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \quad \mu_i = 1 - \lambda_i,$$

$$\text{and } B_i = 6 \cdot \frac{\sigma_{i+1} - \sigma_i}{h_{i+1}} - \frac{\sigma_i - \sigma_{i-1}}{h_i} \cdot \varepsilon_{1i} \cdot \varepsilon_{Ni}, \quad i=1, \dots, N \quad (5)$$

$$\varepsilon_{ij} = \begin{cases} 1, & i \neq j \\ 0, & i = j \end{cases}$$

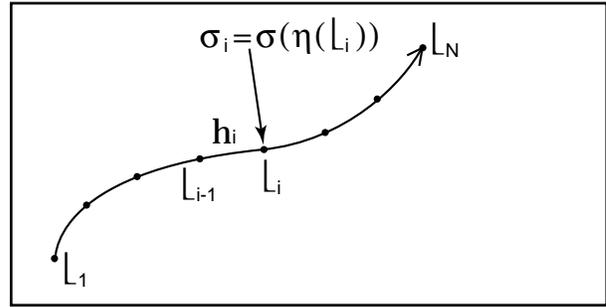


Figure 3. Discretization of segment for spline interpolation of surface charge.

The solution of equation system is:

$$M_i = 6 \cdot \sum_{k=1}^N a_{ik} \frac{\frac{\sigma_{k+1} - \sigma_k}{h_{k+1}} - \frac{\sigma_k - \sigma_{k-1}}{h_k}}{h_k + h_{k+1}} \cdot \varepsilon_{1k} \cdot \varepsilon_{Nk}, \quad (6)$$

Here  $a_{ik}$  are the coefficients of  $A^{-1}$ . Using this expression, we obtain the following equation for surface charge density:

$$\begin{aligned} \sigma(l) = & \sigma_{i-1} f_{3i} + \sigma_i f_{4i} \\ & + h_i^2 (f_{1i} - f_{3i}) \cdot \sum_{k=1}^N a_{i-1,k}^* \sigma_k \\ & + h_i^2 (f_{2i} - f_{4i}) \cdot \sum_{k=1}^N a_{i,k}^* \sigma_k, \\ & i = 2, \dots, N \end{aligned} \quad (7)$$

$$\begin{aligned} \text{where } a_{ik}^* = & \frac{a_{i,k-1} \varepsilon_{1k} \varepsilon_{2k}}{h_k (h_k + h_{k+1})} \\ & - \frac{a_{i,k} \varepsilon_{1k} \varepsilon_{Nk}}{h_k \cdot h_{k+1}} \\ & + \frac{a_{i,k+1} \varepsilon_{N-1k} \varepsilon_{Nk}}{h_{k+1} (h_{k+1} + h_{k+2})}. \end{aligned}$$

Using this equation and the definite integral

$$I_{ij}^m = \int_{l_{j-1}}^{l_j} f_{mj}(l) \cdot G(\xi_i, \eta(l)) dl \quad (8)$$

,  $m=1, 4, i=1, \dots, N, j=2, \dots, N$

we obtain the final system of linear equations:

$$\begin{aligned} \sum_{j=1}^N D_{ij} \cdot \sigma_j = & U_{ci}, \quad D_{ij} \\ = & \varepsilon_{Nj} \cdot I_{ij}^3 + \varepsilon_{1j} \cdot I_{ij}^4 \\ & + \sum_{k=2}^N h_k^2 \cdot (I_{ik}^1 - I_{ik}^3) \cdot a_{k-1,i}^* \\ & + \sum_{k=2}^N h_k^2 \cdot (I_{ik}^2 - I_{ik}^4) \cdot a_{k,i}^* \end{aligned} \quad (9)$$

Solving this system, we obtain a solution with interpolation accuracy  $o(h^4)$ .

### 3.2. Extraction of a Solution Singularity at Sharp Edges

Another difficulty in BEM is that the solution has singularities at sharp edges of conductors. The singularity of a solution in general case has the following dependence [4]:

$\sigma \sim r^{t-1}$ , where  $r$  is the distance from the edge, and  $t > 0$  is the degree of the singularity. For the case  $t < 1$  the charge density can not be described by cubic spline interpolation with acceptable accuracy. The following method is used to avoid this problem.

The conductor contour is divided into smooth continuous parts. The junction points of the parts belong to the edges of conductors. So the whole contour consists of the sum of these parts  $L = \sum L_k$ . On each part  $L_k$  the charge density can be rewritten as:

$$\sigma(l) = \left(\frac{l}{L_k}\right)^{t_{1k}-1} \cdot \left(L_k - \frac{l}{L_k}\right)^{t_{2k}-1} \cdot \sigma^*(l), \quad (10)$$

$$0 \leq l \leq L_k$$

where  $\sigma^*$  is a smooth function and can be fitted by spline with high accuracy,  $t_{1k}$  and  $t_{2k}$  are the degrees of singularity at the beginning and at the end of part  $L_k$ . The singularity degree for a sharp conductor edge with angle  $\theta$  is equal to  $t = \pi / (2\pi - \theta)$  (Figure 4).

For this case we can rewrite integral (8) as:

$$I_{ij}^m = \int_{l_{j-1}}^{l_j} \left(\frac{l}{L_k}\right)^{t_{1k}-1} \cdot \left(L_k - \frac{l}{L_k}\right)^{t_{2k}-1} \cdot f_{mj}(l) \cdot G(\xi_j, \eta(l)) dl, \quad (11)$$

$$m=1,4, i=1,N, j=2,N.$$

and solve the system of linear equation (9) for  $\sigma^*$ .

Integrals (11) are computed numerically using Gauss quadratures. Since the integral kernel contains singularities, for accurate calculation the method of integral kernel singularity extraction is used. The idea of the method is as follows. The integral kernel  $F(x)$  is split into the product of two factors  $S(x)$  and  $R(x)$ , and the integral can be rewritten as the sum of two integrals:

$$I = \int F(x) dx = \int S(x) \cdot R(x) \cdot dx =$$

$$\int S(x) \cdot (R(x) - R(a)) \cdot dx \quad (12)$$

$$+ R(a) \cdot \int S(x) \cdot dx$$

$R(x)$  is a function without singularities.  $S(x)$  has an integrable singularity at point  $a$ , and the integral in the second term (12) can be calculated analytically. The first integral has smooth kernel and can be calculated numerically without problems.

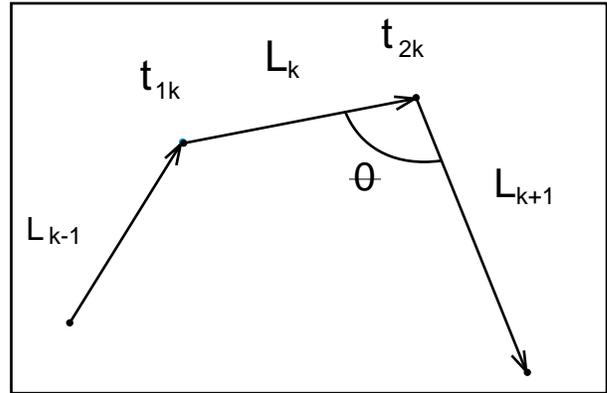


Figure 4. The singularity degree of the solution at the end of the segment  $L_k$  is defined by the angle  $\theta$ .

## 4. Conclusion

The spline approximation of surface charge and extraction of solution singularity in BEM permits to calculate capacitance with high accuracy. Implementation of BEM with described improvements in extraction code can provide accurate estimation of parasitic capacitances.

### Reference:

- [1] U. Choudry and A. Sangiovanni-Vincentelli, "Automatic generation of analytical models for interconnect capacitances," IEEE Trans. Computer-Aided Design, vol. 14, pp. 470-480, 1995.
- [2] N. Arora, K. Raol, R. Schumann and L. Richardson, "Modeling and Extraction of Interconnect Capacitances for Multilayer VLSI Circuits," IEEE Trans. Computer-Aided Design, vol. 15, pp. 58-67, 1996.
- [3] Ivanov V.Y. "Automation of computer design of electronic devices". Preprint CS SBAS USSR, 1977 N-40. (In Russian).
- [4] R. Mittra, S.W. Lee "Analytical Technique in the Theory of Guided Waves", The Macmillian Company, New York, 1971.

# Advanced Layout Editing Using XI-scripts

To provide uniform etching of metal, dummy metal rectangles can be placed over an unused area of the chip.

The following script generates dummy layer with a full array of metal rectangles over the whole cell area. It calls DRC script <dummy.dsf> which sizes the real metal layout up by 2  $\mu\text{m}$  and subtracts it from the dummy layer. Then the dummy layer sized up and down to remove any slivers left over from the subtraction phase.

Resultant dummy layer can be added to real metal in current cell or copied into a new cell.

```
Xs = 4.0; ! X size of dummy metal box
Ys = 10.0; ! Y size of dummy metal box
dist = 10.0; ! spacing between dummy boxes

C1 = get_edited_cell_name();
bb = get_cell_bbox(C1);

IF ((layer_exists("DummyM5")) NEQ TRUE)
THEN (new layer "DummyM5");

layer DummyM5;
SepX = Xs + dist;
SepY = Ys + dist;
nRow = (bb.ysize + Ys)/(SepY);
nCol = (bb.xsize + Xs)/(SepX);

PosX = bb.xpos;
PosY = bb.ypos;
iBoxY = 1;
LOOP BEGIN
IF (iBoxY GTR nRow) THEN (LEAVE LOOP);

iBoxX = 1;
LOOP BEGIN
IF (iBoxX GTR nCol) THEN (LEAVE LOOP);
box (PosX) (PosY) (Xs) (Ys);
PosX = PosX + SepX;
iBoxX = iBoxX + 1;
END;
PosY = PosY + SepY;
PosX = 0;
iBoxY = iBoxY + 1;
END;
drc script ("D:/DRC/dummy.dsf") /whole /flat;
select
delete layer "DummyM5";
delete layer "DMet";
```

----- < dummy.dsf> -----

```
OverSize: Value=2um, Layer=Metal5, LayerR=&ovMet(0);
Dif: Layer1=DummyM5, Layer2=&ovMet, LayerR=&Dummy(0);
UnderSize: Value=1um, Layer=&Dummy, LayerR=&tmp(0);
OverSize: Value=1um, Layer=&tmp, LayerR=DMet(0);
Or: Layer1=DMet, Layer2=Metal5, LayerR=&M5new(0);
// Copy: Layer=&M5new, LayerR= Metal5 (0);
// Dummy metal boxes will be added to layer "Metal5"
in current cell
```

To increase the surface breakdown of a structure the high field emission points must be eliminated. This can be achieved by rounding the corners of any orthogonal mask shape (radius should exceed the junction depth of the diffusion to increase considerably the breakdown voltage of the shallow junction).

Rounded corner shapes can be created in Expert with "Rolled Region" tool or with XI script.

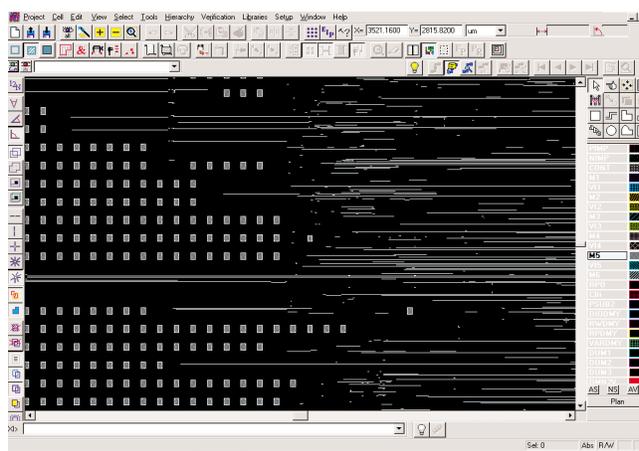


Figure 1. Dummy metal boxes added to layer "Metal5" in current cell.

The following script rounds each corner of selected orthogonal polygons. Both inside and outside corners will receive fillets.

```
----- <round_corner.xis> -----

DEFINE PROCEDURE "point_distance" /REPLACE
PARAMETER pt1
PARAMETER pt2
DO BEGIN
s = abs(pt1.x - pt2.x);
IF (s EQL 0.0) THEN (s = abs(pt1.y - pt2.y));
RETURN (s);
END;

DEFINE PROCEDURE "get_next_vertex" /REPLACE
PARAMETER vertices
PARAMETER cur_idx
DO BEGIN
nVertices = vertices.size;
IF (cur_idx EQL nVertices) THEN (RETURN (vertices[1]));
ELSE (RETURN (vertices[cur_idx + 1]));
END;

DEFINE PROCEDURE "get_prev_vertex" /REPLACE
PARAMETER vertices
PARAMETER cur_idx
DO BEGIN
nVertices = vertices.size;
IF (cur_idx EQL 1) THEN (RETURN (vertices[nVertices]));
ELSE (RETURN (vertices[cur_idx - 1]));
END;

DEFINE PROCEDURE "round_corner" /REPLACE
PARAMETER pt_prev
PARAMETER pt
PARAMETER pt_next
PARAMETER radius
PARAMETER layer_name
DO BEGIN
x = pt.x;
y = pt.y;
IF (pt_prev.y GTR pt.y AND pt_next.x GTR pt.x) THEN
BEGIN
region mode /hole;
box (x) (y) (radius) (radius)
/layer=(layer_name);
region mode /merge;
ellipse (x+radius) (y+radius) (radius) (radius)
/layer=(layer_name);
```

```

 region mode /normal;
END
ELSEIF (pt.x GTR pt_prev.x AND pt_next.y GTR pt.y)
THEN
 BEGIN
 region mode /hole;
 box (x-radius) (y) (radius) (radius)
 /layer=(layer_name);
 region mode /merge;
 ellipse (x-radius) (y+radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
ELSEIF (pt.y GTR pt_prev.y AND pt.x GTR pt_next.x)
THEN
 BEGIN
 region mode /hole;
 box (x - radius) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /merge;
 ellipse (x - radius) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
ELSEIF (pt.y GTR pt_next.y AND pt_prev.x GTR pt.x)
THEN
 BEGIN
 region mode /hole;
 box (x) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /merge;
 ellipse (x+radius) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
ELSEIF (pt.x GTR pt_prev.x AND pt.y GTR pt_next.y)
THEN
 BEGIN
 region mode /merge;
 box (x-radius) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /hole;
 ellipse (x-radius) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
ELSEIF (pt.y GTR pt_prev.y AND pt_next.x GTR pt.x)
THEN
 BEGIN
 region mode /merge;
 box (x) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /hole;
 ellipse (x+radius) (y-radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
ELSEIF (pt_prev.x GTR pt.x AND pt_next.y GTR pt.y)
THEN
 BEGIN
 region mode /merge;
 box (x) (y) (radius) (radius)
 /layer=(layer_name);
 region mode /hole;
 ellipse (x+radius) (y+radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
ELSEIF (pt_prev.y GTR pt.y AND pt.x GTR pt_next.x)
THEN
 BEGIN
 region mode /merge;
 box (x-radius) (y) (radius) (radius)
 /layer=(layer_name);
 region mode /hole;
 ellipse (x-radius) (y+radius) (radius) (radius)
 /layer=(layer_name);
 region mode /normal;
 END
END;

```

```

DEFINE procedure/replace "round_pol_corner"
parameter obj
parameter radius
DO BEGIN
 IF(obj.shape NEQ "Polygon") THEN (RETURN("This
object is not Polygon!"));
 vertices = obj.coords;
 nVert = vertices.size;
 i =1;
 LOOP BEGIN
 IF (i GTR nVert) THEN (LEAVE LOOP);
 pt = vertices[i];
 pt_next = get_next_vertex(vertices, i);
 IF (pt.x NEQ pt_next.x AND
 pt.y NEQ pt_next.y)
 THEN (RETURN("Nonortogonal polygon encontered"));
 IF (point_distance(pt, pt_next) / 2 LSS radius)
 THEN (RETURN("Too big radius value"));
 i = i + 1;
 END;
 i = 1;
 layer_name = obj.layer;
 LOOP BEGIN
 IF (i GTR nVert) THEN (LEAVE LOOP);
 pt = vertices[i];
 pt_next = get_next_vertex(vertices, i);
 pt_prev = get_prev_vertex(vertices, i);
 round_corner(pt_prev, pt, pt_next, radius,
 layer_name);
 i = i + 1;
 END;
 RETURN NIL;
END;

DEFINE command/replace "round_object_corner";
DEFINE action
parameter radius
DO BEGIN
 objseq = search(SEARCH_REGION, {}, WHOLE_CELL,
CROSS_RECT, SEARCH_SELECTED, SEARCH_VISIBLE);
 display ("Selected " &objseq.size &" objects");
 i=0;
 LOOP BEGIN
 i=i+1;
 if (i GTR objseq.size) then (leave LOOP);
 ret = round_pol_corner(objseq[i], radius);
 IF (ret NEQ NIL) THEN (display(ret));
 END;
END;
DEFINE argument
radius/COERCE=(DOUBLE)/positional/is_default/value_re
quired;
complete command;

```

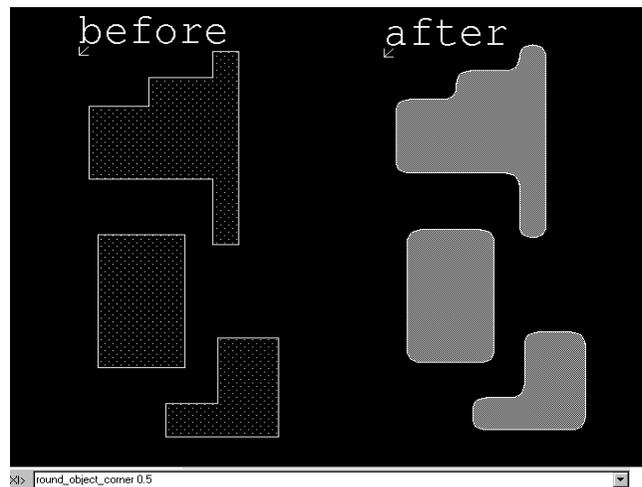


Figure 2. Boxes on the right were selected, then command round\_object\_corner 0.5 was executed.

# Calendar of Events

## June

|                                                                               |
|-------------------------------------------------------------------------------|
| 1                                                                             |
| 2 HITEC - Albuquerque, NM                                                     |
| 3 HITEC - Albuquerque, NM<br>ISPSD - Santa Fe, NM<br>IITC - San Francisco, CA |
| 4 HITEC - Albuquerque, NM<br>ISPSD - Santa Fe, NM<br>IITC - San Francisco, CA |
| 5 HITEC - Albuquerque, NM<br>ISPSD - Santa Fe, NM<br>IITC - San Francisco, CA |
| 6 ISPSD - Santa Fe, NM                                                        |
| 7 ISPSD - Santa Fe, NM                                                        |
| 8                                                                             |
| 9                                                                             |
| 10 DAC - New Orleans, LA<br>IWSM - Kyoto, Japan                               |
| 11 DAC - New Orleans, LA<br>IWSM - Kyoto, Japan                               |
| 12 DAC - New Orleans, LA<br>IWSM - Kyoto, Japan                               |
| 13 DAC - New Orleans, LA<br>IWSM - Kyoto, Japan                               |
| 14 DAC - New Orleans, LA<br>IWSM - Kyoto, Japan                               |
| 15                                                                            |
| 16                                                                            |
| 17                                                                            |
| 18                                                                            |
| 19                                                                            |
| 20                                                                            |
| 21                                                                            |
| 22                                                                            |
| 23                                                                            |
| 24 ADRC - UC Santa Barbara                                                    |
| 25 ADRC - UC Santa Barbara                                                    |
| 26 ADRC - UC Santa Barbara                                                    |
| 27                                                                            |
| 28                                                                            |
| 29                                                                            |
| 30                                                                            |

## July

|                                                                          |
|--------------------------------------------------------------------------|
| 1                                                                        |
| 2                                                                        |
| 3                                                                        |
| 4                                                                        |
| 5                                                                        |
| 6                                                                        |
| 7                                                                        |
| 8                                                                        |
| 9                                                                        |
| 10 Int'l W/S on Active<br>Matrix&TFT Tech - Tokyo                        |
| 11 CS-MAX - Boston, MA<br>Int'l W/S on Active<br>Matrix&TFT Tech - Tokyo |
| 12 CS-MAX - Boston, MA<br>Int'l W/S on Active<br>Matrix&TFT Tech - Tokyo |
| 13 CS-MAX - Boston, MA                                                   |
| 14                                                                       |
| 15 NSREC - Phoenix, AZ                                                   |
| 16 NSREC - Phoenix, AZ                                                   |
| 17 NSREC - Phoenix, AZ                                                   |
| 18 NSREC - Phoenix, AZ                                                   |
| 19 NSREC - Phoenix, AZ                                                   |
| 20                                                                       |
| 21                                                                       |
| 22                                                                       |
| 23                                                                       |
| 24                                                                       |
| 25                                                                       |
| 26                                                                       |
| 27                                                                       |
| 28                                                                       |
| 29                                                                       |
| 30                                                                       |
| 31                                                                       |

## Bulletin Board



### Appointment of New Distributors in the Far East

**IC Microsystems Sdn Bhd**  
Suite S06, 2320 Century Square  
63000 Cyberjaya  
Selangor D.E.  
Malaysia  
Tel No : 603-8319-1919  
Fax No : 603-8319-1918  
Email: sales@icmic.com  
Email: support@icmic.com

**Agamem Technology Shanghai Co Ltd**  
No 668, Beijing E.Road  
Room B804,  
Shanghai 200001  
China  
Tel No : 86-21-53085993  
Fax No : 86-21-53085113  
Email: sales@agamem.com.cn  
Email: support@agamem.com.cn

**K & H (Beijing)Co Ltd**  
8F, Building #2, Hwa Aou Center  
No 4 Beiwa Rd  
Haidian District  
Beijing 100089  
China  
Tel No : 86-10-6872-8672  
Fax No : 86-10-6872-8671  
Email: vincent@kandh.com.cn

**INTERSIGN**  
#20, First Cross, 12th Main  
Vasanthnagar, Bangalore 560052  
Karnataka  
India  
Tel No : 91-80-2260606  
Fax No : 91-80-2912147  
Email: intersign@vsnl.net

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. 10, No. 9, September 1999 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.

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, Spirit, 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

Galina Makouski, Applications and Support Engineer

**Q: (From “Hints and Tips” of Simulation Standards Vol. 10, No.9, 1999) How can I cut exactly the same hole in many layers?**

**A:** In recent versions of Expert a special option is added to do exactly this operation. In addition to “Normal”, “Merge”, and “Hole” options of object creation, option “Edit>>Region Mode>>Hole Selected” allows you to make a hole or cut of any shape in all selected shape objects at the same time. (The corresponding button is added to the Drawing Bar as well.)

**Q: How to skip a single layer during stream-out?**

**A:** There is a common way to control the output of layers during stream-out. This is performed using layer remapping tables, or layer maps. A layer map is a file of simple format: each line contains three parameters: Expert’s layer name, the required GDSII number and the required DATATYPE. A layer map is assigned using the Setup>>GDSII I/O>>Output panel. Only layers listed in the layer map are streamed out. So if you want to skip a single layer, you must prepare a layer map that contains all your design layers but one.

The easiest way to make one is as follows.

- (1) At the layer Bar, make all layers visible
- (2) Make the one you want to skip, hidden.
- (3) Run the “View>>Layer View>>Layer Lists>>Save Visible Layer List” command.

This command will prompt you to select a file name with extension \*.ELL. This file will be the required layer map.

**Q: I found something very strange on OutDistance checking. Please see the attached file, the distance of P2 to Ntub is 2um, but OutDistance checking will treat it as LT 2um.**

My script is

```
OutDistance: Options=(0), Layer1=&P2, Layer2=&hvnNtub,
Type=LT, Value=2um,
```

**A:** From the image I see that the distance is checked between 45-degree segments. If your geometry is on grid, it is impossible for the distance between such segments to be exactly 2um. It could be, e.g., 1.99998082. Under default Expert layout editor settings, this value will be shown as 2.0. You may see the value, if you increase the “numeric display resolution” by increasing the “Decimals displayed” value in the Setup>>Editor/Viewer>>Viewing panel.

Therefore, if you have non-90 geometry to check, you must use “Tolerance” parameter set, e.g., to 0.001um. WARNING: that if your geometry is purely orthogonal, the tolerance is unused, for faster calculations. Therefore if you want ensure that some distance is within, e.g., 0.299 and 0.3001, please use the “Range” distance parameter of the DRC check, rather than set the tolerance.

**Q: When distance between two objects is less than minimum gap, the filler box is created if its sizes are more than minimum width for this layer. (Compass, Synthesis, p. 16) How can we do this in Expert?**

**A:** You may try the DRC command with option “R”, e.g.,

```
Distance: Options=(0,L1=GT 1um,R), Layer=Contact,
LayerR=Filler,
Type=LE, Value=2um;
```

The “R” option is to write error trapezoids into the LayerR layer.

## 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)



## 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*

