

Simulation Standard

TCAD Driven CAD

A Journal for CAD/CAE Engineers

Expert - PC NT Layout Editor

Recent Significant Advances

Introduction

Expert layout editor has been designed to handle multi-million-transistor layouts with any kind of hierarchies, from flat to deep ones, while utilizing computing power of an ordinary Pentium-based PC running under Windows NT. To achieve this goal, the following components of **Expert** were carefully designed:

- A sophisticated geometric database was designed to support fast search and editing operations for huge amounts of geometric data.
- Fast drawing algorithms were fine-tuned to achieve a trade-off between drawing speed and detail level.
- Powerful tools were designed for convenient navigation through chip hierarchy and across design plane: Locator, Chip Rover, Multiview Navigator, and Edit-in-Place tools.
- Highly efficient built-in DRC checks facilitate the user to maintain design correctness, in addition to complete runs of Savage DRC system checks.

Usability of a layout editor to a significant degree depends on its readiness to enhance designer's productivity in terms of daily routine operations: device and simple library cell creation, composing larger layouts from pre-created cells, etc. The initial release of **Expert** provided basic and some advanced editing capabilities. The objective of the new release is to enhance the usability of **Expert** by adding a set of additional powerful editing features.

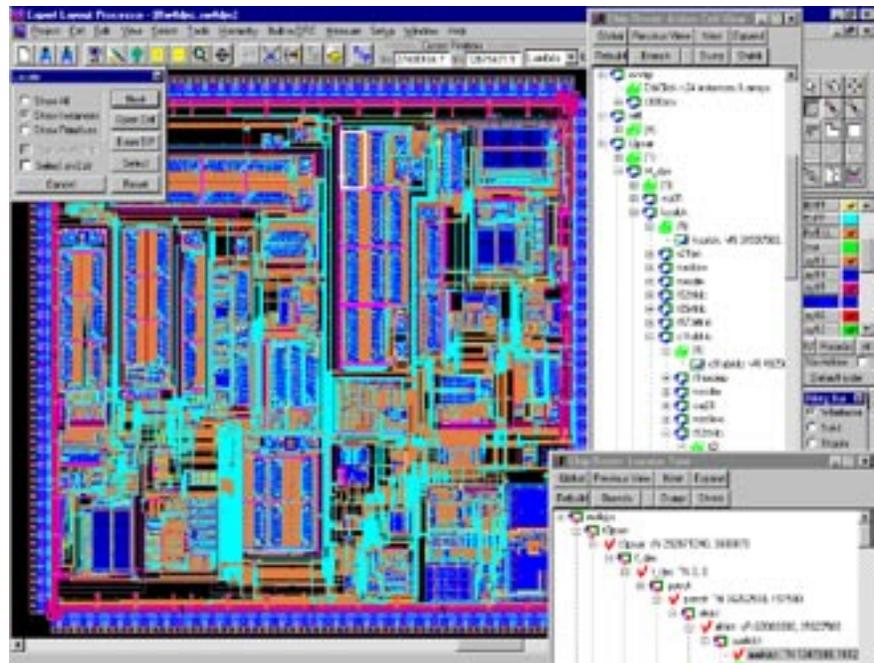


Figure 1. A general view of a multi-million transistor IC with corresponding tree of cell hierarchy.

Continued on page 2....

INSIDE

<i>Minimal Area Design of Power/Ground Nets</i>	4
<i>Application of Scan Line Methodology to Perform Metric Operations in DRC</i>	7
<i>Calendar of Events</i>	10
<i>Hints, Tips, and Solutions</i>	11

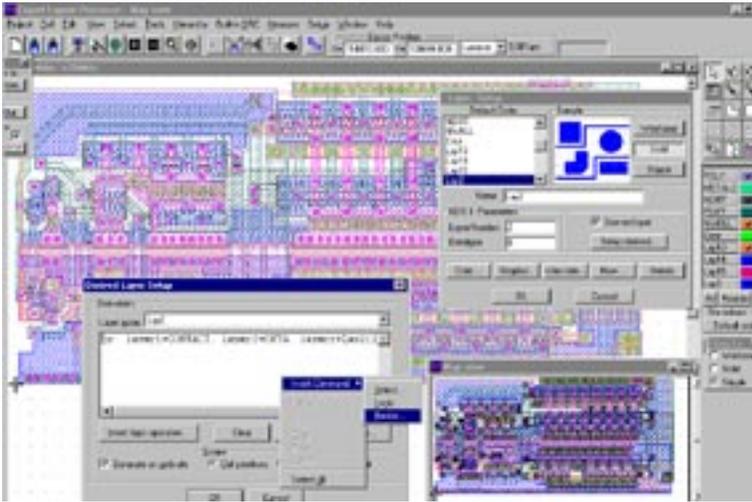


Figure 2. Interactive definition of derived layers.

A Derived Layer is a layer whose contents are produced by certain operations involving other layers. In the current version three types of layer operations are supported:

- selection
- logical
- resizing

Layer Derivation may be performed either via Layer Setup dialog or by writing the corresponding commands into the technology file.

Layer Generation (or Layer Rebuilding) is the process of actual construction of the contents of derived layers according to their definition. It may be invoked in several ways:

- from menu
- from Generation bar
- on activation of a derived layer using Layer bar
- on switching to a window where a derived layer is the active one

Fill patterns (Stipples)

Three drawing modes for layout objects are supported:

- wireframe mode, where only object's boundary is shown
- solid mode, where the whole object is filled by the selected color
- stipple mode, where the object's boundary is shown and its interior is filled in a semi-transparent way by a selectable stipple pattern (hatching, dot filling, etc.)

A partial list of significant improvements in **Expert**, version 1.30 is:

- Undo Operation
- Derived layers
- Fill patterns (stipples)
- Improved Hardcopy
- Improved Modify Operation
- Multi-object Stretching
- Resolving name collisions during cell import
- Improved Numeric Input interface
- Various improvements of visual appearance
- Network-based protection scheme
- Setup for GDSII Read/Write

The release notes contain a detailed description of these and other newly developed features.

Undo Operation

An Undo Operation has been implemented to provide unlimited rollback depth for both Local and Global Undo operations. Both operations maintain their own undo lists. Undo is invoked in one of the following three ways: by menu command Edit>>Undo, by clicking the Undo button at the Command bar, or by pressing Ctrl-Z.

Esc key allows editing to be interrupted.

The Undo feature is applicable only to Create, Modify, Copy, Move operations and to the drawing of zoom and selection boxes. Currently, Undo operation is not available for pan/zoom, select operations, switching among windows, flatten, explode, create cell in place operations. These features will be available in future versions of **Expert**.

Derived Layers

Definitions of derived layers may now be typed into the technology file or input interactively via the improved Layer Setup dialog.

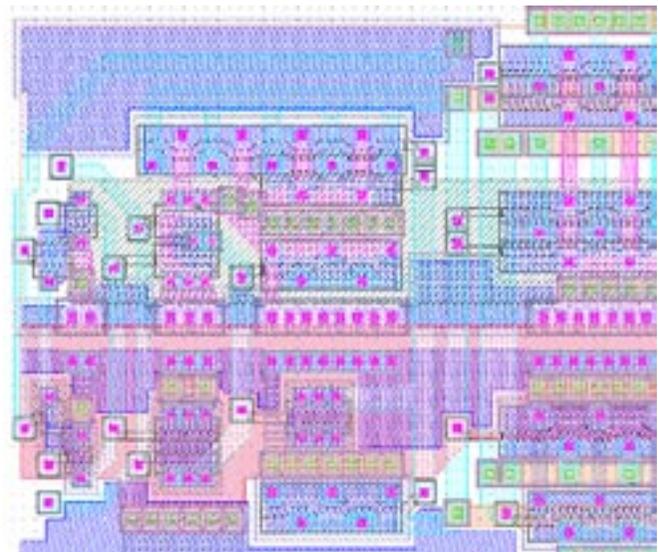


Figure 3. Sample layout display using stipple mode.



Figure 4. Page setup for plotting.

Hardcopy

The group of plotting commands permits you to make a hardcopy of the visible part of the layout. These commands are invoked from the "Project" submenu.

"Plotter Setup" command invokes standard Windows NT dialog to assign the default output device, size and orientation of paper sheets, as well as other device-specific properties. These properties will be used by default for previews and plotting operations.

"Plot preview" command invokes Expert's Page Layout dialog and standard Windows NT preview session to show placement of the plotting layout on paper sheets.

"Page Layout" Dialog is invoked within Plot Preview and Plot commands. It permits definition of the size or scaling of the plotted image. For example to split the drawing into several pages, and to set drawing margins on the page.

"Plot" command invokes standard Windows NT dialog to assign output device using for drawing and then plots the visible part of the layout. The method of drawing (printing or plotting) will be selected automatically according the hardcopy device.

Modify Operation

There are several improvements in the Modify operation for various types of objects. The most important of them is removal of some restrictions for 90° and 45° geometries. All region modifications (except of vertex/side deletion) can be done for 90° and 45° angle modes preserving restrictions on

angles. When a vertex or side is deleted, a warning is issued that angle mode cannot be preserved with this operation.

Stretch Operation

Stretch operation is used to modify several sides of multiple regions and multiple wires at the same time. The user may select several pieces of boundaries of several objects and move them while stretching/contracting object's sides that connect the chosen pieces to the rest.

This operation consists of the following steps:

- Select several region sides and/or several wire segments, possibly from different regions, boxes, and wires. They will move during stretch.
- Specify the restrictions on stretch
- Specify the amount of stretch (stretching shift)

During the stretch segments are classified into movable, fixed and stretchable:

- Selected segments will be moved as whole
- Segments with one end shared by a selected segment will stretch to accommodate the shift of selected segments ("rubber-bending")
- Segments with both ends shared by some selected segments will move as whole
- Segments with no end shared by selected segments will rest in place

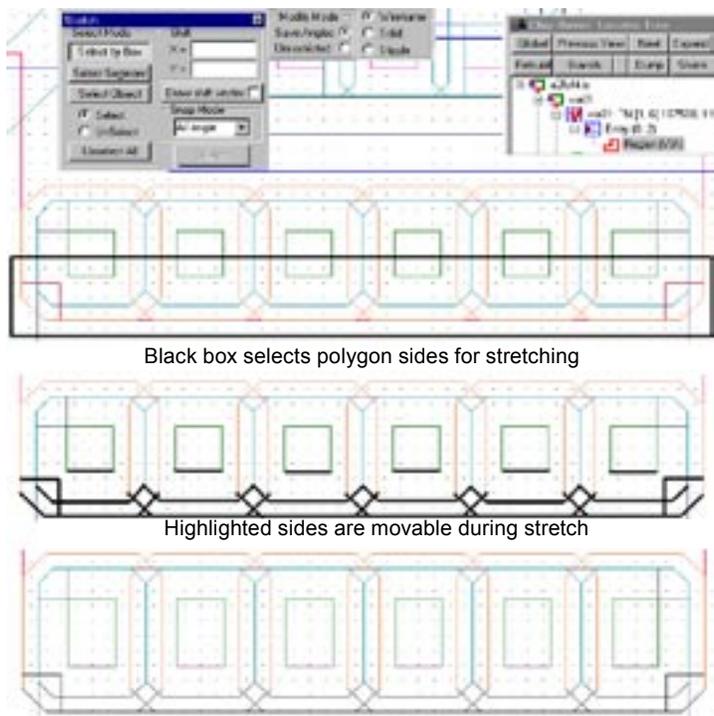


Figure 5. Stretch operation for multiple objects.

Stretch operation is controlled by several settings:

- segments can be selected individually, by box, or by whole objects
- stretching type may be controlled in terms of the original shape, just as for the ordinary Modify operation (Save Angle and Unrestricted modes)
- permitted stretch directions are vertical, horizontal, orthogonal, diagonal, and all-angle modes of stretching

Resolving Name Collisions

The Cell>>Import feature has been improved to process name collisions. When newly imported cells and cells already present in the project have the same names the following possibilities for pairs of colliding cells:

- replace all old cells by the new ones
- ignore all new cells
- rename all new cells by appending a user-defined suffix
- process cells one by one (replace/ignore/rename)

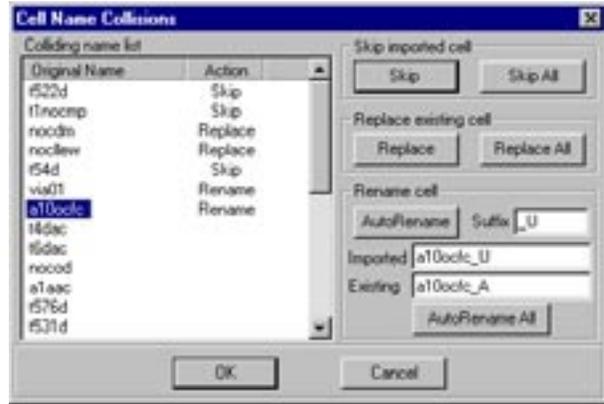


Figure 6. Automated detection and resolution of name collisions during cell import.

Minimal Area Design of Power/Ground Nets.

The value of modern layout editors is increased by including wide set of automatic operations. Expert, a PC-NT layout editor includes many traditional automatic tools for IC layout processing: merging, hollowing, resizing, cutting, clipping out, connectivity extraction, flattening, routing, etc. However, full-custom IC design requires more specific operations. In particular, the correct width calculation for segments of Power and Ground buses could be one of the crucial points in layout design flow. Power/Ground nets are among of basic components of ICs. Total chip area is significantly influenced by the quality of P/G net design.

For full- and semi-custom design, a shape and size of IC functional elements can be arbitrary, so P/G-net layout cannot be designed in advance, and VLSI designers are faced with the problem of minimization of the area for such net. This problem is not of great importance for standard cell design approach because the layout of P/G-nets is predefined for base chip in this case.

This article deals with mathematical model and algorithmic aspects for problem of determining the optimal width of each branch in the power and ground nets for VLSI circuits. The structures of the net topologies are rooted trees. Our objective is to minimize the area of the P/G nets subject to voltage drop constraints.

A basic mathematical model was developed by S.Chowdhury and M.A.Breuer[1]:

Let $T(r)$ be the tree for power or ground net consisting of n branches and having the root in the vertex r . Let $I(j)$, $w(j)$ and $l(j)$ represents, respectively, maximum current flow, width and length, associated with branch j . $V(r)$ is

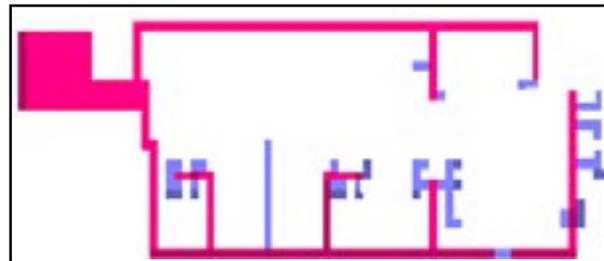


Figure 1: Fragment of ground bus from full-custom IC layout.

set of all vertices of the tree $T(r)$, $|V(r)|=n$. $L(r)$ is set of leaf of the tree $T(r)$ ($|L(r)|=m$ and $L(r)$ is subset of $V(r)$).

The objective is to minimize:

$$\sum_{j=1, \dots, |V(r)|} \{l(j)w(j)\} \quad (1)$$

The voltage drop constraints can be expressed for each j :

$$\sum_{k=p(r,j)} \{f \cdot I(k)l(k)/w(k)\} \leq u(j) \quad (2)$$

where:

- $p(r,j)$ is the path between the root r and leaf j of the tree $T(r)$;
- $u(j)$ is the maximum allowable voltage drop between these two points;
- f is conductivity of a layer.

We ignore some other constraints considered in [1] and consider a more practical one ([2], [3]):

$$w(k) \text{ is in } W(k), \text{ and } W(k) \text{ is a subset of } \quad (3)$$

the positive integer.

As shown in [3], the problem (1)-(3) (we shall call it the Ordinary Problem) is NP-hard. Nevertheless, it is possible to construct efficient algorithms for its solution [2, 3], if some parameters of the problem are bounded by reasonable constants, because the problem is not strongly NP-hard.

Our investigation have shown that this model is applicable only for small P/G nets. The reason of this is the assumption that we have non-zero current $I(j)$ in all branches of nets simultaneously. In practice, only a part of P/G net branches "works" at one time. Consider all (or some most essential) maximal subsets of branches which have non-zero current simultaneously and the list of these subsets consist of M items [4].

Let $\{T(r,j)\}$, $j=1,2,\dots,M$ be the set of the corresponding subtrees of the tree $T(r)$, and $L(r,j)$ be the set of leaves of the subtree $T(r,j)$.

Suppose we know the following vectors:

- current flows
 $I(j) = \{i(j,1), i(j,2), \dots, i(j,M)\}$,
 where $i(j,k)$ is the value of a current in a j -th P/G net branch, when the subtree $T(r,k)$ "works";
- maximum allowable voltage drops
 $U(j) = \{u(j,1), u(j,2), \dots, u(j,M)\}$
 where $u(j,k)$ is maximum allowable voltage drop on the path from the root of the tree $T(r,k)$ to its j -th leaf;
- voltage drops
 $X(j) = \{x(j,1), x(j,2), \dots, x(j,M)\}$,
 where $x(j,k)$ is the value of a voltage drop on a j -th P/G net branch, when the subtree $T(r,k)$ "works".

Moreover, let's introduce the following variables:

- $c(j) = f(j)a(j)$,
 where $a(j) = \text{Card}(\{k=1, M \mid I(j,k) \text{ is not equal } 0\})^{-1}$
 (here "Card" is a cardinality of a set);
- $G(X(j)) = (g(x(j,1)), g(x(j,2)), \dots, g(x(j,M)))$,
 where $g(x(j,k)) = 1/x(j,i)$, when $i(j,k)$ is not equal 0;
 $g(x(j,k)) = 1$, when $i(j,k)$ is equal 0.

Thus, using the vectors and the variables introduced above, we can describe the problem (1)-(3) as follows:

$$\sum_{j=1, \dots, |V(r)|} \{c(j)I(j)G(J)\} \rightarrow \min \quad (4)$$

$$\sum_{k=p(r,j)} \{X(k) \leq U(j), j=1,2,\dots, |L(r)|\} \quad (5)$$

The problem described by (4)-(5) is called Multivariant Problem of sizing P/G nets.

Note that the considered problem is also NP-hard (for $M=1$ it is transformed to the Ordinary Problem (1)-(3)). But since the problem is also not strongly NP-hard (just

as for the Ordinary Problem) there exist algorithms which are efficient for some practical restrictions on input parameters. Several methods developed for the exact solution of the Multivariant Problem are based on the dynamic programming.

Let us have a vector $Y = (y(1), y(2), \dots, y(M))$ which is determined in the M -dimensional space. Let each the vector Y have the correspondent set $G(r,Y)$. Each items of the set $G(r,Y)$ are $\{X(j)\}$, $j=1,2,\dots, |V(r)|$, which satisfy the following inequality:

$$\sum_{k=p(r,j)} \{X(k)\} \leq U(j) - Y, \quad j=1,2,\dots, |L(r)|$$

Let us introduce the following function:

$$F(r,Y) = \min \left(\sum_{\{X(j)\} \text{ is in } G(r,Y)} \{c(j)I(j)G(X(j))\} \right) \quad j=1, \dots, |V(r)|$$

If $G(r,Y)$ is empty then $F(r,Y)$ is equal to infinite number.

Since the value of $F(r,0)$ is exactly the value of function (4) for the problem (4)-(5) in the point of minimum, therefore the function $F(r,Y)$ will be called the determining function of the problem (4)-(5) for the vertex r .

For maximal subtree $T(k)$ of the tree $T(r)$ with the root in the vertex k , the determining functional of the problem (4)-(5) for vertex k is following:

$$F(k,Y) = \min \left(\sum_{\{X(j)\} \text{ is in } G(k,Y)} \{c(j)I(j)G(X(j))\} \right) \quad j=1, \dots, |V(r)|$$

The determining functional of the problem (4)-(5) satisfies the following equation of dynamic programming:

$$F(k,Y) = \min \left(\sum_{\{X(j) \mid j \text{ is in son}(k)\}} \{c(j)I(j)G(X(j)) + F(j,Y+X(j))\} \right)$$

Algorithms being based on an equation of dynamic programming depend on a kind of a representation of the function shown above.

Several algorithms based on the dynamic programming are developed for the exact solution of the problem (4)-(5).

The first method is pseudopolynomial. It gives polynomial solutions in the case when the number of subtrees is restricted by a constant. It is based on the fact that the cost functional of the dynamic programming formulation for the Multivariant Problem is piecewise constant, and its domain can be subdivided into hyperrectangles (k -dimensional boxes). The efficiency of the constructed algorithm is achieved by advanced techniques used for implementing of logical operations on hyperrectangles (see, in particular, [5]). The overall time complexity of considered method is

$$O((Hb)^M \max_{i=1, \dots, |V(r)|} \{W(i)\} |V(r)| M^{(M-2)} \log^{(M-2)}(bH))$$

where:

$$H = \max(u(j,k));$$

$$b \text{ is minimal integer which is divisible on each member of set } W = \text{union}_{i=1, \dots, |V(r)|} (W(i)).$$

The second approach is based on monotonicity of the dynamical programming function. For its calculation an efficient algorithm of determining the maximal (in relation to partial order) elements of set of k-dimensional vectors is used.

The computational complexity of the second approach is

$$O((bH)^{2M} \max_{i=1, \dots, |V(r)|} (|W(i)| |V(r)| M^{(M-2)} \log^{(M-2)}(bH)))$$

Despite the better theoretical (worst-case) efficiency of the first approach, in practice the second one has some advantages, because it can be enhanced with powerful truncation methods. In addition, it requires two times less space.

Further, for the cases when both methods of exact solution become too slow due to specifics of input data, a fast heuristical algorithm for quasioptimal solutions was designed. In particular, when k is large, and there exists essential overlapping between given subtrees T(j). It calculates the value of dynamical programming function only at points of a predefined regular grid associated with the function domain.

It has computational complexity

$$O(|V(r)| H / (C^M)),$$

where C is a constant proportional to grid step.

So, the trade-off between efficiency and quality of problem solution must be taken into account while applying the latter method.

Sometimes, it appears a necessity to get a fast estimate (not exact) solution to the problem (4)-(5). Such solution can be obtained by using a solution of the problem (1)-(3).

Let $p(i) = \{(z(1,i,j), z(2,i,j))\}$, $j=1, \dots, k(i)$ be a set of P/G net segments which connect i-th functional

IC element and corresponding VLSI P/G input pad, where $k(i)$ is a cardinality of the $p(i)$ set; $z(q,i,j)$, $q=1,2$ are planar coordinates of start and end points of j-th segment for i-th functional IC element.

Let $Z(0)$ be a placement coordinate of VLSI P/G input pad, $Z(i)$ - a placement coordinate of P/G input pad for i-th functional IC element.

P/G TOPOLOGY SKETCH will be called such a set $PP = \{p(i)\}$, $i=1,2, \dots, |V(r)|$ that each $p(i)$ (i is in $L(r)$) satisfies the following relations:

- a) $z(2,i,j) = z(1,i,j+1)$, $j=1,2, \dots, k(i)-1$
- b) $z(1,i,1) = Z(0)$
- c) $z(2,i,k(i)) = Z(1)$

The tree $T(r)$ will be called the REALIZATION of P/G TOPOLOGY SKETCH, if every path $p(r,i) = (j(1,i), j(2,i), \dots, j(t(i),i))$, $(j(m-1,i), j(m,i))$ is edge of the tree $T(r)$; $j(m,i)$ is in $V(r)$; $j(1,i) = r$; $j(t(i),i) = i$, i is in $L(r)$ satisfies the following relations:

- 1) $t(i) = k(i)$;
- 2) $l(j(m-1,i), j(m,i)) = d(z(1,i,m), z(2,i,m))$,

where d is a distance in some metric.

Let's introduce two additional notions:

- GG(PP) is a set of all realizations of P/G topology sketch;
- $W = \{w(i)\}$, i is in $V(r)$, is a set of width of all $T(r)$ segments.

Now we assume that the realization GG(PP) of P/G topology is variable therefore we shall write $V(r, T(r))$, $p(k,i)(T(r))$. Basing on the assumption above, we can describe the problem which has the variables: W and $T(r)$:

$$FF(W, T(r)) = \sum_{j=1, \dots, |V(r, T(r))|} \{l(j)w(j)\} \rightarrow \min (6)$$

$$\sum_{k=p(r,j)(T(r))} \{f_l(k)l(k)/w(k)\} \leq u(j), j=1, \dots, |L(r, T(r))|, (7)$$

$$T(r) \text{ is in GG(PP)} (8)$$

MAXIMAL realization of P/G topology is called such that the tree $TT(r)$ being in GG(PP) which satisfies the following condition:

INTERSECTION($p(r,i), p(r,j)$) is empty, if i is not equal to j for all i and j which are in $L(r)$.

So maximal tree $TT(r)$ and the set $w_0 = \{w_0(1), w_0(2), \dots, w_0(n)\}$, where

$$w_0(i) = \sum_{k=p(r,j)} \{f_l(i)l(k)/u(j)\}, i=p(r,j), j \text{ is in } L(r),$$

are the solution of the problem (6)-(8).

Conclusion

Implementation of the algorithm in order to provide optimal P/G nets for designers is an automatic feature that may become necessary in any sophisticated set of CAD tools.

References

1. Chowdhury S., Breuer M. Optimum Design of IC Power/Ground Nets Subject to Reliability Constraints. IEEE Trans. on Computer-Aided Design. - 1988. Vol. 7. N.7. - P.787-796.
2. Azarenok A.S., Rudenko A.G., Stepanec V.J. Two Methods of VLSI Power/Ground Bus Design. Notices of the National Academy of Sciences, Minsk, Ser. Phys. And Math. Sciences, No. 5. - P.93-100, 1988.
3. Rudenko A.G. About Exact Integral Solution of Optimization Problem for VLSI Power/Ground Nets. Notices of the National Academy of Sciences, Minsk, Ser. Phys. And Math. Sciences, No. 3. - P.103-108, 1990.
4. Rudenko A.G. Mathematical Models and Methods of Solution of the Problem of Optimization of IC Power/Ground Nets Widths. Preprint (13(463)) of the Mathematical Institute, Belarussian Academy of Science, 1991
5. Edelsbrunner H. A New Approach to Rectangle Intersection. Part II. // Int. J. Comput. Math. - 1983. - Vol. 13. - P.221-229.

Application of Scan Line Methodology to Perform Metric Operations in DRC

Introduction

This article describes application of the scan-line method to metric operations in DRC. This method is known to be very fast and it is widely used as a powerful engine in computational geometry algorithms including DRC algorithms [1, p.79]. We show here that the same scan-line based engine can be used for almost all DRC operations which opens very attractive possibilities for optimization of the DRC script execution process. Here we present an approach that al-

lows us to the extend scan-line technique capabilities to make it useful for building algorithms of metric operations. The methodology described here is implemented in **Savage DRC**.

Formalization of Various DRC Operations as Checking of Distances between Segments

Formalization of Metric Operations

Among all DRC operations we consider those intended to check metric correlations between different objects of topology. Such operations here are referred to as metric operations. Although every topology consists of polygons we suggest to define DRC operations in terms of segments which are edges of the polygons. If we consider only basic metric operations, such as Width, InDistance and OutDistance it is easy to show that for any of them a logical condition $C(s_1, s_2)$ on segments s_1 and s_2 could be formulated. Then such operation on some set of polygons reduces to checking of distances between segments (edges of initial polygons) which satisfy the C condition. For example, in case of metric operation Width the simplified condition $C(s_1, s_2)$ could be formulated as follows: checking of distance between segments s_1 and s_2 is performed if and only if these segments represent internal edges of the same polygon. Here we consider the "segment" as a data structure which includes not only the pair of points but also the reference to the polygon it belongs to and the side of the segment which lies inside the polygon. Thereafter this data structure is referred to as an extended segment.

Determination of Distance between two segments

We will restrict consideration here only to those DRC operations that check the condition "distance is not less than r units". The distance between two segments s_1 and s_2 is defined here as length of the shortest segment that

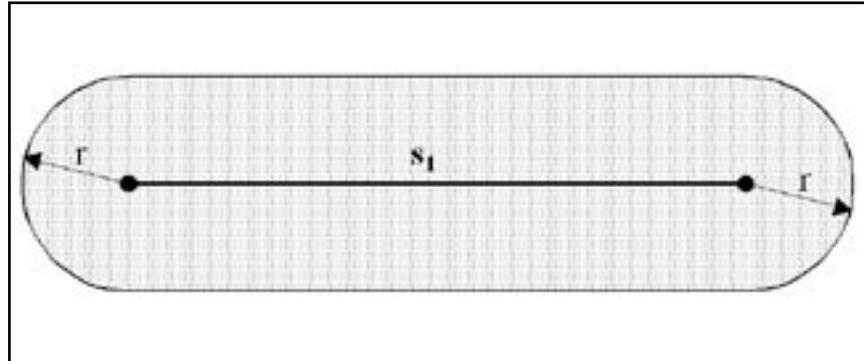


Figure 1. An example of defining neighborhood of a segment.

connects s_1 and s_2 . Then, for the distance between s_1 and s_2 to be not less than r units it is necessary and sufficient that no point of segment s_2 lies in the neighborhood of radius r around segment s_1 (see Figure 1). A similar neighborhood can be easily built for entire polygon, as it equals to external part of the union of neighborhoods of its edges. It could be an internal part of the union for other DRC operations. This kind of neighborhood is most widely used by various DRC programs when it comes to the checking of metric correlations.

Application of Scan-Line Method in Enumeration of Segments Subject to Check

Structure of scanning algorithm

Consider the building of an algorithm that checks DRC metric correlations which is based on scan-line technique. It is known that this technique could be naturally used to build algorithms for DRC operations from Logic and Select groups [1, p.110] or for determination of all the points where polygons of topology intersect each other [1, p.85]. Therefore it can be shown that considering the initial topology as a set of extended segments it is possible to divide the algorithm of DRC operations into two distinct modules: the universal abstract algorithm of scanning of topology with horizontal line and the block of calculation of these operations. The output of the first module is the input of the latter one and the data exchange between these modules occurs at only one point in which the scanning algorithm calls the block of logic operations once for each scanning level passing there piece of information called scan-line status [2, p.11]. The scan-line status is defined as pair (y, S) consisting of vertical coordinate of scan-line and set S of all segments of topology that intersect the scan-line including those segments which lie on the scan-line with one or both of their ends (see Figure 2). Let us assume that coordinates (x_1, y_1, x_2, y_2) of every segment in set S satisfy the condition $y_1 \leq y_2$.

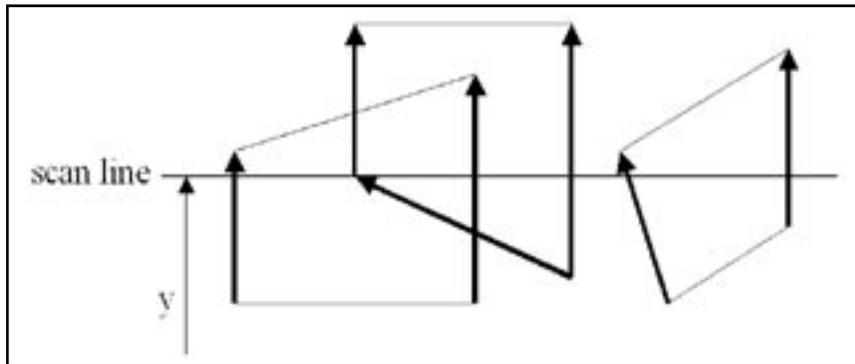


Figure 2. Scan line status at a given moment is defined as the set of segments intercepting the scan line.

Scan levels are defined by endpoints of segments and their intersection points. In order to perform DRC metric checks it is necessary to analyze information within some neighborhood of every object of topology. In our case those objects are extended segments. It is easy to see that the information that constitutes the scan-line status defined above is not sufficient for exhaustive analysis of neighborhoods of all segments from the status. Therefore it is important to find a generalization of scan-line algorithm that allows implementation of Logic or Select operation as well as metric operations. It is also important to inherit the module structure of the original algorithm to the new one.

Scan-stripe technique

The scan approach Savage employs differs from the original one only in organization of information about scan-line status. The scanning itself is carried out the same way as before. But while the original scan-line status (namely, set S) includes only those segments that intersect the line, the generalized status includes segments that have common points with a horizontal stripe which bottom border coincides with the scan-line and which width is equal to r , where r is the parameter of the metric operation. Thereafter this new status is referred to as status of scan-stripe of width r (see Figure 3). The three elements (y, r, S) describe scan-stripe status.

Obviously, when $r = 0$ scan-stripe status coincides exactly with status of corresponding scan-line. As before, scan levels are defined by endpoints of segments and their intersection points. It is easy to demonstrate that in order to reveal for every scanning level with scan-stripe status (y, r, S) all the violations of DRC metric correlations in given topology it is sufficient to consider only neighborhoods of radius r of segments from so called subset of active segments of set S (denoted as $P(S)$) and check if these neighborhoods are intersected

by any other segments from set S. Subset of active segments of set S consists of non-horizontal segments from S that lie inside scan-stripe with one or both of their ends:

$$P(S) = \{ (x1, y1, x2, y2) \text{ from } S \mid \\ x1 \text{ not equal } x2, y \leq y1 \leq y + r \text{ or } \\ y \leq y2 \leq Y + r \}$$

In particular, rule of checking some DRC metric correlation formally defined by condition $C(s1, s2)$ with parameter r could be formulated the following way: for each segment $s1$ from $P(S)$ the distance d is calculated, which is the distance from $s1$ to every segment $s2$ from S that satisfies the condition $C(s1, s2)$, $s1$ not equal $s2$, and compare this distance d with r . Violation of relation $d \geq r$ indicates that the metric correlation being checked is violated on segments $s1$ and $s2$. When $r > 0$ the scan-stripe status contains sufficient information to perform all required checks on segments from set S.

Options of metric operations

It is clear that most of standard geometrical options of DRC metric operations, such as "Check parallel/non-parallel segments only", "Check vertical/ horizontal segments only", "Check segments with/without projections only" can be immediately implemented within this technique, as it uses pairs of segments. Thus, in order to implement these options it is only necessary to formulate more complex condition $C(s1, s2)$ to select segments subject to check. However, it is more difficult but also possible to implement such polygon-based options of metric operations as "Do/don't check overlapping polygons" by adding some piece of information to the extended segment data structure. It is worth to mention that certain minor extensions of data structure that represents a segment allow one to implement such DRC features as conjunctive rules or check in selected area of layout only etc.

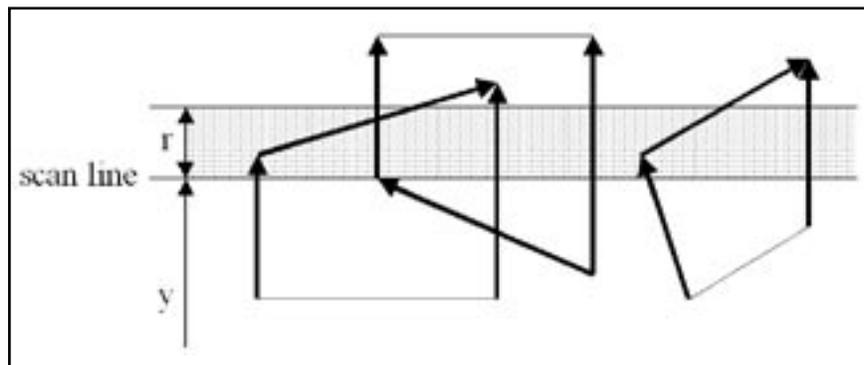


Figure 3. Scan stripe status at a given moment is defined as a set of segments intercepting the scan stripe.

Optimization Considerations

As mentioned above, one of the main advantages of DRC algorithms based on scan-line technique is the significant DRC script optimization possibilities it offers. Analyzing the information flow in the graph of informational dependencies of some DRC script we search for situations that allow us to make use of the following optimization methods:

- boolean sub-formulas extraction;
- merging of sequential operations;
- merging of informationally independent operations.

The first one is well known optimization method that consists in finding coinciding sub-formulas within all logical formulas of given script and separating them into individual boolean operations. For instance, the sequence of formulas below:

$$\begin{aligned}A &= (B \text{ .or. } C) \text{ .and. } (D \text{ .xor. } E); \\F &= (B \text{ .or. } C) \text{ .dif. } G;\end{aligned}$$

could be transformed by optimizer into the following sequence:

$$\begin{aligned}T &= B \text{ .or. } C; \\A &= T \text{ .and. } (D \text{ .xor. } E); \\F &= T \text{ .dif. } G;\end{aligned}$$

On the one hand, this optimization removes the unwanted repeated calculation of sub-formulas thus contributing to increase of script execution speed. On the other hand, it replaces two initial script commands with three commands that generally may lead to increase of per-operation preprocessing time which is very noticeable in scan-line method based algorithms. Experimental results show that in our scan-line method based DRC system this kind of optimization yields substantial positive effect on overall script execution speed. However, it is only useful with formula-based DRC operations, namely - operations from Logic group.

The second method of optimization employed consists in splitting the graph of informational dependencies into chains of operations which sequentially depend on each other and merging operations in every single chain into smaller number of more complex operations. The straightforward example of such optimization can be built from Logic operations: sequence of operations

$$\begin{aligned}A &= B \text{ .or. } C; \\D &= E \text{ .and. } F; \\G &= A \text{ .xor. } D;\end{aligned}$$

is transformed by optimizer into single operation:

$$G = (B \text{ .or. } C) \text{ .xor. } (E \text{ .and. } F);$$

which can be carried out by single scanning pass. Although the more complex operation requires longer preprocessing time, but according to our benchmarks this optimization method gives a largedecrease of script execution time by removing those excessive per-operation

preprocessing stages. The most important thing is that in the DRC system that is completely based on scan-line method, not only for Logic operations but operations of any kind may be merged with each other and executed in one scan-line pass. The last simple optimization method gives considerable improvement specifically on metric operations of DRC. Informationally independent metric operations satisfying certain conditions may be executed simultaneously in one pass of a scan-line algorithm. Of course, it is applicable to Logic operations as well.

Conclusion

The techniques described above were thoroughly tested on real layouts in **Savage**. The scan-line algorithms and it proved to be reliable and effective on layouts with millions of transistors. Employment of the optimization methods we mentioned allows users to obtain up to three times DRC script performance increase in comparison with non-optimized script execution on most of practical examples.

References

- [1]. V. Feinberg, Geometrical problems of VLSI computer graphics, Radio i Sviaz, Moscow, 1987.
- [2]. F. P. Preparata, M. I. Shamos, Computational geometry. An introduction, Springer-Verlag New York Inc., 1985.

Calendar of Events

December

1
2
3
4
5
6
7
8 IEDM 97, Washington D.C.
9 IEDM 97, Washington D.C.
10 IEDM 97, Washington D.C.
11
12
13
14
15
16
17
18
19 UTMOST W/S - Yokohama
20
21
22
23
24
25 Christmas Day
26
27
28
28
30
31

January

1 New Years Day
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 SmartSpice W/S - Yokohama
24
25
26
27
28
28
30
31

Bulletin Board



Silvaco to Demonstrate Latest Advances in Ultra Deep Sub-Micron EDA at ASP-DAC'98

Silvaco will be exhibiting its advanced TCAD Driven CAD fully integrated family of EDA tools in booth C-201 at ASP-DAC'98. The conference is being held on February 10-13 at the Pacifico Yokohama in Yokohama Japan. Users will have the opportunity to meet with Silvaco Japan's engineers and see demonstrations of Silvaco's innovative simulation, characterization, and design & verification software products. Among the highlights will be the introduction of new Silvaco product releases including SmartSpice v1.5.4 and v1.3.0 of the Expert PC-NT Layout Editor.



Visit Silvaco at ISSCC'98

Silvaco will host a hospitality suite at ISSCC'98 in the Marriott, San Francisco on February 5-7. Conference participants are invited to visit and see the latest advances in sub-0.25u circuit simulation and IC design & verification software. Silvaco's engineers will be demonstrating powerful new technology based tools for highly accurate LPE, analog and mixed-signal circuit simulation, and Windows-NT based layout editing and design rule checking.



1998 TCAD Workshop Series to be Launched in Phoenix

A series of Process and Device simulation workshops will be given in Phoenix on Feb. 11-12 and 25-26. Details on the topics to be covered are available at www.silvaco.com, or users may contact Silvaco's Southwest Office at:

az_sales@silvaco.com.

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. 8, No. 12, December 1997 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, TCADDrivenCAD, Virtual Wafer Fab, Analog Alliance, Legacy, ATHENA, ATLAS, FAST ATLAS, ODIN, VYPER, TSUNAMI, RESILIENCE, TEMPEST, CELEBRITY, Manufacturing Tools, Automation Tools, Interactive Tools, TonyPlot, DeckBuild, DevEdit, Interpreter, ATHENA Interpreter, ATLAS Interpreter, Circuit Optimizer, MaskViews, PSTATS, SSuprem3, SSuprem4, Elite, Optolith, Flash, Silicides, SPDB, CMP, MC Deposit, MC Implant, Process Adaptive Meshing, S-Pisces, Blaze, Device 3D, Thermal 3D, Interconnect 3D, Blaze3D, Giga3D, MixedMode3D, TFT, Luminous, Giga, MixedMode, ESD, Laser, FastBlaze, FastMixedMode, FastGiga, FastNoise, MOCASIM, UTMOST, UTMOST II, UTMOST III, UTMOST IV, PROMOST, SPAYN, SmartSpice, MixSim, Twister, FastSpice, SmartLib, SDDL, EXACT, CLEVER, STELLAR, HIPEX, Scholar, SIREN, ESCORT, STARLET, Expert, Savage, Scout, Guardian and Envoy are trademarks of Silvaco International.

Hints, Tips and Solutions

Mikalai Karneyenka, Applications and Support Engineer

Q1: When I plot my design on a laser printer some regions are not plotted at all.

Q2. Stipple patterns on screen and on plot look differently.

A: In both cases the reason is the same. Unfortunately, color capabilities of screen and plotting devices often do not match. In particular, some non-saturated colors clearly visible on the screen are mapped into white color on drawing. We are planning to add color remapping capabilities for plot setup. At the moment, however, if you are not satisfied with colors of drawing, you may do the following.

- Select screen colors that are mapped to plotter colors in a satisfactory way by trial printing of a set of sample color boxes.
- Prepare an alternative technology file either using a text editor or from within **Expert** as follows:
- Change layer colors using Layer Setup;
- Save the technology file (using menu command Project>>Save Technology) under another name. In fact, you should have as many alternative tch files as you have different plotting devices.
- If you want to print the project only once, just change layer colors and print. If you want to print in several separate sessions, you must:
- Save your project/cell as gdsII file;
- Read from gdsII file using an alternative technology file;
- Plot

Q: How can I select an object if it is completely covered by other layout objects?

A: There are two ways to do this.

- (1) If you know the layer of the required object, you may hide the overlaying layers.



Figure 1. Menu for locating obscured objects.

- (2) If you do not know the layer or the objects are overlapping in the same layer, or if you simply do not want to mess with layer visibility, you may use the Locate operation as follows:

- On the Locate bar check the box "Top Level Only". Under this option Locator will pick objects that are normally picked by Select operation, i.e., in the top level of the current cell.
- Draw a selection box;
- Click right mouse button (or Next button on the Locate bar) several times until you see the required object highlighted.
- Click "Select" button on the Locate bar.

As you see from the Locate bar, there is an option to make the chosen object selected in the corresponding cell when you enter Edit-in-Place from Locate operation.

Q: I run DRC, find one or two violations, correct them, then I re-run DRC to check whether my corrections worked. However re-running on the whole design is time-consuming. How can I run DRC over the section of the layout in the vicinity of the introduced changes only?

A: Now you may do this in one of the two ways:

- (1) You may select objects by box and run a script on selected objects. However this run will work only on top-level objects of the current cell. Instances will not be processed by such check. You may explode the instance several times until the selected objects are flattened and then run the check. This approach has a number of drawbacks. You must backup the original design and the DRC will not be hierarchical.
- (2) You may clip out the selected region (using the Tools>>Clip Out command) and run DRC on the clipped piece. The coordinates in the clipped piece are the same as in the original design, so that you can locate errors in the original, since errors are reported as segments with coordinates of their ends. The drawbacks are spurious error reports along the clip boundary and "semiautomatic" error location. However you will get hierarchical DRC.

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
<http://www.silvaco.com>



Join the Winning Team!

Silvaco Wants You!

- SPICE Application Engineers
- TCAD Application Engineers
- CAD Application Engineers
- Software Developers

fax your resume to:
408-496-6080, or
e-mail to:
personnel@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

4701 Patrick Henry Drive, Building 2
Santa Clara, CA 95054

Telephone: (408) 567-1000
Fax: (408) 496-6080
URL: <http://www.silvaco.com>