Simulation Standard

Connecting TCAD To Tapeout

A Journal for CAD/CAE Engineers

Schematic Driven Layout

Introduction

Expert schematic driven layout (SDL) is a CAD tool which increases the productivity of layout design by automating cell generation and providing visual cues to assist the wiring process. In the current implementation, schematic driven layout is used to automatically create IC layouts based on information from a netlist. Cell instances are created from existing cells. P-cells are instanced with parameters from the netlist file. Schematic driven layout automatically transfers net names, instance names and connectivity information from netlist to the layout. Automatic instantiation of cells with built-in connectivity support dramatically speeds up layout designer's productivity and accuracy.

Operation Flow

Expert schematic driven layout operation flow is depicted in Figure 1.

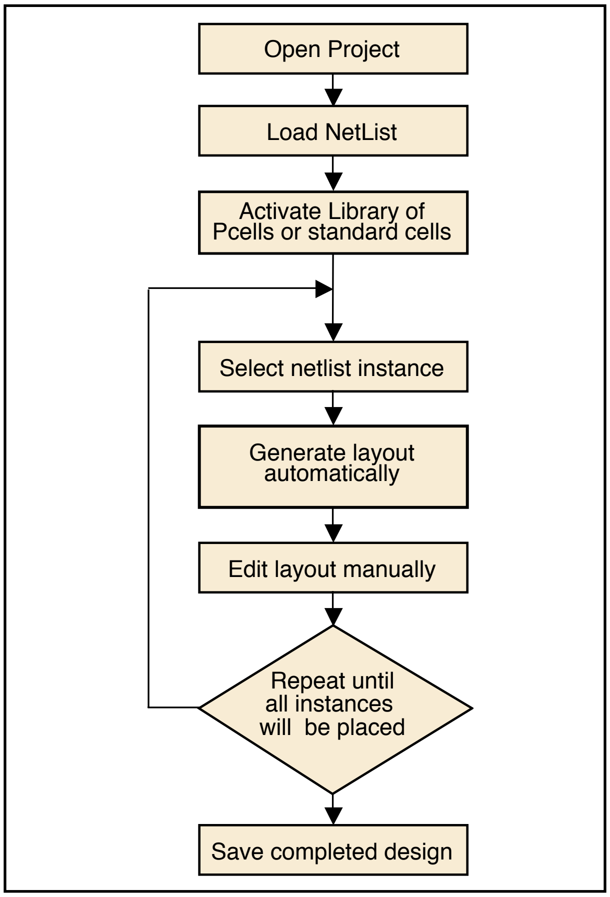

Open Project
Load NetList
Activate Library of Pcells or standard cells
Select netlist instance
Generate layout automatically
Edit layout manually
Repeat until all instances will be placed
Save completed design

Figure 1. ***Expert*** schematic driven layout operation flow

Preparing for Schematic Driven Layout

An existing ELD project file must be opened for the automatically generated cells to be placed. Use the 'Tools>>SDL>>Load Netlist' menu command is used to load the file with the netlist. Circuit information will be displayed in the netlist panel with the circuit hierarchical tree (Figure 2). To run Schematic driven layout a library of predefined standard cells (resistors, capacitors, transistors, diodes, and other finished devices) must be placed in your custom cell. These standard cells can be located either in the current project or in the activated cell library. Cell libraries can be activated by 'Libraries>>Library setup' dialog panel. If the name of a standard cell differs from the name of the corresponding netlist instance, you have to provide a mapping file with correspondences between names. Each line of the mapping file should contain the subcircuit name and the layout cell name separated by a space or tab character:

AND2	and2
MUX4	LIB1::MUX_4

Continued on page 2 ...

INSIDE

Interactive P-cell Generation 4
Guardian DRC vs. Other DRC Systems, I 7
Calendar of Events 10
Hints, Tips, and Solutions 11

SILVACO INTERNATIONAL

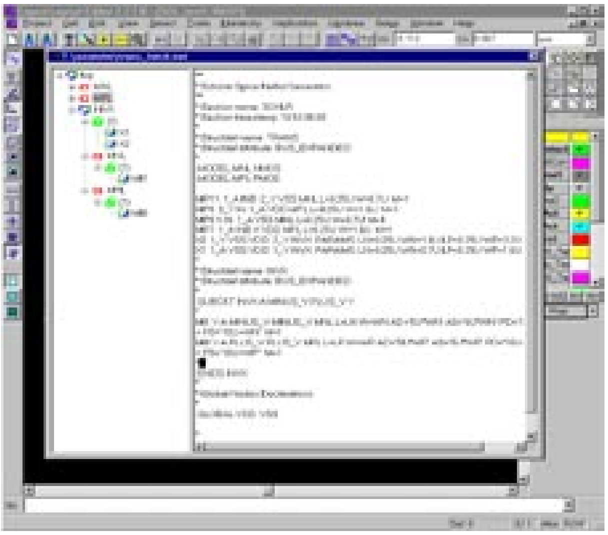


Figure 2. Netlist panel.

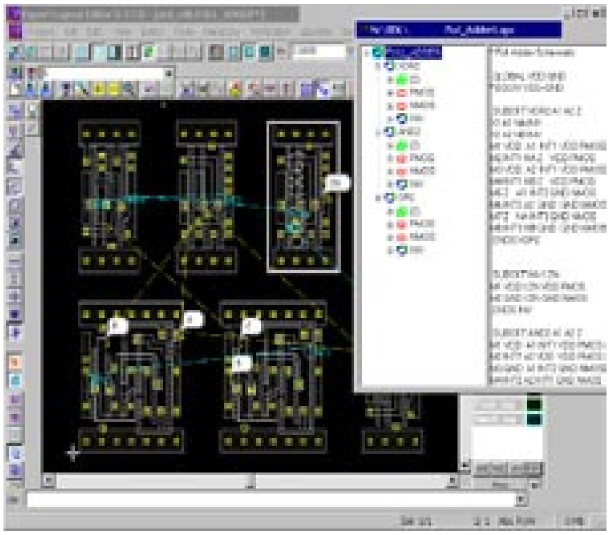


Figure 3. Initial placement with highlighted nets.

Note that the library name for a cell from a library should be included in the cell name ('LIB1::MUX4' means cell MUX4 from library LIB1).

Initial Placement

Go to the circuit hierarchy tree in the netlist panel. Position your cursor at the location of the subcircuit you want to lay out and click the right mouse button. Select 'Create' command in the popup menu. Automatic cell generation will be performed. Cell area is computed from the areas of the individual components plus the area required for wiring. If an individual component has already been laid out, the actual area is used. If no layout exists, the area of a component is estimated from component parameters. The position of the devices will be calculated automatically to fill out the estimated cell area. The cell view window displays the geometry of the created layout (Figure 3).

If the selected subcircuit is hierarchical and the underlying instances not calculated yet, they will be laid out as well.

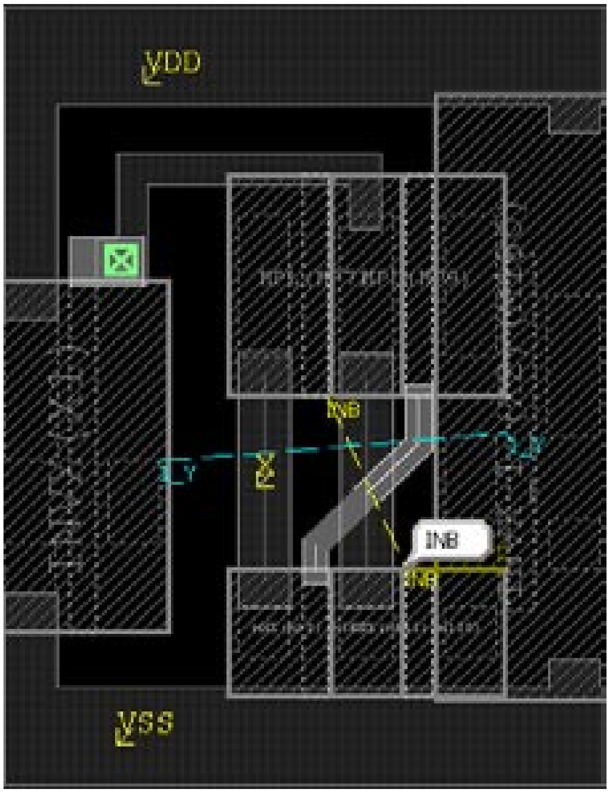


Figure 4. Highlighting the unfinished nets only.

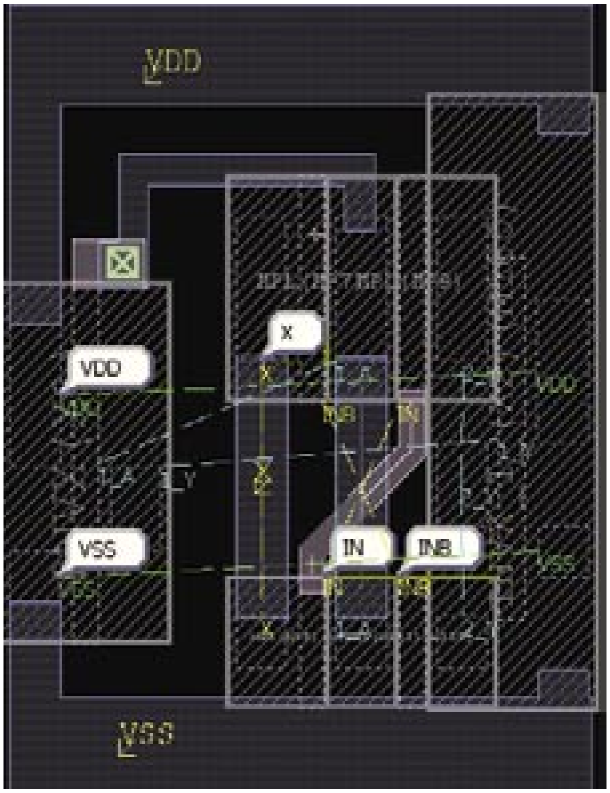
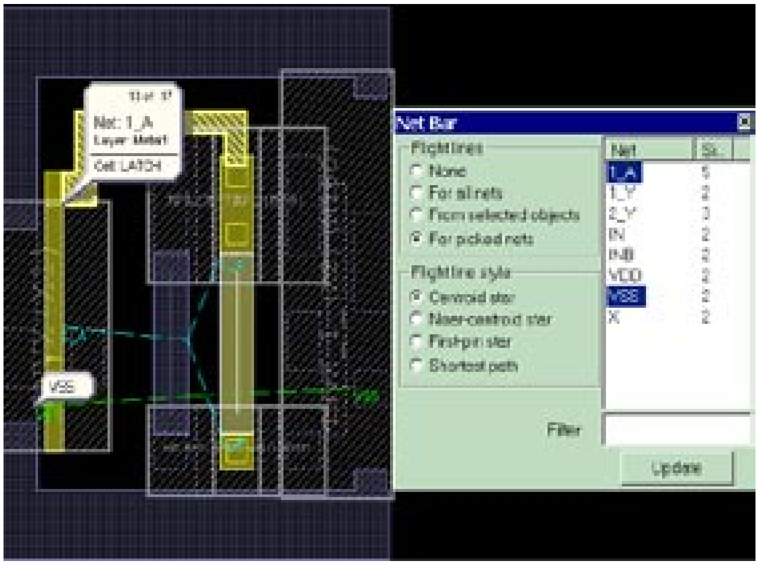


Figure 5. All nets are highlighted.



Net Bar

Flight lines
○ None
○ For all nets
○ From selected objects
● For picked nets

Flight line style
● Centroid star
○ Near-centroid star
○ First-pin star
○ Shortest path

Net	Si..
1_A	5
1_Y	2
2_Y	3
IN	2
INB	2
VDD	2
VSS	2
X	2

Filter

Update

Figure 6. Net Bar panel for managing of connectivity drawing.

Layout Editing

Once the automatic placement is finished editing we can be started. ***Expert*** features a user-friendly interface for editing the generated design layout. Electrical connectivity is highlighted (Figure 3) to simplify design wiring. ***Expert*** shows "flight lines" to indicate which pins belong to the same net and should be connected to each other. The flight lines are redrawn automatically during layout editing.

There are several ways to do wire routing in your design. ***Expert*** 'Wiring' tool may be used to automatically place a wire connection between selected contacts. Pins can be connected manually by a wire object. To perform wire routing user's should choose a wire tool in the ***Expert*** design bar and then simply click at each point at which the connection should be. Other parameters (e.g. layer, thickness, wire joint and end styles) for how the wire should be drawn may also be specified. ***Expert*** multiwire tool is used to create interconnects between pins using wire segments lying in different layers and connected by special contact cells. Any other primitive layout object to make connection between pins may also be used.

Expert submenu 'Tools>>SDL' contains many useful commands and options to simplify cell routing. The following list contains helpful hints for interconnect operations:

- Use 'Tools>>SDL>>Update Connectivity' command and 'Tools>>SDL>>Show unfinished Nets only' option to check nets which are not connected yet (Figure 4).
- 'Tools>>SDL>>Show Global Nets' option is used to hide or show Global nets. Global nets have a lot of connections (Figure 5), therefore turning its visibility off decreases the congestion of flight lines and simplifies design editing.
- Dialog panel 'Tools>>SDL>>Nets Bar' is used to choose flight line style and select nets for which flight lines are to be shown (Figure 6).
- 'Tools>>SDL>>Show Ports' option is to show pin geometry inside a cell instance. This is useful when design is being edited in the 'lazy' instance view mode.
- Check option 'Tools>>SDL>>Show Shorted Nets Only' to show which nets overlap (Figure 7).

Conclusion

We have described the first phase of development of ***Expert*** schematic driven layout.

In the future, ***Expert*** schematic driven layout tool will use circuit description from Silvaco's ***Scholar*** schematic editor. It will provide the initial positions for the generated layouts to resemble the placement of the schematic instances.

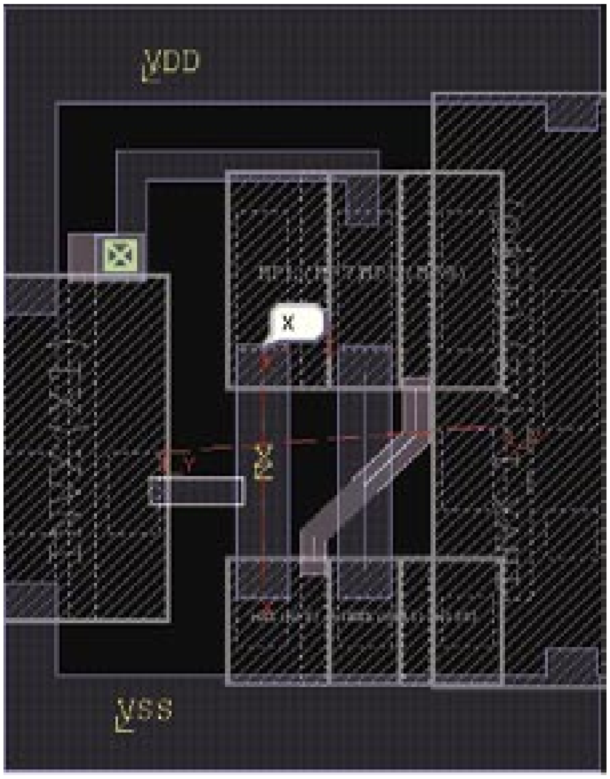


Figure 7. Highlighting of shorted nets only.

Interactive P-cell Generation

1. Abstract

Parameterized cells (P-cells) significantly increase the productivity of layout designers. Silvaco International's tool ***Expert*** has been designed to aid layout designers through its ability to support and manipulate P-cells based on its ***LISA*** script language. This simulation standard will demonstrate the capability of Expert and will identify its inherent benefits to the layout engineer.

In previous versions of ***Expert*** layout editor new P-cells could be created only by direct input of the text of the ***LISA*** procedure that defines a p-cell, into the special P-cell panel (Figure 1). We shall call this approach "textual P-cell generation". To simplify writing of ***LISA*** scripts, it is possible to draw an instance of an intended P-cell in ***Expert*** editor, save it into XI-script format, and then edit the script manually to substitute the actual numeric values by parameters and formulas.

While these approaches are acceptable for small P-cells, it becomes inconvenient with more complex P-cells and when large numbers of P-cells are required, e.g., by developers of cell libraries and PDKs.

Therefore, "interactive parameterized cell generation" is a new feature added into Expert. This feature increases user's productivity of creating P-cells. It generates a parameterized cell with LISA xi-scripts from a geometric cell based on interactive functions "Stretch", "Repetition", and "Layer". After a P-cell is generated, parameter values can be modified to produce the desired cells.

Rules for P-cell generation are applied to cell objects directly or via user defined control blocks. Several rules can be applied to one object /block as well as one rule can be applied to several objects/blocks. Control blocks can be of any shape (box, region, etc.).

The "Stretch" function is used to stretch or move selected objects in one of the following directions: left, right, up, down, left-up, left-down, right-up, right-down. Objects completely inside Control blocks will move, and partially overlapped by blocks will stretch.

The "Repetition" function is used to repeat the selected objects in the specified direction with specified steps.

The "Layer" function is for the assignment of used layers.

An existing library cell can be used as the base for P-cell generation, or a new cell with base geometry can be created according to the specification.

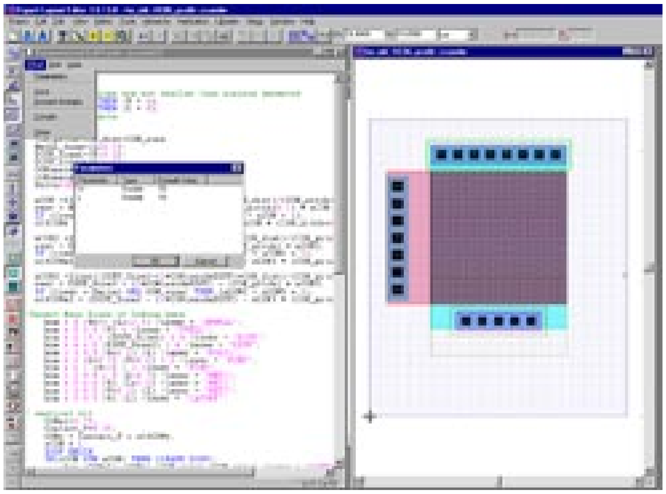


Figure 1. Textual P-cell generation in ***Expert***.

2. Example of P-channel MOSFET P-cell Generation

P-cell generation of P-channel MOSFET (Figure 2 and Figure 3) is illustrated by the following step-by-step example.

Specification:

Contact Size	0.4 um
Repetition step for Contact	0.4 um
Spacing between Contact and Poly	0.2 um
Spacing between Contact and P-Act	0.2 um
Extended of Poly for P-Act	0.4 um
Cell Origin	Center of Poly Gate

2.1. Implement the gate length and multiple gates

The following steps show the implementation of "length" parameter (Gate length of MOSFET) by the "Stretch" function. Generating multiple gates is one of the applications for adding combined stretching and adding repetition functions.

1) Run Tools >> P-cell generate >> Create Control Blocks

2) Run Edit >> Create Objects >>Box to enter the stretch box.

3) Select the created box and run Tools >>P-cell generate >>Set Rules.

4) Click the "Add Stretch" button and enter the following parameters in the dialog:

   Direction : right
   Parameter : length + (gates-1) * (length + 0.4) - 0.4
   where 0.4 is the default gate length in microns.

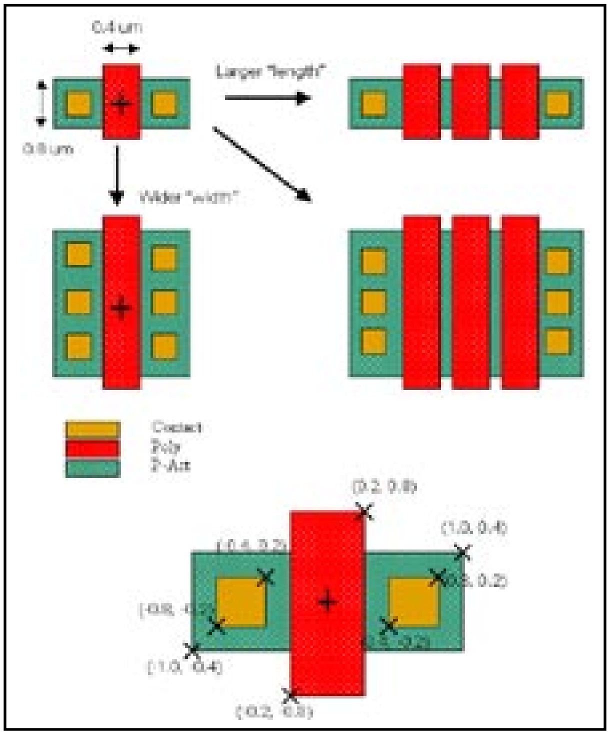


Figure 2. P-cell specification.

5) Select an object in Poly layer and run Tools >> P-cell generate >> Set Rules.

Then click the "Add Repetition" button to enter the following parameters in the dialog:

Direction	:	right
Column	:	gates
Delta	:	length + 0.4

6) Run Tools P-Cell generate >> Create Control Blocks, and create a new stretch box enclosing the right side segment of Poly layer then select the created box, and run Tools >> P-Cell generate >> Set Rules, click the "Add Stretch" to enter the following parameters:

Direction	:	right
Parameter	:	length - 0.4

2.2. Implement the gate width and contact array

We need to add a few steps for implementation of "width" parameter (Gate width of MOSFET) and the corresponding contact array by "Repetition" function.

1) Create one more control block and run Tools >> P-cell generate >> Set Rules. Then click the "Add Stretch" button and set the following parameter in the dialog:

Direction	:	up
Parameter	:	width - 0.8

This "0.8" means that the default gate width.

2) Run Select >> By Click / Drag and select the contacts.

3) Run Tools >> P-cell generate >> Set Rules, click the "Add Repetition", and set the parameters as follows:

Direction	:	up
Row	:	floor (width / 0.8)
Delta	:	0.8

2.3. Implement parameterize the layer for diffusion area

In this step we will use the "Layer" function.

1) Select a diffusion area in the layer "P-Act".

2) Run Tools >> P-cell generate >> Set Rules, and click Add Layer to specify a parameter name "diff_layer":

2.4. Parameters of P-cell

Run Tools >> P-cell generate >>Parameters, and set the parameters as follows:

length	Type = double,	Default value = 0.4
width	Type = double,	Default value = 0.8
diff_layer	Type = string,	Default value = P-Act
gates	Type = integer,	Default value = 1

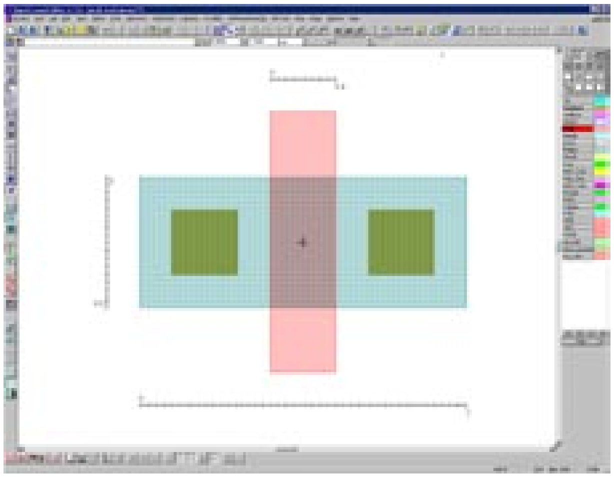


Figure 3. Base geometry.

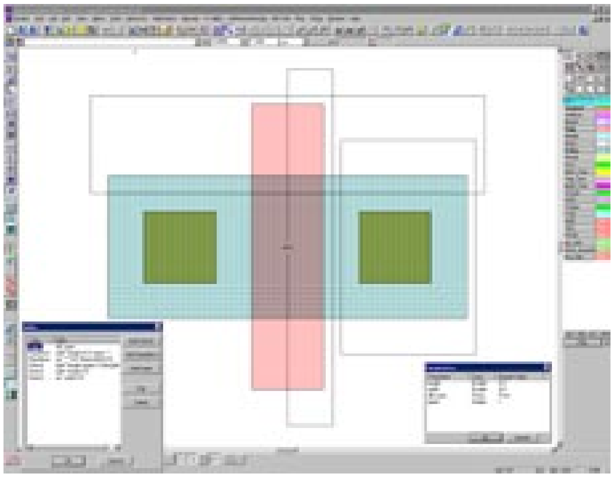


Figure 4. Base geometry with control blocks.

2.5. Generate P-cell

Run Tools >> P-cell generate >> Compile to execute compiling. A new parameterized cell will be created with a suffix "_pcg". The resulting P-cell will be opened with the corresponding script:

```
DEFINE PCELL "pmos_pcg" /REPLACE
    PARAMETER length /TYPE = (Double) /DEFAULT = (0.4)
    PARAMETER width /TYPE = (Double) /DEFAULT = (0.8)
    PARAMETER diff_layer /TYPE = (String) /DEFAULT = ("P-Act")
    PARAMETER gates /TYPE = (Integer) /DEFAULT = (1)
    BODY BEGIN
    $$j0 = -1;
    LOOP BEGIN
         $$j0 = $$j0 + 1;
        IF ($$j0 GEQ (floor(width / 0.8))) THEN (LEAVE LOOP);
        box (-0.8) (-0.2 + $$j0 * (0.8)) (0.4) (0.4) /layer = "Contact";
    END;
    $$j1 = -1;
    LOOP BEGIN
        $$j1 = $$j1 + 1;
        IF ($$j1 GEQ (floor(width / 0.8))) THEN (LEAVE LOOP);
        box (0.4 + (length + (gates - 1) * (length + 0.4) - 0.4))
         (-0.2 + $$j1 * (0.8)) (0.4) (0.4) /layer = "Contact";
    END;
    $$i2 = -1;
    LOOP BEGIN
        $$i2 = $$i2 + 1;
        IF ($$i2 GEQ (gates)) THEN (LEAVE LOOP);
        box (-0.2 + $$i2 * (length + 0.4)) (-0.8)
         (0.4 + (length - 0.4)) (1.6 + (width - 0.8)) /layer = "Poly";
    END;
    box (-1) (-0.4) (2 + (length + (gates - 1) * (length + 0.4) - 0.4))
        (0.8 + (width - 0.8)) /layer = (diff_layer);
END;
```

Now we can create several instances of the resultant P-cell with various parameter values (Figure 5).

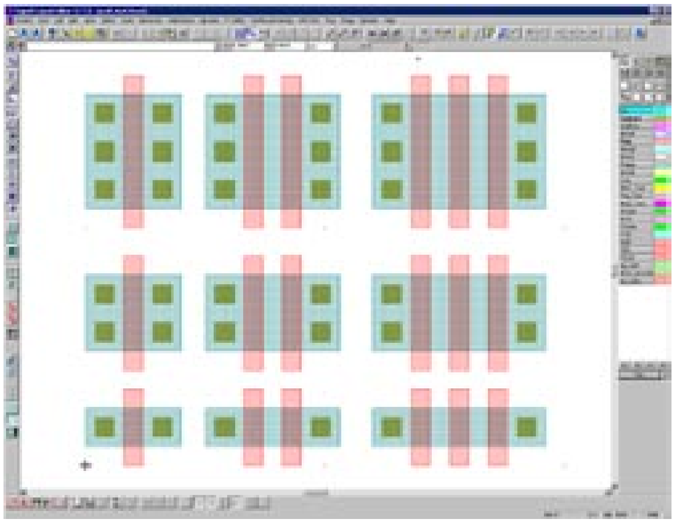


Figure 5. Instances of resultant P-cell.

3. Conclusion

The benefit of interactive P-cell generation is the increase of the user's productivity for P-cell libraries creation. Once P-cells are generated, they can be simply modified to become the required cells. This feature delivers the power of P-cells without spending much time creating and modifying P-cells manually.

Guardian DRC vs. Other DRC Systems, I

1. Introduction

This article begins a series of articles that describe the compatibility of ***Guardian*** DRC system[1] in terms of functionality: command set, syntax and implementation with other leading industrial DRC systems. The purpose of this series is to facilitate the usage of ***Guardian*** DRC system by the designers who are already familiar with some other DRC systems.

Although basic DRC operations performed by various DRC systems correspond to "natural" operations possible with geometric shapes: Boolean operations, selection, resizing, distance checking, the implementations vary both in syntax and in the way of actual execution. This sometimes may lead to wrong expectations about certain functionalities, unless the designers are aware of these differences..

Guardian DRC system includes automatic DRC rule translators for several popular DRC formats. Nevertheless the knowledge of differences in commands may be useful not only when writing new DRC rules from scratch. For example, it may prove useful for optimization of rule sets, since automatic translation sometimes does not deliver the best possible implementation of certain checks, especially when converting from systems with limited command set. For example, Dracula(TM) DRC system has very limited capabilities to operate with separate edges from shape boundaries. Also, often it is necessary to put additional checks into the DRC deck, to comply with local technology variations and/or limitations of design methodologies..

The goal of ***Guardian*** DRC is to provide all the functionality required for performing modern DRC operations. When compared to other DRC systems, ***Guardian*** DRC either provides direct counterparts of commands or converts them into groups of several commands. In some cases this way opens possibilities for an additional optimization of execution time.

2. Compatibility with Calibre™ DRC System

This section is a general overview of the correspondence of Calibre and ***Guardian*** DRC commands. Detailed comparisons of particular commands will be provided in subsequent articles of the series.

2.1. Overall Execution and Syntax

The most important differences in DRC rule set execution are the following.

- Calibre DRC syntax allows layer generation statements to be written later in the DRC rule file, after their use in other commands. DRC rule compiler automatically defines proper order of execution of DRC commands. In ***Guardian*** DRC this possibility can be turned on by the setup directive "free_layer_definition_order".
- Calibre merges original layers on input to the DRC system. In ***Guardian*** DRC this possibility can be turned on by the setup directive "merge_input".
- The syntax of ***Guardian*** DRC is based on pairs "parameter-value", and commands have an explicit terminator, a semicolon. This provides an additional flexibility in writing down the rule set, at the price of brevity.
- Newer versions of ***Guardian*** DRC have a simplified syntax for writing the constraints for numerical values in checks and selection operations.
- Usually ***Guardian*** DRC runs integrated into ***Expert*** Layout Editor [2]. By default, ***Guardian*** returns all layers, generated during DRC execution, unless their names start with the ampersand (&). This behavior may be flexibly controlled by the setup directive "Update_Layout".
- DRC error reporting in Guardian DRC differs from that of Calibre. This will be covered in a separate article.
- Both Calibre and ***Guardian*** use layer names in DRC rule sets, that are independent of the names in a layout system. Calibre can define its layers as combinations of original layers. In ***Guardian*** DRC, to achieve the same functionality, the rule set must explicitly contain "OR" or "COMBINE" commands.

2.2. Particular Types of Commands

The commands are classified here in the following main groups: setup directives, connectivity/antenna checks, metric checks and all other operations. Although the Metric Checks group constitute only four commands, the total number of different actual operations performed by the set commands is comparable to the number of all commands, since metric checks are controlled by a large number of parameters.

2.2.1. Metric DRC Checks (See Table 1)

The "**Rectangle enclosure**" command of Calibre is in fact a more efficient implementation of enclosure checks for rectangular shapes. Currently ***Guardian*** DRC performs these operations by general means.

The way of actually performing measurements between line segments in ***Guardian*** DRC is described in detail in [3]. The described metrics correspond to those used in Calibre DRC system.

An important difference arises from the fact that ***Guardian*** DRC does not perform automatic merging of input layers. In certain cases this can lead to time optimization, if it is known that certain layers are created from non-overlapping shapes, e.g., contact or via layers. Therefore to ensure consistency of metric check, one must be sure that the input layers are merged. Notice that many DRC operations with layers provide merged output.

CALIBRE	Guardian DRC	Note
External	OutDistance, Distance	+
Internal (two-layer)	OvDistance	+
Internal (single-layer)	Width	+
Enclosure	InDistance	+
Enclosure…outside also	CompDistance	
Rectangle Enclosure		-

Table 1. Metric DRC Checks

2.2.2. Parameters of Metric Checks (See Table 2)

Metric checks in Calibre and ***Guardian*** DRC differ in the way of the output of the results. This difference will be covered in detail in one of the subsequent articles.

Another difference is in the reporting error segments. The way Calibre does it corresponds to ***Guardian's*** option "Report error subsegments". This option may be set either for each DRC command individually, or globally, by a setup directive.

CALIBRE	Guardian DRC	Note
'[' <layer> ']' , '(' <layer> ')'	ElayerR, CelayerR	[+](-)
region	Option=(RP)	=
region extents		-
region centerline		-
abut [range]	abut	+
singular	sing	+
overlap	cross	+
square	metric=square	=
opposite	metric=opposite	+
opposite symmetric	metric=symmetric	+-
opposite extended <val>	metric=extended	+
measure coincident	T	=
measure all	(~)VI	-
notch | space	N' | no notch	+
connected | not connected	C'| C	=
paral also | only | not paral	-|P|P'	=
perp also | only | not perp	perp	+
acute also | only | not acute	acute	+
obtuse also| only | not obtuse	obtuse	+
projecting [range] | not projecting	O[range], O'	+[-]
angled[range]	slanted	+
corner to corner[range]	C2C	+
corner to edge	C2E	+
[not] corner	C2CE, C2CE-	+
intersecting only	NoMeasure	+
inside also		-
outside also	(~)CompDistance	~+
exclude false notch	irlevant	

Table 2. Parameters of Metric DRC Check Commands.

Legend:
= already existed in Guardian DRC
- not implemented yet
+ added
~ difference in implementation exist

2.2.2. Connectivity/Antenna Checks (See Table 3)

Antenna checks of Calibre are converted into pairs of operations: obtaining of layer parameters and the subsequent checking of them. This way of implementation allows for optimization of execution time when the same parameters for certain layers are used in several antenna checks.

CALIBRE	Guardian DRC	Note
text layer	Attach_Labels	+
connect	Connect	=+
sconnect	Connect …(S)	=+
disconnect	Connect_order:;	+
attach	Attach_Labels	=
label order	Label_Order	+
net, not net	Select…Label (C)	=
net area	Get_Node_Params, Check_Node_Params	+
net area ratio	Get_Node_Params, Check_Node_Params	+
net area ratio print	Check_Node_Params (LogR)	+
polynet	Connect (one-layer)	+
stamp	Select…stamp	=
ornet	Combine	+

Table 3. Connectivity/antenna DRC Checks

Legend:
= already existed in Guardian DRC
+ added

2.2.3. Other DRC Operations (See Table 4)

The notion of "check" of Calibre rule files is converted into preprocessor "block" notion of ***Guardian*** DRC script file.

CALIBRE	Guardian DRC	Note
@	comment	+
include	#include	+
#define/#undefine	#define/#undefine	+
#ifdef/#ifndef	#Ifdef/#ifndef	+
#else	#Else	+
#endif	#Endif	+
layer		~+
group		~+
flag acute, offgrid, skew, nonsimple (path)	input data checks	-+
snap offgrid	Snap_Input ~	+
and	And	+
or	Or	=
xor	Xor	=
not	Dif	=
inside	Select…Inside	=
outside	Select…Outside	=
cut	Select…Cut	=
enclose	Select…Enclose	=
touch	Select…Touch	=
interact	Select…Overlap	=
area	Parea	=
perimeter	Perimeter	+
vertex	Select…Vertex	=

rectangle	Contours	+
donut	Contours…holes	+
enclose rectangle	Contours	-
holes	Contours…opt=holes	+
size	Size…cut	+
size.. truncate	Size…cut	+
size..overlap only		-
size..overunder	underSize+oversize	+
size..underover	overSize+undersize	+
size..inside of	Size…within	+
size..outside of	Size…without	+
size..step	Size…step	+
grow	Size...T,B,L,R	=
shrink	Size...T,B,L,R	=
expand edge	Size_edge	+
exp edge.. corner fill	Opt=(Fill)	-
density	Density	+
extent [<layer>]	Substrate	=[+]
extents	boxes	+
extent cell		-
rectangles		-
expand text	size…list=()	+
[not] inside cell		-
inside edge	select_edges…Inside	+
outside edge	select_edges…Outside	+
coincident edge	select_edges…Coincide	+
coincident inside edge	select_edges…Coincide	+
coincident outside edge	select_edges…Coincide	+
touch edge	select_edges…Touch	+
touch inside edge	select_edges…Touch	+
touch outside edge	select_edges…Touch	+
with edge	Select..Touch	+
with text	Select…Label	+
convex edge	Select_edges…corners	+-
angle	Slope	+
length	Length	=
path length	Path_length	+
drawn acute, offgrid, skew	Check_*_Input: on;	+
offgrid	Offgrid	+
COPY		
copy	Copy	=
flatten	Copy	+
merge	Copy	+
push	Copy	+
shift, magnify, rotate	Copy	--
snap	copy	+

Table 4: Other DRC Operations

Legend:
= already existed in Guardian DRC
- not implemented yet
+ added
~ difference in implementation exist

2.2.4. Setup Commands (See Table 5)

Guardian DRC provides the setup for execution of the DRC script in three ways: through user interface, through command line parameters of the batch mode of execution and by means of setup directives of the DRC script itself.

CALIBRE	**Guardian DRC**	**Note**
Precision	Database_Precision	+
Resolution	Grid_Resolution	+
layer resolution	Grid_Resolution	+
title	Title	+
unit length	Units	=
variable	Variable	+
polygon		
text		
layer map	Layers	+
layout primary	Top_cell	=
layout window	Window	+-
layout windel	Exclude_window	+-
layout window clip		-
exclude cell	Exclude_cells	=
virtual connect name		
virtual connect colon		
virtual connect box name		
virtual connect box colon		
virtual connect report		
layout text		
layout depth [primary|all]	Setup: primitives | whole	+
layout allow duplicate cells		-
expand cell		
expand cell text		
flatten cell		hier
flatten inside cell		hier
push cell		hier
hcell		hier
layout base cell		hier
layout rename cell		hier
layout process box record		UI
layout magnify		
layer directory		UI
layout error on input		
layout merge on input		
layout property text		
layout property placement name		
text print maximum		
layout rename text		
drc tolerance factor	tolerance	+
drc exclude false notch		hier
drc maximum results		UI
drc check text		
drc summary report		
drc results database	output_to_edb/cell	
drc maximum vertex		
drc keep empty		
drc select check	Exec	+
drc unselect check	skip	+
drc check map		
drc print area	Iarea	--
drc print perimeter		
drc map text		hier
drc incremental connect		+
drc cell text		hier
drc cell name		hier
drc magnify results		
drc results database precision	Output_Precision	--
drc maximum cell name length		

Table 5. DRC Setup Commands

Legend:
= already existed in ***Guardian*** DRC
- not implemented yet
+ added
~ difference in implementation exist

hier marks commands related to hierarchical execution, not supported in current implementation of ***Guardian*** DRC

UI marks commands implemented only by means of user interface of ***Guardian*** DRC

3. References

[1] *Guardian User's Manual*, Silvaco International.

[2] *Expert and ExpertViews User's manual*, Silvaco International.

[3] Measurements in Spacing Checks of Guardian DRC, S*imulation Standard,* vol. 13, no 9.

Calendar of Events

December

1

2

3

4

5 ISDRS - Washington DC

6 ISDRS - Washington DC

7 ISDRS - Washington DC
IEDM - Washington DC

8 IEDM - Washington DC

9 IEDM - Washington DC

10 IEDM - Washington DC

11

12

13

14

15

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

17

18

19

20

21

22

23

24

25

26

27

28

29 EDS Fair - Yokohama, Japan

30 EDS Fair - Yokohama, Japan

31

Bulletin Board

Silvaco at IEDM

Silvaco exhibited and participated in Washington, DC at the 2003 IEEE International Electron Devices Meeting (IEDM), the leading forum for the presentation of research and development in the area of electron devices and their applications. As a leader in TCAD, Silvaco was participating in the discussion on strained silicon, silicon-on-insulator, high-k and multi-gate devices.

Silvaco Joins Compact Modeling Council

The Compact Model Council exists to promote the international, non- exclusive standardization of compact model formulations and the model interfaces. They visualize standardized compact models for all major technologies so that customer communication and efficiency can be improved with standard interfaces so that models can be tested faster and implemented easier. Silvaco, with a market leadership position in both SPICE simulation and SPICE modeling software, joins other leaders in the semiconductor, EDA, and foundry industries to work toward these goals.

*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. 13, No. 12, December 2003 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

Galina Makovsky, Applications and Support Engineer

Q: Sometimes I have problem to skip cells when importing cells with the same names from reference library. Once we skip one cell, expert will complain other cells already exist in the new library.

A: To process name collisions for several cells when you try to skip some of them:

1. click 'All Existing' button in Cell Name Collision dialog. The name of all existing cells become highlighted in grey.
2. Hold 'Ctrl' and mouse click on existing(grey) cell name(s) for the cells you want to stay without changes. Cell name background becomes white.
3. Press "Replace". New Cell Name disappears from the 'existing' cell lines that will be replaced and stays for the cells that won't be replaced.
4. For cell that you want to skip from importing, click on a Copied cell Old name($$ImportLib$$ line), then press "Skip". If you have several cells in the list, you can hold 'Ctrl' key while clicking on cell names, then press "Skip".

Now for all Existing cells to be replaced New Cell Name field is empty and a replacement cell has New Cell Name in $$ImportLib$$ line. For all Existing cells NOT to be replaced, New Cell Name field has a cell name and New Cell Name in $$ImportLib$$ line is empty.

If you have only one New Cell Name for each cell pair, press OK. If for some cell both New Cell Names are empty, or the same cell name repeated twice in New Cell Name field, press "Retry" and repeat steps 1 - 4.

Q: I have set up device and connection layers and then tried to run Hipex. But I got an error message saying that there are no executable DRC commands in DRC scripts. What kind of scripts I need to prepare?

A: Please check an option 'Rebuild Derived Layers' in Hipex>>Options>>Layout. It should be OFF if you don't need to generate any derived layers. If it is ON, Hipex looks for derived layers in technology or for DRC script for layer generation, depending on setting in Hipex>>Options>>Netlist_Technology_Definition>>Derived_Layers ('Use Current Technology' or 'Use DRC Script File').

Q: I'm having some difficulty setting the instance grid. I know how to change MinGridStep for instanceGridParams in technology file, but is there a way to change it interactively inside Expert?

A: There are several kinds of grids in EXPERT. Their sizes and visibility are defined in Grid Setup, which is invoked from three different places, for different purposes.

1. Default grid settings are set from Setup>> Technology>> Grid. These grid settings are applied to all editing operations, if they are not overriden.
2. Individual layer grid settings are set from the Layer Setup panel (Setup>>Technology>>Layer/Wire Setup menu command). They are applied to that particular layer.
3. Cell instance placement grid is set as a parameter of the cell instancing dialog and is applied to cell instance/array placement. In Create Instance dialog set Individual_Grid ON and press Set_up_Grid. This Grid Setup dialog is for instances grid settings. If Individual_Grid is OFF, these settings are disabled and grid settings from Setup>>Technology>>Grid Setup are used for instances.

Q: I get the warning "Polygonal font file 'C:\...' is not found". Do you have a polgfont.gds file that I should be referencing?

A: We always include <polgfont.gds> in each ***Expert*** version...\Silvaco\lib\Expert\3.6.12.R\x86-nt\polgfont.gds. It looks like your Silvaco installation is not in C:\Silvaco, so in Setup>>Editor/Viewer>>Editing>>Polygonal_font_file browse for polygonal font file from ...\Silvaco\ .You can use the provided <polgfont.gds> file or create your own with another name.

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

Join the Winning Team!

- PROCESS AND DEVICE APPLICATION ENGINEERS
- SPICE APPLICATIONS ENGINEERES
- CAD APPLICATIONS ENGINEERES
- SOFTWARE DEVELOPERS

EMAIL TO: CAREERS@SILVACO.COM

SILVACO
INTERNATIONAL

USA Headquarters:

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

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

sales@silvaco.com
www.silvaco.com

Contacts:

Silvaco Japan
jpsales@silvaco.com

Silvaco Korea
krsales@silvaco.com

Silvaco Taiwan
twsales@silvaco.com

Silvaco Singapore
sgsales@silvaco.com

Silvaco UK
uksales@silvaco.com

Silvaco France
frsales@silvaco.com

Silvaco Germany
desales@silvaco.com

*Products Licensed through Silvaco or e*ECAD*