Simulation Standard

Connecting TCAD To Tapeout

A Journal for CAD/CAE Engineers

Resistor and Capacitor L/W Parameters in LVS Comparison

Introduction

In the design of VLSI circuits the situations occur very often when a single schematic device is implemented in the layout by the group of several devices connected in parallel or series. Such groups of devices must be reduced to a single device in the layout and the circuits must be compared in terms of the single devices taking into account their geometrical characteristics. In this article we introduce some methods of calculation and comparison of geometrical parameters of resistors and capacitors connected in parallel or series, when LVS verification is performed.

Resistor L/W parameters

Guardian LVS reduces the group of parallel resistors to a single resistor, if the **Parallel merge** option in **Models Settings** panel for resistors is on. All resistors in the group must have the same type, the same number of terminals and each terminal of a resistor must be connected to the same or swappable terminal of another resistor (see Figure 1). All optional (bulk) terminals of resistors must be connected to the same net.

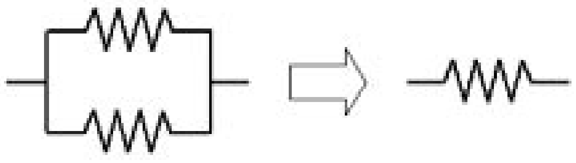


Figure 1. Parallel Resistor Reduction

Guardian LVS calculates, along with resistance, the width and length of resultant resistor using the following formulas:

$$W = \sqrt{P * Q}$$

$$L = \sqrt{\frac{P}{Q}}$$

where P and Q are calculated as

$$P = W_1 * L_1 + W_2 * L_2 + \cdots + W_n * L_n$$

$$Q = \frac{W_1}{L_1} + \frac{W_2}{L_2} + \cdots + \frac{W_n}{L_n}$$

where W_i and L_i. are width and length of the *i-th* resistor, respectively.

Guardian LVS can reduce also a group of serial resistors to a single resistor, if the **Series merge** option in **Models Settings** panel for resistors is on. All resistors in the group must have the same type and the same number of terminals (see Figure 2). All positive and negative terminals must be connected in series. As the positive and negative terminals of resistor are always swappable, the positive-to-positive, positive-to-negative, negative-to-positive, and negative-to-negative connections are admissible for the reduction. All optional (bulk) terminals of resistors must be connected to the same net.

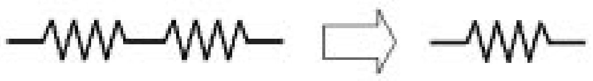


Figure 2. Serial Resistor Reduction.

Continued on page 2 ...

INSIDE

SILVACO INTERNATIONAL

The width and length of resultant resistor are calculated as follows:

$$W = \sqrt{\frac{P}{Q}}$$

$$L = \sqrt{P * Q}$$

where P and Q are calculated as

$$P = W_1 * L_1 + W_2 * L_2 + \cdots + W_n * L_n$$

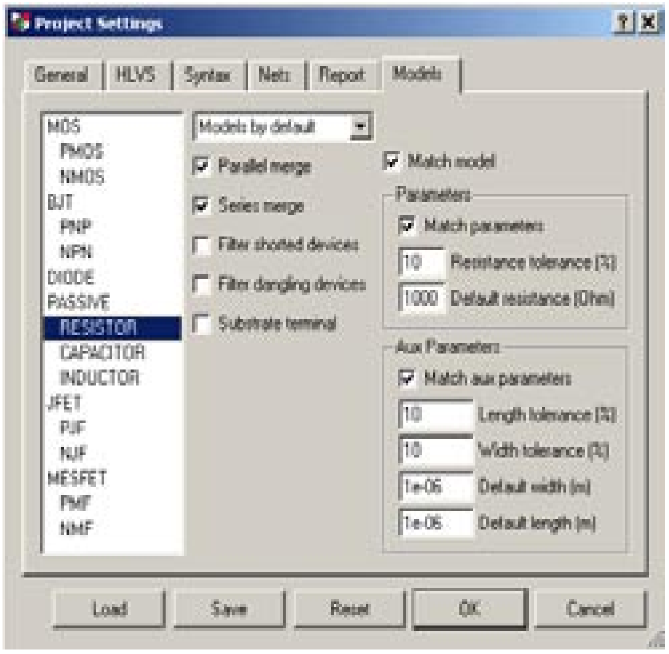
$$Q = \frac{L_1}{W_1} + \frac{L_2}{W_2} + \cdots + \frac{L_n}{W_n}$$

where W_i and L_i are width and length of the i-th resistor, respectively.

The parameters of matched resistors can be checked during netlist comparison using **Match parameters** and **Match aux parameters** options (see Figure 3). There are three combinations to check the resistor parameters:

- check only resistance values of matched resistors;
- check the length and width of matched resistors;
- check the resistance, length, and width of matched resistors.

You can change the tolerance and the default values of resistance, length, and width parameters. Initial values for default length and width and their tolerance are 1e-6m and 10%, respectively.



Models by default

[✓] Parallel merge
[✓] Series merge
[ ] Filter shorted devices
[ ] Filter dangling devices
[ ] Substrate terminal
[✓] Match model

Parameters
[✓] Match parameters
10 Resistance tolerance (%)
1000 Default resistance (Ohm)

Aux Parameters
[✓] Match aux parameters
10 Length tolerance (%)
10 Width tolerance (%)
1e-06 Default width (m)
1e-06 Default length (m)

Figure 3. Resistor Parameters

If ***Guardian LVS*** finds a mismatch for at least one parameter, all parameters of matched resistor pair are reported in **Parameter Error** report like:

```
L1: R: top:X9:RH14 ......... = (#)  RX:XI2:XI3:XI0<2>:RI#32
       R=1300                       R=1300
       L=62u  W=5u                  L=50u  W=5u
```

Note that **Match parameters** and **Match aux parameters** options help LVS to resolve node automorphism. When there are equal groups of devices that can't be distinguished from each other, additional information is needed for matching. Comparing device parameters helps LVS to match devices. But, if all devices within a group have the same parameters, LVS can't differentiate between group members. In this case, you can set the initial correspondence points to resolve node automorphism and reduce verification time.

Capacitor L/W Parameters

Guardian LVS can reduce a group of parallel capacitors to a single capacitor, if the **Parallel merge** option in **Models Settings** panel for capacitors is on. All capacitors in the group must have the same type, the same number of terminals. All positive, negative, and optional (bulk) terminals must be connected to the same nets (see Figure 4). Note that positive and negative terminals can be swapped, if **Swap terminals** option is on.

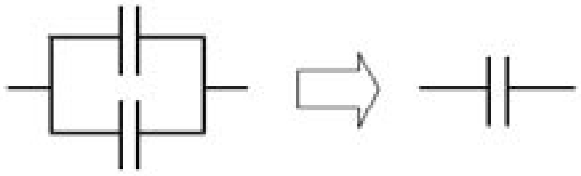


Figure 4. Parallel Capacitor Reduction

Guardian LVS calculates, along with capacitance value, the width and length of resultant capacitor using the following formulas:

$$W = \frac{P - \sqrt{P^2 - 16 * A}}{4}$$

$$L = \frac{P + \sqrt{P^2 - 16 * A}}{4}$$

where A and P are area and perimeter of resultant capacitor calculated as:

$$A = L_1 * W_1 + L_2 * W_2 + \cdots + L_n * W_n$$

$$P = 2 * (L_1 + W_1 + L_2 + W_2 + \cdots + L_n + W_n)$$

where W_i and L_i are width and length of the i-th capacitor, respectively.

Continued on page 6 ...

Cross-Sectional Viewer

Introduction

With the increase in complexity of IC layouts and processes the designer must have knowledge not only of layout design, but of simulation and manufacturing as well.

Cross-Sectional Viewer is a tool in the ***Expert*** layout editor that simulates the cross-sections of ICs along an arbitrary drawn cut-line on the layout. Cross-Sectional Viewer is a link between the layout and the resulting device and it helps the designer to understand its spatial structure. Cross-sectional drawings are useful for detecting parasitic coupling, undesired metal stacking, and other design and fabrication problems.

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

Cross-Sectional Viewer Setup

For the simulation of fabrication steps specific parameters must be specified in the technology file. For simplicity we assume that each fabrication step is associated with a separate layer in the layout.

The user can create and modify process simulation parameters interactively using the Cross-sectional Viewer setup panel of the Preferences window of ***Expert***. (see Figure 1)

The different steps to follow for setting up of the Cross-Sectional Viewer are as follows.

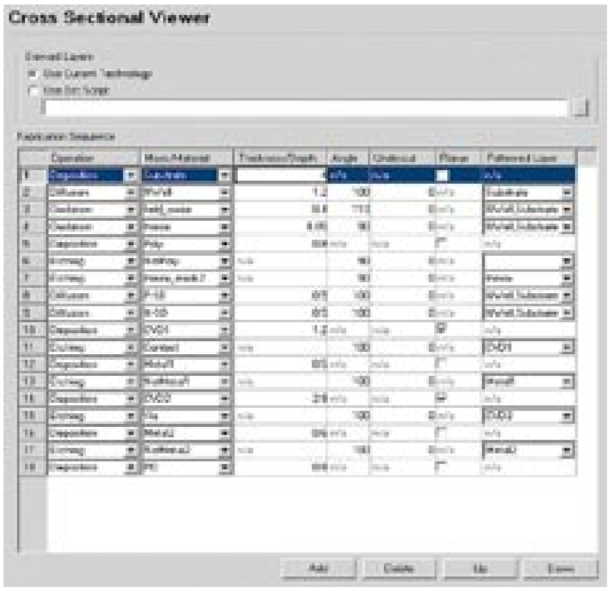
Derived Layers

Before defining the process steps in the **Cross-sectional Viewer setup,** the user has to check that all these steps can be defined with existing layers of the layout. Otherwise, the missing layers have to be generated generally using boolean operations on other layers.

For example, if the user needs to define etch operation on a deposit of Metal1, the technology file should contain a layer which will be the reverse mask of Metal1.

So one must define the corresponding derived layer in ***Expert*** technology as follows:

- In **Layer Setup** panel (menu: **Setup>>Technology>> Layer setup**), create a new layer and define it as 'derived'.



	Operation	Mask/Material	Thickness/Depth	Angle	Undercut	Planar	Patterned Layer
1	Deposition	Substrate	[illegible]	n/a	n/a	☐	n/a
2	Diffusion	NWell	1.2	100	0 n/a		Substrate
3	Oxidation	field_oxide	0.4	[illegible]	0 n/a		NWell Substrate
4	Oxidation	[illegible]	0.05	90	0 n/a		NWell Substrate
5	Deposition	Poly	[illegible] n/a	n/a	☐		n/a
6	Etching	EtchPoly	n/a	90	0 n/a		
7	Etching	[illegible]	n/a	90	0 n/a		[illegible]
8	Diffusion	P-SD	0.5	100	0 n/a		NWell Substrate
9	Diffusion	N-SD	0.5	100	0 n/a		NWell Substrate
10	Deposition	OVD1	1.2 n/a	n/a	✓		n/a
11	Etching	Contact	n/a	100	0 n/a		OVD1
12	Deposition	Metal1	[illegible] n/a	n/a	☐		n/a
13	Etching	EtchMetal1	n/a	100	0 n/a		Metal1
14	Deposition	OVD2	[illegible] n/a	n/a	✓		n/a
15	Etching	Via	n/a	100	0 n/a		OVD2
16	Deposition	Metal2	[illegible] n/a	n/a	☐		n/a
17	Etching	EtchMetal2	n/a	100	0 n/a		Metal2
18	Deposition	[illegible]	[illegible] n/a	n/a	☐		n/a

Figure 1. Cross-sectional viewer setup.

- In the **Derivation setup panel,** enter the boolean operation defining the "Etch_Metal1" layer representing the mask of etching:

```
Dif: Layer1=Substrate, Layer2=Metal1,
LayerR=Etch_Metal1;
```

- Activate menu **Derivation>>Accept derivation** of this editor in order to validate this definition and its syntax.

In this way the layer "Etch_Metal1" will be included in the technology: it can be used during the definition of process steps of the **Cross-sectional Viewer** and it will be generated automatically when launching a Cross-Sectional-Viewer like all other derived layers of the technology file.

Fabrication Sequence

The second step of the **Cross-sectional Viewer** setup is the Fabrication sequence table definition. It allows defining one-by-one all the steps of the process beginning from the bottom (substrate) to the top of the chip.

The available commands in order to define a process step are as following:

- Create a process step using the **Add** button. In order to add a process step, the user must first select the step above the place where he wants to add a process step. The use can then click the **Add** button and a new step is added in the table and the user can set the parameters of this step.

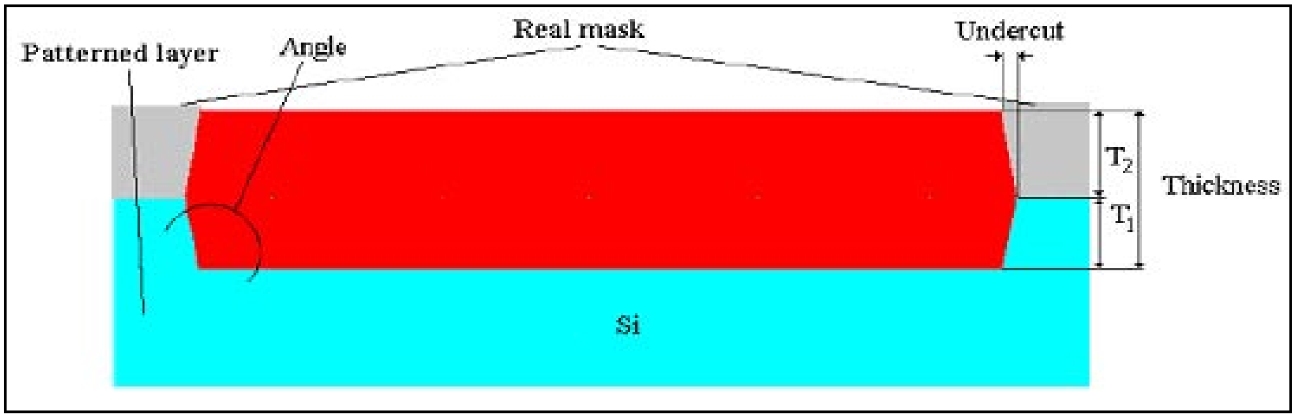


Figure 2. Local oxidation parameters.

- Define the kind of **Process operation**:

  A process step is defined first by the operation it consists in. Listed below are the process operations and the meaning of necessary parameters. There are four different kinds of process operations that are available:

  - DEPOSITION
  - ETCHING
  - IMPLANTATION / DIFFUSION
  - OXIDATION

- **Mask/Material** layer:

  The layer you will choose will represent either the material (in the case of a deposit or an oxidation), or the mask (in the case of an etching or an implant/ diffusion.).

  Note: in the case of a deposit, the geometry of the layer does not matter: the deposit will be done on all off the surface anyway.

- **Thickness/Depth** parameter:

  This value will define the thickness of the material in the case of a deposit or an oxidation. In the case of a etching or an implant, this will define the maximum depth where the etch or the implant will end.

- **Angle** value:

  This is equivalent to the angle between the horizontal base of the material and its side. See Figure 2 (oxidation case) and Figure 4 (etching case) for examples.

- **Undercut** value:

  This parameter allows the user to introduce a shift between the border of the geometry of the mask layer and the place where the process operation will be applied. See Figure 2 (oxidation case) and Figure 4 (etching case) for examples.

- **Planar** option:

  This option only concerns the deposit operation. Checking this option makes the current deposit planar. The thickness of the deposit will be calculated from the top of the last planar deposit. If the option is not checked the deposit will be conformal.

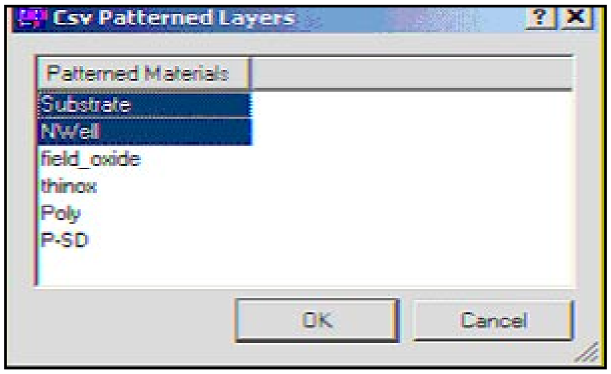
- Patterned Layers list:

  This is available for etching, implantation or oxidation only. In the **Patterned Layer** drop-down list, you can choose which material(s) will be processed: in other words this feature allows a selective etching, implantation or oxidation.

  When selecting the **"Edit..."** field of this list, the **Csv patterned layers** window will appear (see Figure 3). In this window the user is prompted to select the list of materials that can be etched/implanted/oxidized during this process step.

  Note: Thanks to **Patterned Layers** list and **Depth** parameters the user can define precisely the etching/ implant/oxidation results:

- If the process step is independent of the material the user can select all the layers of the Patterned layer list. In this way only the Depth parameter will influence the process operation.



Patterned Materials
Substrate
NWell
field_oxide
thinox
Poly
P-SD

Figure 3. CSV patterned layers window.

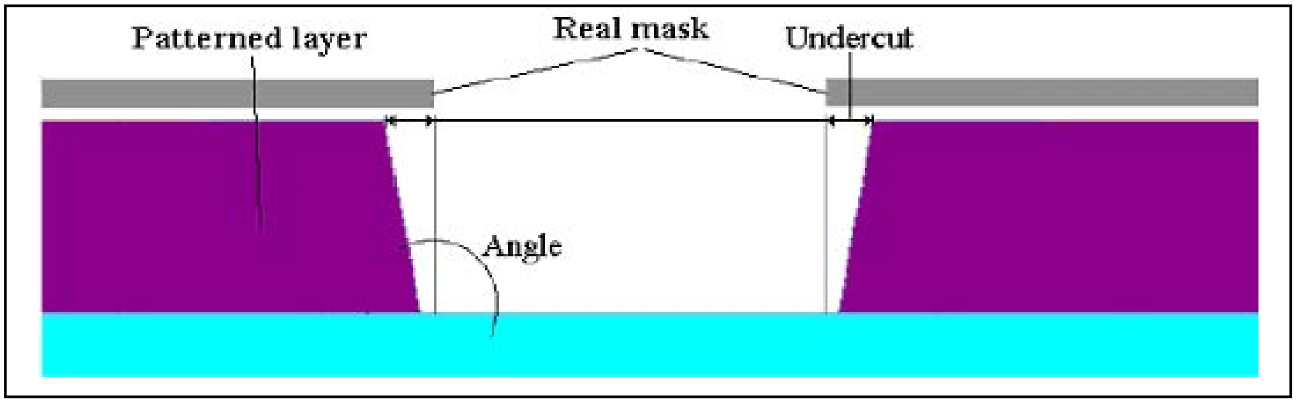


Figure 4. Etching step parameters when using “Patterned layers”.

- If the process step depends only on the material(s) to be etched/implanted or oxidized, the user has to set a high value for the Depth parameter, so it will not influence the result. (See etching Figure 4 for an etching example where the Patterned layer is the purple one)
- If the etch/implant or oxidation is done on specific material(s) and with a particular depth, the user will have to specify both a Patterned layer list and Depth value.

Using the Cross-Sectional Viewer

Once the user has defined Cross-Sectional Viewer setup information, he owns a technology file which can be used for any layout of the same technology for generation of cross-sections.

The Cross-Sectional-Viewer consists of two separate dock-windows for its current use:

- The **Cross-Sections** window (see Figure 5) is used to manage the generation of new cross-sections in the opened cell of the layout, and also to switch between existing cross-sections.
- The **Cross-Sectional View** window is the output window (see Figure 6) where cross-sections are displayed with all of the visualization commands of the Cross-Sectional Viewer.

Cross-Sections Generation

Cross-sections can be generated in this way:

- One must click the green ‘+’ button of the **Cross-Sections** window,
- Specify a cut-line in the active cell of the layout either by mouse click, or with XI script command: ‘cross view (start_x) (start_y) (end_x) (end_y)’
- After drawing the cut-line, the processing is automatically launched. **The Cross-Sectional View** window displays the calculated cross-section, and the coordinates of the cut-line are displayed in the **Cross-segments** list of the **Cross-Section** window.

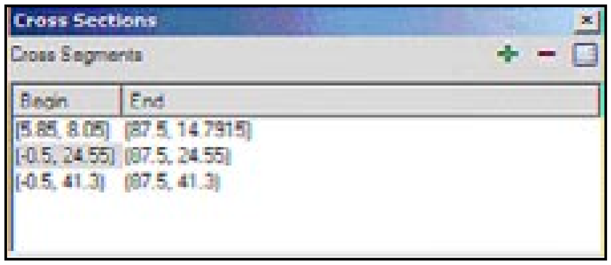
The user can repeat this step as many times as he needs to create several cross-sections: they will be stored as long as the project is opened. The cut-line of the active cross-section is highlighted in red (see complete example in Figure 7).

Cross-Sections Visualization

When generating a cross-section, it appears in the **Cross-sectional view** window (see Figure 6).

Beside coordinates, axes, and zoom functions, advanced features are available in this window in order to improve visualization of the materials stacking along the cross-section:

- **Step-by-step** view: The Cross-sectional Viewer has a feature that allows the user to view the individual fabrication processes one step at a time with an actual sequence of process steps. There are four buttons managing this step-by-step view in the **Cross-sectional view** window:
  - **First Operation** and **Last Operation** of the fabrication sequence,
  - **Previous Operation** and **Next Operation** buttons allow seeing respectively the material stack-



Begin	End
[5.85, 8.05]	[87.5, 14.7915]
[-0.5, 24.55]	[87.5, 24.55]
[-0.5, 41.3]	[87.5, 41.3]

Figure 5. Cross-sections window.

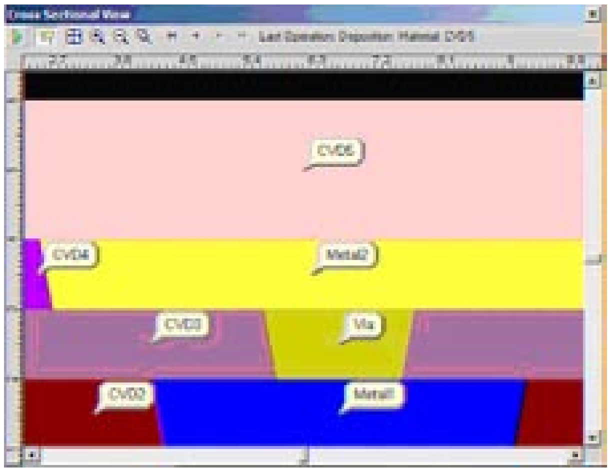


Figure 6. Cross-sectional view window.

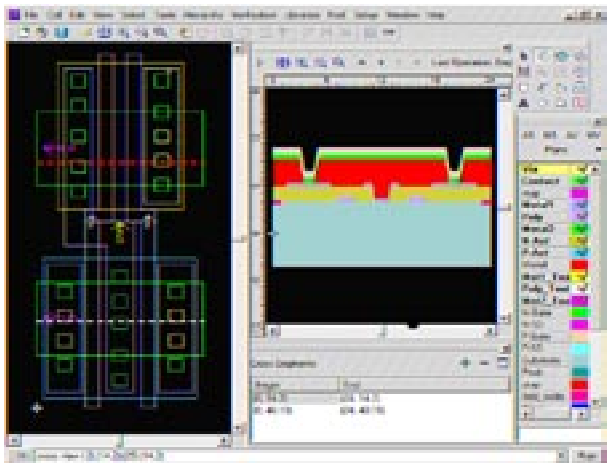


Figure 7. Cross-Sectional Viewer integration in Expert GUI.

ing before and after the current process step. The current process step is detailed on the right side of the **step-by-step** buttons with the **Material** name and the **Operation** name.

- **Cross-sectional view update:** It may be useful to update a cross-section after some changes in the layout or in the process parameters. By clicking the **Update** button (green arrow) in the **Cross-sectional view** window, the user can regenerate the selected cross-section without having to redraw the cut-line.
- **Material labels** button: This feature displays the names of materials wherever you inspect the cross-section. (see Figure 6)

The Cross-Sectional-Viewer and all the features described in this article are available now in the 2006 release of ***Expert*** (version 4.2.8.R) on Windows, Solaris, and Linux RedHat.

...continued from page 2

Note that *L* and *W* parameters of resultant capacitor can be calculated, if $P^2 - 16 * A \geq 0$.

Guardian LVS reduces a group of serially connected capacitors to a single capacitor, if the **Series merge** option in **Models Settings** panel for capacitors is on. All capacitors in a group must have the same type, the same number of terminals.

The positive and negative terminals must alternate within series, unless **Swap terminals** option is on. All optional (bulk) terminals must be connected to the same nets (see Figure 5).

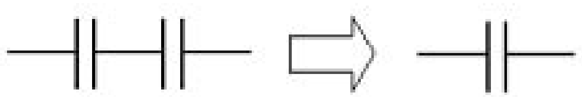


Figure 5. Serial Capacitor Reduction

There are no formulas for calculation of the width and length of capacitor obtained by serial reduction. Only the capacitance value can be calculated.

The parameters of matched capacitors (same as resistors) can be checked during netlist comparison using Match parameters and Match aux parameters options. If these options are on, the capacitance, length, and width of matched resultant capacitors, connected in parallel, are compared and reported, as in example below:

```
L0: C:($)  TN007:CI157 ......... = ($) TN007:CI157
           C=1.01507E-011              C=1.01507E-011
           L=268.478u W=30.6359u       L=268.478u W=30.6359u
```

If the length and width can't be calculated (for example, for serial capacitor reduction), the capacitance values of matched resultant capacitors and the length and width of capacitors forming the resultant capacitor are compared and reported like:

```
L0: C:($)top:C1 ............ = (@#) S3:XS2:C2
          C=1.97464E-012           C=1.97464E-012
          top:C1                   S3:XS2:C2
          L=123.038u  W=36.212u    L=103.038u  W=37.212u
          top:C3                   S3:XS2:C2
          L=103.038u  W=37.212u    L=103.038u  W=37.212u
```

Conclusion

Described methods of calculation and comparison of resistor and capacitor geometrical parameters are realized in Guardian LVS verification tool. This allows designers to execute more detailed comparison of the netlists representing the layout and the schematic of a circuit.

Using *Expert's* Pcell Feature to Generate Complex Shaped Layers

One of the strengths of ***Expert*** is to generate complex shaped layers such as spiral inductors, parabolic waveguides, etc. For the case of the spiral inductor, it has already been well-documented in Celebrity example package as "spiral.eld", from which interested users may learn or use.

In some specific applications, e.g. photonics design, designers may need to generate parabolic-shaped layers such as waveguides. For example, a customer is designing a layer as seen in Figure 1.

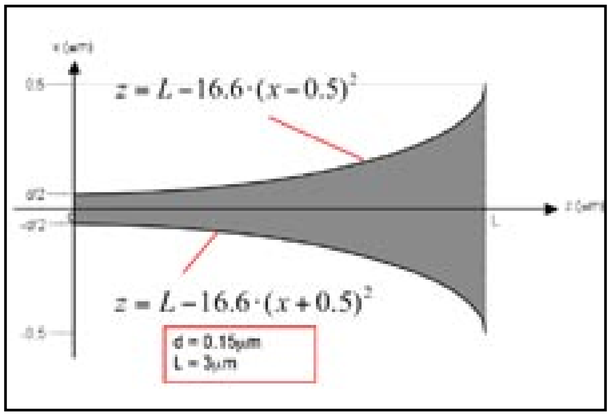


Figure 1. Parabolic shaped layer to be drawn.

Layers like this may appear complicated but by using ***Expert's*** Pcell feature, they can be implemented with ease. The idea is based on using many small rectangles to approximate the actual shape as seen in Figure 2. By choosing a proper resolution, the generated layer will be much closer to the desired shape. The error will be so small that it can be neglected under current process technologies.

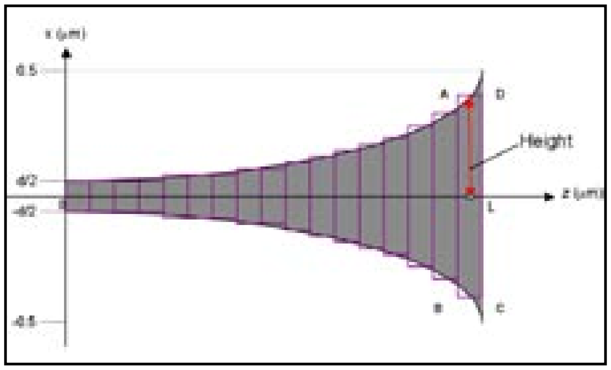


Figure 2. Rectangular shaped approximation, for demonstration, actual resolution should be higher to minimize deviation.

Knowing L = 3um and d = 0.15_m, from $z = L - 16.6\bullet(x\text{-}0.5)^2$ we have

$$x = 0.5 \pm \sqrt{\frac{L-z}{16.6}}\,\mu m = 0.5 \pm \sqrt{\frac{3-z}{16.6}}\,\mu m.$$

For any rectangle, this equation can be used to estimate its height, with respect to the center's z-coordinate.

```
DEFINE PCELL "parabolic_single" /REPLACE
  PARAMETER height /TYPE = (Double) /DEFAULT = (0)
  PARAMETER i /TYPE = (Double) /DEFAULT = (0)
  PARAMETER layer_name /TYPE = (String) /DEFAULT = ("Layer6")
  BODY BEGIN
  units /um;
  i = 0.00;
  loop begin
    if (i GEQ 100) then (leave loop);
    height = 0.5 - sqrt((3 - (0.015 + 0.03 * i) )/16.6);
        layer (layer_name);
    box  (0.03 * i) (-height) 0.03 (2 * height);
    i = i + 1;
  end;
END;
```

Figure 3. Text input of Xi script panel.

The Pcell Xi-Script for the case of L = 3μm will be like Figure 3.

In this Xi script, we define a parameter "height" for each rectangle. The parameter "i" is the loop parameter. In the "if…then…" statement, "100" is the iteration time, i.e. the number of rectangular shapes. In this example, since L=3um, each rectangle will have a width of 0.03um. To minimize deviation, users can set the iteration time to be larger. By substituting the z value, the height for each can be written as:

$$x = 0.5 - \sqrt{\frac{3 - (0.015 + 0.3\bullet\, i)}{16.6}}\quad \mu m$$

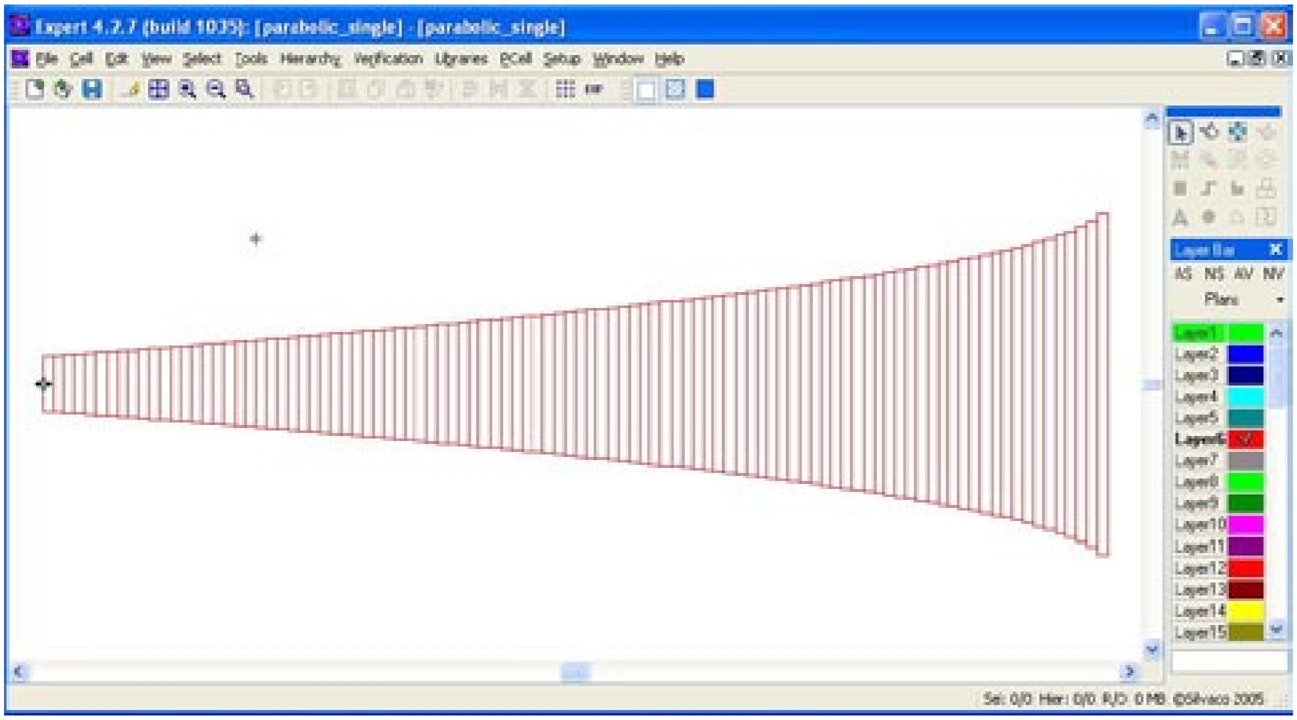

Expert 4.2.7 (build 1035): [parabolic_single] - [parabolic_single]
File Cell Edit View Select Tools Hierarchy Verification Libraries PCell Setup Window Help
Layer Bar
AS NS AV NV
Plans
Layer1
Layer2
Layer3
Layer4
Layer5
Layer6
Layer7
Layer8
Layer9
Layer10
Layer11
Layer12
Layer13
Layer14
Layer15
Sel: 0/0 Hier: 0/0 R/O 0 MB ©Silvaco 2005

Figure 4. Generated layer.

After the "height" of the rectangle is found, we can proceed to draw the rectangular shape for this "i" value with the "box" statement, where each () contains the following parameters:

(0.03 * i) (-height):	Coordinates of the bottom-left corner.
0.03 (2 * height):	Width and height of the box.

Repeat for all 100 shapes until the loop is finished. This operation is quite similar to the "for" loop in C language.

After running the Xi script, ***Expert*** will generate a layer as described, as shown in Figure 4.

A concern about the file size then arises. The layer generated using the above method produces a large file size, especially when the layer size and iteration time are large. For certain designs, there might be hundreds of these layers needed. To overcome this problem, we need to do the following:

1. Export the current project into gds format from **Expert** ⇒**File** ⇒ **Export** ⇒, select output format into **GDS** ⇒ **Save**.
2. Re-open/import that gds format and edit from **Expert** ⇒ **File** ⇒ **Import**, select all small rectangles and merge all from **Expert** ⇒ **Tools** ⇒ **Merge** selected.
3. Go to library setup from **Expert** ⇒ **Library** ⇒ **Setup**, add and activate this project we just edited as one of the libraries.
4. In layout where these layers need to be added, use "add instance" features, and add layers.

With the above steps, the file size is reduced greatly. Take for example of a layer with L=1000um and iteration time=10000, in Step 1, the unmerged *.eld file size is 987KB. In Step 2 the merged *.eld file has the size of 393KB. While in the layout where this layer has been added by the "add instance" feature, the file size has reduced to 81KB.

Pcell is indeed a very useful feature of ***Expert***. Through these simple procedures, the complex layer such as parabolic shapes with varying lengths can be generated without suffering from inaccuracy or big file sizes. By applying this feature, the customer will reduce both effort and cost expended. Most customers using other layout editors will have to use a separate photonic tool to generate those layers, export them to GDSII, before importing them into the layout editor. One does not need to switch between various tools when using ***Expert***.

Calendar of Events

December

1
2
3
4
5
6
7 FSA PDK Checklist Announcement
8
9
10
11
12
13 IEDM - San Francisco, CA
14 IEDM - San Francisco, CA
15 IEDM - San Francisco, CA
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

January

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 Electronic Imaging-San Jose, CA
17 Electronic Imaging-San Jose, CA
18 Electronic Imaging-San Jose, CA
19 Electronic Imaging-San Jose, CA
20 Electronic Imaging-San Jose, CA
21
22
23
24
25
26
27 Electronic Design and Solution Fair - Yokohama, Japan
28 Electronic Design and Solution Fair - Yokohama, Japan29
30
31 Design Con - Santa Clara, CA (Jan 31 - Feb 2)

Bulletin Board

Leading Foundries Spur Widespread Adoption of Fabless Semiconductor Association Mixed-Signal/RF PDK Checklist

The Fabless Semiconductor Association (FSA), the voice of the global fabless business model, announced today that its Mixed-Signal/RF PDK Checklist, released in March 2004, has been adopted as standard practice into top-tier foundry PDK development workflows and delivered with multiple foundry process design kits (PDKs). The checklist documents the contents of a PDK, which is a set of data files that enable analog circuit and layout designers to efficiently design a semiconductor chip using a set of electronic EDA tools and a selected foundry process.

Foundries currently adopting the checklist include 1st Silicon, austriamicrosystems, Jazz Semiconductor, PolarFab, Tower Semiconductor, TSMC, UMC and X-FAB. Additionally, EDA vendors and fabless semiconductor members of the FSA's PDK working group that are also adopting the checklist include Agilent Technologies, Cadence Design Systems, HPL, Mindspeed Technologies, OK Initiative and Silvaco.

"Adoption of the FSA's PDK Checklist facilitates the clear communication between foundry, fabless customer, EDA vendor, intellectual property (IP) developer and design service providers," said Ken Brock, chair of the FSA's PDK working group and vice president of marketing at Silvaco. "The widespread industry adoption of the checklist has motivated the subcommittee to spin off another working group and partner with the FSA modeling subcommittee to deliver the SPICE Model Checklist."

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

The Simulation Standard, circulation 18,000 Vol. 14, No. 12, December 2004 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.

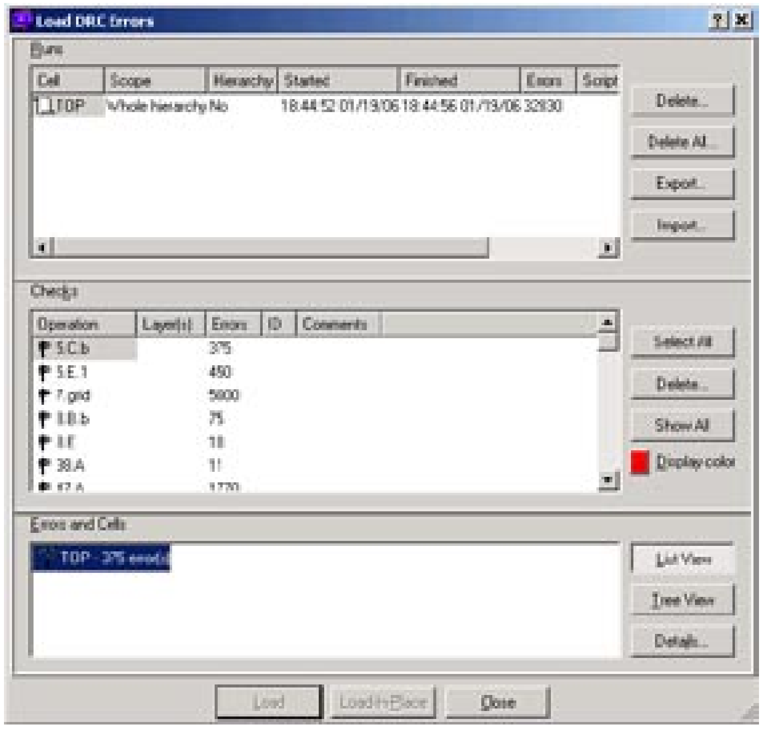
Hints, Tips and Solutions

SILVACO PDK Development Group

Q. How can I view a Foundry DRC run in *Expert*?

A. Before a Foundry will start creating the masks to fabricate a design, they normally will do some sort of drc verification with legacy tools. For the Analog Designer, it is always important to beable to understand the feed-back from the Foundry. Silvaco makes that easier than ever since ***Expert*** can import two other commonly used error database formats: Calibre and DRACULA. Just request the error databases from the foundry, and you can save countelss hours locating the errors that you didn't know you made!

First Open up the TOP level cell, in which the Foundry performed the DRC run. Make sure that the error database has the proper file extension, then select the import tab on the load DRC errors Dialog. You can now load and navigate through the errors, with your fa-miliar and helpful DRC navigator from Silvaco International.



Cell	Scope	Hierarchy	Started	Finished	Errors	Script
TOP	Whole hierarchy	No	18:44:52 01/19/06	18:44:56 01/19/06	32830	

Operation	Layer(s)	Errors	ID	Comments
5.C.b		375		
5.E.1		450		
7.grid		5800		
8.B.b		75		
8.F		18		
38.A		11		
47.A		1770		

TOP - 375 error(s)

Figure 1.

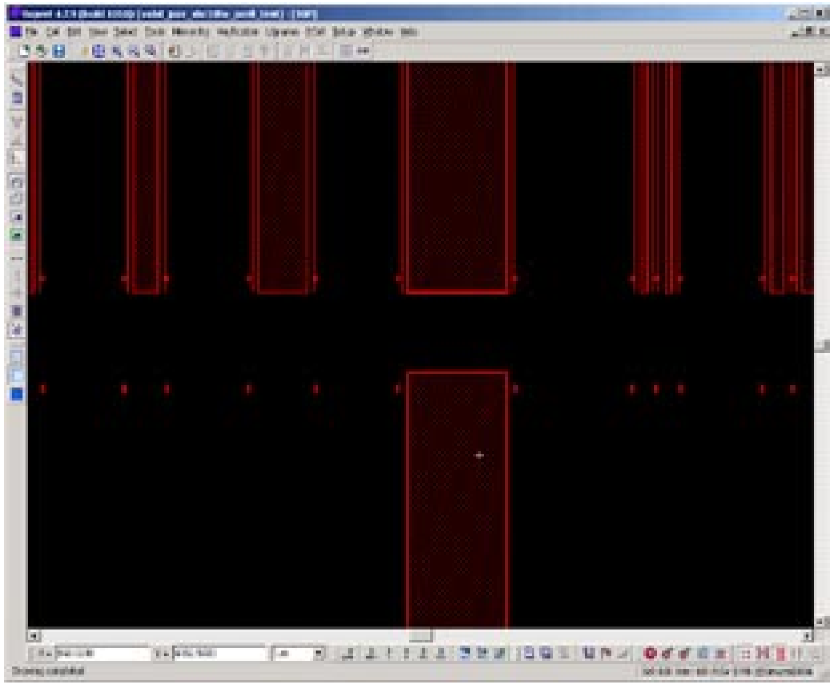
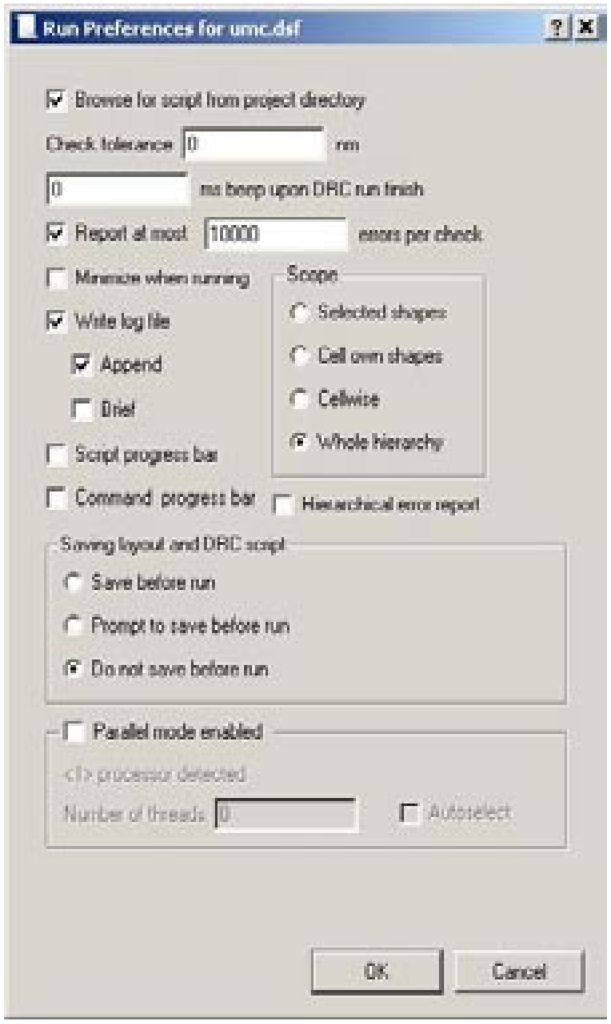


Figure 2.

Q. How can I optimize DRC runtime?

- ***Log file slows run time***: Writing the DRC log file takes a lot of time. The speed of the run time can be drastically improved if the "Write log file" option is disabled. One can also limit the total number of errors to be reported. In the figure below the total number of errors are 10,000. These options can be modified at Setup → Current DRC script run preference.
- ***Running only sections of DRC that you need:*** Executing only sections of DRC code or only on parts of layout also help to speed up run time. The scope of the DRC run in the above figure is set as "Whole Hierarchy".
- ***DRC guard***: DRC guard is used for checking the rules real time. Small DRC decks can be setup while doing routing so that an errors are detected immediately. For example, simple rules like width and spacing of metals can be coded in DRC guard.


Run Preferences for umc.dsf

- [✓] Browse for script from project directory
- Check tolerance 0 nm
- 0 ms beep upon DRC run finish
- [✓] Report at most 10000 errors per check
- [ ] Minimize when running
- [✓] Write log file
  - [✓] Append
  - [ ] Brief
- [ ] Script progress bar
- [ ] Command progress bar
- Scope: Selected shapes / Cell own shapes / Cellwise / [✓ Whole hierarchy]
- [ ] Hierarchical error report
- Saving layout and DRC script: Save before run / Prompt to save before run / [✓ Do not save before run]
- [ ] Parallel mode enabled
  - <1> processor detected
  - Number of threads 0
  - [ ] Autoselect

Figure 1.

- ***#DEFINE /#IFDEF adding***: Implementing the DRC with Conditional Branching Directives like #define, #ifdef #ifndef, #else, #endif, and Block grouping Directives like #EXEC, #SKIP, #stop makes it possible to execute specific parts of the script. Conditional Branching directives enables grouped statements to be executed based on values specified provided they are nested properly.

Example:

```
#define M5 1
#define M6 0
#ifdef M6
... statements ignored here
#else
#ifndef M5
...statements ignored here
#else
...statements executed here
#endif
 ... statements executed here
 #endif
 ... statements executed here
 #ifdef M4
 ... statements ignored here, because M4 is undefined
 #endif
```

- ***Layout styles to improve productivity (partitioning the job)***: Make sure that individual cells (building blocks) are DRC clean, before reusing them. This way the total number of errors can be cut down. Implementing the layout using a proper hierarchical structure enables faster DRC cleaning and editing if needed.
- ***Reducing redundant operations:*** Having repetitive code in the DRC deck slows down the run time. Temporary layers that will be used frequently can be defined as global variables, so that redundancy is eliminated.

Call for Questions

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

Hints, Tips and Solutions Archive

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

Your Investment is Safe

20 Years and Growing

Financially Rock-Solid

Fiercely Independent

We are NOT For Sale

SILVACO
INTERNATIONAL

USA Headquarters:

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

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

sales@silvaco.com
www.silvaco.com

Contacts:

Silvaco Japan
jpsales@silvaco.com

Silvaco Korea
krsales@silvaco.com

Silvaco Taiwan
twsales@silvaco.com

Silvaco Singapore
sgsales@silvaco.com

Silvaco UK
uksales@silvaco.com

Silvaco France
frsales@silvaco.com

Silvaco Germany
desales@silvaco.com

e*ECAD PARTNER

*Products Licensed through Silvaco or e*ECAD*