Simulation Standard

Connecting TCAD To Tapeout

A Journal for CAD/CAE Engineers

Expert Parametric Wires

Introduction

Expert Parametric Wire (Pwire) is a complex group of objects containing, a single wire named master wire, any number of subparts such as enclosure wires, offset wires and sets of rectangles. Pwire objects enable extremely quick and efficient creation and editing of guard rings and shielded paths which are increasingly important due to higher integration density of IC designs.

The number of subparts for each type, its properties and placement with respect to the master wire is defined by a set of pwire parameters. Pwire parameters for the master wire and each subpart are defined in the 'Parametric Wire' Numeric Input Form. Once the parametric values are entered you can create a parametric wire in the ordinary way by point-and-click mouse operation to enter points in the current cell window. When you create a Pwire group the last one is treated as a single object in many edit operations such as modify, stretch, copy, paste, move, duplicate, etc.

You can store a particular set of Pwire parameters into the ***Expert*** technology (internal binary *.eld or *.sld project file or external ASCII *.tcn file) as a template. The Pwire template is a predefined set of pwire parameters that define a particular Pwire, for example, a guard ring. The Pwire template can be chosen in 'Parametric Wire' Numeric Input Form (see Figure 1). Once you select the Pwire template each field in the Numeric Input form will be filled with the matching parameter value stored in the template. You can make changes for some parameters values before entering points for Pwire in a cell window.

Creating Pwires

You can create a Pwire object using the ***Expert*** menu command Edit>>Create Object>>Parametric Wire or Xi command PWIRE.

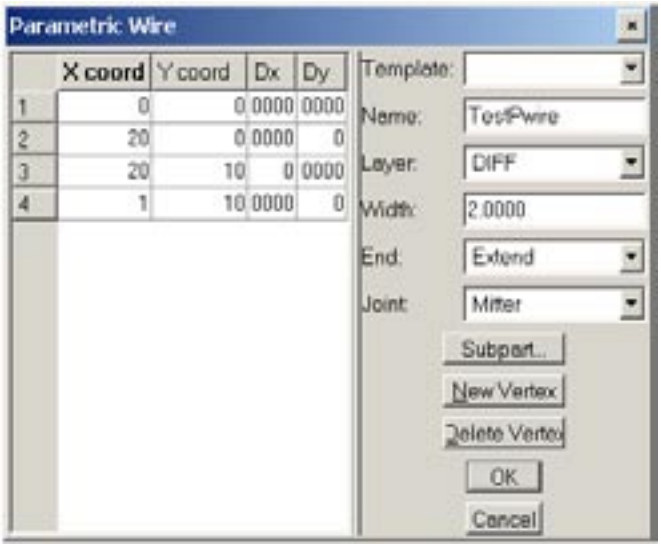

Figure 1. Pwire Creation Numeric Input Panel

Edit>>Create Object>>Parametric Wire Command

Once you select the command, the numeric input dialog similar to the ordinary wire numeric input panel pops up (see Figure 1). But there are a couple of additional fields and buttons.

Continued on page 2 ...

INSIDE

Expert's Netlist Driven Layout 6

Calendar of Events 10

Hints, Tips, and Solutions 11

SILVACO INTERNATIONAL

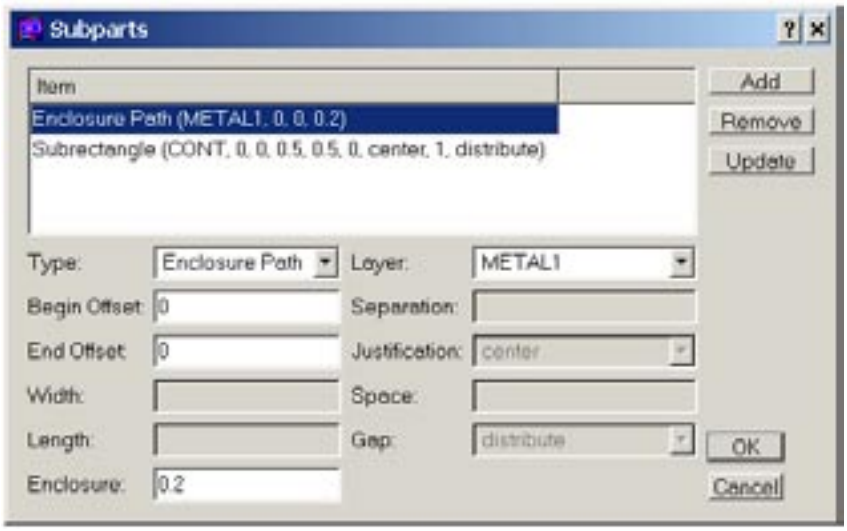

Figure 2. Subparts Dialog Panel

Template allows selecting template stored in the technology with predefined pwire parameters.

Name is to assign a unique name for the created pwire object. If the name is not specified then a system defined unique name will be assigned.

Subparts… button opens 'Subparts' dialog panel to allow you to add, remove, or update subparts of Pwire object (see Figure 2).

Input fields Layer, Width, End, Joint defines attributes of the master wire. Vertices for master wire can be added by a mouse click or using New Vertex button.

If you are using an existing template from the 'Template' drop down list then parameter values for master path and all subparts will be filled in automatically. You can make any changes of predefined parameters or create a new Pwire from scratch, without using a template.

Subparts dialog panel allows add, remove, and modify subparts for the current Pwire.

Pwire object can include any number of subparts such as enclosure wires, offset wires and sets of rectangles. Pwire subparts list is shown on the top part of the Subparts dialog panel. When you select existing subpart then values of its parameters will be automatically filled in the corresponding fields. You can edit the values of parameters and click 'Update' button to apply changes to the selected subpart. 'Delete' button is to delete the highlighted subpart.

To add a new subpart first of all you have to select the subpart type: Enclosure Path, Offset Path, or Subrectangle. Then set up for a specific selected subpart type parameter and click Add button. A new subpart will be added to the current Pwire and the subpart list will be updated. Click Ok button to apply changes to the subparts, and Cancel button to discard any changes.

You can create the following types of Pwire subparts.

Enclosure path is a wire with vertices coincident with vertices of the master wire and wire width calculated by formula

```
Width = MasterWireWidth – (2 * Enclosure),
```

Where the Enclosure is defined in 'Enclosure' field from 'Subparts' dialog panel.

You have to choose the layer (drop down list 'Layer') for the Enclosure path. In addition you can specify how

Separation	Center	Left	Right
0	Vertices coincide	Left edge master wire coincide with right edge of offset wire	Right edge master wire coincide with left edge of offset wire
>0	Shift left (separation value is distance between centerlines))	Shift left (separation value is distance between left edge of master wire and right edge of offset wire)	Shift right (separation value is distance between right edge of master wire and left edge of offset wire)
<0	Shift right (separation value is distance between centerlines)	Shift right (separation value is distance between left edge of master wire and right edge of offset wire)	Shift left (separation value is distance between right edge of master wire and left edge of offset wire)

Table 1. Offset path separation

Separation	Center	Left	Right
0	Box center on master wire centerline	Left edge master wire coincide with right edge of box	Right edge master wire coincide with left edge of box
>0	Shift left (separation value is distance between box center and master wire centerline)	Shift left (separation value is distance between left edge of master wire and right edge box)	Shift right (separation value is distance between right edge of master wire and left edge of box)
<0	Shift right (separation value is distance between box center and master wire centerline)	Shift right (separation value is distance between left edge of master wire and right edge of box)	Shift left (separation value is distance between right edge of master wire and left edge of box)

Table 2. Subrectangle separation

the starting and ending points of the Enclosure path are located with respect to the starting and ending points of the master wire. Use 'Begin Offset' and 'End Offset' fields from 'Subparts' dialog to change beginning and ending points of the enclosure path. Starting and ending edges of the enclosure path are extended by specified values if the offset values are positive, and truncated if negative.

Offset path is a wire with edges parallel to the master wire edges and a shift defined by 'Separation' and 'Justification' parameters (see Table 1).

The width of the offset wire is defined by 'Width' parameter. Use drop down list 'Layer' from 'Subparts' dialog to set the layer for the offset wire. In addition you can specify 'Begin Offset' and 'End Offset' parameters to extend or truncate the beginning and ending wire segments correspondingly (see description above).

Subrectangle is the set of boxes lying on the specified layer. The size of the box is defined by 'Width' and 'Length' parameters, where box width is measured across the master wire and the box length along the master wire centerline. The distance between boxes are defined by 'Space' and 'Gap' parameters. Space defines minimal distance between boxes. Gap can be one of two types: Minimum or Distribute. Minimum means that distance between boxes is fixed and equal to space parameter value. The remaining space after the last placed box if there is one is left empty. If you chose Distribute Gap then unused space is evenly distributed between placed boxes, and actual space between boxes is in the range:

```
space value < actual space
< space value + box length.
```

Parameters 'Separation' and 'Justification' is used to place boxes with offset from master wire (see Table 2).

Xi command PWIRE

You can use Xi command PWIRE to create complex object Pwire. Xi command PWIRE has the following format

```
PWIRE {vertices} [/name = pwire_name]
[/template = pwire_template_name]
[/layer = layrname] [/width = value]
[/extend | /flushend | /roundend] [/
extjoint | /filljoint | /roundjoint |
/miterjoint]
[/centeroffset | /leftoffset | /
rightoffset]
[/inwires = {inwires}] [/offwires =
{offwires}] [/boxes = {boxes}]
```

Here {vertices}, /layer, /width, and wire end and joint style arguments are similar to arguments for Xi command 'WIRE' and specify vertices, layer and other wire attributes for master wire of Pwire object. Additional arguments are to define the specific pwire parameters.

Named argument /name specifies a unique name for the pwire object.

Named argument /template allows you to use pwire templates stored in the technology.

Named arguments /inwires, /offwires, /boxes describe pwire subparts (see Xi command PWIRE_TEMPLATE for detailed description of arguments).

Suppose we have Pwire template "GuardRing_2" stored in the technology. Than the following Xi command will create a new Pwire object in the current cell.

```
pwire {-10,-10, -10,20, 20,20, 20,-10, -8,-10} /template = "GuardRing_2";
```

Storing Pwire templates in the *Expert* Technology

A set of predefined Pwire parameters can be stored in the technology with a unique name and used for quick creation of Pwire objects. Use Setup>>Technology>>Pwire Templates command to add, remove, or modify Pwire templates. Xi command PWIRE_TEMPLATE is also for adding pwire templates to your technology. The command has format

```
PWIRE_TEMPLATE pwire_template_name
[/layer = layername] [/width = value]
[/extend | /flushend | /roundend] [/extjoint | /filljoint | /roundjoint | /miterjoint]
[/centeroffset | /leftoffset | /rightoffset]
[/inwires = {inwires}] [/offwires = {offwires}] [/boxes = {boxes}],
```

where named arguments /inwires, /offwires, and /boxes describe enclosure paths, offset paths, and subrectange subpart of pwire object. There are set of predefined xi attributes used for definition of pwire subparts, such as:

PWA_LAYER	defines subpart layer;
PWA_WIDTH_DELTA	defines Enclosure path width;
PWA_WIDTH	defines Offset path or Subrectangle witdh;
PWA_LENGTH	defines Subrectangle length;
PWA_OFFSET	defines offset path separation from master wire;
PWA_OFFSET_TYPE	defines offset separation type;
PWA_MIN_GAP	defines space between boxes for subrectangle subpart;
PWA_BOX_DISTR	defines box distribution for subrectangle subpart;

Each subpart is described as a sequence of attributes with values of subparts parameters. For example, the following Xi command adds Pwire template "GuardRing_2" with two enclosure paths and two subrectangles.

```
wire_template "GuardRing_2" /layer = "METAL1" /width = 2 /offset = 1
/inwires = ({{attr_create(PWA_LAYER,"METAL2"),
       attr_create(PWA_WIDTH_DELTA, 0.2)},
       {attr_create(PWA_LAYER, "NWELL"),
       attr_create(PWA_WIDTH_DELTA, 0.5)}})
/boxes = ({{attr_create(PWA_LAYER, "CNTG"),
       attr_create(PWA_WIDTH, 1),
       attr_create(PWA_LENGTH, 1),
       attr_create(PWA_OFFSET, 0),
       attr_create(PWA_OFFSET_TYPE, POT_CENTER),
       attr_create(PWA_MIN_GAP, 1),
       attr_create(PWA_BOX_DISTR, BDT_EVEN)},
       {attr_create(PWA_LAYER, "CONT"),
           attr_create(PWA_WIDTH, 0.5),
       attr_create(PWA_LENGTH, 0.5),
       attr_create(PWA_OFFSET, 0),
       attr_create(PWA_OFFSET_TYPE, CENTER),
       attr_create(PWA_MIN_GAP, 2),
       attr_create(PWA_BOX_DISTR, BDT_EVEN)}});
```

Pwire template can be stored in and loaded from the ***Expert*** external ASCII *.tcn technology file (see example below).

```
PwireTemplate
  {
  PwireTemplateName = "GuardRing_1"
  MasterWireLayer = "METAL1"
  MasterWireWidth = 2
  MasterWireJoint = EXTEND
  MasterWireEnd = EXTEND
  MasterWireOffset = 0
  MasterWireOffsetType = CENTER
  PwInWire
    {
    PwLayer = "METAL2"
    PwBeginOffset = 0
    PwEndOffset = 0
    PwWidthDelta = 0.2
    }
  PwInWire
    {
    PwLayer = "NWELL"
```

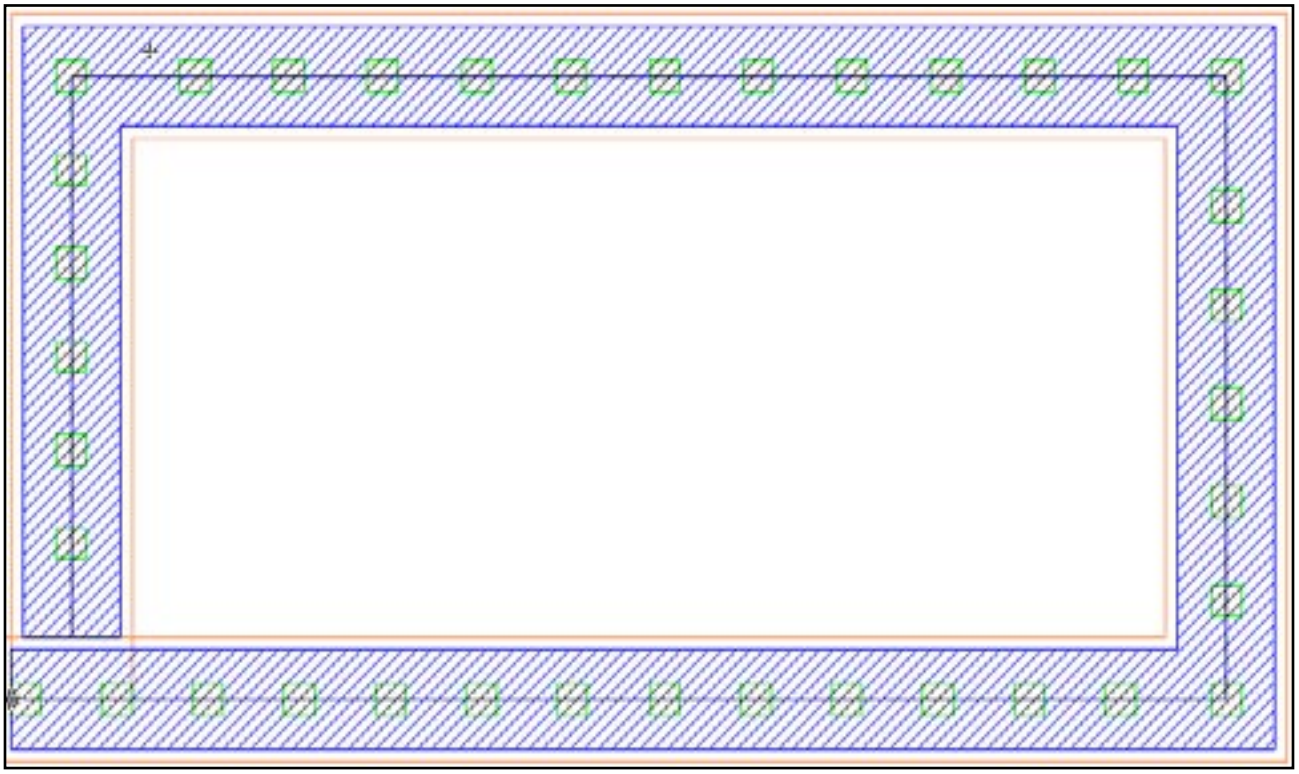

Figure 3. Example of Pwire.

```
    PwBeginOffset = 0
    PwEndOffset = 0
    PwWidthDelta = 0.5
    }
  PwBox
    {
    PwLayer = "CNTG"
    PwBeginOffset = 0
    PwEndOffset = 0
    PwWidth = 1
    PwLength = 1
    PwOffsetType = CENTER
    PwDistrType = EVEN
    PwOffset = 0
    PwMinSpace = 1
    }
  PwBox
    {
    PwLayer = "CONT"
    PwBeginOffset = 0
    PwEndOffset = 0
    PwWidth = 0.5
    PwLength = 0.5
    PwOffsetType = CENTER
    PwDistrType = EVEN
    PwOffset = 0
    PwMinSpace = 2
    }
}
```

Example of Creating a Pwire

The Pwire with parameters shown on Figure 1 and Figure 2 looks like this in Figure 3.

Conclusion

Expert Pwire engine is a convenient and powerful feature of the layout editor. It speeds up and improves the quality of IC layout designs.

Expert's Netlist Driven Layout

Expert features a powerful Netlist Driven Layout (NDL) function to assist the user in creating a layout. It increases the productivity of layout design by automating cell generation and providing visual cues to assist in the wiring process. In this example, a latch circuit layout will be built based on developed child cells.

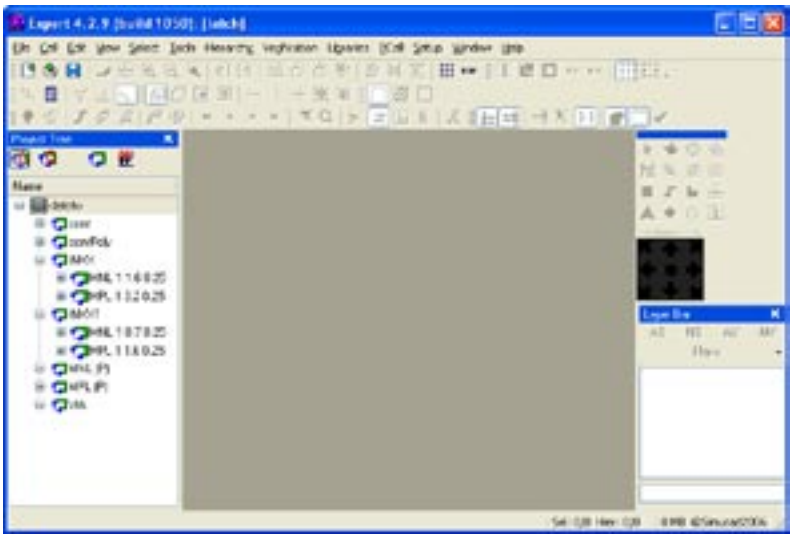

Figure 1. Loading project that includes child cells.

1. Open **<latch.eld>** project file from ***Expert***. In this project, there are some cells (such as INVX, MNL, MPL) etc developed and ready to be used to generate the latch circuit. See Figure 1.
2. Make sure that NDL options 'Show Nets', 'Show Pins' and 'Show Ports' are ON from 'Tools→ Netlist Driven Layout' or toolbar.
3. Use 'Tools→ NDL→ Load Netlist' to load ***Gateway*** netlist **<latch1.net>**.
4. Click '+' in the Netlist Rover sub-panel from Netlist Panel, i.e. Netlist Editor Window to expand circuit hierarchical tree. See Figure 2.
5. Select "MNL" in the tree, then click right mouse button to open pop-up menu. There already is a cell "MNL" in this project, so users can click on 'Open' to see the P-cell with default parameters and corresponding XI-script (displayed in separate window). The cell has two Metal1 pins D and S, and Poly box assigned pin G.
6. Select cell "INVX" in the circuit hierarchical tree, and click right mouse button. Select 'Open' in pop-up menu to open existing view "INVX" and inspect its pins. Switch from 'Flat View' into 'Lazy View' via 'View→ Cell View→ Lazy' or from Hierarchy toolbar to see ports in instances MPL (MI0) and MNL (MI1). See Figure 3.

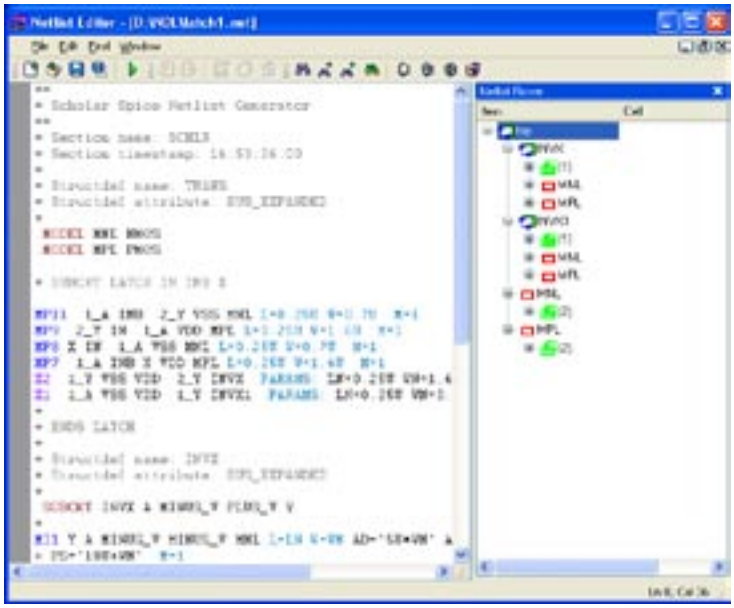

Figure 2. Importing Netlist.

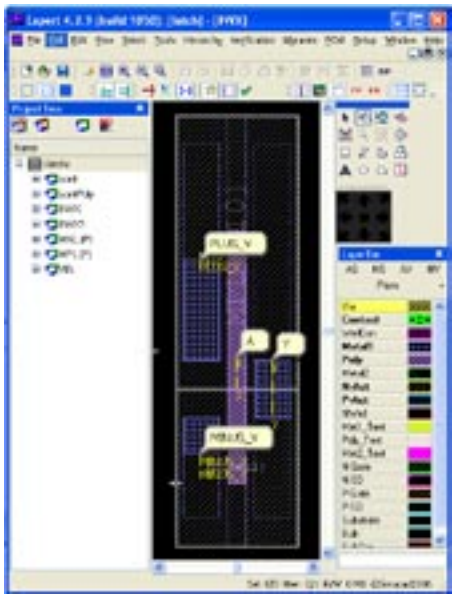

Figure 3. Viewing of circuit element with 'Open' command from Netlist Rover.

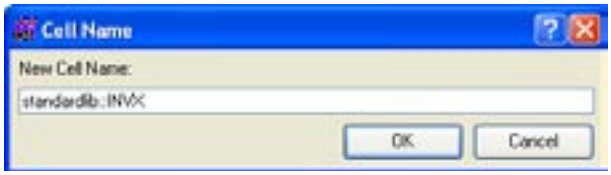

Figure 4. Mapping subcircuits/devices to the cells from the other library/project.

7. Select cell "top" in the netlist hierarchy tree. As there is no cell "top" in the project as shown in Figure 3, the user has to right mouse click to activate the pop-up menu with 'Open', 'Create' and 'Map'.

8. 'Map' command opens a dialog which allows setting correspondence between model name (or subcircuit name) in netlist with cell name in layout. Also, if the user would like to use standard cell(s) e.g. INVX from another library/project e.g. "standardlib" instead of current work library/project, please key in following syntax (Figure 4):

   ```
   Lib/Prj_name_of_cell::cell_name
   ```

   Note that library "standardlib" should be an active library/project, which can be set from 'Expert→ Library→ Set up'. For this example no mapping is necessary, since the user is using child cells from the same library/project.

9. Click right mouse button at "top" again, then select 'Create' command in the popup menu. NDL tool uses cells that are already in the project MNL, MPL, INVX and INVX1 to create new cell "top". MNL and MPL are P-cells, whose instance parameters L and W are selected based on netlist <latch.net>. See Figure 5.

10. Note that the default cell name "top" can be changed at will. Before creating the circuit layout, in the netlist panel, right click "top" and select 'Map' in pop-out menu, key in any preferred cell name (e.g. "latch"). If the user would just like to create this circuit within the current working directory, there is no need to key in the library name since the default parent project will be the current working directory. See Figure 6.

11. After the NDL run, the new cell "top" (or the cell name just chosen by 'Map' command) opens automatically. ***Expert*** shows flight lines to indicate which pins belong to the same net and should be connected to each other, as seen in Figure 5.

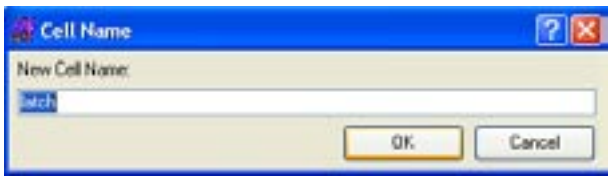

Figure 6. Assigning the generated cell a meaningful name in current project.

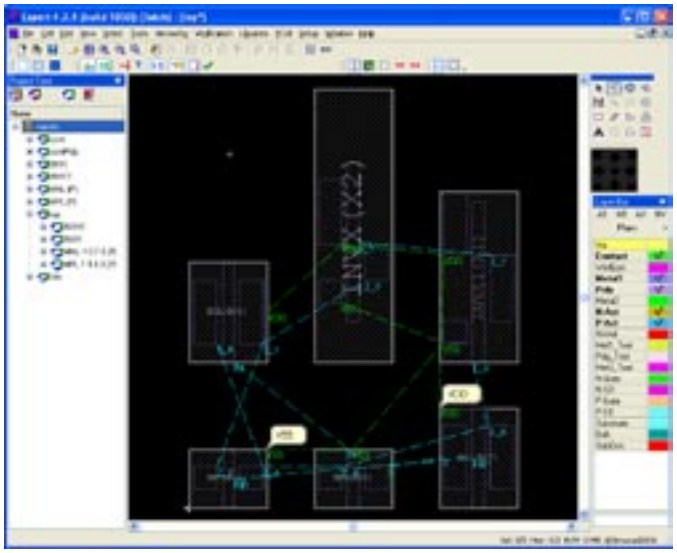

Figure 5. Initial placement with highlighted nets.

12. Next, the user may need to compare instances names and parameters in layout with netlist data, and inspect connectivity info which are represented by flight lines, and compare with net names in the netlist. Following functions can be used during inspection.

    - To switch views, perform 'View → Cell View → Flat / Lazy'. In Lazy view ports of cell instances appear in dashed lines.
    - Blue flight lines indicate internal nets, external connections are yellow.
    - 'Tools → NDL → Show Global Nets' shows green flight lines for global nets VDD and VSS connections. Global nets have a lot of connections; therefore turning its visibility off decreases the congestion of flight lines and simplifies design editing.

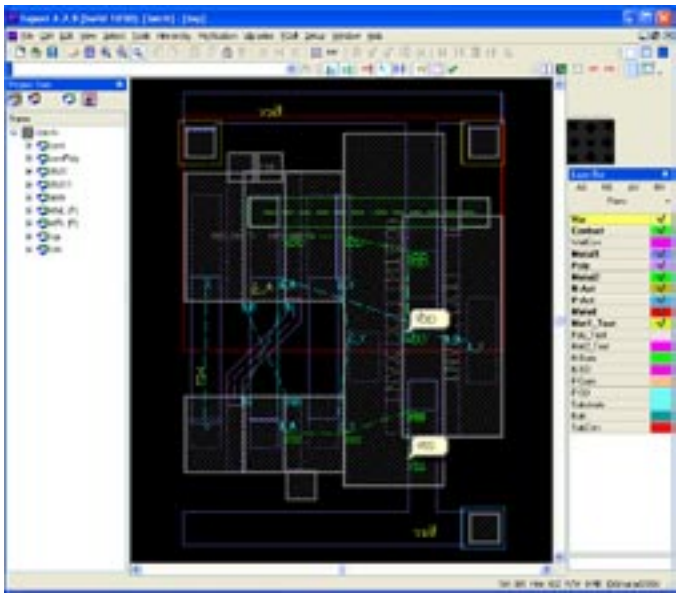

Figure 7. Edited Layout view before re-extraction.

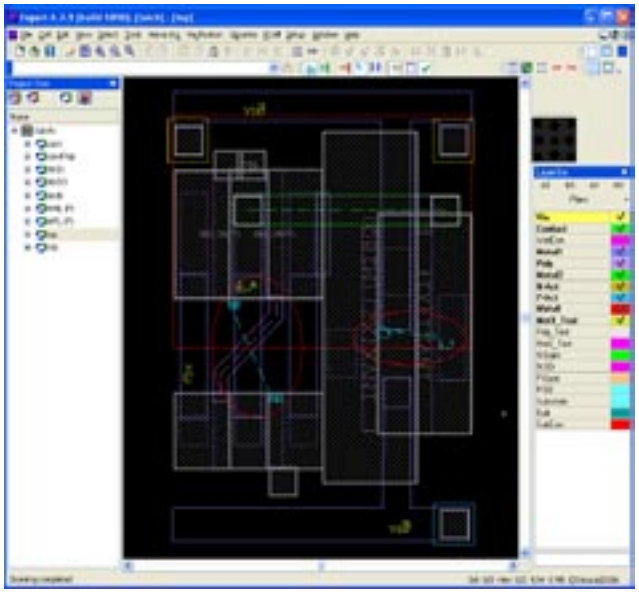

Figure 8. Edited Layout view after re-extraction to show Unfinished Nets Only.

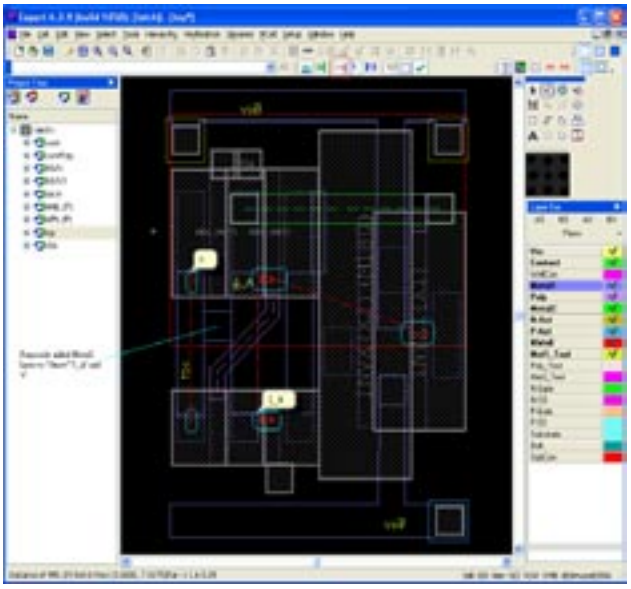

Figure 9. Edited Layout view after re-extraction to show 'Shorted Nets Only'.

- Note that cells MPL and MNL do not have VDD and VSS pins, so global flight lines end at the cell instance bounding box.

13. To improve initial placement, select Move/Flip/Rotate at any instance. Meanwhile the user can manage nets of generated layout via 'Tools→ NDL→ Nets'. Add connections as indicated by the flight lines.

14. Edit the cell "top" until it looks like the following Figure 7, which has improved placement and wiring added. Till this point the layout may appear messy. To avoid this, the user can check 'Tools → NDL → Unfinished Nets Only', then press 'Update Connectivity' or use 'Reextract' button in NDL tool bar to show flight lines that include unconnected pins.

15. 'Tools → NDL → Closest Points Only' shows parts of unfinished nets that represent missing connections (use 'Update Connectivity' or 'Reextract' to update display). With this feature, ***Expert*** NDL can provide users with updated connectivity from time to time, assisting user to finish the connectivity. It can be seen from Figure 7 and Figure 8 that the connectivity has been updated after reextraction.

16. Add a box in Metal1 on the left to intentionally short nets 'X' and '1_A' as seen in Figure 9. After 'Reextract' or 'Update Connectivity', flight lines for both nets 'X' and '1_A' displayed in red.

    Check option 'Tools→ NDL→ Show Shorted Nets Only' to show flight lines for shorted nets only to simplify looking for shorts and fixing the problem nets.

17. Continue to clear unfinished nets and shorted nets, if any. The layout view after having rectified above-mentioned unfinished and shorted connections is shown in Figure 10.

    To read more about Netlist Driven Layout feature, user may refer to ***Expert*** and ***ExpertView*** User Manual. If the reader would like to have these demo files (i.e. netlist, ***Expert*** .eld files, etc), please contact the nearest local Silvaco support team.

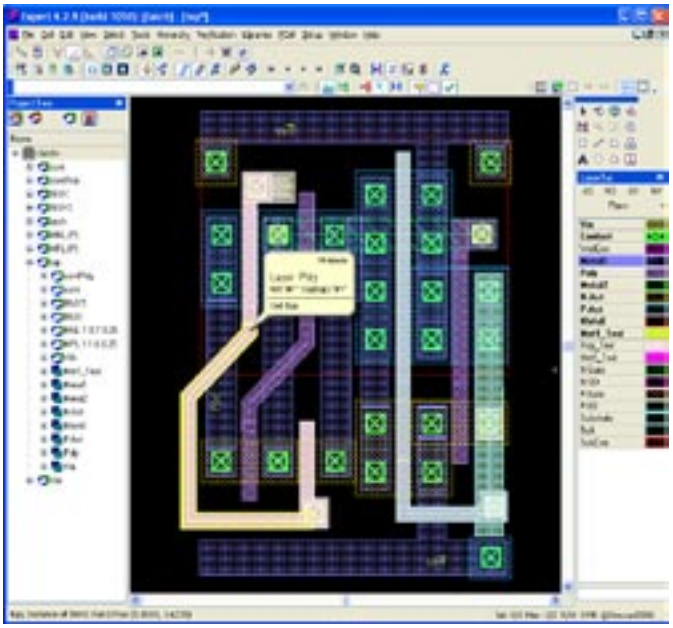

Figure 10. After having rectified unfinished and shorted connections.

Calendar of Events

June

1	
2	
3	
4	
5	
6	IITC – Burlingame, CA
7	IITC – Burlingame, CA
8	
9	
10	
11	
12	
13	DAC - San Diego CA
14	DAC - San Diego CA
15	DAC - San Diego CA
16	DAC - San Diego CA
17	DAC - San Diego CA
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	

July

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	NSREC - Seattle WA
12	NSREC - Seattle WA
13	NSREC - Seattle WA
14	NSREC - Seattle WA
15	NSREC - Seattle WA
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	

Bulletin Board

DAC – Design Automation Conference (DAC)

Come and see Silvaco at the largest annual EDA conference in San Diego. Our best applications engineers will be on hand to demonstrate our latest software products to solve your IC design automation problems.

NSREC Nuclear & Space Radiation Effects

NESREC is an important conference for Silvaco's radiation aware design flow that handles Single Event Effects (SEE). Silvaco International is one of only a few vendors offering commercial analysis with SEE-analysis features. Silvaco a few years ago expanded its SmartSpice circuit simulator to accommodate SEE-aware Spice models. The company also recently completed a Defense Advanced Research Projects Agency contract to enhance its Harmony AMS mixed-signal simulator with soft-error-rate and other radiation-effect capabilities. That tool is commercially available, but some of Silvaco's other technologies are available only to military contractors.

Memories have led the way when it comes to speed, low power, and density, and thus have become the proverbial canary in the coal mine when it comes to SEEs, according to Christopher Nicklaw, senior staff engineer at Silvaco. But the canary is no longer singing. "This is one of those unique times in which military electronics' needs are coexisting with the commercial requirements," says Nicklaw. "There is an agreement in both camps that this is a problem that we need to look into."

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***

Hints, Tips and Solutions

Galina Makovsky, Applications and Support Engineer

Q: Is there a way to turn-off all of the generated layers so that the layout modifications can be done without them cluttering things?

A: When a project is loaded, ***Expert*** automatically creates default layer plans with the names **FULL__**, **DATA__**, **CONNECT__**, **INPUT__**. These layer plans are automatically updated when you open a cell or select one of these plans from the layer plan drop-down list in the Layer bar.

FULL__ layer plan shows all layers present in the technology.

DATA__ layer plan shows only layers that contain geometry in the current cell.

CONNECT__layer plan shows only layers present in the connectivity sets.

INPUT__ layer plan shows all layers present in the technology except generated layers. All layers that have flag Derived or Scratch are hidden.

Generated layers that appear in layout after extraction are either Derived or Scratch layers. Use layer plan **INPUT__** to hide all generated layers at once (Figure. 1).

Tools >> Derived Layers >> Clean Derived Layers in **Whole Project** or in **Cell Hierarchy** deletes all objects from derived layers. If extraction technology uses Scratch layers, use **Tools >>Scratch Layers >> Clean Scratch Layers** to delete objects from generated layers (Figure. 2). If some modifications will be done to layout, derived layers should be rebuild anyway.

Q: I used the Recovery Log in previous versions of *Expert* to keep a record of all operations executed by the user for recovery purposes, but I can't find option "Write Recovery Log" in the new *Expert* QT.

Figure 2. Clean Scratch Layers command

Figure 1. Default Layer Plans

A: ***Expert*** Qt version has a new recovery engine to replace the recovery log in the previous versions.

The old recovery log facility had the following disadvantages:

1. User has to be familiar with ***Expert*** Scripting Language (advanced Expert user) to execute recovery script.
2. The recovery log became inconsistent after saving particular cells (not whole project).
3. Inability to recover the data from linked libraries.

The new recovery engine is based on recovery info saved inside ***Expert*** database file. Therefore the recovery data is written independently into the current project and activated libraries. To activate ***Expert*** recovery feature, check the option "Recovery info" on the "Data Safety" page of the ***Expert*** setup panel and specify interval in minutes you want to save recovery info (Figure. 3). ***Expert*** will automatically write recovery data at the

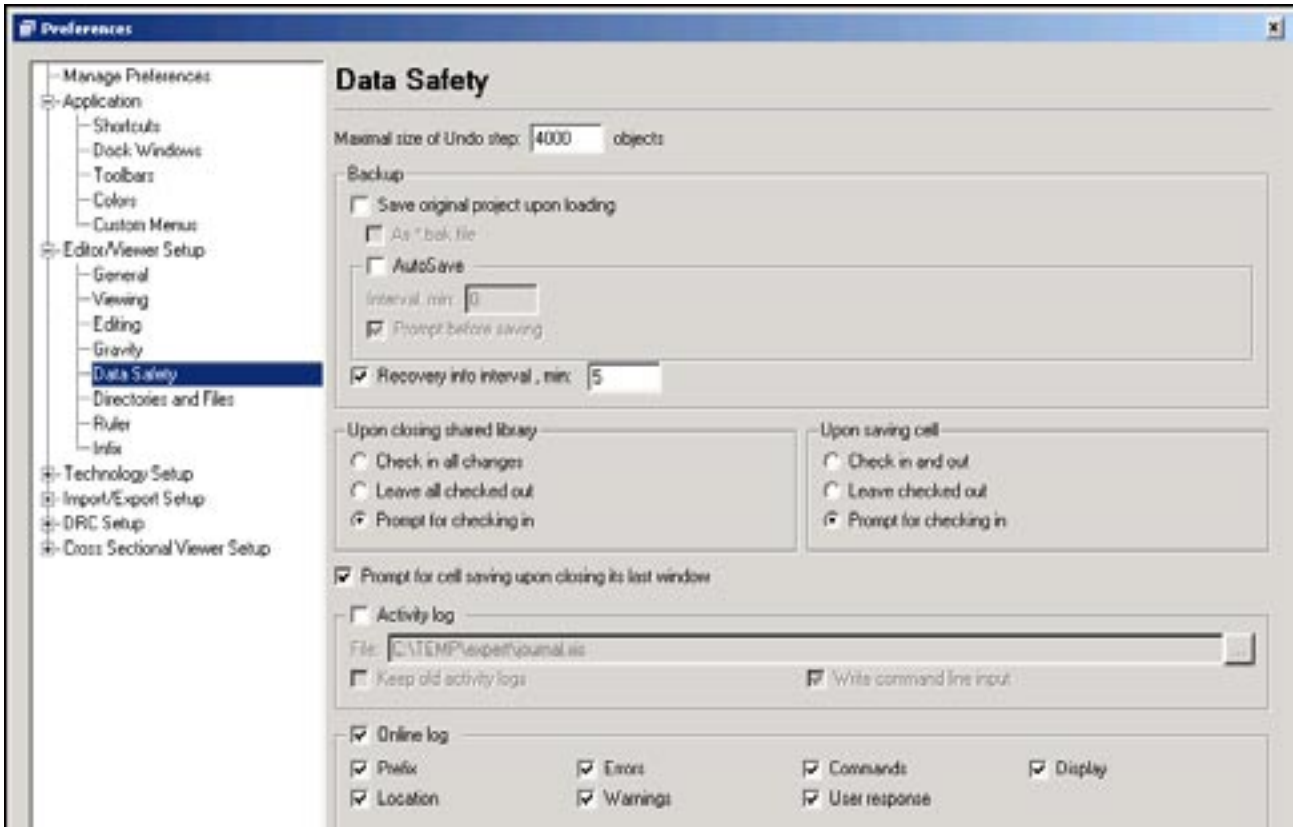

Figure 3. Data Safety

entered interval.

NOTE. Saving recovery data does not replace the Project>> Save command you must use for saving your project.

If ***Expert*** is terminated abnormally (***Expert*** freezes, computer hangs or you lose power unexpectedly) you will be able to recover unsaved data. Restart ***Expert*** and load the project file, which may contain unsaved information. When unsaved data are detected the message box will pop up asking whether you want to save (discard) recovered data or continue loading project with recovered information. The last choice is allowed to open and examine recovered cells (they will be marked as modified). You can save or discard whether specific cells or whole project by Save project or Close project without saving.

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

Analog/MS EDA Design Leader

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

*Products Licensed through Silvaco or e*ECAD*