Virtuoso® Layout Editor

Version 5.0

Lab Manual December 16, 2002

cādence

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Cadence Trademarks

Alanza℠
Allegro®
Ambit®
Assura™
BuildGates®
Cadence® (brand and logo)
CeltIC™
ClockStorm™
CoBALT™
Concept®
Connections®
Diva®
Dracula®
ElectronStorm™
Fire & Ice®
First Encounter®
FormalCheck®
Gate Ensemble®
HDL-ICE®
how big can you dream?™
Integration Ensemble™
MegaSim™
Mercury™
NC-Verilog®
OpenBook® online documentation library
Orcad®
Orcad Capture®
Orcad Layout®
Palladium™
Pearl®
PowerSuite™
PSpice®
Q/Bridge®
QPlace®
Quest®
Quickturn®
Radium™
SignalStorm™
Silicon Ensemble®
SoC Encounter™
SourceLink® online customer support
SPECCTRA®
SPECCTRAQuest™
Spectre®
TtME®
Vampire®
Verifault-XL®
Verilog®
Verilog®-XL
Virtuoso®
VoltageStorm™

Other Trademarks

All other trademarks are the exclusive property of their respective owners.

Confidentiality Notice

No part of this publication may be reproduced in whole or in part by any means (including photocopying or storage in an information storage/retrieval system) or transmitted in any form or by any means without prior written permission from Cadence Design Systems, Inc. (Cadence).

Information in this document is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

RESTRICTED RIGHTS LEGEND Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

UNPUBLISHED This document contains unpublished confidential information and is not to be disclosed or used except as authorized by written contract with Cadence. Rights reserved under the copyright laws of the United States.

Table of Contents
Virtuoso Layout Editor

Module 1 The Design Environment

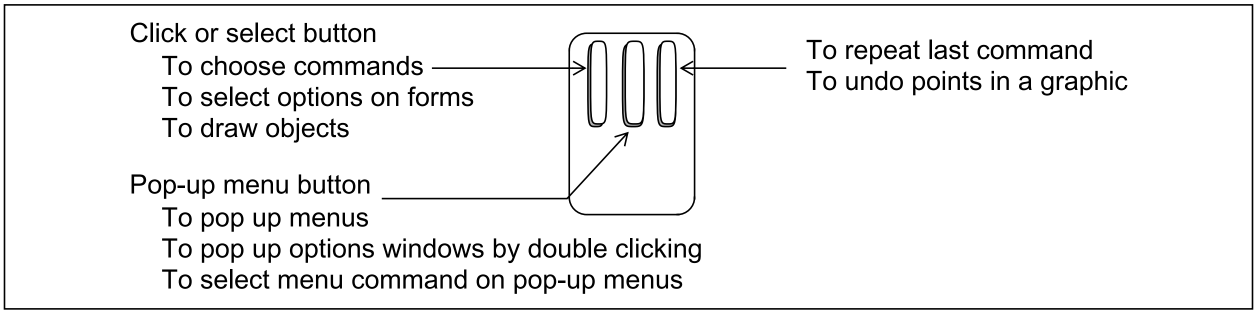
Module 5 Using Assura

App. D Microwave Design

Mouse Use and Terminology

Term	Action	Icon Example
click	Quickly press and release the specified mouse button. On menus and forms, you use the **left** mouse button most of the time.	**Click left**
double click	Rapidly press the specified mouse button twice.	2 **Double click**
Shift-click Control-click Shift-Control-click	Hold down the appropriate key or keys and click a specified mouse button.	Shift **Shift- click right**
draw through	Define a box by clicking the mouse at one corner of the box, moving to the diagonally opposite corner with the mouse button held down, and releasing the button.	→ **Draw through**
pop up	Press the **middle** mouse button.	**Click middle**
pull down	Move the mouse cursor to the menu name on the menu banner, press and hold the **left** mouse button, move the cursor down to highlight the menu selection, release the mouse button to execute the selection.	
Enter	Type a command in a window and press **Return** to execute the command.	
Select	Position the cursor over a command and press the **left** mouse button. *Choose* or *pick* are synonyms for select.	

The basic uses of mouse buttons are shown in this graphic.


Click or select button
To choose commands
To select options on forms
To draw objects
Pop-up menu button
To pop up menus
To pop up options windows by double clicking
To select menu command on pop-up menus
To repeat last command
To undo points in a graphic

Labs for Module 1

The Design Environment

Lab 1-1 Using the Cadence Online Documentation System

Objective: Learn to use the new HTML-based Cadence online documentation.

The Cadence® Online Documentation uses your web browser to read the Cadence product manuals for the Virtuoso® Layout Editor and all other tools.

1. Type in the UNIX window:

   ```
   cdsdoc &
   ```

   The Cadence Documentation window will take a moment to open.

2. In the Cadence Documentation window, double-click on the **Virtuoso Layout Editor** product name.

   The field expands to show the documentation relevant to the Layout Editor.

3. Double click on the **Virtuoso Layout Editor User Guide**.

4. Double click on **Table of Contents**

   The document page is loaded into your Internet browser. This may take a moment. At the top of the window is a built-in toolbar that lets you move forward and backward through the chapters. The document also has hyperlinked cross-references.

5. In the Cadence Documentation window, click on the **Search** button.

   The Search window appears. Use the *All of the following* buttons or select a specific Family, Product or Book to narrow your search. You can also reduce your search further by using the boolean operators such as AND, OR, and NOT.

   Note: *You can also start the Cadence Documentation from the Help button in any window of the application. Once the netscape browser opens, click on the Library button located on the toolbar to bring up the Cadence Documentation window.*

6. Type in the *Search for* field:

   ```
   .cdsinit
   ```

7. Choose **Custom IC Layout** from the *Families* section , **Virtuoso Layout Editor** from the *Products* group, **Virtuoso Layout Editor User Guide** in the *Books* section and click **Go**.

   The results show a number of documents titles to choose from. You can click on the cross-referenced text to find specific information about the *.cdsinit* file.

8. Iconify the Netscape Search Results window.

End of Lab

Lab 1-2 Opening the Design

Objective: Use the Open File form and the Library Manager to open a design window.

You can open a design window by using either the Open File form or the Library Manager. The Library Manager allows you to edit the data inside of the library, such as changing properties or renaming cells.

Starting the Cadence Software

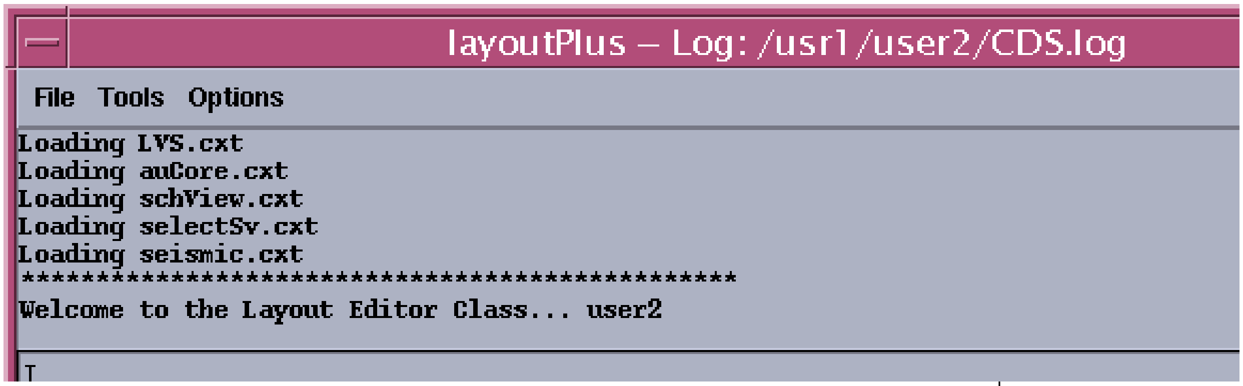
1. Change to the working directory in an xterm window. Enter:

```
cd Layout
```

2. Start the Virtuoso® Layout Editor software in background mode. In the same xterm window, enter:

```
layoutPlus &
```

You may see a *What's New in 5.0.x* window appear. Select **Edit—Off at Startup** from this window to prevent this window from appearing again. Select **File—Close** in the window to close it in this session.



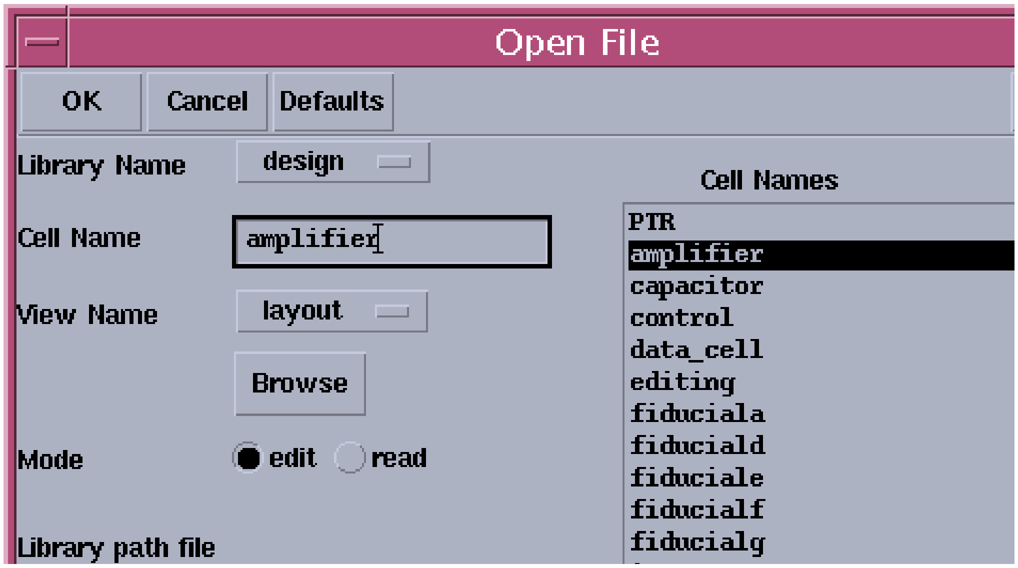
```
Loading LVS.cxt
Loading auCore.cxt
Loading schView.cxt
Loading selectSv.cxt
Loading seismic.cxt
*************************************************
Welcome to the Layout Editor Class... user2
```

When you start the *layoutPlus* executable, the first tool window displayed is the Command Interpreter Window (CIW). From the CIW you can open designs, set global options, view the history of the current session, and input SKILL expressions. It is always a good idea to have the CIW visible to view any messages that are given from commands that you have invoked.

Viewing the Fields in the Open File Form

1. Move your cursor into the CIW and select **File—Open**.

   The Open File form appears.



Library Name design

Cell Name amplifier

View Name layout

Mode ● edit ○ read

Library path file

Cell Names

PTR
amplifier
capacitor
control
data_cell
editing
fiduciala
fiduciald
fiduciale
fiducialf
fiducialg

2. In the Open File form, the Library Name is set to **design**. Click on the Library Name **design** to view the listing of libraries available for editing.

3. From the Cell Names field choose **amplifier**. The listing of Cell Names is alphabetical, so you may need to use the scroll bars.

4. In the View Name field click on the listing of available views and select **layout**.

5. You will now choose a different cell to open.

6. Click on **Browse**.

Opening the Layout

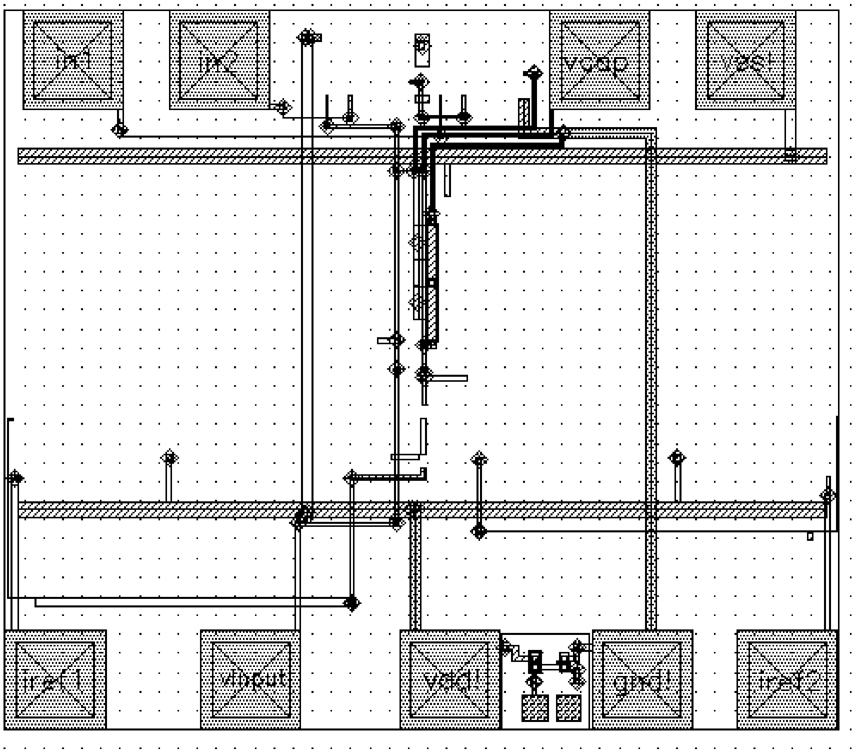
1. In the **Library Browser —Open File** form, click on the *pk44chip* cell.

   Notice the Cell Name in the Open file form has updated to *pk44chip*.

2. Click **OK** in the Open File form.

   The *pk44chip layout* cellview opens.

   The *fiducial layout* cellview is flashing. This cellview is located in the *subdesign* library which is not defined in the *cds.lib* file. The *cds.lib* file specifies which libraries are available during the editing session.



   Notice that there are empty areas of the design. As you go through the course you will be completing these areas and finishing the *pk44chip* design.

Summary

In this lab, you learned the following:

- How to start the Layout Editor software.
- How to view the fields in the Open File form.
- How to open a design window.

Lab 1-3 Setting the Library Path Editor

Objective: Learn how to add a reference library to the Library Path Editor.

The subdesign library is missing from the library path. Once you add the subdesign library to the path, the *pk44chip* layout cellview will be updated.

Opening the Library Path Editor

1. From the CIW, select **Tools—Library Path Editor**.

   The Library Path Editor form appears.

   You define the libraries that you want to use during the editing session in the *cds.lib* file. The Library Path Editor form reads the information from the *cds.lib* file. This file must reside in the directory where you start the software (the working directory). When you invoke *layoutPlus,* the *cds.lib* file is read.

2. In an xterm window type:

   ```
   more ~/Layout/cds.lib
   ```

   Does the cds.lib file match the library path in the Library Path Editor form?

Adding the *subdesign* Reference Library

You will add the subdesign library to the Library Path Editor. The library, *subdesign*, along with the libraries, *design*, *pcells* and *RODpcells* are all placed within a directory called *cdslibs*.

1. In the **Library Path Editor** form select **Edit**—**Add Library**.

2. From the *Directory* field, double click on the directory *cdslibs*.

   This is the directory location of the library subdesign.

3. In the *Library* text field click on **subdesign** and click **OK**.

   The *subdesign* library and the path is added to the *cds.lib* file.

4. Select **File—Save**.

5. This information is saved to the *cds.lib* file. In an xterm window type:

   ```
   more ~/Layout/cds.lib
   ```

   Is the cds.lib file updated with the new library path?

   Note: *To view the changes in the cellview you must redraw the design window.*

6. In the *pk44chip* design window, select **Window—Redraw**.

   Is the fiducial layout cellview still flashing?

Viewing the Library Path Information

View the exact locations of the libraries that are defined in the *cds.lib* file.

1. In the Library Path Editor form, click on the *design* library.

2. Select **View—Library Info**.

   A text box appears with the Library name, path, and where the path is defined in the *cds.lib* file.

3. Click on **Close** in the text box.

4. From the Library Path Editor form, select **File—Exit**.

5. From the *pk44chip* design window, select **Window—Close**.

Summary

In this lab, you learned the following:

- How to add a reference library to the *cds.lib* file using the Library Path Editor.

Lab 1-4 Learning to Use the Library Manager

Objective: Use the Library Manager to create, copy, delete, and organize the cells and libraries.

The Library Manager lets you open a design and edit the information inside of your library. You can use the Library Manager to create, add, copy, delete, or organize the cells and libraries you use in a design project.

1. To open the Library Manager select **Tools—Library Manager**.

   The Library Manager form appears.

2. Resize the Library Manager window horizontally, so that you can view all of the fields clearly.

Viewing the Contents of the *design* Library

1. Click on the *design* library, and turn on the **Show Categories** option.

   Notice that the Category field is updated. Categories are a way of grouping cells together. If you have many cells in your library, you may want to put cells that are of a similar type into a category.

2. In the Category Field:

   a. Click **Uncategorized**.

   All cells that are not in categories are shown.

   b. Click **Everything**.

   All cells are shown, even if they are in a category.

   c. Click on **symDevices**.

   All symbolic devices are shown.

Filtering Cell and View Information

To learn more about the cells that are in the design library use the filter option to find specific cell information.

1. In the Category field click on **Everything**.

2. Find the **amplifier** cell in the *Cell* section of the Library Manager.

3. Click on **amplifier** to expand the views.

   What are the view names that are associated with the cell amplifier?

4. In the Library Manager select **View—Filters**.

   The View Filter By form appears.

5. In the View Filter By form change the following:

Cell Filter	`a*`
View Filter	`l*`

6. Click **OK**.

   What cells and views are listed in the Library Manager now?

7. To clear the filter select **View—Filters**.

8. Remove the text from the text fields and click **OK**.

Creating a New Category

You will create a category in which all of the fiducial cells will be placed.

1. Select **File—New—Category**.

   The New Category form appears.

   What cells are not in categories?

2. In the Category Name text field type:

   ```
   fiducials
   ```

3. In the **Not in Category** of the Cells section, click on **fiduciala**.

   Is this listing in alphabetical order?

4. To add the rest of the cells press the **Control** key and click on the following cells:

   fiducaild

   fiducaile

   fiducialf

   fiducialg

5. Click on the arrow pointing to the **In Category**.

6. Click **OK** in the New Category form

   What happened in the Category section of the Library Manager?

7. Expand the category **fiducials**.

   Do you see the cells that you put into the fiducials category?

Creating a New Library

You will create a new library called *myLib*.

1. From the Library Manager form, click on **File—New—Library**.

2. In the Name field of the New Library form type:

   ```
   myLib
   ```

3. Click **OK**.

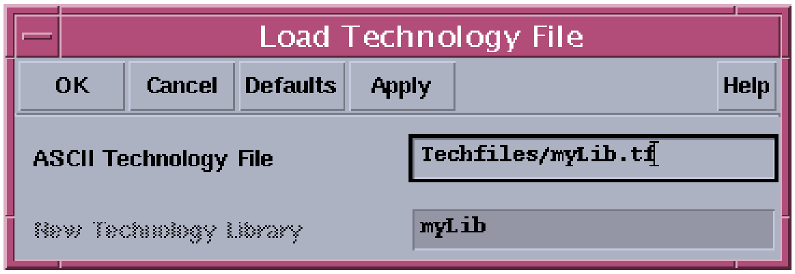
   The Technology File for New Library form appears.

   You have the option of compiling a new technology file, attaching to an existing technology file, or if you do not plan to design layouts, you do not need a technology file. For this example, you will compile a new techfile.

4. Click **OK** to the default, **Compile a new techfile**.

   The Load Technology File form appears.

5. Type *Techfiles/myLib.tf* in the ASCII Technology File text field.



ASCII Technology File: Techfiles/myLib.tf

New Technology Library: myLib

6. Click **OK**.

   A new design library called *myLib* is created and is now listed in the Library Manager. A popup form apears to show the Technology file loaded successfully.

7. Click on **Close** in the popup form.

8. In the *Library Manager* form, Click on the library *myLib* and view the list of cells.

   The cells in *myLib* are symbolic devices. Symbolic devices are defined in the technology file. The cells in *myLib* were defined in the techfile you compiled, *myLib.tf*.

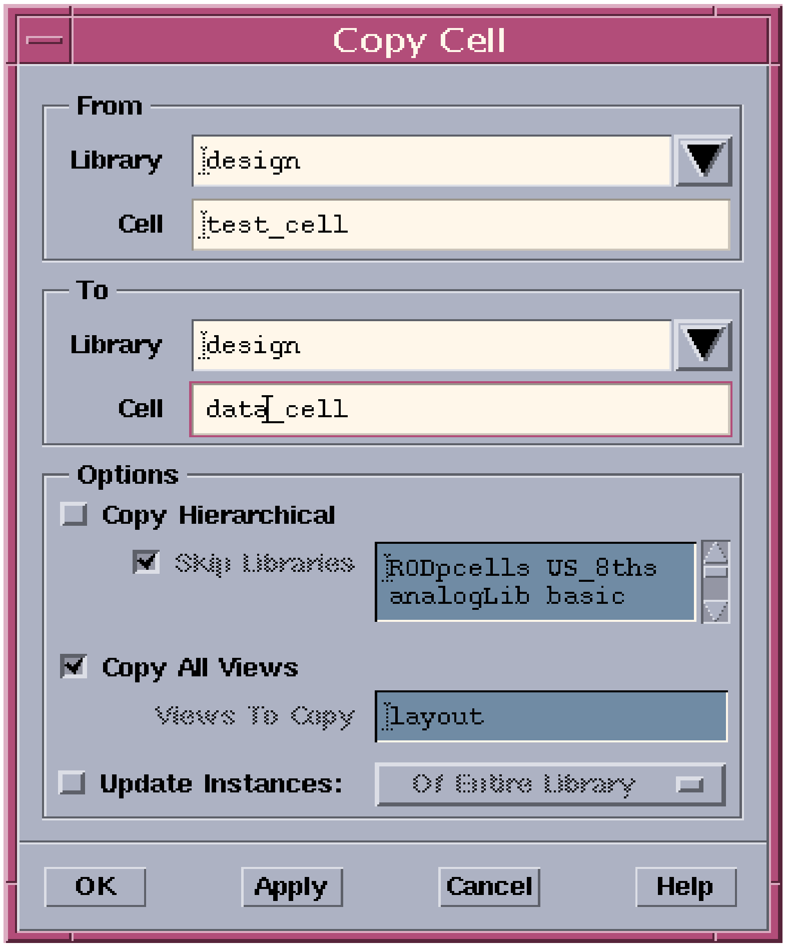
Copying a Cell

Use the Library Manager to copy the cell, *test_cell* from the *design* library to a new cell called *data_cell*.

1. In the Library Manager form click on *test_cell* from the *design* library with the middle mouse button.

2. On the highlighted *test_cell*, hold down the middle mouse button and select **Copy** from the pull-down menu.

   The Copy Cell form appears. The selected library and cell are displayed in the From Library and From Cell fields.



Copy Cell

From

Library: design

Cell: test_cell

To

Library: design

Cell: data_cell

Options

☐ Copy Hierarchical

☑ Skip Libraries: RODpcells US_8ths analogLib basic

☑ Copy All Views

Views To Copy: layout

☐ Update Instances: Of Entire Library

3. In the **To** section of the form, enter *data_cell* in the Cell text field.

4. Click **OK**.

   The **Copy Problems** form appears. The *data_cell* already exists in the design library.

5. Overwrite the *data_cell* by clicking on the **Overwrite All** button.

   By choosing Overwrite All you are replacing every occurrence of the *data_cell* throughout the design library. Use this command with caution.

6. Click **OK** in the Copy Problem form.

Deleting a Cell

You need to delete the cell, *test_cell* from the design library.

1. Select the **test_cell** from the *design* library.

2. Notice that the library, cell and view are highlighted. Click on the **layout** view to remove the highlighting.

   If you had left the layout view highlighted only the cellview *test_cell layout* would have been deleted, and the cell *test_cell* would remain. In this example you want to delete the cell, which will delete all of the views.

3. Select **Edit—Delete**.

   The Delete Cells form appears.

   Which cell is highlighted to be deleted?

4. Click **OK** in the Delete Cells form.

5. Click **Yes** in the *Delete Confirmation* dialog box.

   Where is the message in the Library Manager that tells you that the cells have been deleted?

6. **Window—Close** *pk44chip layout.*

Summary

In this lab, you learned the following:

- How to filter cells in the design library.
- How to create a category.
- How to use the library manager to copy a cell.
- How to create a library.
- How to delete a cellview from a library.

Lab 1-5 Creating a Bindkey

Objective: Learn to define your own bindkey.

Bindkeys are another way of invoking commands. Instead of using the pull-down menus, you press a key on the keyboard that corresponds to a command.

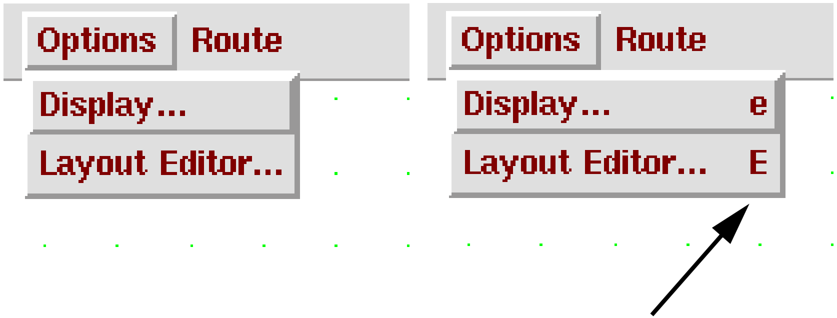
1. In the Library Manager form click on the **pk44chip** cell.

2. On the **layout** cellview, click and hold down the middle mouse button.

3. Open the cellview by selecting **Open** from the pull-down menu.

4. To verify that the bindkeys are not loaded click once on the pulldown menu **Options** from the window banner.

   When the bindkeys are not loaded, you will not see them next to each of the commands.



Bindkeys loaded

Loading the Bindkey File from the *.cdsinit*

You will add the bindkey path to the *.cdsinit* file. Then you will edit the bindkey file to add a bindkey for the **Zoom [z]** command.

1. In an xterm window enter:

   ```
   vi ~/Layout/.cdsinit
   ```

   The *.cdsinit* file opens in the *vi* editor.

   Note: *Appendix F in the Training Manual is a reference guide to the vi editor.*

2. Use the **down arrow** key (or the **j** bindkey) to scroll down to the *Bindkey* section.

   The commands for loading the bindkey files are commented out, so you must remove the comments and then reload the *.cdsinit* file from the CIW.

   You see the following syntax for loading the two bindkey files.

```
;load(prependInstallPath(“samples/local/schBindKeys.il”))
;load(“leBindKeys.il”)
```

3. To remove the semicolon (;) from each line so these files can be loaded by the *.cdsinit* file:

   a. Scroll down by using the **j** bindkey.

   b. Stop at the line:

   ```
   ;load(prependInstallPath("samples/local/schBindKeys.il"))
   ```

   c. Press **x** to delete the semicolon.

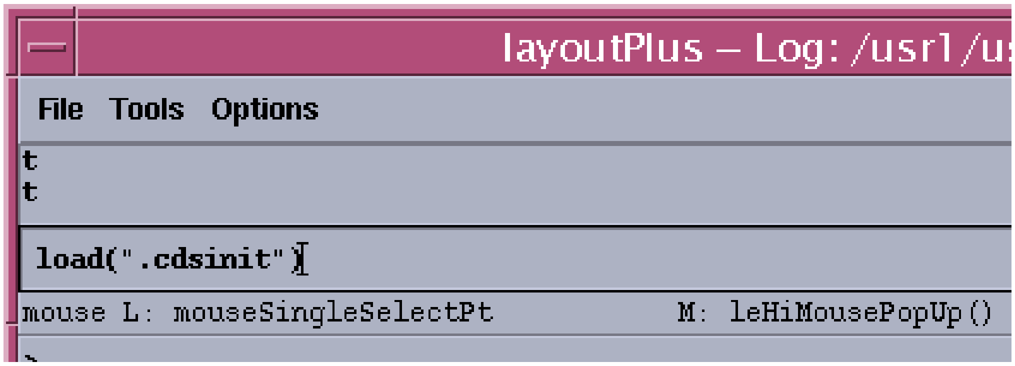
   d. Press the **j** bindkey again to move to the next line and press **x** to delete the semicolon from the second line.

4. To save your edits enter:

   ```
   :wq
   ```

5. To load the edited *.cdsinit* file, move your cursor into the CIW, and enter:

```
    load(“.cdsinit”)
```



```
t
t
load(".cdsinit")
mouse L: mouseSingleSelectPt
```

Verifying the Bindkeys

Now you will verify that the bindkeys have been loaded correctly.

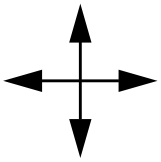
1. Place your cursor in the design window and press the **[e]** bindkey to bring up the Display Options form.

2. **Cancel** the Display Options form.

3. Press the arrow keys on the right keypad of your keyboard to pan around the window.

   Remember that when you pan to the left, your viewport moves to the left, but the design appears to move to the right.

   Each of the arrow keys pans the layout window four possible directions.

   a. Use the up arrow key to pan up.

   b. Use the left arrow key to pan to the left.

4. View the entire design by pressing the **[f]** bindkey.

Bindkey Definitions

View the Bindkey Aliases using the Show Bind Keys command.

1. In the CIW, select **Options—Bindkey**.

   The Key or Mouse Binding form is displayed.

2. Click on the **Application Type Prefix** cyclic field and select **Layout**.

3. To show the bindkeys, click on the **Show Bind Keys** button.

   A Layout Bindkeys text window opens. Different environments have their own bindkey definitions. These are the bindkey definitions for the layout environment. You can scroll, search, and save this file for later printing.

   In the left column are the keyboard key names, and in the right column are the SKILL functions bound to these keys.

4. Select **File—Close Window**.

5. **Cancel** the Key or Mouse Binding form.

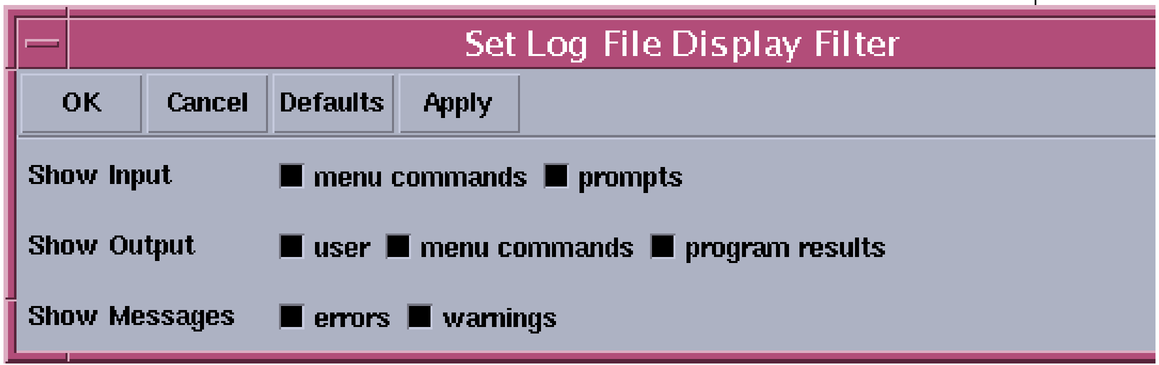
Finding the SKILL Syntax

You will add a bindkey to the *leBindkey.il* file which will allow you to zoom into an area. First you must find the SKILL syntax for the **Zoom In** command. You can find syntax for any command by invoking the command from the pull-down menu and then viewing the output field of the CIW for the corresponding SKILL syntax.

At this time you are filtering command information so that it does not appear in the output field of the CIW. You will change the Log File Display Filter so that you can view all command information and SKILL syntax in the output field.

1. Select **Options—Log Filter**.

2. In the Set Log File Display Filter, turn on all options as shown below.



Show Input ■ menu commands ■ prompts

Show Output ■ user ■ menu commands ■ program results

Show Messages ■ errors ■ warnings

3. Click **OK**.

   Notice that output field in the CIW is updated.

4. From the *pk44chip* design window, select **Window—Zoom—In**.

5. In the CIW and you should see the following:

   ```
   hiZoomIn()
   ```

   This is the SKILL syntax that you will add to the *leBindKeys.il* file.

Adding SKILL to the *leBindKeys.il* File

1. In an xterm window type:

   ```
   vi ~/Layout/leBindKeys.il
   ```

   The *leBindKeys.il* file opens in the *vi* editor.

2. In the xterm window type:

   ```
   110 G
   ```

   This will take you to line 110.

3. Press the **[o]** key to open a line.

4. Type in the following:

   ```
   bk(“Layout” “<Key>z” “hiZoomIn()”)
   ```

5. Press **[Esc]** to get out of edit mode.

6. To save your edits enter:

   ```
   :wq
   ```

7. To re-load the edited *leBindkey.il* file, load the *.cdsinit* file. In the CIW, enter:

   ```
   load(".cdsinit")
   ```

8. To zoom into a specific area, press the **[z]** bindkey.

   Do you see a prompt in the CIW?

   Note: *If you do not see the prompt you might have typed the bindkey information incorrectly. Check the leBindKeys.il file.*

9. Click on the upper-left corner of the area you want to display.

10. Click on the lower-right corner of the area you wish to display.

11. From the design window, select **Window—Close**.

Summary

In this lab, you learned the following:

- How to find if a bindkey is loaded.
- How to change the Log Filter.
- How to add a bindkey to the *leBindKeys.il* file.

Lab 1-6 Changing the User Preferences

Objective: Learn how to change the text size and use the Window Controls.

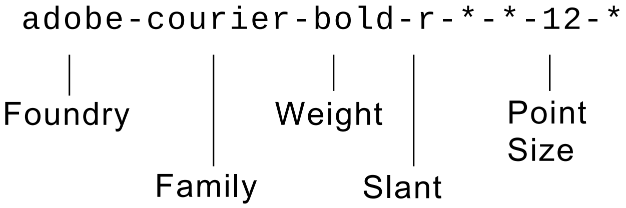
Changing the Text Size

In the last lab you had to read information from the output field of the CIW. The text size may be too small for you to read. You will learn how to change the text size in the following steps.

1. From the CIW, select **Options—User Preferences**

2. Look at the current Text Font setting at the bottom of the User Preferences form.

   What is the Text Font setting?



```
adobe-courier-bold-r-*-*-12-*
```

- Foundry refers to the designers of the typeface.
- Family is the style of the typeface.
- Weight is the typeface width.
- Slant refers to the angle.
- Point Size is the size of the type.

Finding Available Fonts

1. In an xterm window enter:

```
xlsfonts | more
```

The available fonts are listed.

Note: *To advance in the file, use the space bar to scroll a page at a time and the return key to scroll one line at a time.*

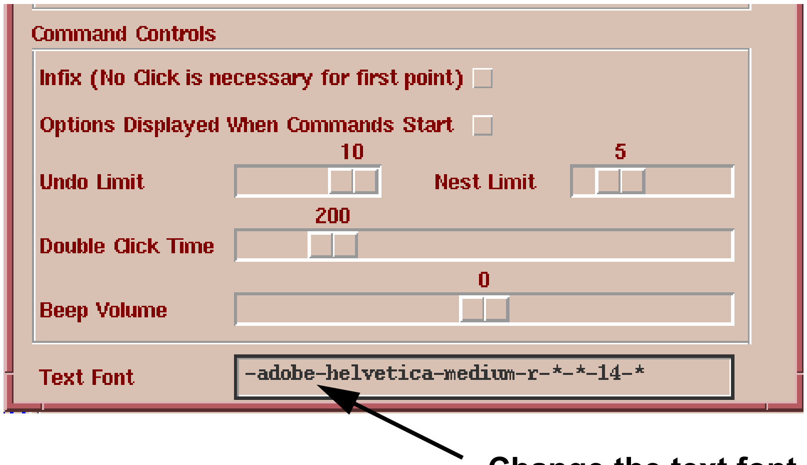
2. Look at the Helvetica fonts.

For this lab you will be using the following font:

```
-adobe-helvetica-medium-r-normal--14
```

Changing the Font

1. In the *User Preference* form, change the **Text Font** field as follows:



Command Controls

Infix (No Click is necessary for first point) □

Options Displayed When Commands Start □

Undo Limit 10

Nest Limit 5

Double Click Time 200

Beep Volume 0

Text Font -adobe-helvetica-medium-r-*-*-14-*

Change the text font

2. Click **OK**.

Notice that the CIW output field changes to the new font.

To update the LSW, you would have to exit out of *layoutPlus*, add the following line to your *.cdsinit* then re-start the software to see the changes.

```
hiSetFont("text"
"-adobe-helvetica-medium-r-*-*-14-*")
```

Changing the Window Controls

1. From the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`tutorial`
View Name	`layout`

2. Click **OK**.

3. From the CIW select **Options—User Preferences**.

4. Set the *Icon Bar* to **None**.

5. Turn the *Prompt line* and *Status Line* off and click **OK**.

6. Close the design window.

   Note: *You must close the windows and reopen them for the changes to be effective.*

7. Re-open the *tutorial* cellview.

   The *tutorial layout* design window is now updated to the user preferences you have just set.

8. Close the *tutorial* design window.

Summary

In this lab, you learned the following:

- How to find available fonts.
- How to change the font size.
- How to change the window controls.

Lab 1-7 Using the Layer Selection Window (LSW)

Objective: Learn how to use the LSW to create shapes.

The *fiduciald layout* cellview is not complete. You will be using the **Create Path** command to complete the interconnect. You will use the LSW to change to the current drawing layer and set the visibility of layers.

Setting the Current Drawing Layer

You will create a path on the *poly dg* layer using the LSW.

1. From the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`fiduciald`
View Name	`layout`

2. Click **OK**.

   The *fiduciald layout* cellview opens for editing.

3. Select **Create—Path** or press the **[p]** bindkey.

4. Look at the LSW and find the current drawing layer.

   Is the current drawing layer poly dg?

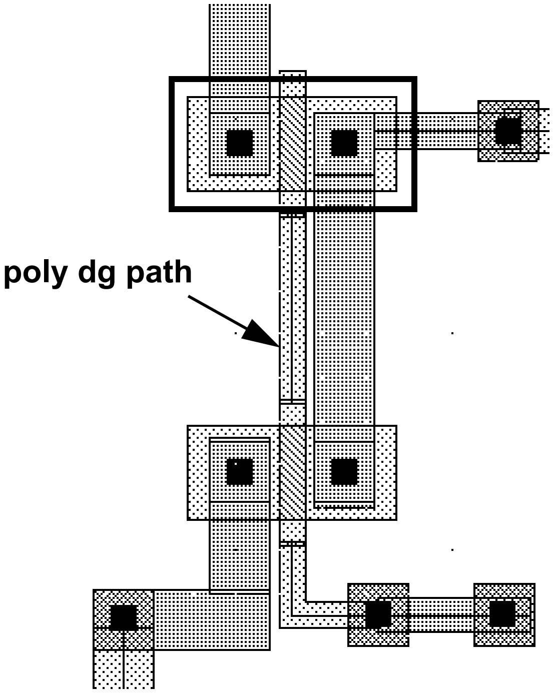
   The current drawing layer is set to *pwell dg*. You will change the current drawing layer to *poly dg*.

5. To set the current drawing layer, click with the left mouse button on the **poly dg** layer.

   Is the poly dg layer displayed at the top of the LSW?

6. To create the poly interconnect click at **X= 6.3**, **Y=12.7**

   Note: *For the X/Y locations look at the window banner of the design window.*



7. Then click **X= 6.3**, **Y= 8.5**.

8. To complete the path press the **Return** key.

9. Press the **[ESC]** key.

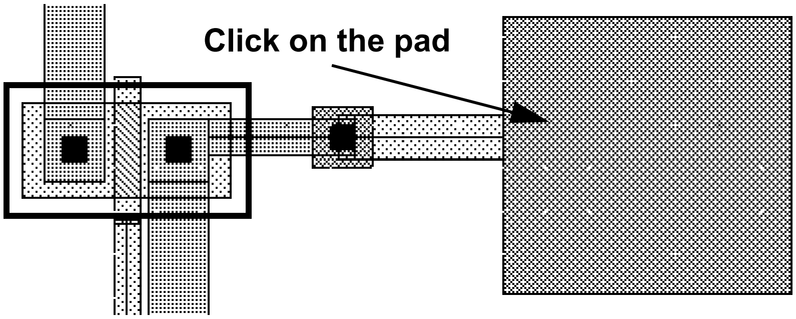
Setting the Current Drawing Layer Using the Layer Tap Command

Another way to set the current drawing layer is to use the **Layer Tap** command, which has the capability of cycling through all valid layers under a single point. The **Layer Tap** command prompts you to point at the figure to be tapped. The LSW makes the layer of the object you point to the current drawing layer.

You will be creating the *metal2 dg* interconnect between the output and the pads.

1. From the **LSW** select **Edit—Layer Tap** or press the **[t]** bindkey.

2. In the design window click on one of the magenta pads.

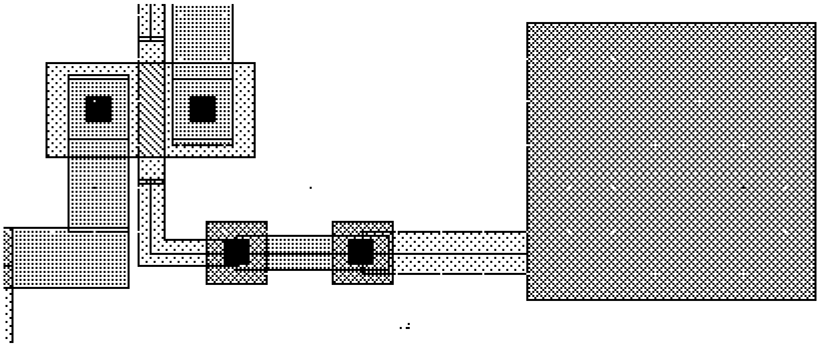


What is the current drawing layer?

3. Click again on the pad to set the current drawing layer to metal2.

4. Select **Create—Path [p]**.

5. To create the metal2 interconnect click at **X=11.2**, **Y= 3.5**, then **X =15**, **Y= 3.5**.



6. To complete the path press the **Return** key.

7. Select **Cancel** in the Create Path form.

Making Layers Visible

One of the features of the LSW is setting visibility.

1. With the middle mouse button, click on the *metal1 dg* layer in the LSW. If *metal1 dg* is your current drawing layer, make another layer, such as *poly dg,* the current drawing layer. The current drawing layer cannot be set to invisible.

   Both the name and the color box of this layer are grayed out. Any layer that is invisible is unselectable.

2. Move your cursor into the design window, then select **Window—Redraw [^r]** to redraw the cellview.

   The *metal1 dg* layer is no longer visible.

3. Move your cursor to the LSW, then with the left mouse button click **NV** (None Visible).

   All layers except the drawing layer are grayed out. You cannot make the drawing layer invisible.

4. Move your cursor into the design window, then select **Window—Redraw [^r]** to redraw the cellview.

   The current drawing layer is the only layer that is visible.

5. With the left mouse button, click **AV** (All Visible).

   Again, all layers are visible. Their names and color boxes are no longer grayed out.

   Although the layers are visible in the LSW, you need to redraw your screen before you can see the layers in the cellview.

6. Move your cursor into the design window, then select **Window—Redraw [^r]** to redraw the cellview.

   All layers are visible.

Making One Layer Visible

1. Move your cursor into the LSW and **Shift** click with the middle mouse button on the *ndiff dg* layer.

   Notice that all the layers are invisible except for *ndiff*, and the current drawing layer is now *ndiff*.

2. Move your cursor into the design window, then select **Window—Redraw [^r]** to redraw the cellview.

   The *ndiff dg* layer is the only layer that is visible.

3. To make another layer visible click with the middle mouse button on another layer in the LSW.

4. Click on **AV** to make all layers visible.

5. Move your cursor into the design window, then select **Window—Redraw [^r]** to redraw the cellview.

6. Select **Design—Save**.

Summary

In this lab, you learned the following:

- How to change the current drawing layer.
- How to set visibility.
- How to use the Layer Tap command.

Lab 1-8 Using the Display Resource Editor

Objective: Learn how to create a new packet and edit existing packets in the *display.drf* file.

A packet has attributes of color, linestyle, and stipple and controls how layers in your design are displayed and plotted.

Editing a Packet

The contact fill needs to be changed to an “x.” You can make these changes in the Display Resource Editor.

1. In the LSW select **Edit—Display Resource Editor**.

2. From the Layers section, select the *contact drawing* layer.

   You may need to use the scroll bar.

3. In the Fill Style section select the **X** fill.

   Notice that the color box updates for contact drawing.

4. Select **Apply** at the bottom of the Display Resource Editor form.

5. Redraw the *fiduciald* design window.

   The *contact dg* layer updates.

6. In the Display Resource Editor form, select **File—Save**.

   The Save form appears. You will save the changes to the existing *display.drf* file.

7. In the Files field select **display.drf**.

   Notice that the Selection field updates.

8. Click **OK**.

9. Click **Yes** in the DRE Save_popup dialog box that appears.

10. Select **File—Exit**.

11. Close the *fiduciald layout* design window.

Editing the *display.drf*

You are given a list of layers that you will be using to create your design. One of the layers and packet information is missing from your techfile and *display.drf* file.You will first create the packet and then assign the packet to the new layer.

1. In an xterm enter:

   ```
   vi ~/Layout/display.drf
   ```

2. To find the packet definition in the *display.drf* enter the following, then press **Return**.

   ```
   /drDefinePacket
   ```

3. You will copy the packet named *border* to create your new packet. To find the *border* packet, enter the following and press **Return**.

   ```
   /border
   ```

4. To copy the *border* packet press the **Shift y** key on the *border* line.

5. To paste the copied line press the **p** key.

   You now have two lines that are exactly the same.

6. Move your cursor over the word *border* and press the **x** key until the packet name is gone.

7. Press the **i** key to insert the new packet name. Enter:

   ```
   mypacket
   ```

8. Press the **[Esc]** key to get out of edit mode.

9. To save your changes enter:

   ```
   :wq
   ```

   You have just added a new packet to your *display.drf*. Next you will load the ASCII text *display.drf* file into the Display Resource Editor.

Loading the *display.drf*

1. In the CIW select **Tools—Display Resource Manager**.

2. In the Display Resources Tool form, select **Edit**.

3. In the Display Resources Form select **File—Load**.

4. In the Load form click on *display.drf* in the Files section.

   This adds the *display.drf* to the end of the path in the Selection text field.

5. Click **OK** in the Load form.

Changing the Packet Attributes

After loading in the new packet info, you will change the attributes of the packet, such as the color, linestyle, and fill.

1. In the Display Resource Editor change the *Application* to **DRE**.

   By changing the Application to DRE you only view the packet names. Virtuoso layout tools specifies that you view the layer purpose pair and the packet that is assigned to that layer purpose pair.

2. Scroll through the Packets section until you find *mypacket*.

   Packets are listed alphabetically. All upper case packets are listed first, and then the lower case packets.

3. Click on **mypacket**.

   What colors, stipples, and linestyles are highlighted?

4. Choose a new Fill Color, Outline Color, Stipple, and Line Style.

5. Once you have made the changes, click **Apply** at the bottom of the Display Resource Editor form.

6. To save the changes select **File—Save**.

7. Select *display.drf* from the Files field.

8. Click **OK**.

9. Click **Yes** in the *DRE Save_popup* dialog box.

10. To close the Display Resource Editor, select **File—Exit**.

Summary

In this lab, you did the following:

- Created a new packet.
- Learned how to change the attributes of the new packet.
- Learned how to load the *display.drf* into the Display Resources Editor.

Lab 1-9 Creating a Layer Purpose Pair

Objective: Create a new layer purpose pair and set valid layers.

You will learn how to add a new layer purpose pair and then assign a packet to it. You will be using the same packet that you created in the last lab.

Adding a Layer Purpose Pair

1. From the CIW, select **Tools—Technology File Manager**.

2. In the Technology File Toolbox select **Edit Layers**.

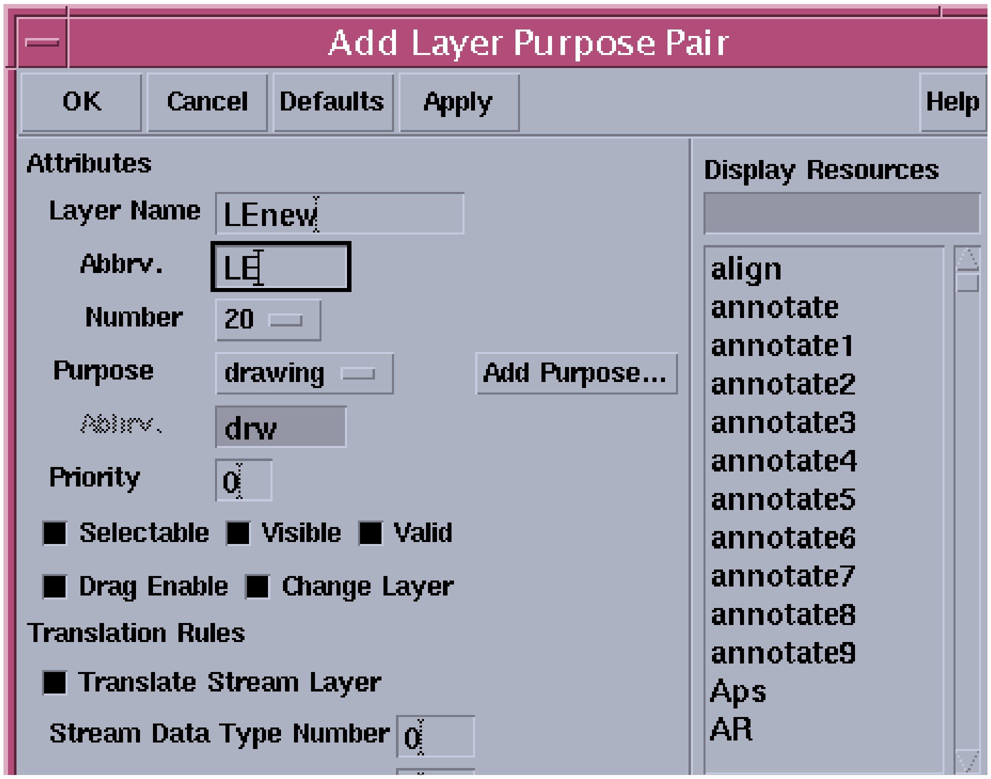
3. In the Layer Purpose Pair Editor make sure that the Technology Library is set to **design**.

4. In the Filter section, click on **Both**.

   The *User* and *System* layers are now visible.

5. In the Layer Purpose Pairs section, select **Add**.

   The Add Layer Purpose Pair form appears.



Attributes

Layer Name LEnew

Abbrv. LE

Number 20

Purpose drawing

Abbrv. drw

Priority 0

■ Selectable ■ Visible ■ Valid

■ Drag Enable ■ Change Layer

Translation Rules

■ Translate Stream Layer

Stream Data Type Number 0

6. In the Layer Name field type in:

   `LEnew`

7. In the Abbrv. field type in:

   `LE`

8. Set the Number to **20**.

9. Set the Purpose to **drawing**.

10. Notice the priority of the *LEnew* layer. By default it is set to 0. You will change this in the next portion of this lab.

    Note: *The priority of a layer controls the order in which applications display the layers. This way you can specify which routing layers appear on top of other routing layers.*

11. Select *mypacket* from the Display Resources section.

12. Click **OK** in the Add Layer Purpose Pair form.

    Do you see your new layer?

Swapping Layer Priorities

You can switch the priority and relative position of two layers in the Layer Purpose Pair Editor form.

1. In the Layer Purpose Pair Editor, with the left mouse button, click to select the *LE new* layer you just created.

2. With the middle mouse button, select the very last layer in the form: *text*.

3. Click the **Move** button.

   The new layer is now placed below layer *text*.

4. Click on the *LEnew* layer and press the **Edit** button. The priority of the layer has been changed.

   The priority numbers assigned to the pairs are updated in virtual memory to reflect the new order.

   Note: *If you have specified in the technology file a subclass called leLswLayers you can change the order of the layers in the LSW with out effecting the priority of the layers. If you do not define leLswLayers, the layout editor displays layers in priority order as specified in the techlayerPurpose Priorities subclass of the Layer Definitions class of the technology file.*

5. Click the **Save** button and then **OK** in the Layer Purpose Pair Editor form.

6. Close the forms and design windows. Save data when requested.

Finding the Layer Purpose Pair

The LSW displays only those layers in your technology file that are defined as valid. These are the only layers you can use to create shapes. System layers are typically not displayed in the LSW.

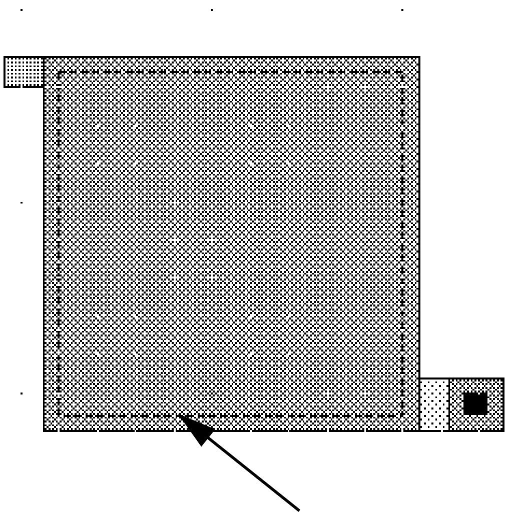
There is a shape in the capacitor layout cellview that is drawn on a layer that is not specified in the LSW, therefore the shape is not selectable. The layer will be made selectable by adding it to the LSW.

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`capacitor`
View Name	`layout`

The *capacitor layout* cellview opens for editing.

2. Notice the rectangle drawn within the capacitor is not selectable.



Try selecting the dashed shape

You are not able to select the shape because the layer is not specified in the LSW.

3. In the LSW, select **Edit—Set Valid Layers**.

The Set Valid Layer form displays all the layers defined in your technology file. The toggle button to the right of each layer indicates whether the layer is valid or invalid. If the button is on (filled in) the layer is valid and appears in the LSW.

4. In the Set Valid Layer form turn on *All Valid* and click **OK**.

   Making all layers valid displays all layers from the Technology File in the LSW.

5. Select the rectangle shape.

6. To find out what layer purpose pair the shape has been drawn on select **Edit—Properties [q]**.

   The layer that the shape is drawn on is *thinox dg* layer.

7. **Cancel** the Edit Properties form.

8. Select **Edit—Select—Deselect All [^d]**.

Adding the Layer to the LSW

1. In the LSW select **Edit—Load**.

2. Turn on Load From Techfile and click **OK**.

   The Load From Techfile option will only load the layers that are defined as valid from the technology file into the LSW.

3. Click **OK** in the dialog box.

4. In the LSW select **Edit—Set Valid Layers**.

5. Turn on the toggle box next to the layer purpose pair *thinox dg* and click **OK**.

   Can you select the shape now?

6. Close the design window.

Summary

In this lab, you did the following:

- Learned how to find and add a layer purpose pair to the LSW.

Lab 1-10 Editing the Technology File

Objective: Create and edit a writable ASCII technology file.

The technology data includes layer definitions, device definitions, design rules, design application rules, display parameters, and plotter parameters—all of the information that defines the framework for creating designs.

Creating an ASCII file from a Binary Technology File.

You can dump all or a portion of a technology file. In this lab you will dump the Layer Definitions class of the technology file to create the layer purpose pair textually.

1. Move your cursor into the CIW and select **Tools—Technology File Manager—Dump**

2. In the Dump Technology File form change the Technology Library to **design**.

   The technology file can reside inside of a single library, or you can have multiple libraries attached to one technology library. We are using the first example in class.

3. Turn on **layerDefinitions**.

   Note: *You only want to dump the classes that you will be editing. The technology file can be large. If you only dump a small portion of the file and you make a mistake while editing it, you can find your error with less effort then going through the entire techfile.*

4. In the ASCII Technology File text field enter:

   ```
   layerdef.tf
   ```

5. Click **OK**.

   An xterm window appears with the Layer Definition class of the technology file.

Viewing Sections of the Layer Definition Class

1. To view the user defined purposes enter:

   ```
   /techPurposes
   ```

   These purposes are defined by the user. In class we are using the system purpose called drawing.

2. To view the user and system defined layers enter:

   ```
   /techLayers
   ```

   The user defined layers are displayed first.

3. To view how layers are prioritized enter:

   ```
   /techLayerPurposePriorities
   ```

   The layers listed first are displayed first. Layers that are listed later are displayed on top of layer purpose pairs already displayed.

4. To view the layer purpose pair and its attributes enter:

   ```
   /techDisplays
   ```

   In the *techDisplays* section you define the layer purpose pair and the packet that it uses. Other attributes such as visibility and selectability are also set.

   You also specify if the layer purpose pair is valid. Validity specifies if the layer purpose pair will show up in the LSW.

5. When you are done viewing the layer definitions class of the technology file, close the xterm window by entering:

   ```
   :q!
   ```

Editing the *symContactDevice* in the Technology File

All *Contact Types* are defined textually in the technology file under the *devices* class. The contact definition specifies the layers that make up the contact, the size of the contact, and the enclosure of the layers around the contact.

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`nor3`
View Name	`layout`

2. Click **OK** in the Open File form.

3. Select **Create—Contact [o]**.

4. In the Create Contact form change the Contact Type to **PTAP**.

5. Move your cursor back into the design window and click at **X=9.8**, **Y=0.8**.

   Does this contact seem a little large?

   This contact exceeds the design rules. You will have to edit the technology file to change the size of this contact.

Editing the Contact Definition in the Technology File

You are going to edit the PTAP contact that you have just placed in your design. You will change the enclosure of *pdiff* around contact to 0.5 micron, and the *metal1* enclosure around contact to be 0.4 micron.

1. Select **Tools—Technology File—Dump**

2. Change the Technology Library to **design**.

3. Turn off **layerDefinitions** and turn on **devices**.

4. In the ASCII Technology File text field enter:

   ```
   device.tf
   ```

5. Click **OK**.

   An xterm window appears with the Device class of the technology file, which contains the definitions of all the symbolic devices used in the design library.

Editing the Contact Definition

1. To search for the PTAP contact enter:

   ```
   /PTAP
   ```

   The editor takes you to the section where the contacts are defined. What you see are the values for the arguments used to define a contact.

   The syntax for creating a contact is:

```
SymContactDevice(
(name viaLayer viaPurpose layer1 purpose1 layer2 purpose2
w l (row column xPitch yPitch xBias yBias) encByLayer1
encByLayer2 legalRegion)
```

Note: *A PTAP contact contains the contact layer enclosed by two layers: pdiff and metal1.*

```
(PTAP contact drawing pdiff drawing metal1 drawing
0.6 0.6 (1 1 1.2 1.2 center center) 2.0 0.2 (inside pwell
drawing))
```

The encByLayer1 maps back to the first layer that is defined. In this example it is pdiff.

The encByLayer2 maps back to the second layer that is defined. In this example it is metal1.

2. Change the *encByLayer1* value to 0.5.

3. Change the *encByLayer2* value to 0.4.

4. To save the file type:

   ```
   :wq!
   ```

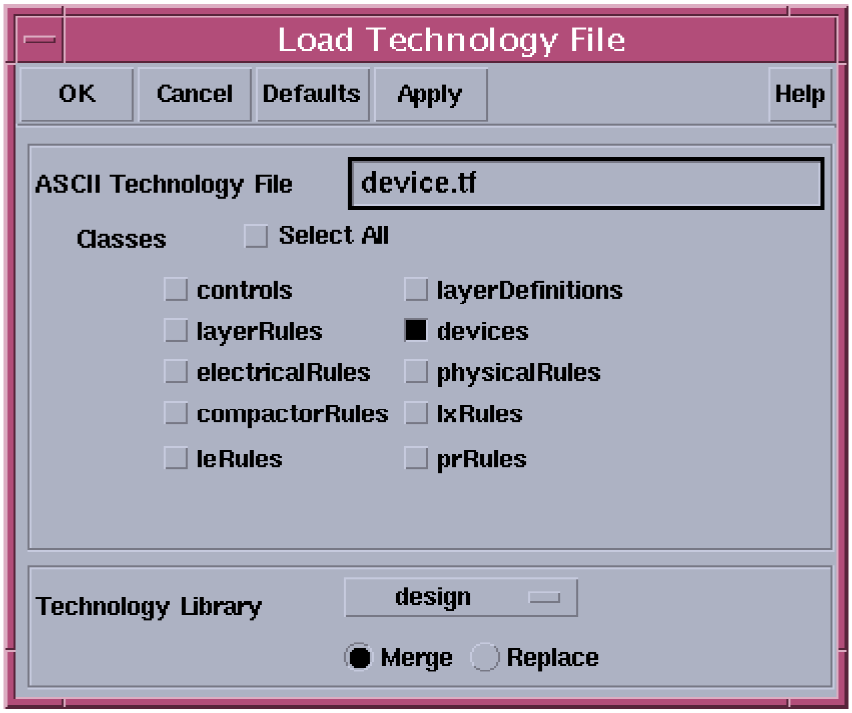
   Your edits to the *device.tf* file are saved.

Loading the Edited Technology File

You will load the technology file that you just edited into the design library's technology file.

1. Select **Tools—Technology File—Load**.

2. Change the following in the Load Technology File form and click **OK**.



ASCII Technology File: device.tf

Classes ☐ Select All

☐ controls ☐ layerDefinitions
☐ layerRules ■ devices
☐ electricalRules ☐ physicalRules
☐ compactorRules ☐ lxRules
☐ leRules ☐ prRules

Technology Library: design

● Merge ○ Replace

The CIW reports:

```
Technology file device.tf compiled
successfully.
```

3. Redraw the *nor3* cell to view the results of the edits.

4. Select **Tools—Technology File—Save**.

   Until you save the technology file changes, the edits are stored in virtual memory.

5. In the Save Technology File form change the Technology Library to **design** and click **OK**.

   The Technology Save form appears asking if it is OK to save this technology file to disk. Click **OK**.

6. Redraw the *nor3* cellview **Window—Redraw [^r]**.

   The PTAP contact is now DRC correct.

7. Select **Design—Save** and **Window—Close** the *nor3* cellview.

Technology File Rule Tables

Rule tables allow entry of conditional rule sets. You will create a new library and view the table entries.

1. Open the *Technology File Manager* by clicking **CIW—Tools—Technology File Manager**

2. Click on **New** in the *Technology File Tool Box* form.

3. In the **New Technology Library** form , enter *mpu2Lib* in the **Technology Library Name** field.

4. Enter skill/mpu2.tf in the Load ASCII Technology File field.

5. Confirm the Load ASCII Technology File field boolean is enabled

6. Click OK .

7. The load Technology File form changes to display the compile results with a message TEchnology file skill/mpu2.tf loaded successfully.

8. Close the popup form.

Review the New Library Technology Table Entries

1. Click on **Technology File Tool Box—Edit Rules**

2. In the *Technology File Edit Rules* form, set **Technology Library** to *mpu2Lib*

3. Select **Physical** under the *classes* section of the form.

4. Select **File—Edit**

5. The *Technology File Physical Rules* form appears.

6. Click 1 Layer Table to select the 1 Layer Table rules.

7. Click the Rule cyclic field and see the name for each table rule in the 1 Layer Table.

8. Set the **Rule** field to *ACCURRENTDENSITY RMS*

9. Click **Edit table** in the lower right of the form.

10. The **Technology File Edit Physical Rules Table** form appears

11. Click on the top Table Entry.

12. Notice the fields **Index1, Index2, Value Type and Value**

13. For a given *Frequency* and *Width* there is a *Value* associated by the table entry.

14. Click your way down the list

15. Notice how *Frequency* stays the same but the *Widths* and their corresponding *Values* change.

16. Close the Technology File forms.

Summary

In this lab, you did the following:

- Learned how to dump a class of the technology file.
- Learned how to edit a *symContactDevice* in the technology file.
- Viewed layer rule table entries

Lab 1-11 Display Options

Objective: Learn to use the settings on the Display Options form to change the window environment variables.

Displaying Detail or Outlines of Instances

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`tutorial`
View Name	`layout`

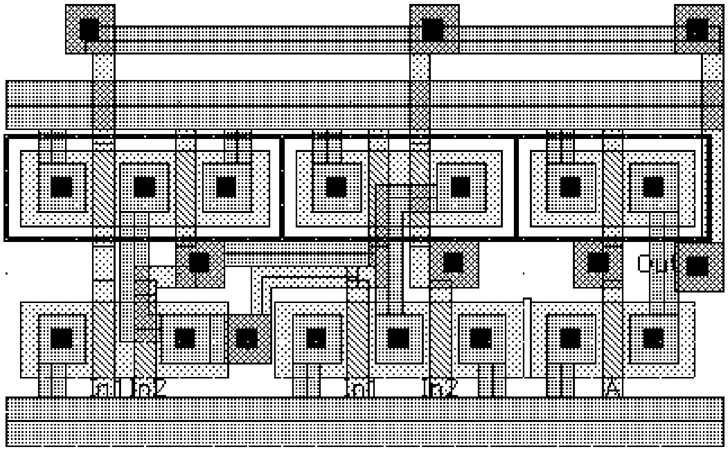
The *tutorial layout* cellview opens for editing.

2. Select **Options—Display [e]** from the menu banner.

The Display Options form appears.

Display Levels controls how much detail inside a cell instance is viewed. *From* shows the lowest level of hierarchy that appears. *To* shows the highest level that appears. The current level is 0, and by default is shown. The highest level displayed is 32.

Note: *The [^f] bindkey sets the display levels to From 0 and To 0. The [F] binkey sets the display levels at From 0 and To 32.*

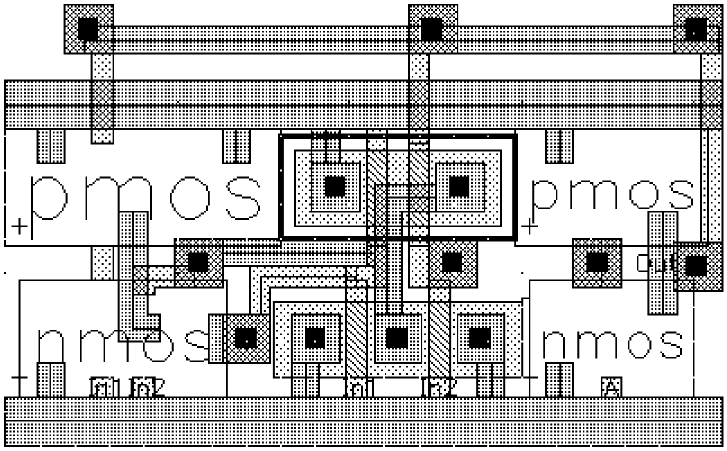


3. Change the Display Levels to:

   ```
   From: 0
   To: 1
   ```

4. Click **Apply**.

   The cellviews *pmos* and *nmos* are at level 2, therefore only the outline of the instance is displayed. All the detail of *2nor* is displayed because it is at level 1.

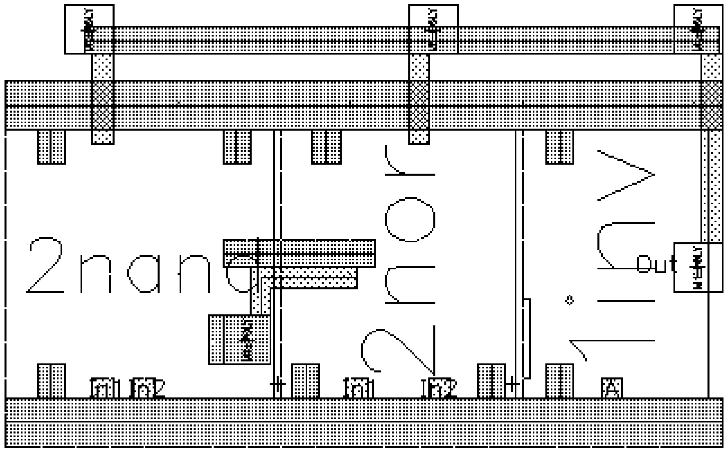


5. Change the Display Levels to:

   ```
   From: 0
   To: 0
   ```

6. Click **Apply**.

   The cellviews *2nand*, *2nor* and *inv* are at level 1. Only the outline of these instance are displayed.



Setting Which Design Objects Appear

1. In the Display Options form, change the Display Stop Levels to:

   ```
   From: 0
   To: 32
   ```

2. Turn on the **Axes**, **Nets**, and **Label Origins**.

3. Click **Apply**.

   Notice the changes in the design window. The X and Y axes has appeared. The Nets *In1* and *In2* have flight lines showing the pin connections and the label origins are now visible.

4. Turn off **Pin Names** and **Path Borders**.

5. Click **Apply**.

   The pin names are no longer visible and the paths borders have disappeared with only the center lines showing.

Setting the Filter Size and Style

The filter size and style can effect the amount of time it takes for a design window to redraw. The smaller the number in the filter size, the more objects are displayed.

Note: *Turn the **Path Borders** back on and click **Apply**.*

1. Under Filter, change the **Size** to **15** and the **Style** to **empty**.

2. Click **Apply**.

3. Redraw the design window **[^r]**.

   Less detail is displayed. Try Zooming out and see how the filter allows even fewer objects to be displayed.

4. **Cancel** the Display Options form.

5. Close the *tutorial* cellview

Using Area Display

1. Open design amplifier layout
2. Click **Ctrl-f** to set the display levels to top level only.
3. Select **Window—Area Display—Set**

   The **Set Area Display** popup form appears.
4. Set *Display Levels From 0 To 32*
5. You are prompted to *Point to the first corner of the view area*
6. Enter two individual coordinates surrounding the top left two PNP devices to view their hierarchy (such as X 0 Y 99 and X 37 and Y 81).

   Note: *Click each corner to describe the area for display.*
7. The data within the area you described is now available for editing.
8. Try this on the lower right *spcres* device.
9. Use the bindKey *shift f* to view all the cellview levels.
10. Use the bindKey *control f* to hide all the levels.
11. Notice your area displays have been retained.
12. Select **Window—Area Display—Delete All**
13. Close the design windows.

Summary

In this lab, you did the following:

- Learned how to set the Display Levels.
- Learned how to change the setting for view objects.
- Learned how to change the filter size and style
- Learned how to use Area Display.

Lab 1-12 Editor Options

Objective: Learn about settings on the Editor Options form.

Setting Gravity

1. Open the *nor3 layout* cellview from the *design* library.

2. Select **Options—Layout Editor [E]** from the menu banner.

   The Editor Options form is displayed. Notice that **Gravity** is turned off. By default gravity is turned on, but for classroom purposes, gravity is off.

3. Move the cursor into the layout window and notice how the cursor snaps to the 0.1 grid that is set.

4. In the Editor Options form, set Gravity On and change the Aperture to **1.0**.

5. Click **Apply**.

   The aperture controls how far your cursor can snap between objects. If centerline were the type selected, the cursor would snap to a path centerline within a 1 Micron radius of the cursor position.

6. Move the cursor to the layout window and notice that it now snaps only to the edges of objects (unless you move the cursor too far away from the object.)

7. In the Layout Editor Options form, click **none** under Gravity Controls.

8. Select **centerline** then click **Apply**.

9. Move the cursor into the layout window and notice how the cursor snaps only to the centerline of the path.

10. In the form, click **vertex** under Gravity Control, then click **Apply**.

11. Move the cursor into the layout window and notice that the cursor now snaps to the centerline of the path and to the vertices of the other objects.

12. Select **none**.

13. Select **pin**, then click **Apply**.

    Notice that the cursor only snaps to the *A*, *B,* and *C* pins.

14. Select **none**, and then set **midpoint** and **nexus** on.

15. Click **Apply**.

    Midpoint snaps the middle of an edge of an object.

16. In the Editor Options form, set Gravity On to **off** then click **OK**.

17. Close the design window.

Setting Conic Sides

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`sample_npn`
View Name	`layout`

2. Select **Options—Layout Editor [E]**.

   Notice that the default for **Conic Sides** is 10. This means when a conic (circle, donut, or ellipse) is converted to a polygon, it is created as a polygon with 10 sides.

3. In the layout window, click on the green ndiff layer circle to select it.

4. Select **Edit—Other—Convert To Polygon**.

   The circle is converted to a polygon with 10 sides.

5. Select **Edit—Undo [u]** to undo the conversion.

6. In the Editor Options form, change the number of Conic Sides to 4, then click **Apply**.

7. In the layout window, select **Edit—Other—Convert To Polygon**.

   The circle is converted to a polygon with four sides (a square).

8. Select **Window—Close** to close the cellview. You don't need to save the changes.

Summary

In this lab, you did the following:

- Learned how to use Gravity.
- Learned how to change the setting for Conic Sides.

End of Lab

Labs for Module 2

Layout Editor Basics

Lab 2-1 Viewing Your Designs

Objective: Learn different ways to view your design.

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`pk44chip`
View Name	`layout`

Using the Pan Command to Move Around Your Design

1. To pan the design, select **Window—Pan** or press the **[Tab]** key. The CIW prompts you:

```
Point at the center of the desired display:
```

2. Click on the *in1* pad.

   The display of the cellview changes so that the point you chose appears at the center of the window.

3. To repeat the **Pan** command press the right mouse button.

   What is the prompt in the CIW?

4. Click with the left mouse button on the *vcap* pad.

   Note: *If the vcap pad is not visible, click on the far right side of the design window.*

5. To move back to the *in1* pad select **Window—Utilities— Previous View [w]**.

6. Select **Window—Fit All [f]** to fit the design to the window.

Moving Around Using the Zoom Command

1. Select **Window—Zoom—Out by 2 [Z]**.

   Scales the current window by 0.5.

2. To return to the original cellview display size, select **Window—Zoom—In by 2 [^z]**.

   Scales the current window by 2.0.

3. To zoom in a specified amount, select **Window—Zoom—In [z]**.

   The CIW prompts:

   ```
   Enter the first corner of the box you wish to
   enlarge.
   ```

4. Click on the upper-left corner of the area you want to display.

   The CIW prompts:

   ```
   Enter the other corner of the box you wish to
   enlarge.
   ```

5. Click on the lower-right corner of the area you want to display.

6. Close the design window.

Summary

In this lab, you did the following:

- Learned to use the **Pan** command.
- Learned to use the **Zoom** commands.

Lab 2-2 Selecting Objects

Objective: Learn different ways of selecting objects.

Using the Mouse to Select Objects

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`select_obj`
View Name	`layout`

The *select_obj layout* cellview opens for editing.

2. Select **Options—Display** and turn on **Dynamic Highlighting and click OK**.

Move the cursor over the shapes in the design. Notice that the shapes are highlighted with a dotted line. This is called Dynamic Highlighting.

3. Click with the left mouse button to select any object.

To show that the shape is selected, the system highlights the shape in bright white.

Notice the number of selected objects in the window banner *(F)Select:1*

(F) indicates full select mode. Full select mode only lets you select the entire object(s). You cannot select an edge(s).

4. Use the **[^d]** bindkey to deselect all objects.

The shape is now deselected and the white highlighting is removed.

How many objects are shown as being selected in the window banner?

Choosing Between Objects

1. Click to select one of the contacts of the inverter.

2. To select the *metal1* shape drawn around the contacts, click at the same location.

   How many objects can you select at one time by just clicking?

   You can only select one object at a time by clicking on an object.

3. Use the **[^d]** key to deselect the metal1 shape.

Adding or Subtracting Objects to the Select List

1. Select one of the contacts of the inverter.

2. To select another contact, press and hold the **Shift** key and click the left mouse button.

   You can continue to select objects one at a time using this method.

3. To remove an object from the select set, press and hold **Control** and click the left mouse button on a selected object.

Using the Drag Box to Select Objects

1. To select a group of objects, press and hold the left mouse button and drag the cursor to draw a selection box around the group.

   All previously selected objects are deselected. All objects fully inside the selection box are selected.

   How would you deselect these objects?

Using Partial Selection

1. To select a single vertex, press the **F4** key and click on the corner of a green *ndiff* layer shape.

   The **F4** bindkey toggles between partial and full selection. The window banner shows *(P)Select:1.*

2. To select an edge, click on the edge of the *ndiff* layer rectangle.

3. To select a group of edges only, draw a selection box around parts of any objects.

4. Deselect the objects.

Making Layers Selectable Using the LSW

1. With the left mouse button, click **NS** (none selectable) in the LSW.

2. Try to select any layer.

   Only the resistor instances are selectable. The names of all layers including the drawing layer are grayed out to indicate that they are not selectable.

3. Press **Control [f]**.

4. Turn off the **Inst** button in the LSW.

5. Try to select one of the resistor instances.

   The only object selectable is the *A* pin on the inverter.

6. Turn off the **Pin** button in the LSW.

7. Try to select the *A* pin on the inverter.

   None of the layers, instance or pins are selectable.

8. With the <u>right</u> mouse button, click on the name of any layer to make that layer selectable.

   If you click the same layer again with the <u>right</u> mouse button, it becomes unselectable again.

9. With the left mouse button, click **AS** (All Selectable) and turn on the **Inst** and **Pin** buttons.

   All layers are selectable again, and their names are no longer grayed out.

Making One Layer Selectable

1. Move your cursor into the LSW and **Shift** click with the right mouse button on the *metal1 dg* layer.

   Notice that all the layers are unselectable except for *metal1*.

2. To make another layer selectable, click with the right mouse button on another layer in the LSW.

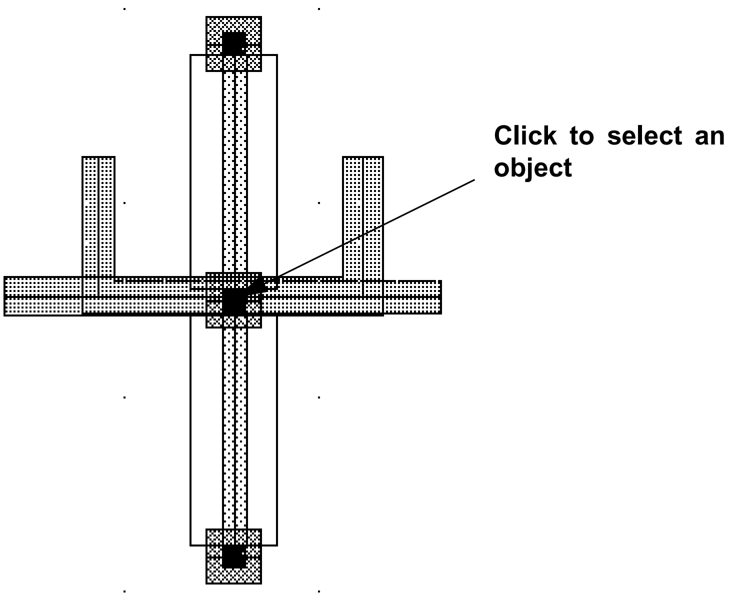
3. Click on **AS** to make all layers selectable

Selecting Objects Under One Selection Point

You can select individual objects that are stacked under one selection point.

1. Press **Shift [f]**.

2. Place you cursor over the center contact of the two overlapping resistor instances and click to select an object.



3. Click again on the same contact.

   You are able to cycle the selection of two objects.

4. With one of the objects still selected, press the **[^y]** bindkey.

   Another object is now selected.

5. Continue to press the **[^y]** key.

   You are now able to cycle the selection of all the objects.

6. Close the design window.

Summary

In this lab, you did the following:

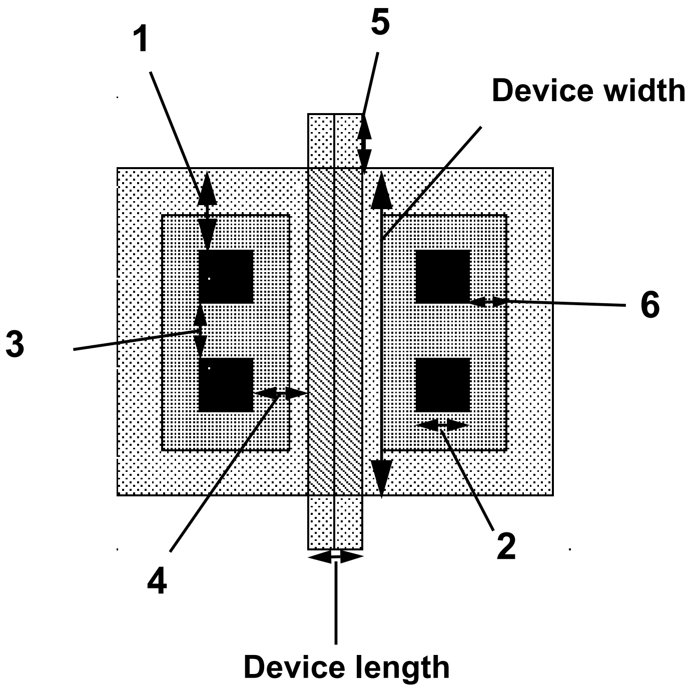
- Learned how to select objects.
- Learned how to deselect objects.
- Learned how to set the LSW for select ability.
- Learned how to find a layer that is not in the LSW.
- Learned how to make a layer valid.

Lab 2-3 Using The Basic Commands

Objective: Use the Create and Edit commands to draw different shapes that make up the nmos and pmos cellviews.

Use the following design rules to create the correct spacial relationships between layers of the *nmos* device.

Design Rules: *nmos*



1. ndiff overlap of contact 0.9u
2. contact minimum width 0.6u
3. contact spacing 0.6u
4. contact to gate spacing 0.6u
5. poly extension 0.6
6. metal overlap of contact 0.4u

Create the *nmos* Cellview

1. In the CIW, select **File** — **Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`nmos`
View Name	`layout`

2. Click **OK**. The design is empty.

3. Referring to the design rules, continue with the following steps to draw the *nmos* cellview. The device size is:

```
width 3.6
length 0.6
```

Note: *If you have difficulty creating the nmos device, there are instructions at the end of this lab to automatically generate the nmos cellview.*

4. Select the *poly drawing* layer as the current layer.

5. Use the **Create—Path [p]** command to create the gate portion of the nmos.

6. Click left to enter the first point of the path.

Note: *Hint: Look at the X and Y coordinates in the top left corner of your design window. Start the path at the coordinates X:0 and Y:0 and end the path at coordinates X:0 and Y:4.8.*

7. To finish the path, press the **Return** key.

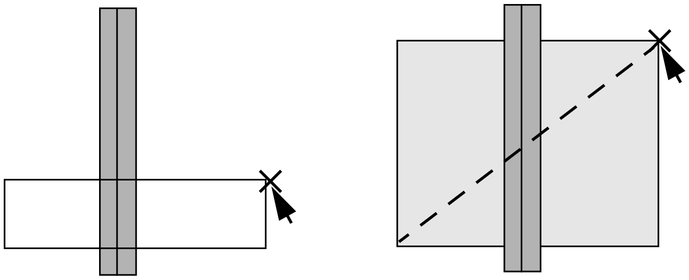
   You can also double-click the mouse to finish the path. To undo a point you have digitized, use the Back Space bindkey.

8. Select the *ndiff drawing* layer as your current drawing layer and use the **Create—Rectangle [r]** command to draw the diffusion area of the nmos.

9. Click left to enter the first corner of the rectangle.

10. Click left on the opposite corner to finish the rectangle.

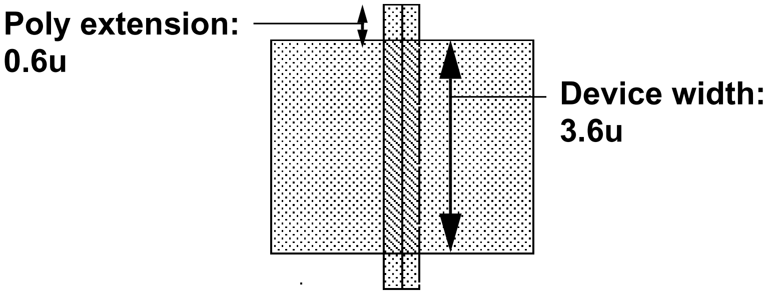
    Do not be concerned with the exact placement or size of the rectangle. You will adjust the size later using the **Stretch** command.



Adjusting the Size of the *ndiff* Rectangle and the *poly* Path

1. Select **Window—Create Ruler [k]**.

    Note: *To remove the rulers, select **Window—Clear All Rulers [K]**.*



Use the **Create Ruler** command to measure the width of the nmos device.

2. Click to enter the starting point of the ruler.

3. Press the **Return** key or click to end the ruler.

    In the Create Ruler Options form **[F3]**, you have the option of using a Multi-segment Ruler, changing the snap mode or keeping the ruler until you delete it using the **[K]** bindkey.

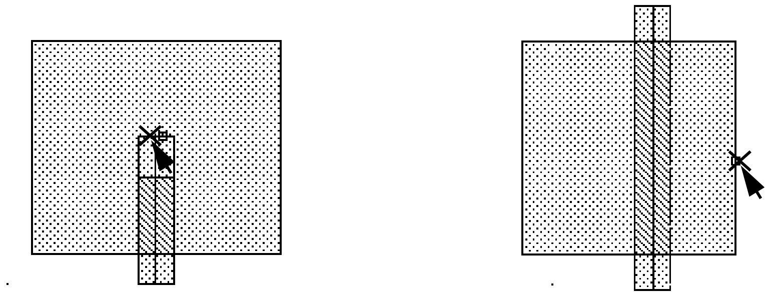
4. Select **Edit—Stretch [s]**.

   When you invoke the command, it automatically invokes the partial **(P)** selection mode.

5. Click on, or area select the object(s) you want to stretch.

   The objects or parts of objects you are stretching will by highlighted with a dashed line.

6. Drag your cursor to the new location and release the mouse button.

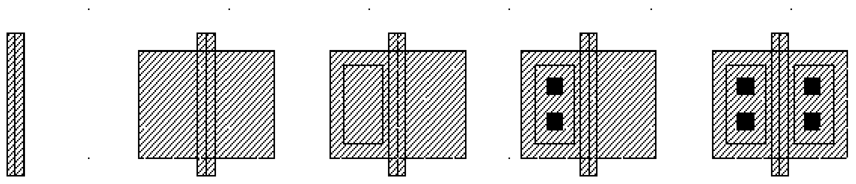


When stretching a path, make sure you have only the mid-line of the path highlighted.

Creating the Source and Drain

1. Draw a *metal1* rectangle for the source connection of the nmos device.

2. Using the contact drawing layer, draw two rectangles for contacts on top of the *metal1* rectangle.



3. Select the *metal1* rectangle and the two contacts.

   For selecting multiple objects, drag a selection box around the objects or select one object and press the **Shift** key, continuing to click on addition objects to add to your selection.

4. Select the **Edit—Copy [c]**.

   You will copy the *metal1* rectangle and the two contacts to create the drain connection of the nmos device.

5. Click left on the highlighted objects.

6. Click left to place the objects.

7. Use the **Ruler** and **Stretch** commands to adjust the size of the nmos device in accordance to the design rule.

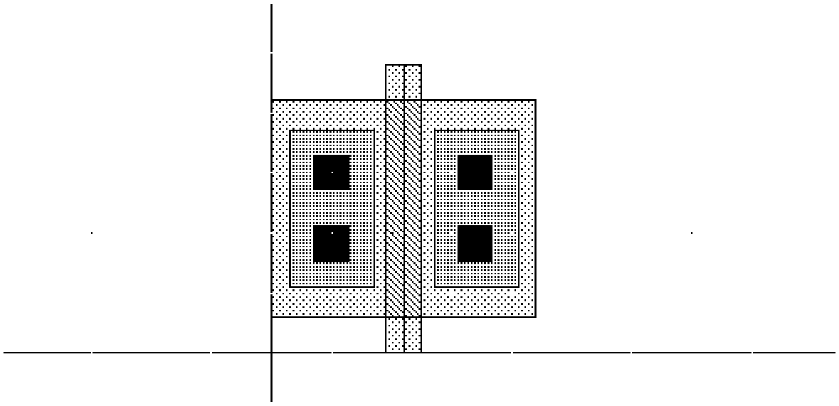
8. Select **Options—Display [e]** and turn on **Axes**.

9. Select the **Edit—Move [m]**.

10. Select all of the objects that make up the *nmos* device, either by using a selection box or by using the **Shift** key to add individual objects to your selection.

11. Once the objects are selected and highlighted, click left on the selected objects.

12. Click left to place and align the selected objects with the X and Y axes.



13. When you finish the *nmos*, select **Design—Save** and close the window.

    Note: *If you had difficulty creating the nmos device, you can automatically generate the device by following these steps:*

a. With the empty *nmos layout* cellview open, type in the CIW window:

```
hiReplayFile("LOG/nmos.log")
```

b. Press the **Return** key.

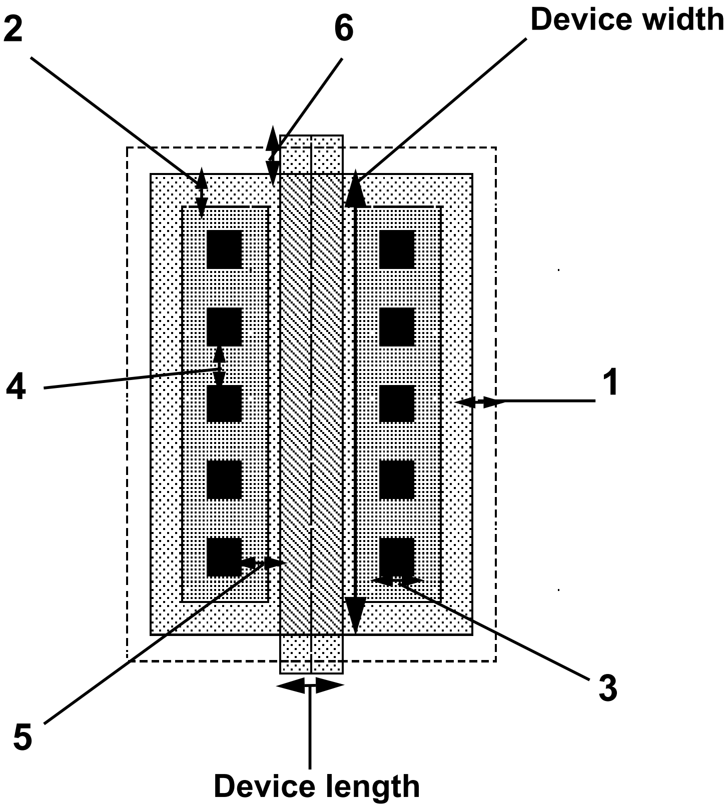
c. Select **Window—Fit All**.

Working with an Existing Cellview

You will copy the *nmos* you just created to a cellview called *pmos* and then change the layers and sizes to make a *pmos* device.

Use the following design rules to create the correct spacial relationships between layers of the *pmos* device.

Design Rules: *pmos*



1. nwell overlap of pdiff 0.4u
2. pdiff overlap of contact 0.9u
3. contact minimum width 0.6u
4. contact spacing 0.6u
5. contact to gate spacing 0.6u
6. poly extension 0.6u
7. metal overlap of contact 0.4u

Copying the *nmos* Cellview to the New *pmos* Cellview

1. From the CIW, select **Tools—Library Manager**.

2. Click on the *design* library and highlight the *nmos* you just created from the Cell section of the Library Manager.

3. Click and hold down the middle mouse button, then select **Copy** from the pull-down menu.

4. In the empty Cell text field of the To section, enter:

   ```
   pmos
   ```

5. Click **OK**.

6. Open the *pmos* cellview from the *design* library.

Creating the *pmos* Cellview

1. Referring to the design rules, continue with the following steps to draw the *pmos* cellview. The device size is:

   ```
   width 7.2
   length 1.0
   ```

   Note: *If you have difficulty creating the pmos device, there are instructions at the end of this lab to automatically generate the pmos cellview.*

2. Select the *ndiff dg* rectangle and select **Edit—Properties [q]**.

   The properties of an object are the user defined information that make up the object. For example; the properties of a rectangle that you can edit are the layer and the X and Y coordinates. The path properties that you can edit are the layer, type of end cap, width and the X and Y coordinates.

3. In the Edit Rectangle Properties form, click on the *ndiff dg* layer name. All the valid layers will appear. Select the *pdiff dg* layer and click **OK**.

4. Select the poly gate path and select **Edit—Properties [q]**. Change the **Width** to 1.0 and click **OK**.

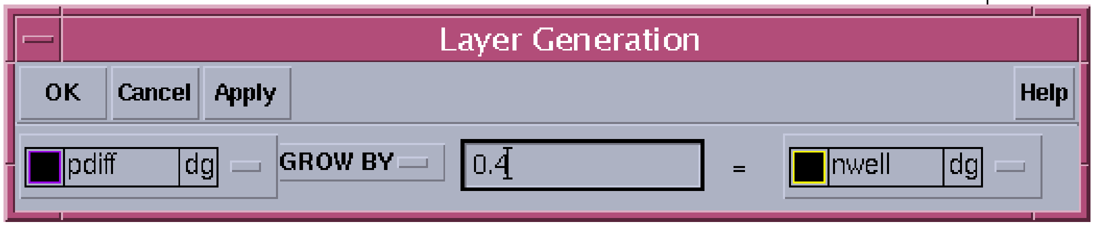
   Note: *The length of the gate is determined by the width of the path.*

5. Select the *pdiff dg* rectangle.

6. Select **Create—Layer Generation.**

7. Change the form as shown and click **OK**.

   This will create a rectangle on layer *nwell dg* that is 0.4 larger than the *pdiff* rectangle.

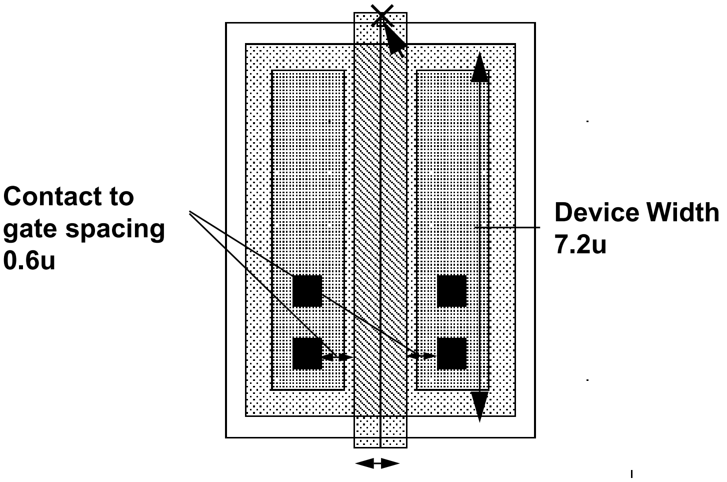


Layer Generation

pdiff dg GROW BY 0.4 = nwell dg

8. In the **Options—Display [e]** form, change Edit Snap Mode to **diagonal** and click **OK**.

9. Stretch the device in the X and Y direction to create a 7.2 width and a 1.0 length pmos.

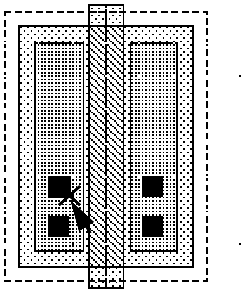


10. Press **[Esc]** to cancel the **Stretch** command.

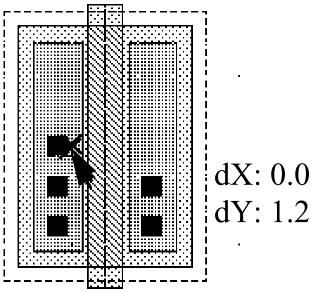
11. To create a flat array of contacts, select the contact on the left and second from the bottom.

12. Select **Edit—Copy [c]**.

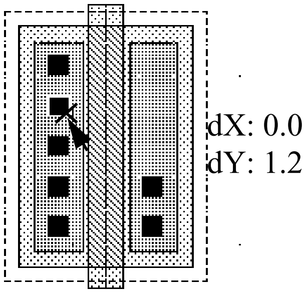
13. In the Copy option form, enter **2** for Rows and **3** for Columns.



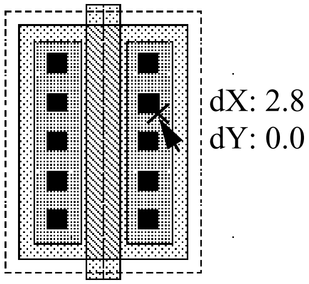
Click the selected contact



Click to place the first contact of the array



Click to place the columns



Click to place the rows

14. Using the dX: and dY: coordinates on the window banner, place the first contact.

    Contact to contact spacing is 0.6 and the contact width is 0.6. The placement of the first contact is a total of 1.2 in the Y direction.

15. Place the contact columns.

    The placement of the second contact is also 1.2 in the Y direction.

16. Place the contact rows.

    Align the contact rows with the existing contacts on the left. The placement is 2.8 in the X direction.

17. When you finish the *pmos*, select **Design—Save** and close the window.

Note: *If you had difficulty creating the pmos device, you can autogenerate the device by follow these steps:*

a. With the empty *pmos layout* cellview open, type in the CIW window:

```
hiReplayFile("LOG/pmos.log")
```

b. Press the **Return** key.

c. Select **Window—Fit All**.

Summary

In this lab, you learned the following commands:

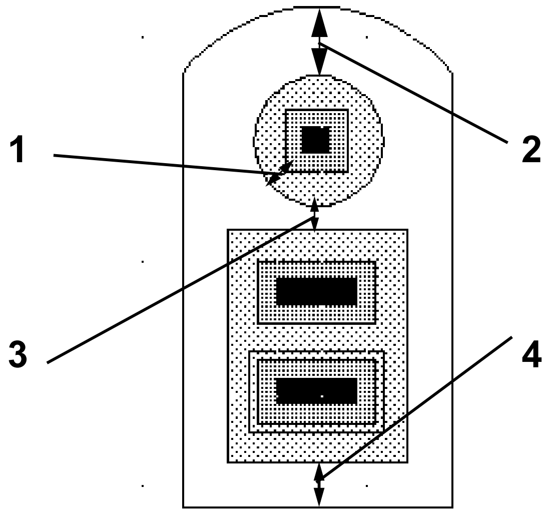
- Rectangle
- Path
- Copy
- Create Ruler
- Stretch
- Move
- Layer Generation

Lab 2-4 Creating Polygons and Circles

Objective: Drawing the collector and buried layer for the npn device using the Create Circle and Create Polygon commands.

Use the following design rules to create the correct device size and the correct spacial relationships between layers of the *npn* device.

Design Rules: *npn*



```
1. ndiff overlap of contact 1.0u
2. buried overlap of ndiff 1.5u
3. ndiff to pbase spacing 0.5u
4. buried overlap of pbase 1.0u
```

Finish Drawing the *npn* Cellview

1. In the CIW, select **File—Open**, and enter the following values in the form:

Library Name	`design`
Cell Name	`npn`
View Name	`layout`

2. Referring to the design rules, continue with the following steps to finish the *npn* cellview.

Using the Create Circle Command to Create the Collector

You will use the **Create Circle** command to create the collector of the npn.

1. Change the current drawing layer to *ndiff dg*.

2. Select **Create—Conics—Circle**.

3. Click for the center placement of the circle, X: 3.0 and Y: 8.0.

4. Look at the dX: and dY: coordinates of your cursor. Move your cursor in the X direction 1.5u.

5. Click to finish the circle. The diameter of the circle should be 3.0u.

6. Select **Options—Layout Editor**.

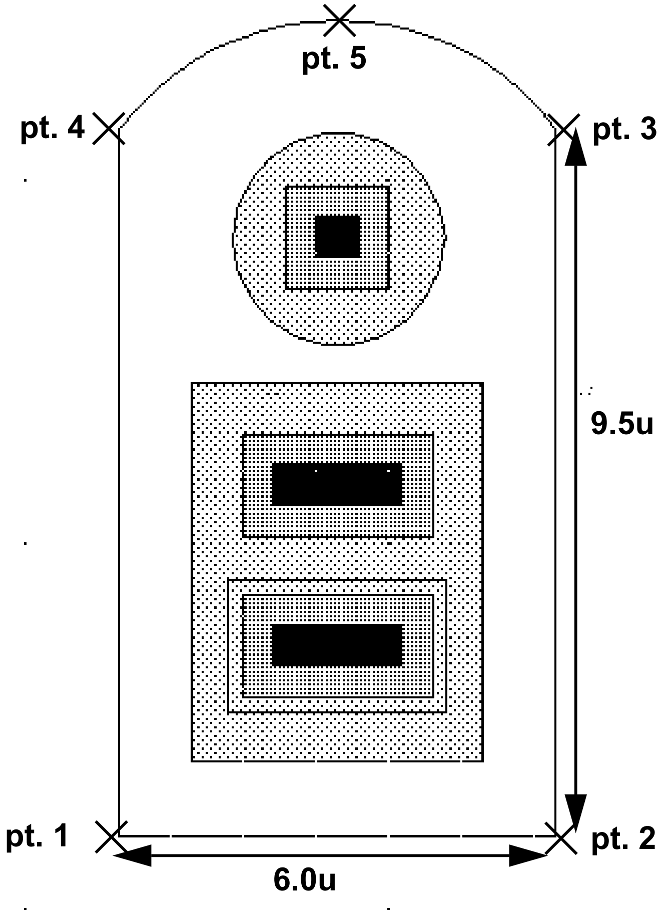
7. Change Conic Sides to **200** and click **OK**.

   Once you convert the circle to a polygon it will have 200 vertices.

8. To change the conic shape to a polygon, draw a drag box around the circle and select **Edit—Other—Convert to Polygon**

Using the Polygon Command

You will use the *Polygon* command to draw the buried layer of the *npn*.



1. Select **Create—Polygon [P]**.
2. Change the current layer to *buried dg*.
3. To start the polygon, click the *first point* in the lower left corner.

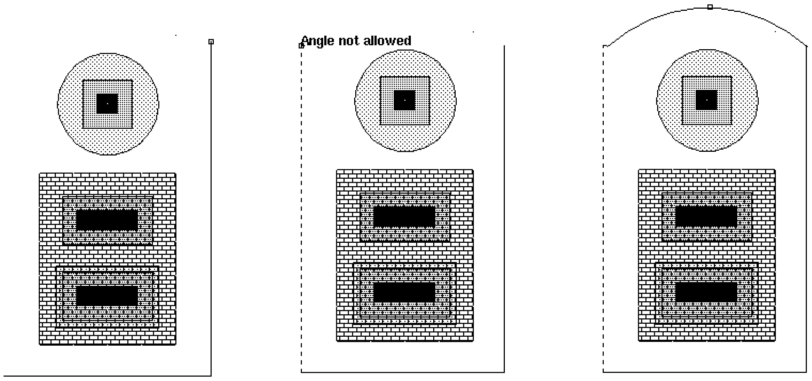
   Buried layer overlap of pbase layer is 1.0u.
4. Click to enter the *second point*, 6.0u in the X direction.

   As you enter each point a dotted line will show how the polygon will close if you stop at that point.
5. Move 9.5u in the Y direction and click to enter the *third point*.

6. Move your cursor into the Create Polygon form and click on **Create Arc**.

7. Move 6.0u in the X direction and click to enter the *fourth point*.

   The cursor will tell you that this angle is not allowed. You can ignore this message for the moment.



8. Center your cursor over the *ndiff* circle (3.0 in the X direction), then move in the Y direction 1.5u. Click the *fifth point* to create the arc.

9. Press the **Return** key to end the polygon.

10. **Save** the *npn* design and close the window.

    Note: *If you had difficulty creating the npn device, you can automatically generate the device by follow these steps:*

    a. With the empty *npn layout* cellview open, type in the CIW window:

    ```
    hiReplayFile("LOG/npn.log")
    ```

    b. Select **Window—Fit All**.

Summary

In this lab, you did the following:

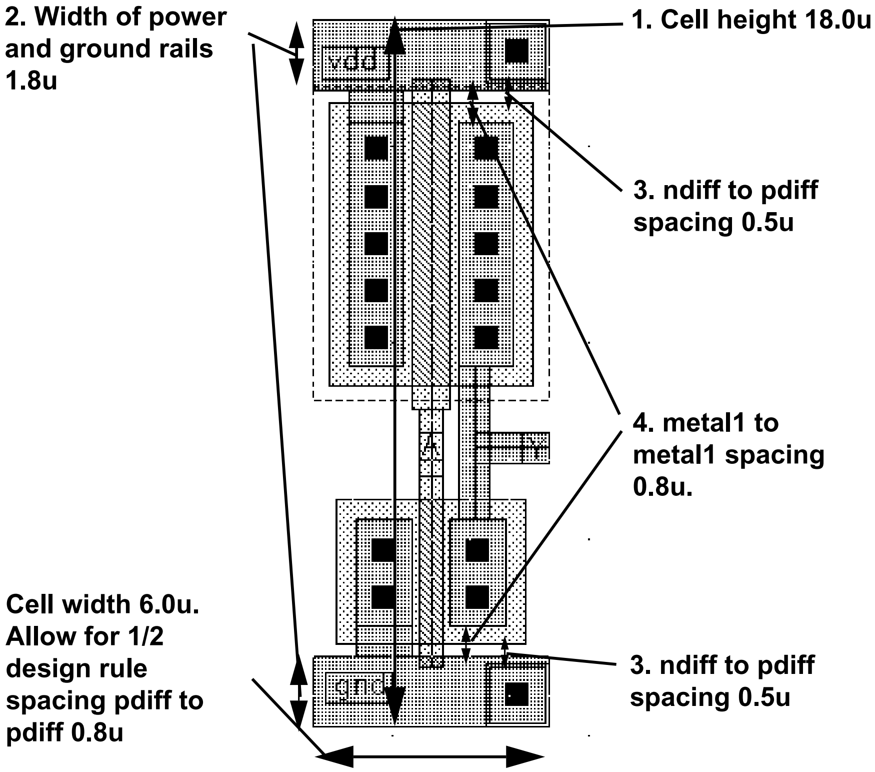
- Learned how to create a circle.
- Learned how to create a polygon.

Lab 2-5 Creating an Inverter

Objective: Use the *nmos* and *pmos* cellviews to create an inverter.

Use the following design rules to create the correct device size and the correct spacial relationships between layers of the *inverter* device.

Design Rules: inverter



2. Width of power and ground rails 1.8u

1. Cell height 18.0u

3. ndiff to pdiff spacing 0.5u

4. metal1 to metal1 spacing 0.8u.

Cell width 6.0u. Allow for 1/2 design rule spacing pdiff to pdiff 0.8u

3. ndiff to pdiff spacing 0.5u

1. Cell height 18.0u
2. Width power and ground 1.8u
3. ndiff to pdiff spacing 0.5u
4. metal1 to metal1 spacing 0.8u
5. metal1 width 0.8u

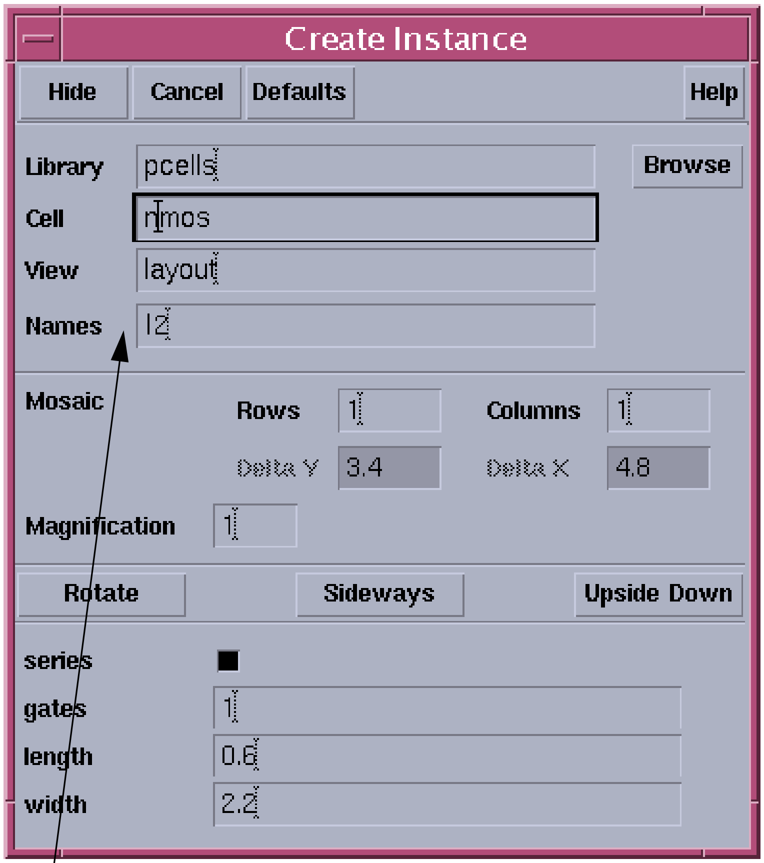
Placing an Instance

The power and ground rails have already been drawn for the inverter. You will place an instance of an *nmos* and *pmos* and draw the interconnect for inverter to finish the cellview.

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`inverter`
View Name	`layout`

2. Select **Create**—**Instance [i]**. Change the following:



Library	pcells
Cell	nmos
View	layout
Names	I2
Rows	1
Columns	1
Delta Y	3.4
Delta X	4.8
Magnification	1
series	■
gates	1
length	0.6
width	2.2

Do not change the instance name. The name of the instance is assigned. Each instance is numbered sequentially as you place it.

3. Click to place the nmos instance in the design window.

Note: *Do not be concerned with the exact placement of the instances. You will arrange them later.*

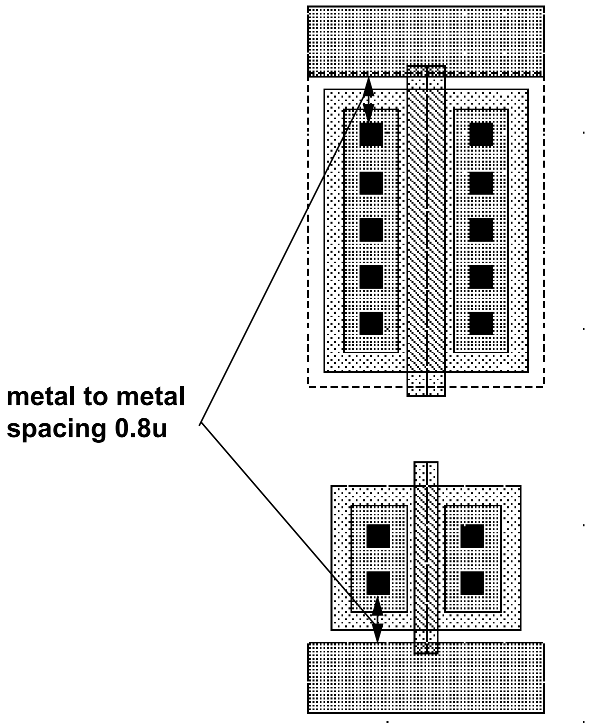
4. In the Create Instance form, change the Cell text field to:

   ```
   pmos
   ```

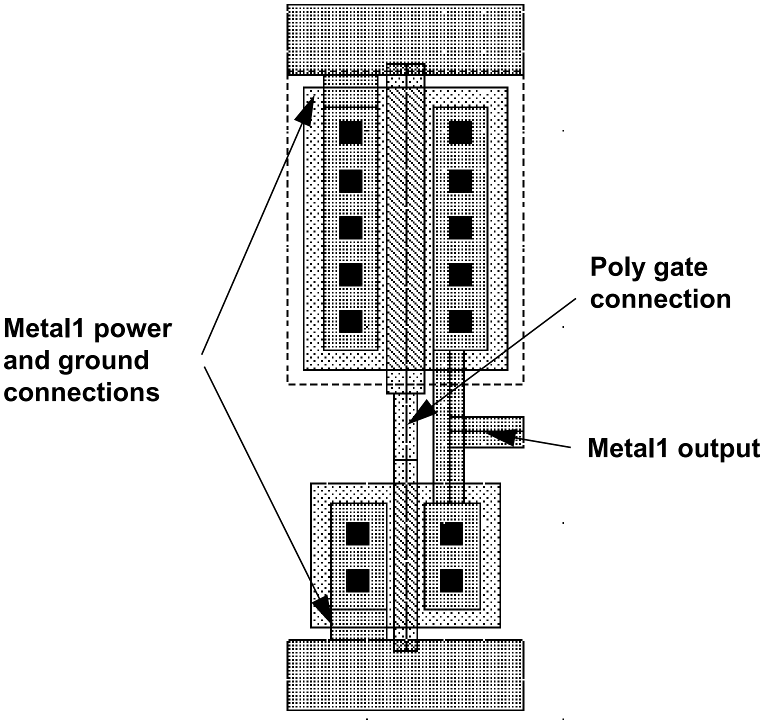
5. Place the *pmos* instance and **Cancel** the Create Instance form.

6. Adjust the placement of the *nmos* and *pmos* instances.

   The metal1 to metal1 spacing will determine the position of the *nmos* and *pmos*.



7. Use the **Create—Path** and change the current layer to *poly dg* to make the gate connection.



8. Use *metal1 dg* to make the output connection of the inverter.

9. Use the **Create—Rectangle** command to create the *metal1* connection for the power and ground rails

Creating Contacts

You will use the **Create Contact** command to place the NTAP and PTAP contacts. These contacts are created textually in the technology file, but placed using the **Create Contact** command.

1. Select **Create—Contact [o]**.

2. In the Create Contact form change the Contact Type to **PTAP**.

3. Place the PTAP on the metal1 ground rail.

   The position the PTAP is based on *ndiff* to *pdiff* spacing.

4. Place a NTAP contact on the power rail and click **OK**.

   The position the NTAP is based on *ndiff* to *pdiff* spacing.

Summary

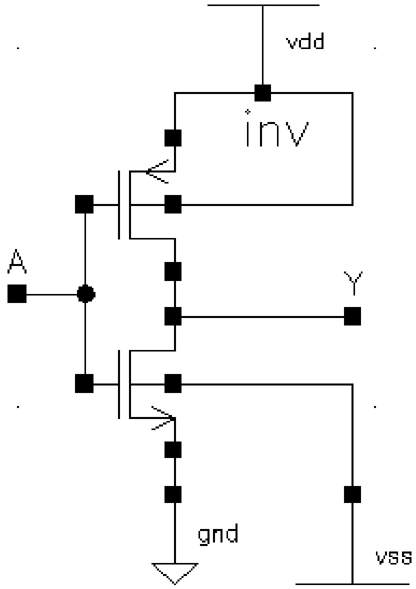
In this lab, you did the following:

- Learned how to place an instance.
- Learned how to place contacts.

Lab 2-6 Creating Pins

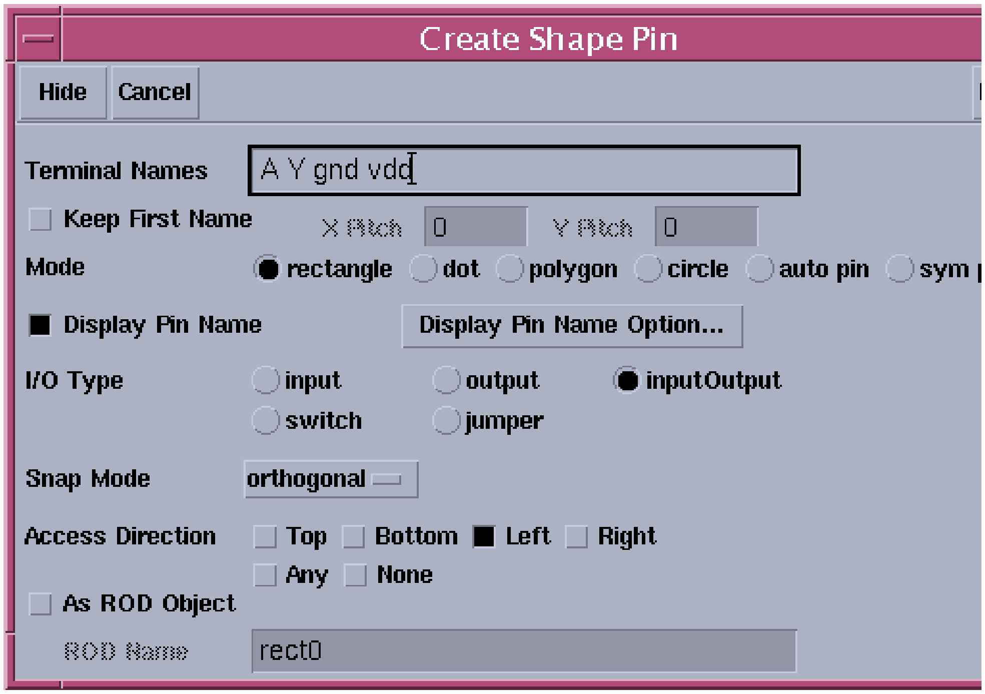
Objective: Place pins on the inverter cellview.

There are pins on the schematic for the input (A), output (Y), power (vdd) and ground (gnd). The corresponding pins must be placed in the layout in order to add the connectivity to run Virtuoso® Custom Router, Assura™ Verification or Diva® LVS, Virtuoso®-XL layout editor, and layout synthesis tools.



1. Select **Create—Pin [^p]**.

2. In the Create Shape Pin form change the following:



Terminal Names: A Y gnd vdd

□ Keep First Name X Pitch: 0 Y Pitch: 0

Mode: ● rectangle ○ dot ○ polygon ○ circle ○ auto pin ○ sym

■ Display Pin Name

I/O Type: ○ input ○ output ● inputOutput ○ switch ○ jumper

Snap Mode: orthogonal

Access Direction: □ Top □ Bottom ■ Left □ Right □ Any □ None

□ As ROD Object

ROD Name: rect0

3. Select *poly pn* layer for the *A* pin.

4. Draw a rectangle shaped pin on the poly gate.

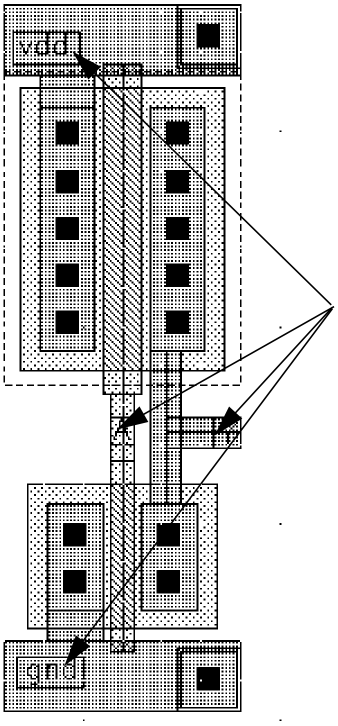
5. Click to place the label on the poly gate.

   The labels automatically appear because you turned the **Display Pin Name** option on in the Create Shape Pin form.

   Notice that there is a dashed line from the label to the edge of the pin. The dashed line denotes that the label is attached to the pin. If you were to move the pin the label would move with the pin.

6. Select *metal1 pn* as the current drawing layer.

7. Place the *Y*, *gnd* and *vdd* pins.



Place the A, Y, vdd and gnd pins

If you did not select **Display Pin Name** from the Create Shape Pin form, you could create pin labels using the **Create Label** command, then use the **Edit Attach** command to attach the labels to the pins.

8. Save your design by selecting **Design—Save**.

9. Close the design window.

Summary

In this lab, you did the following:

- Learned how to create pins.

Lab 2-7 Creating the *control* Block

Objective: Use the Create and Edit commands to create the control block.

You will create the control block using the *inverter* cellview you created in the previous lab and the *nand2* from the *design* library.

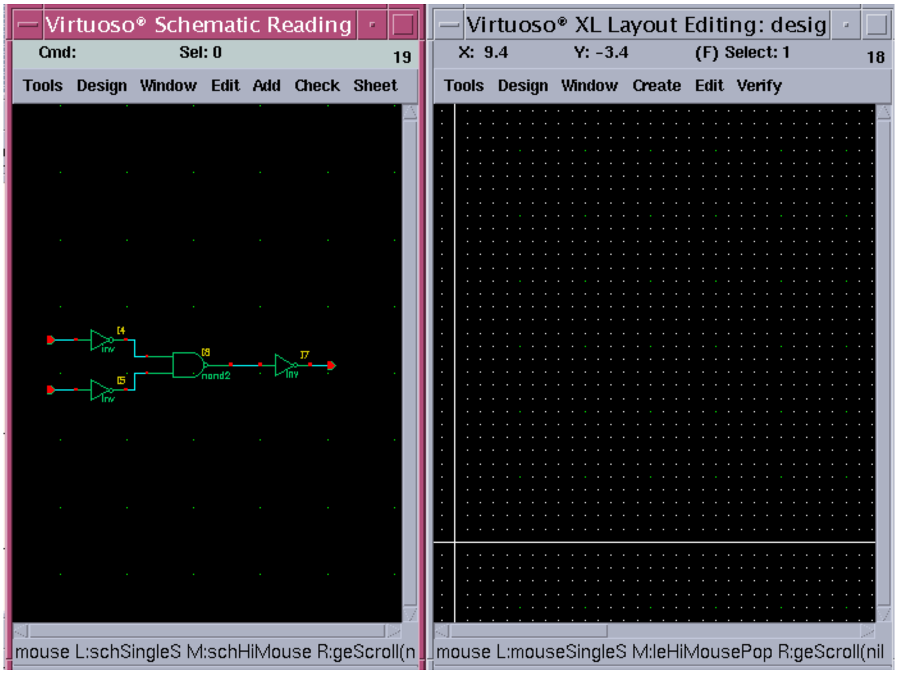
Open the Control Block Layout and Schematic Views

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`control`
View Name	`layout`

2. Open the corresponding schematic cellview named *control schematic*.

3. Resize the windows so that you can view both the schematic and layout window as shown below.



Virtuoso® Schematic Reading
Tools Design Window Edit Add Check Sheet
I4
inv
I5
inv
I8
nand2
I7
inv
Virtuoso® XL Layout Editing: desig
Tools Design Window Create Edit Verify

Cmd: Sel: 0 19

mouse L:schSingleS M:schHiMouse R:geScroll(n

X: 9.4 Y: -3.4 (F) Select: 1 18

mouse L:mouseSingleS M:leHiMousePop R:geScroll(nil

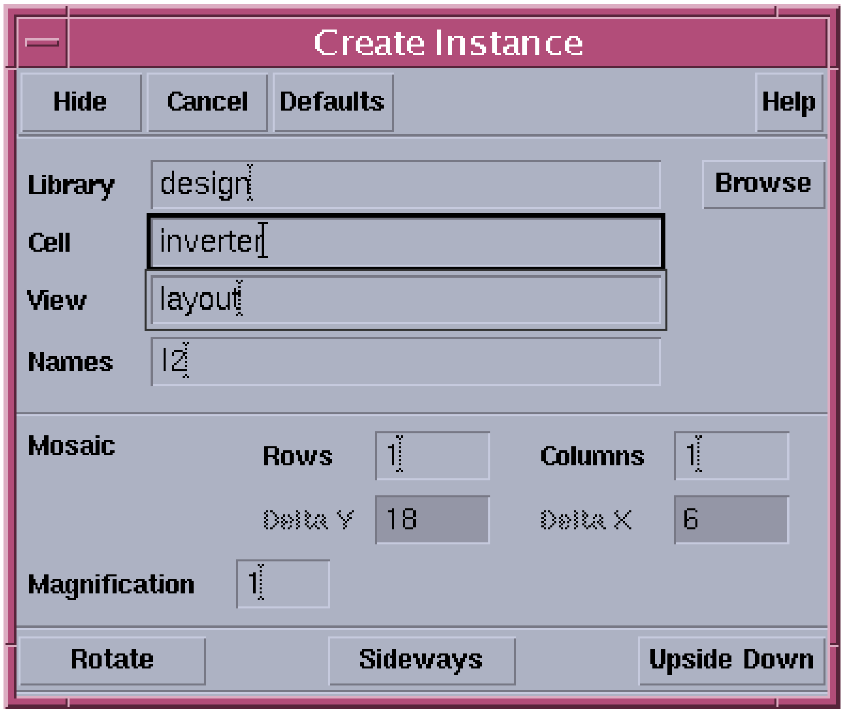
Do you need more room on the screen to work in both windows?

On the User Preference form you can remove the fixed menu, prompt line, and status line from both the schematic and the layout windows.

Placing the *inverter* Instance

Place two instances of the *inverter layout* cellview from the *design* library.

1. Select **Create** — **Instance [i]** and change the form as shown:



Create Instance

Library: design

Cell: inverter

View: layout

Names: I2

Mosaic

Rows: 1 Columns: 1

Delta Y: 18 Delta X: 6

Magnification: 1

2. Place the lower-left corner of the instance at **X=0**, **Y=0**.

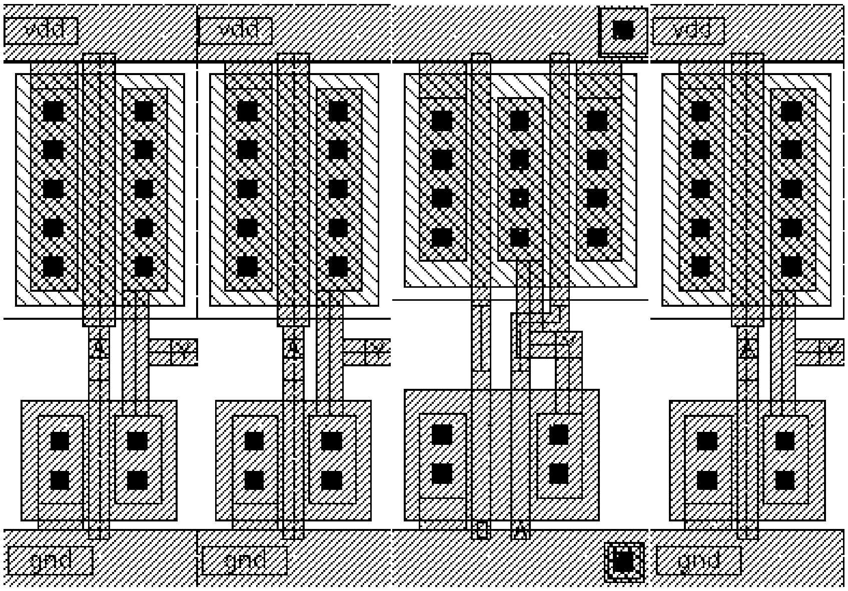
3. Place the next *inverter* instance at **X=6.0**, **Y=0**.

4. In the Create Instance form, change the *Cell* name to **nand2**.

5. Place the lower-left corner of the *nand2* instance at **X=12.0, Y=0.0**.

6. Place the last *inverter* cellview instance at the coordinates **X=20.0, Y=0.0**

   Your placement of cells should look like the following:



7. Press the **[Ecs]** key to cancel out of the **Create Instance** command.

8. Select **Design—Save**.

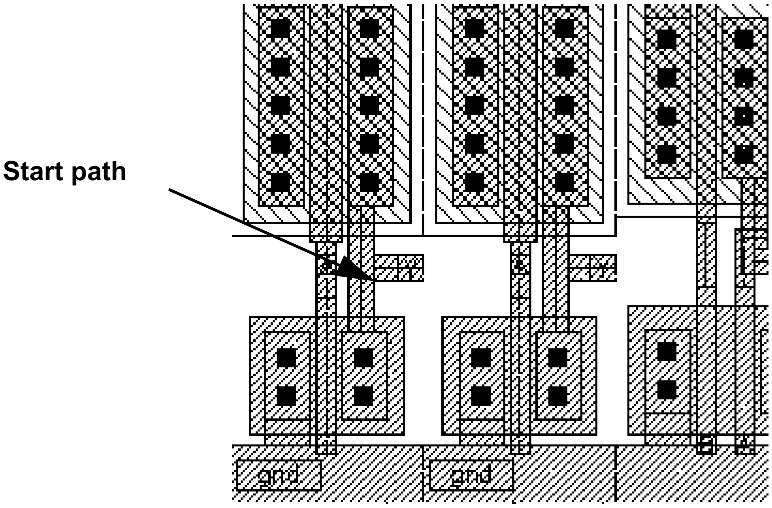
Creating Interconnect Using Path Stitching

You will use Path Stitching to make the connection from the *Y* pin of the first *inverter* instance to the *A* pin the *nand2*.

1. Select *metal1 dg* layer as the current layer.

2. Select **Create—Path [p]**.

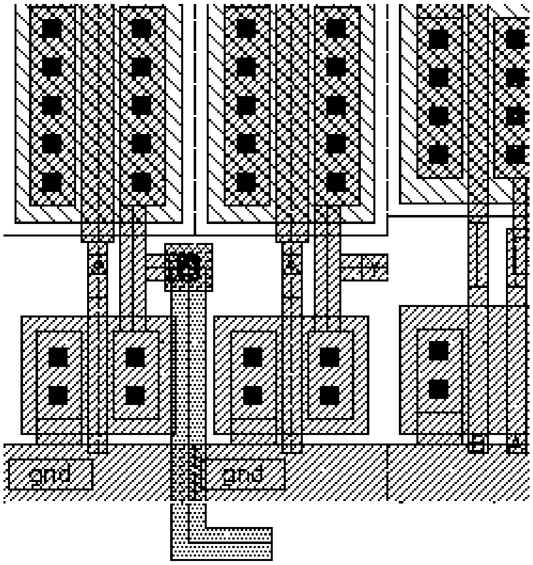
3. When starting the path, click to the right of the *Y* pin of the first *inverter* instance.



4. In the **Create Path** form, use the **Change To Layer** button to change to the *metal2 dg* layer.

5. Place the metal1/metal2 via on the Y pin of the first *inverter* instance.

6. Bring the *metal2* path down and past the *metal1* ground rail of the *inverter* instance and click to instantiate a point.



7. In the Create Path form change back to *metal1 dg* layer.

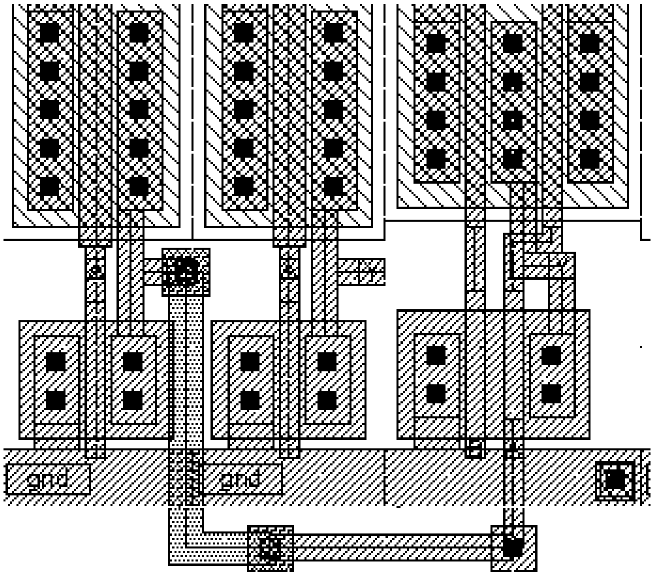
8. Place the metal1/metal2 via.

9. Continue the path in *metal1 dg* layer to the right.

10. In the **Create Path** form, change to layer *poly dg*.

11. Place the poly contact directly below the *A* pin of the *nand2*.

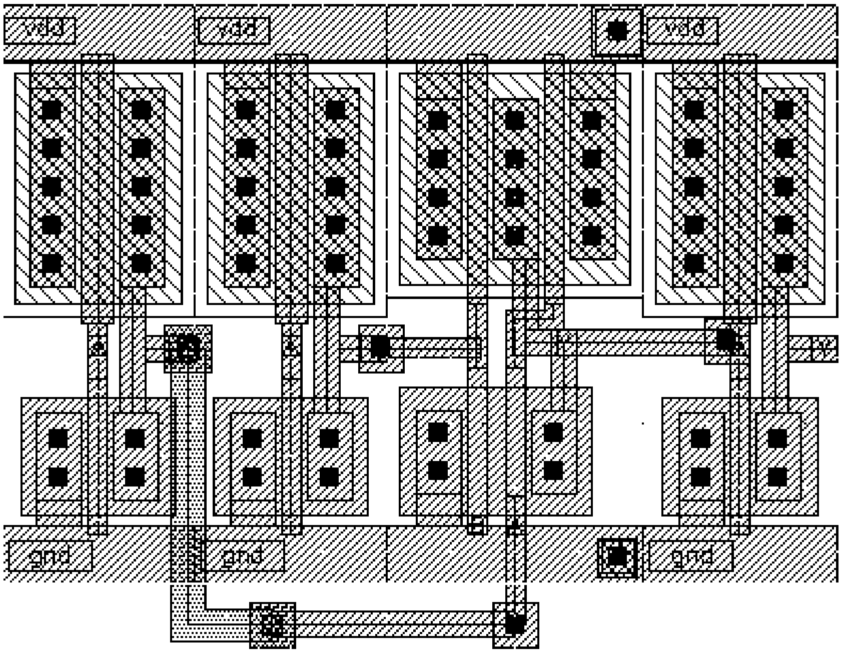
12. Continue the *poly dg* layer path up and overlap the pin of the *A* gate.



13. Press the **Return** key to end the path.

14. Stretch the poly path over *gate B* and press the **Return** key.

15. Wire the cell as shown below using path stitching.



16. Select **Design—Save** and close the design window.

Summary

In this lab, you did the following:

- Learned how use Path Stitching.

Lab 2-8 Editing the *pk44chip* Cellview

Objective: Place the instances inside the *pk44chip* cellview.

Placing the *control* Block

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`pk44chip`
View Name	`layout`

2. Select **Create—Instance [i]**.

3. Enter the following into the Create Instance from.

```
Library design
Cell control
View layout
```

4. To place the instance by typing in the coordinates, type in the CIW:

```
110.4:164.4
```

5. Leave your cursor in the CIW window and press the **Return** key.

6. Place another instance of the *control layout* cellview at **X=83.8**, **Y=164.4**.

   Does this look like a correct placement for the control layout cellview?

   You need to adjust the placement of the control block placed at X: 83.8 and Y: 164.4. You will use the Reference Point option to help you when you move the instance.

7. Press **[Esc]** to cancel the **Create Instance** command.

Displaying the Reference Point

1. Select **Options—Layout Editor**.

2. Turn on the **Display Reference Point** and click **OK**.

   You won't see a reference point yet. It is displayed when you enter a reference point for the **Move** command.

3. Zoom in to the lower-left corner of the control cell.

4. Select the *control layou*t cellview.

5. Select **Edit—Move**.

6. In the Edit Move form change the Snap Mode to **anyAngle**.

7. The CIW prompts you to "Point at the reference point for the move."

8. Click on the selected control block.

   You see the reference point displayed here. Notice that the dX: and dY: in the window banner resets to 0:0. The reference point you entered is now a temporary origin for the move. You might have to move the cursor a little to see the change.

Moving an Object with the Key Pad

1. Press and hold the **Shift** key and use the **left arrow** on your keypad (on the right of your keyboard). Watch the dX: and dY: in the window banner to see exactly how far the cursor moves. Move the instance:

   ```
   dX: -3.0 dY: 0.0
   ```

2. Press the **Return** key to place the instance.

3. To turn off the reference point select **Options—Layout Editor**.

4. Turn off **Display Reference Point** and click **OK**.

Placing the *amplifier* Block

1. Place an instance of the *amplifier layout* cellview outside of the cell boundary.

   Next, you will rotate the amplifier layout cellview before you place it at the correct X/Y location.

2. Select the amplifier layout cellview.

3. Select **Edit—Properties [q]**.

4. In the Edit Instance Properties form select **Attribute**.

5. Change the Rotation field to **MX** and click **OK**.

6. Move the amplifier layout cellview into the cell boundary and place the lower-left corner of the cellview at **X=0.5**, **Y=55.7**.

7. Place another instance of the amplifier cellview outside of the cell boundary.

8. Select the instance, press **[q]** to bring up the properties form and change the Rotation field to **R180**. Click **OK**.

9. Place the lower-left corner of the cellview at **X=116.8**, **Y=55.7**.

10. Save the design.

Summary

In this lab, you did the following:

- Learned how to set the Reference point.
- Learned how to rotate a cellview using the Edit Properties form.

Lab 2-9 Additional Editing Commands

Objective: Learn how to use additional editing commands.

1. In the CIW, select **File—Open**, then enter the following values in the form:

Library Name	`design`
Cell Name	`editing`
View Name	`layout`

Using the Split Command

1. Select **Window—Zoom—In [z]**, then zoom in on the box labeled *Split.*

2. Select **Edit—Other—Split [^s]**.

3. Change the Snap Mode to **anyAngle**.

4. Select all of the objects in the *Split* box. You should have six objects selected.

5. The CIW prompts you to point at the first point of the split line. Create the split line by clicking on pts. 1 through 4. Double click on *pt. 4*.

   Segments that can be stretched are highlighted in yellow.

6. The CIW prompts you to point at the reference point for the stretch. Click on one of the contacts and move the cursor up. The highlighted segments stretch and the contacts move.

7. Click to complete the stretch.

8. To deselect the objects, use the **[^d]** bindkey.

9. Fit the window by selecting **Window—Fit All**.

Using Reshape

1. Select **Window—Zoom—In [z]**, then zoom in on the box labeled *Reshape.*

2. Select **Edit—Reshape [R]**.

3. In the Reshape form change the Reshape Type to **rectangle**.

4. With the left mouse button, click on the *green* diffusion rectangle to indicate the object to be reshaped.

   The rectangle is highlighted in yellow.

5. To create the new shape, click on *pt. 1* and finish by clicking on *pt. 2*.

6. Toggle between the shapes by clicking on the right mouse button.

7. When the shape you want to keep (the new shape) is highlighted in yellow, click anywhere in the cellview with the left mouse button.

   The shape you select is converted to a polygon.

8. Next you will reshape a path. Select the U shaped poly path.

   Notice that the Reshape Type has automatically changed to line.

9. Click on *pt. 3* to start the reshape.

10. Click on *pt. 4* and *pt. 5*, then double click on *pt. 5*.

11. Toggle between the shapes with the right mouse button.

12. When the shape you want to keep (the new shape) is highlighted in yellow, click anywhere in the cellview with the left mouse button.

13. To cancel the **Reshape** command, press **[Esc]** key.

14. To view the entire drawing, select **Window—Fit All [f]**.

Deleting Vertices

In this section you modify shapes by deleting vertices.

1. Zoom in on the box labeled Delete Vertex.

2. Press the **F4** key.

   The window banner should show *(P) Select:0* for partial selection mode.

3. Click on *pt. 1* to select the vertex.

   Only the vertex is selected.

4. Select **Edit—Delete** or press the **Delete** key.

5. The selected vertex is deleted and you have reshaped the polygon.

Rotating Shapes

In this section you will rotate a polygon at any angle.

1. Change back to Full selection mode by pressing **F4**.

2. Zoom in on the box labeled *Rotation.*

3. Select the red polygon.

4. Select **Edit—Other—Rotate[O]**.

5. To start the rotation, click with the left mouse button on the selected polygon.

   Notice that as you move your cursor, the Rotation Angle changes in the Rotate form.

6. Move your cursor into the Rotate form and change the Angle Snap To **.1 Degree** and enter **30** in the Rotate Angle field.

7. Leave your cursor in the form and click **Apply**.

   The polygon is automatically rotated at 30 degrees.

8. Click **Cancel**.

Chopping Objects with the Rectangle Option

1. Zoom in on the box labeled Chop Rectangle.

2. Select the blue path.

3. Select **Edit—Other—Chop [C]**.

   The Chop form appears.

4. On the Chop form make sure that the Chop Shape is set to **rectangle**.

5. Click on *pt. 1* and drag the cursor to *pt. 2* to draw the “cutter” shape.

   The path is now cut. The data inside the cutter is deleted because **Remove Chop** is turned on. Notice that the remaining shapes are still paths.

6. Press **[Esc]** key.

Chopping with the Polygon Option

1. Zoom in on the box labeled Chop Polygon.

2. Select the yellow rectangle.

3. Select **Edit—Other—Chop [C].**

4. In the Chop form, set the following:

Chop Shape	`polygon`
Snap Mode	`L90XFirst`

5. Click on *pt. 1* and *pt. 2*.

6. To complete the polygon cutter shape, double click on *pt. 3* with the left mouse button.

   The rectangle is converted to a polygon. Notice that the polygon cutter shape you drew is cut out of the middle of the new polygon you created.

7. Press the **[Esc]** key.

Chopping with the Line Option

1. Zoom in on the box labeled Chop Line.

2. Select the green shape.

3. Select **Edit—Other—Chop [C]**.

4. In the Chop form, set the following:

Chop Shape	`line`
Snap Mode	`diagonal`

5. With the left mouse button, click on points 1-5.

6. To complete the line, double click on *pt. 6* with the left mouse button.

   The shape is cut into two shapes, but nothing is removed because you used a line cutter shape. **Remove Chop** has no effect when you use the line option.

7. To exit the **Chop** command, press **[Esc]** key.

Merging Objects

1. Zoom in on the box labeled *Merge.*

2. Select **Edit—Merge [M].**

3. Press and hold **Shift** and use the left mouse button to select both the polygon and the donut.

   Notice that the objects are merged into one polygon.

4. Deselect the polygon by using the bindkey **Control [d]**.

5. To exit the **Merge** command, press **[Esc]** key.

Attaching Objects

1. Zoom in on the box labeled *Attach.*

2. Select **EditOther—Attach/Detach [v]**.

3. With the left mouse button, click on the green rectangle to select the figure to be attached.

   The green rectangle is highlighted.

4. With the left mouse button, point to the purple rectangle to select the figure to which the green rectangle is to be attached.

   The green rectangle is now a child of the purple rectangle.

5. To verify that the objects are attached, select the purple rectangle and move it.

   Edits applied to the child do not affect the parent.

6. To de-attach the two shapes select **Edit—Other—Attach/Detach**.

7. Click on the green rectangle, then click in an empty area of the design window.

   Try moving the purple rectangle again, and notice that the green rectangle does not move with it.

8. To exit the **Attach** command, press the **[Esc]** key.

Sizing Objects

1. Zoom in on the box labeled *Size.*

2. Select **Edit—Other—Size**.

   The Size form is displayed. A positive number oversizes an object, and a negative number undersizes it.

3. With the left mouse button, click on *pt. 1* and drag your cursor to *pt. 2* to draw a selection box around the green diffusion rectangle and red poly path.

4. In the **Size** form, change the following:

Size Value	`.5`

5. Click **Apply**.

   Both objects are oversized by .5 microns. Notice that the path converts to a polygon.

   You can experiment with various values for oversizing or undersizing these objects.

6. When you are finished, click **Cancel** to close the Size form.

7. To deselect the objects, use the bindkey **[^d]**.

Converting Objects to Polygons

1. Zoom in on the box labeled *Convert to Polygon.*

2. Select **Edit—Other—Convert to Polygon.**

3. With the left mouse button, click on *pt. 1* and drag your cursor to *pt. 2* to draw a selection box around the *path, ellipse*, and *donut* shapes.

   These shapes are now converted to polygons. Notice that the path no longer has a centerline.

4. Press **Escape** to exit the **Convert to Polygon** command.

Making a New Cell

1. Zoom in on the box in the cellview instance, labeled *Make Cell.*

2. Sele**ct Edit—Hierarchy—Make Cell** then enter the following values:

Library Name	`design`
Cell Name	`mycell`
View Name	`layout`

The **Replace Figures** option causes the objects you select to be replaced with the new cell you create.

3. With the left mouse button, click on *pt. 1* and drag your cursor to *pt. 2* to draw a selection rectangle.

All objects in the *Make Cell* box are highlighted.

4. Click **OK** in the Make Cell form.

A new cell named *mycell* is created containing the highlighted objects.

To view the top level, use the **Control [f]** bindkey.

Control [f] is a bindkey that displays level 0 of the design hierarchy.

Flattening Objects

1. Zoom in on the box labeled *Flatten.*

2. Press the **Control [f]** bindkey.

3. Select **Edit—Hierarchy—Flatten** and set the following:

Flatten Mode	`one level`

4. Turn on **Flatten Pcells**.

5. With the left mouse button, select the *inv* instance, then click **Apply**.

   The cell is now flattened up through one level of hierarchy.

6. To flatten the entire cell change the display level to 20 by pressing the **Shift [f]** bindkey.

7. In the Flatten form change the *Flatten Mode* to **displayed levels**.

8. Drag a selection box around the cell.

9. Click **OK**.

   Can you select any of the hard data?

10. To exit the **Flatten** command, press **[Esc] key**.

Modifying a Corner

1. Zoom in on the box labeled *Modify Corner*.

2. Select **Edit—Other—Modify Corner**.

3. Press **F4** to change to partial selection.

4. Click on *pt. 1* to select the vertex.

5. Click **Apply** in the Modify Corner form.

   This modifies the corner to a radial corner.

6. In the Modify Corner form change the Type of Corner to **chamfer**.

7. Click on *pt. 2* to select the vertex.

8. Click **OK** in the Modify Corner form.

   This creates a chamfer cut.

Using Move to Mirror Shapes

1. Zoom in on the box labeled *Mirror*.
2. Select **Edit—Move**.
3. Select the inverter instance.
4. Select **Upside Down** in the Move form.

   The cell mirrors on the X axis.
5. Select **Sideways** in the Move form.

   The cell mirrors on the Y axis.
6. Click to place the instance.
7. Select the instance again.
8. Press the **Shift** bindkey and click the right mouse button.
9. The first click mirrors the instance along the X axis.
10. Click right again and the instance mirrors along the Y axis.
11. Press the **[ESC]** key.

Yank and Paste

1. Zoom in on the box labeled *Yank.*
2. Select both instances.
3. Select **Edit—Other—Yank [y]**.
4. Click and drag a rectangle from *pt. 1* to *pt. 2*.
5. Select **Edit—Other—Paste [Y]**.

6. Click to place the flatten data you copied.

7. Close the design window.

Summary

In this lab, you learned the following commands:

- Split
- Reshape
- Rotate
- Chop
- Size
- Yank and Paste

You learned the following tasks:

- Delete vertices
- Merge objects
- Attach objects
- Convert an object to a polygon
- Make a new cell
- Flatten an object
- Modify corners

End of Lab

Labs for Module 3

Creating Pcells

Lab 3-1 Installing Sample Pcells

Objective: Install the sample pcells from the Cadence software directory.

The Cadence documentation includes a reference guide to installing the sample cells. In your own work environment, you can open this reference guide to find the script used to start the pcell installation process. For classroom purposes you will start the install process by loading a file which has been provided for you.

1. In the CIW, enter;

   ```
   load(“./skill/spcInstall”)
   ```

2. Press the **Return** key to invoke the install script.

3. The Welcome to the Sample Parameterized Cell Installation window will appear. Click on the **Next** button to start the installation process.

   Note: *Use Next to go to the next step. Clicking OK ends the install process.*

4. Select the *spcres* and *spcpnp* devices to download and click **Next**.

5. In the Select Destination Library form, make sure the *RODpcells* library is selected. Click **Next**.

6. The layers shown below are undefined. In the Define Layers form, select the appropriate layers to be used for the sample pcells. For example, the layer *cont* is used in the sample pcells, but in the RODpcells technology file the layer is called *contact*.

`poly`	`poly`
`cont`	`contact`
`metal1`	`metal1`
`buried`	`buried`
`coll`	`ndiff`
`iso`	`isolati`

7. Click **OK** to the Define Layers form.

8. In the Verify Layer Definitions form, click **Next**.

9. In the Define Rule form there are two layers rules that do not have any physical rules defined in the technology file. Add these values to the Rules form.

   ```
   minWidth poly 0.6
   minSpacing poly 0.6
   ```

10. Click **OK** to the Define Rules form.

11. In the Verify Required Rules form, click **Next**.

12. In the Verify Source Directory form, click **Next** to accept the path to the source directory.

13. In the Loading Status form, verify the devices to be loaded:

    spcres

    spcpnp

14. Click **Next** in the Load Status form.

15. In the Initialization Choices form, choose the *libInit* file to be used for loading the pcells into your library and click **Next**.

16. In the Save Technology File form, choose the *RODpcells* library from the cyclic field and click **OK**.

17. Click **Yes** to the prompt, *Save the technology file*.

18. To complete the installation process, click **OK** to the Installation Complete form.

19. Open the Library Manager and verify that the new device masters *spcres* and *spcpnp* exist in the *RODpcells* library.

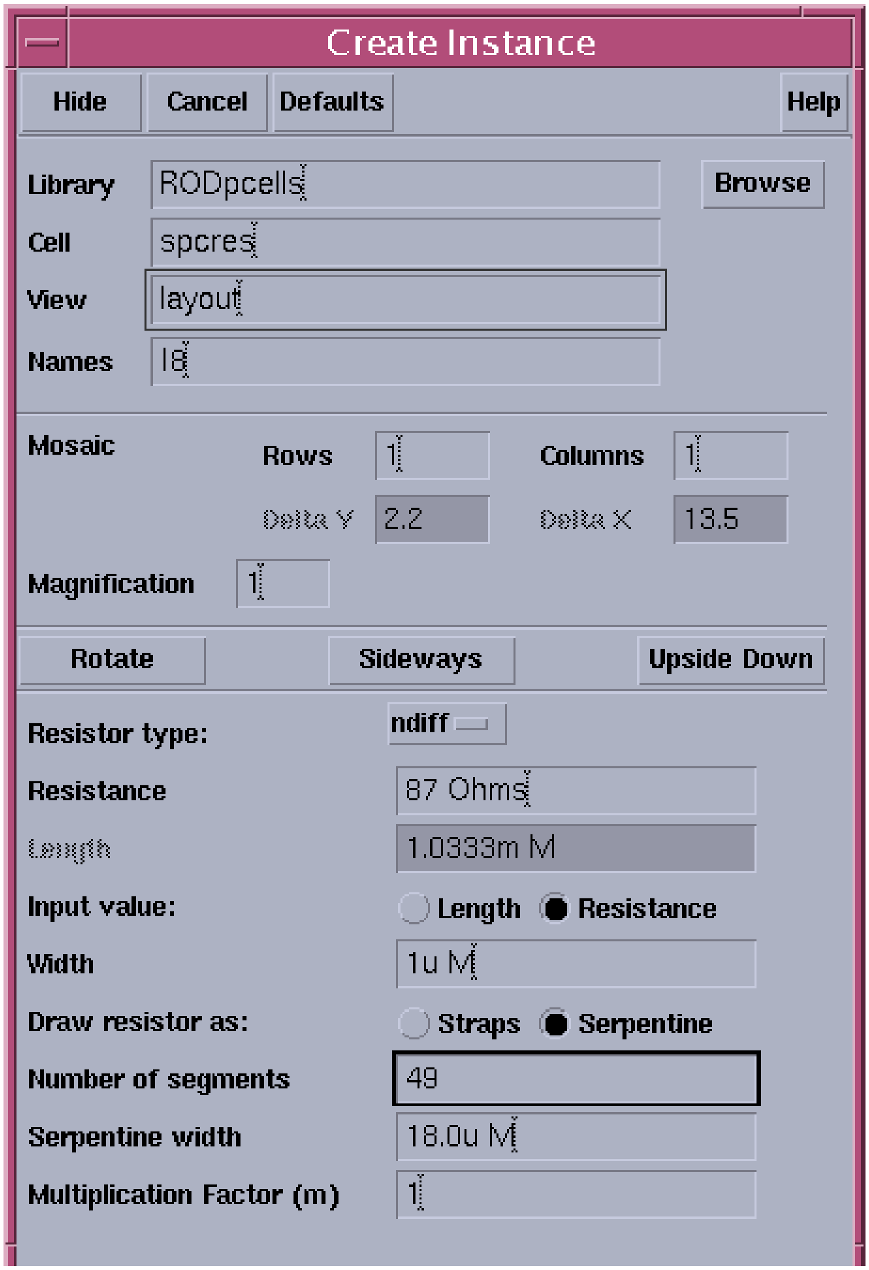
Lab 3-2 Changing the Parameters of Pcells

Objective: Place the *spcres* device from the sample pcells and modify the parameters.

The devices *spcres* and *spcpnp* sample pcells are now located in the *RODpcells* library. You will place the *spcres* device into the *pk44chip* cellview and change the parameters.

1. Open the *pk44chip* cellview if you have closed it from the previous lab.

2. Select **Create—Instance** and change the form as shown.

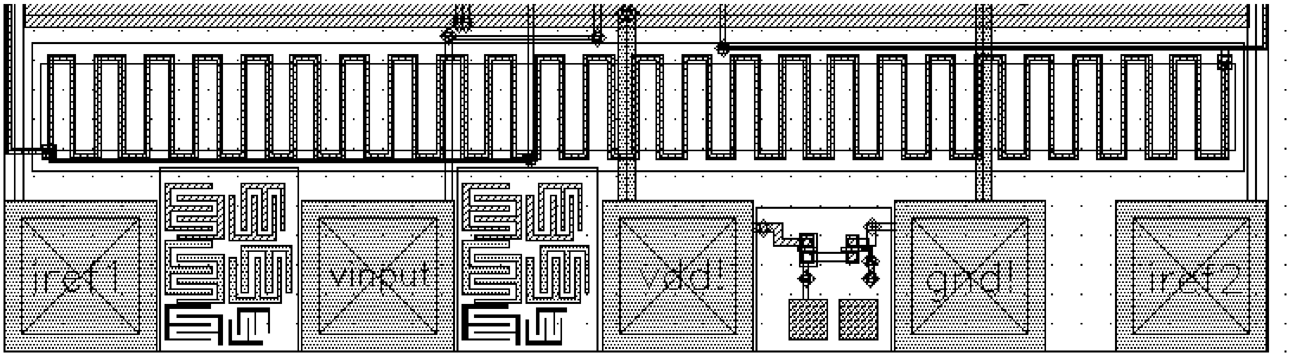


Library	RODpcells
Cell	spcres
View	layout
Names	I8
Mosaic Rows	1
Columns	1
Delta Y	2.2
Delta X	13.5
Magnification	1
Resistor type:	ndiff
Resistance	87 Ohms
Length	1.0333m M
Input value:	○ Length ● Resistance
Width	1u M
Draw resistor as:	○ Straps ● Serpentine
Number of segments	49
Serpentine width	18.0u M
Multiplication Factor (m)	1

Notice how the instance form changes for the *spcres* cellview. Since the *spcres* is a pcell, the form expands to give options to change the Resistor type, Resistance, Width, etc.

3. Click on the **Rotate** and the **Upside Down** buttons in the Create Instance form.

4. Place the *spcres* instance at the coordinates X=211.0, Y=51.0.



5. Press the **Esc** bindkey to cancel the command.

6. Save the *pk44chip* cellview.

Lab 3-3 Stretchable Pcells

Objective: Use the stretch handles assigned to pcells to change the parameters.

A stretch handle is a relative object design (ROD) point handle that is assigned to one or more of the parameters of a pcell. By using the stretch command, you can change the parameters of a pcell. These stretch handles appear as small diamonds on placed pcell instances.

You will place two instances of the *pnp* pcell that you loaded from the sample pcells directory in an earlier lab.

1. Select **Create—Instance** and change the form to the *spcpnp layout* cellview from the *RODpcells* library.

2. Before placing the first instance, click on the **Upside Down** button in the **Create Instance** form.

3. Place the upper-left corner of the first instance at **X=104.0**, **Y=92.0**.

   Note: *You may enter the coordinates in the CIW input area as **104.0:92.0** and then press **Return** to place the instance.*

4. Place the next instance at **X=103.0**, **Y=76.0**.

5. Press the **Esc** key.

6. Select **Options—Display**.

7. In the Options Display form, turn on **Stretch Handles** and click **OK**.

   The cellview updates to show the diamond shaped stretch handles on each device.

8. Select **Edit—Stretch [s]** and click on the stretch handle of the first *pnp* instance you placed. The device highlights and the system displays the value of the parameter next to the upper-right corner of the pcell.

   Note: *You can use **area select** to select the stretch handles.*

9. Stretch the handle to create an emitter width of *3.6*.

   As you drag a handle, the system displays an outline showing how the pcell is changing. This is according to the frequency defined for regenerating the pcell. The system updates the associated parameter, regenerates the pcell, and displays the updated information.

10. Select the stretch handle of the second *pnp* instance and create an emitter width of *5.0*.

11. Save the *pk44chip* cellview.

Lab 3-4 Creating Rectangle Arrays

Objective: Use SKILL to create rectangle arrays within a described area.

1. In the Library Manager, select **File—New—CellView**.

   The Create New File form appears.

2. Enter the following:

Library	`design`
Cell	`skill_test`
View	`layout`

3. Move the cursor out of the form for a moment.

   The Tool field updates to *Virtuoso.*

4. Click **OK**.

5. In the CIW input pane enter:

   ```
   load(“skill/multiPartRect.il”)
   ```

6. Use the bindkey **f** to fit the screen.

   You now see the contact array within the *metal1* rectangle.

7. Press **Control-a** to select all the shapes.

8. Select **Edit—Move**.

   The Move form appears.

9. Enter *20* in the Y area for Delta moves.

10. Click **Apply XY**.

    The array of contacts has been moved up 20 microns.

11. Press **Shift-z** to zoom out so you can see the relative movement of the arrayed shapes.

12. Cancel the Move form.

13. In the CIW enter:

    ```
    load(“skill/fillBBoxWithRects.il”)
    ```

14. Press **Control-f** to fit the screen.

    You now see the array of metals near *0:0* and the contact array at *0:20*.

15. View the SKILL programs you just executed.

16. Close the design window.

Summary

In this lab, you did the following:

- Used *rodCreateRect* to make multipart rectangles with enclosures
- Used *rodFillBBoxWithRects* to create an array of rectangles
- Used a relative *Move* command option

End of Lab

Labs for Module 4

Creating and Editing Designs

Lab 4-1 Using Hierarchy Commands

Objective: Use the Hierarchy commands to edit the *pk44chip* design.

You need to edit the amplifier cellview. There are not enough p-type substrate contacts to meet the design specifications. Also, the resistor and npn need to be adjusted. You will use the **Edit in Place** command to correct the amplifier layout cellview.

1. Open the *pk44chip layout* cellview in the *design* library if you have closed it from the previous lab.

Viewing the Design Hierarchy

Before you edit the design take a look at the hierarchy of the design.

1. Select **Design—Hierarchy—Tree**.

   The form is set to Current to bottom. This will give you a hierarchy listing from the top level all the way to the bottom of the design.

2. Click **OK**.

   A text window displays the listing of the complete hierarchy of the design. Any cell name that is indented is placed in the cell above. The number in parenthesis is how many times the cell is placed in the instance.

3. To close the text window, select **File—Close Window**.

Using the Mark Net Command

The **Mark Net** command extracts the metal and via layer information from the technology file and traces a net through the hierarchy without having to use a schematic.

1. Select **Connectivity—Mark Net**.

2. A prompt will appear in the CIW.

   ```
   Point at a net.
   ```

3. Click on the *metal2* of the *vss!* pad at the top-right of the design window.

   The *vss!* net highlights throughout the hierarchy. Notice there is only one p-type contact on this *vss!* net. You will add additional PTAP contacts in the next section of the lab.

4. Press the **[Esc]** key to remove the highlight.

Editing in Place

In the *pk44chip* design there are two instances of the *amplifier layout* cellview. You need to edit only one of the instances.

1. Select **Design—Hierarchy—Edit in Place [x]**.

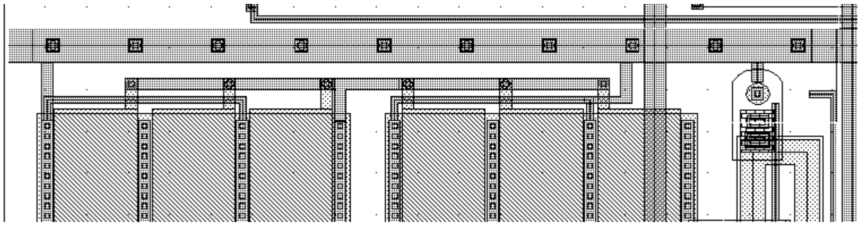
2. The CIW prompts you to "Point at a shape in the cellview to be edited-in-place." Click on either the *metal1* power or ground rail of the amplifier cell.

   Note: *The amplifier cellview contains pcells. If you attempt to* ***Edit in Place*** *a pcell you will get the error message, "Cannot edit-in-place a parameterized cell."*

   Does the window banner change?

   The window banner should read *Virtuoso Layout Editing: design amplifier layout*.

3. Use the **Create—Contact** command to add substrate contacts (**PTAP**) approximately every *10u* as shown below.



4. To view only the amplifier cell that you are editing, and no data around that cell, select **Options—Display**.

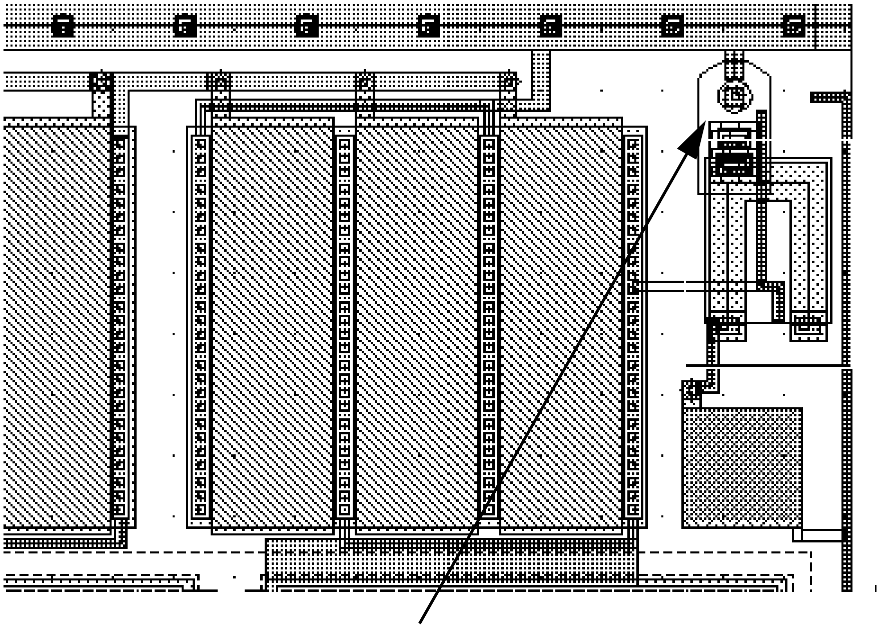
5. Turn off **EIP Surround** and click **OK**.

   Do you see any data around the amplifier layout cellview?

Rotating the *npn* Device

The *npn* device has been placed at an incorrect orientation in the amplifier cell. You will change the orientation of the *npn*.

1. Select the *npn* instance as shown below.



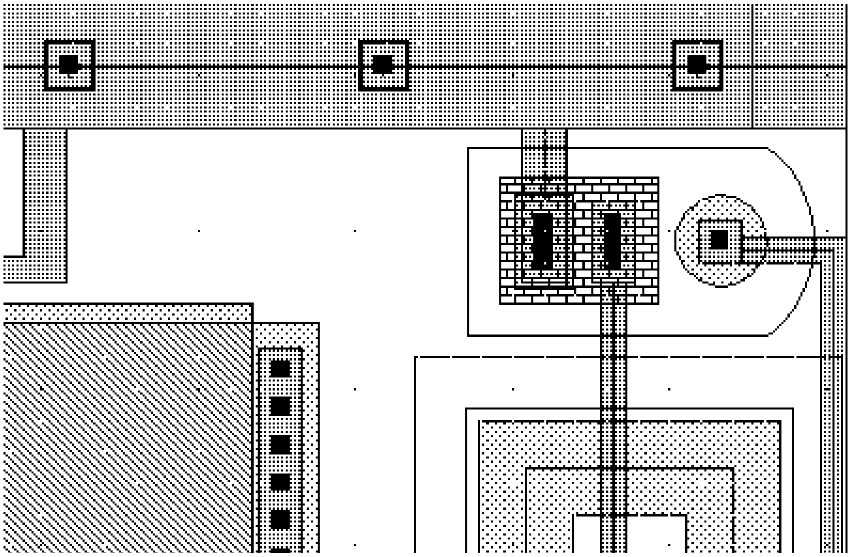
Select the npn instance

2. Select **Edit—Move [m]**.

   You cannot rotate instances using the **Edit Rotate** command, because instances can only be placed orthogonally. To rotate instances use the **Edit Move** or the **Edit Properties** command.

3. Click with the left mouse button on the selected *npn* instance.

4. With your right mouse button click on the instance until you have rotated the *npn* as shown below.



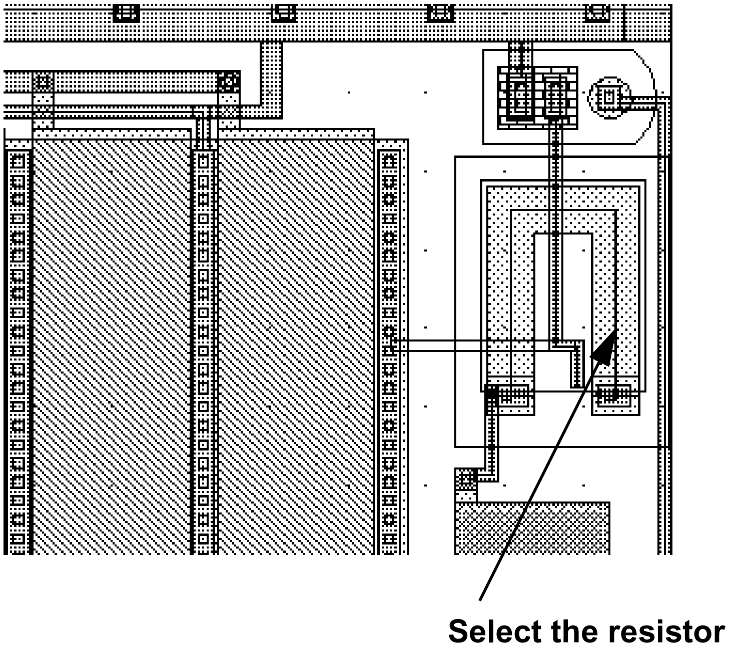
5. Deselect the *npn*.

   Another way to rotate instances is to use the **Edit Properties** command. In the Edit Instance Properties form use the **Rotation** button to rotate the selected instance.

Editing Pcell Properties

You need to change the width of the pcell resistor *spcres*, from 3.0u to 1.0u. Use the **Edit Properties** command to make the changes.

1. Select the resistor in the upper-right hand corner of the design as shown below.



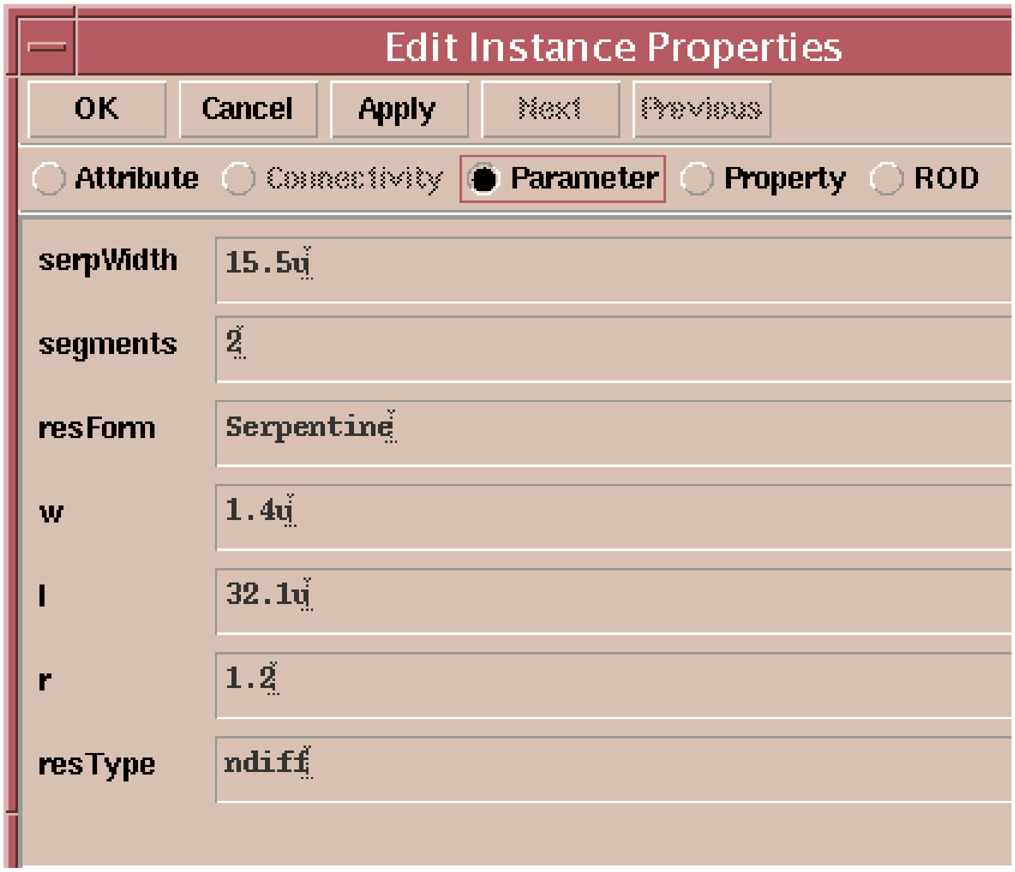
Select the resistor

2. Select **Edit—Properties [q]**.

3. In the Edit Instance Properties form select **Parameter**.

   The form changes to display the pcell parameters.

4. Change the parameters *serpWidth* and *w* on the Instance form as follows and click **OK**.



○ Attribute ○ Connectivity ● Parameter ○ Property ○ ROD

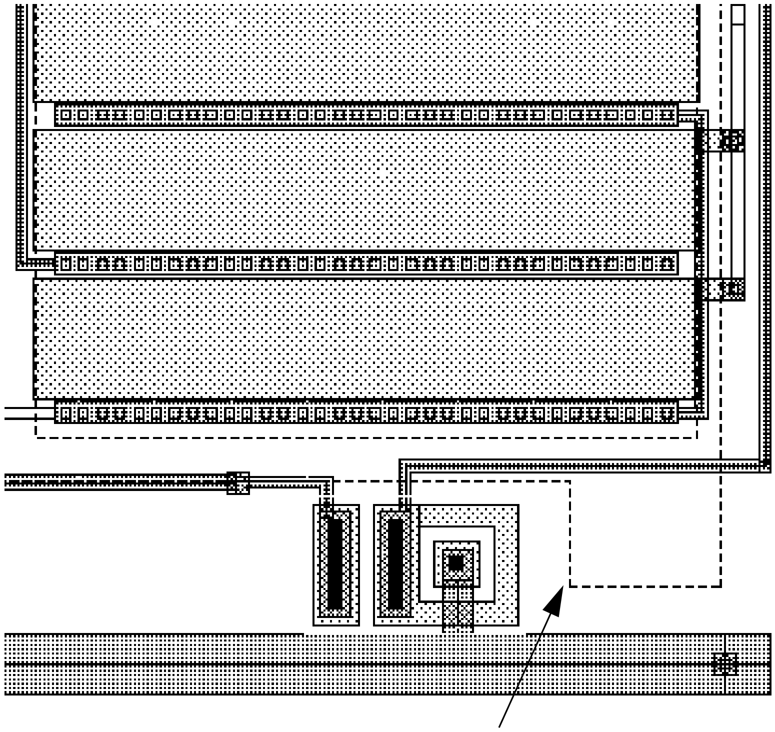
serpWidth	15.5u
segments	2
resForm	Serpentine
w	1.4u
l	32.1u
r	1.2
resType	ndiff

5. Deselect the resistor.

Chopping the Well

You need to edit the *nwell* in the lower right-hand corner. Instead of redrawing the nwell, use the **Edit Chop** command.

1. Select the large *nwell* as shown below.

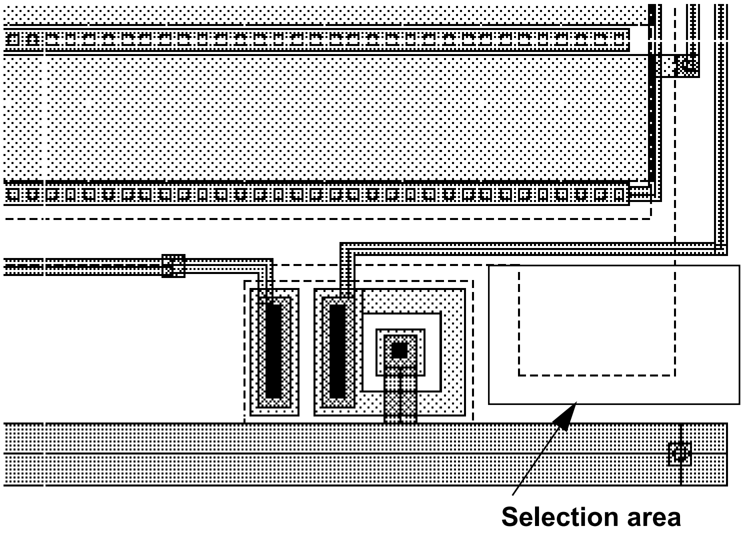


Select the nwell

2. Select **Edit—Other—Chop [C]**.

3. In the Chop form change the Chop Shape to **Rectangle**.

4. In the design window draw a rectangle around the portion of the *nwell* to be removed.



5. Save your design.

Returning Up the Hierarchy

Now that you have completed your edits, you want to return back up to the *pk44chip layout* cellview.

1. Select **Design—Hierarchy—Return [B]**.

   Does the window banner change?

   You should now be at the top level of the design. The window banner should read, *Virtuoso Layout Editing: design pk44chip layout.*

Editing Multiple Cell Placements

1. Select **Options—Display** and change the Display Levels to:

   ```
   From: 0
   To: 0
   ```

2. Click **OK**.

3. Select the *fiducial layout* instance located between the *iref1* and *vinput* pads, in the bottom-left corner of the design window.

   You will be editing only one of the instances. You must flatten the one instance of *fiducial* cellviews, make the edits, and then make it a new cell, with a new name.

4. Select **Edit—Hierarchy—Flatten**.

   You can use the default of *one level* for the *Fatten* mode because all the data inside of the fiducial cellview is at level one of the hierarchy.

   Is the data flattened?

Editing the Flattened Data

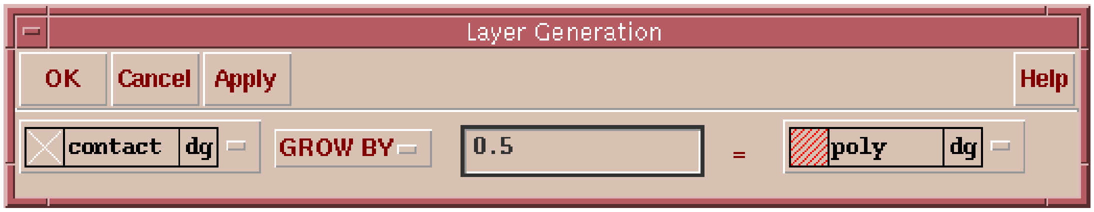
Use the **Layer Generation** command to edit the flattened data.

1. Select **Create—Layer Generation**.

2. In the flattened data select the white contact shapes. You should have four objects selected.

   Note: *Because the highlight layer and the contact layer are both white, you may not see the four objects selected. Look at the window banner to make sure they are selected.*

3. Change the Layer Generation form as shown and click **Apply**.



contact dg GROW BY 0.5 = poly dg

   Notice that a new shape has been created on the *poly dg* layer. This shape is merged together into one shape because .5 microns was added to each side.

4. Select **Edit—Undo [u]**.

5. Change the **GROW BY** value in the Layer Generation form to **0.2**.

6. Click **OK**.

7. Press the **[Del]** key to delete the original *contact dg* shapes that are still selected.

Making a Cell

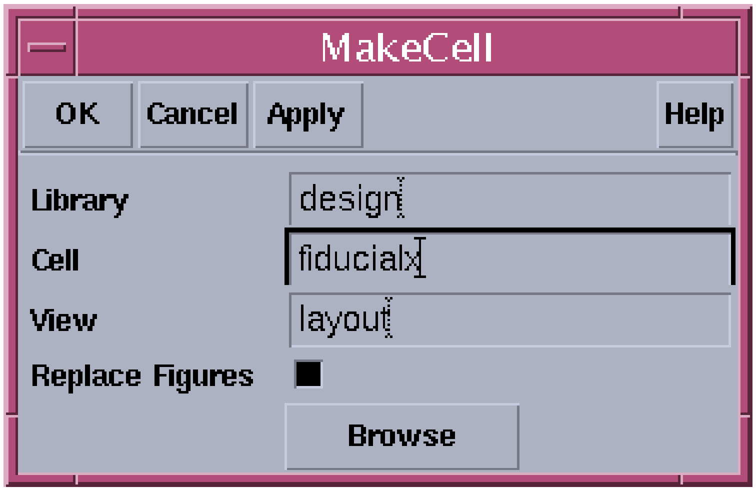
Now that you have made the edits, you can select the flattened data and make a new cell using the **Make Cell** command.

1. Select all of the objects in the *fiducial layout* cellview.

   You should have 14 objects selected. This is including the text and border drawn around the flatten objects.

2. Select **Edit—Hierarchy—Make Cell**.

3. Change the following in the MakeCell form and click **OK**.



Library: design
Cell: fiducialx
View: layout
Replace Figures ■

The Replace Figures option replaces the hard data with the *fiducialx layout* cellview.

Summary

In this lab, you did the following:

- Learned how to use the Mark Net command.
- Learned how to flatten an instance.
- Learned how to use the Layer Generation form.
- Learned how to make a cell from hard data.

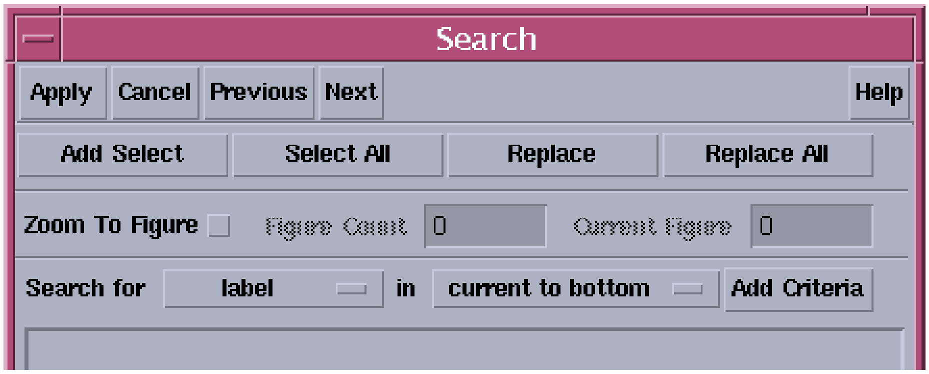
Lab 4-2 Searching the Database

Objective: Use the Edit Search command to find labels in the *pk44chip* design.

Searching for a Label

1. Select **Edit—Search [S]**.

2. In the Search form, change the following options.



Zoom To Figure □ Figure Count 0 Current Figure 0

Search for label in current to bottom

3. Press the **Apply** button.

   How many labels are found?

   The Figure Count field tells you how many labels were found during the search.

4. Select **Add Criteria**.

5. In the text field enter:

   ```
   iref*
   ```

6. Click **Apply**.

   How many labels were found in this search?

7. Change the text field to:

   ```
   iref$
   ```

8. Click **Apply**.

   Only one label is found this time. The **$** limits the search to the label *iref*.

Replacing the *iref* Label

The *iref* label is placed within the amplifier cell. You will replace the *iref* label using the Search form.

1. At the bottom of the Search form change the Replace option to **text**.

   A text field appears.

2. In the text field enter:

   ```
   amp_iref
   ```

3. Select **Replace All** at the top of the Search form.

   A dialog box appears:

   ```
   Open read-only cells for edit to replace ‘label’?
   ```

4. Click **Yes**.

   Another dialog box appears:

   ```
   Modifying ‘label throughout hierarchy. Is this
   okay?
   ```

5. Click **Yes**.

6. Click **Cancel** in the Search form.

   The label updates to *amp_iref*.

7. **Save** the design and close the window.

Summary

In this lab, you did the following:

- Learned how to use the Search and Replace commands.

Lab 4-3 Importing and Exporting a Design

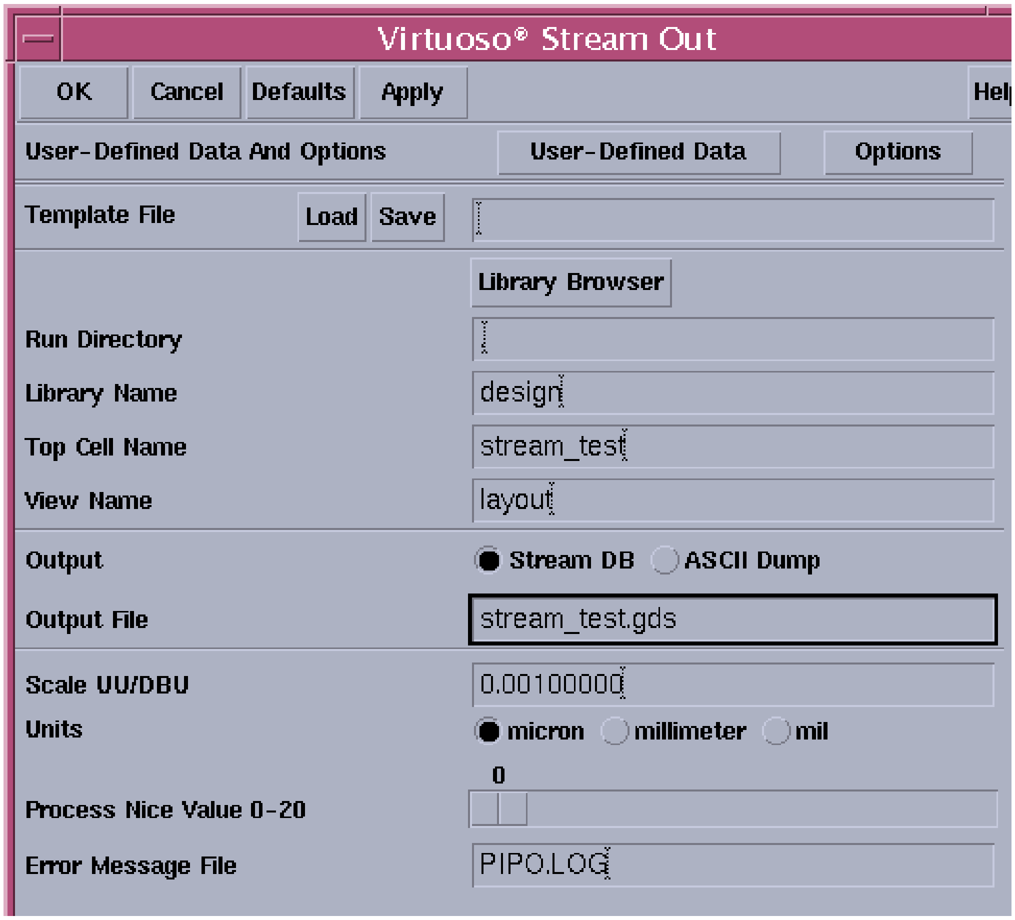
Objective: Learn how to translate a cellview using the basic options.

Opening a Cellview

1. Open the *design stream_test layout* cellview.

Streaming Out the Design

1. Select **File—Export—Stream** from the CIW menu and change the following:



User-Defined Data And Options

Template File

Run Directory

Library Name: design

Top Cell Name: stream_test

View Name: layout

Output: ● Stream DB ○ ASCII Dump

Output File: stream_test.gds

Scale UU/DBU: 0.00100000

Units: ● micron ○ millimeter ○ mil

Process Nice Value 0-20: 0

Error Message File: PIPO.LOG

2. In the **Stream Out** form, click on **OK** to start the stream translator.

   You should see these messages in the CIW.

   ```
   PIPO STRMOUT is running (PID = ipc:-1)...
   PIPO STRMOUT (PID = ipc:-1) completed
   successfully.:
   ```

   Note: *If you did not get the successful completed dialog box, check the strmout.log file to determine what is wrong. Then rerun the translator with corrected input.*

3. In the STRMOUT PopUp Message dialog box click **OK**.

4. Open a new xterm window, and enter:

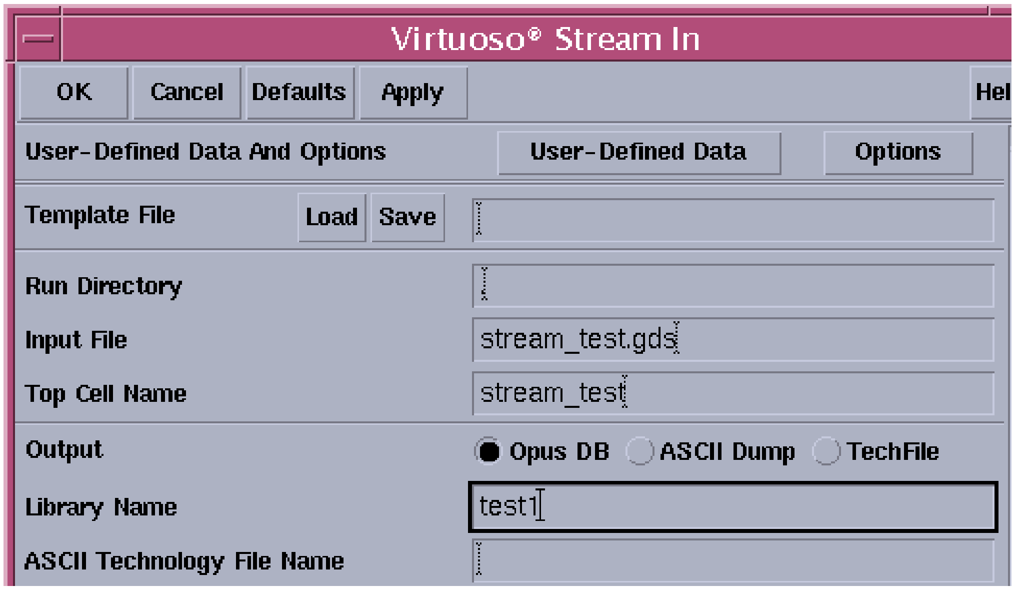
   ```
   cd ~/Layout
   ```

5. Verify that the run went as expected by viewing the log file.

   ```
   more PIPO.LOG
   ```

Streaming In *stream.strm*

1. Select **File—Import—Stream** from the CIW menus and change the following:



Run Directory	
Input File	stream_test.gds
Top Cell Name	stream_test
Output	● Opus DB ○ ASCII Dump ○ TechFile
Library Name	test1
ASCII Technology File Name	

2. In the **Stream In** form, click on **OK** to start the stream translator.

Verifying the Translation

You should see these messages in the CIW.

```
PIPO STRMIN is running (PID = ipc:-1)...
PIPO STRMIN (PID = ipc:-1) completed
successfully.:
```

1. From the Library Manager form, select **View—Refresh**.

2. Open the *stream_test* cellview from the *test1* library.

   Stream automatically adds the new library that you created during translation to the *cds.lib* file.

   Do you notice anything different about this design from the original?

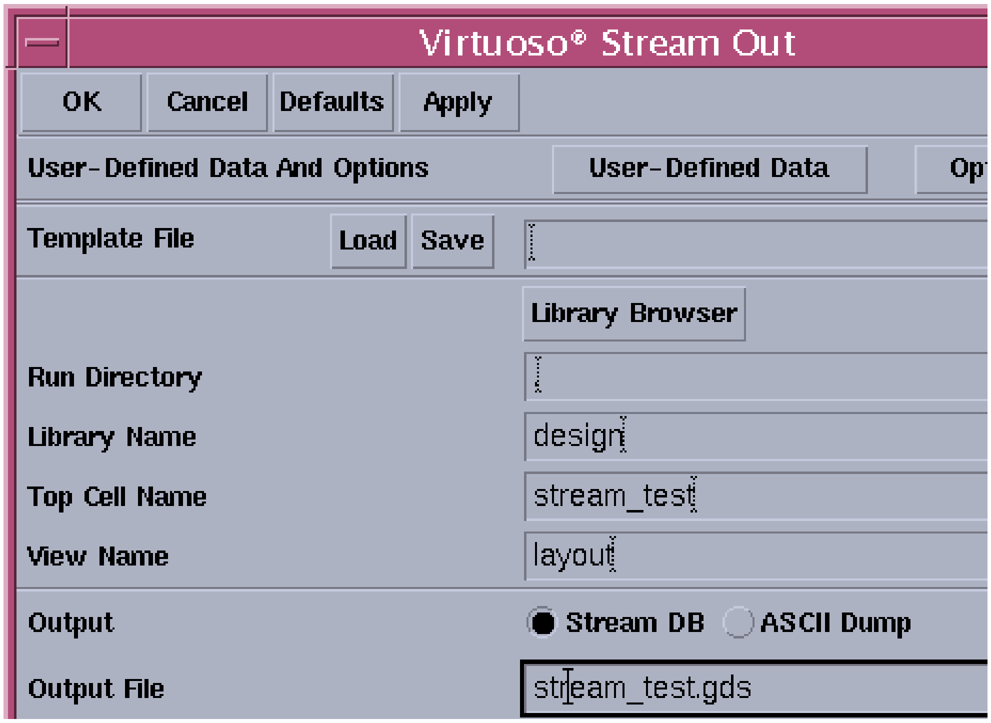
   Notice that the layers in the LSW do not have names. This is because you did not specify a layer mapping file, or a technology file. The default technology file was used when creating the *test1* library.

   In the next lab you will use a layer mapping file and a technology file.

3. Close the design window.

Streaming Out with a Map File

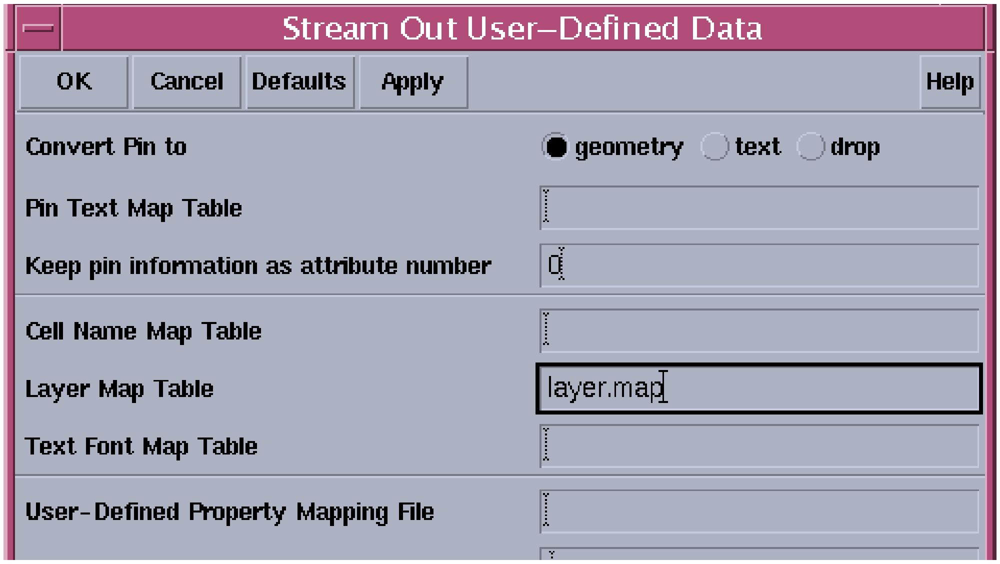
1. Select **File—Export—Stream** and change the following:



Template File	
Run Directory	
Library Name	design
Top Cell Name	stream_test
View Name	layout
Output	● Stream DB ○ ASCII Dump
Output File	stream_test.gds

2. Click on the **User-Defined Data** button.

3. Change the following in the *Stream Out User-Defined Data* form and click **OK**.



Convert Pin to	● geometry ○ text ○ drop
Pin Text Map Table	
Keep pin information as attribute number	0
Cell Name Map Table	
Layer Map Table	layer.map
Text Font Map Table	
User-Defined Property Mapping File	

4. In the **Stream Out** form, click on **OK** to start the stream translator.

   You should see these messages in the CIW.

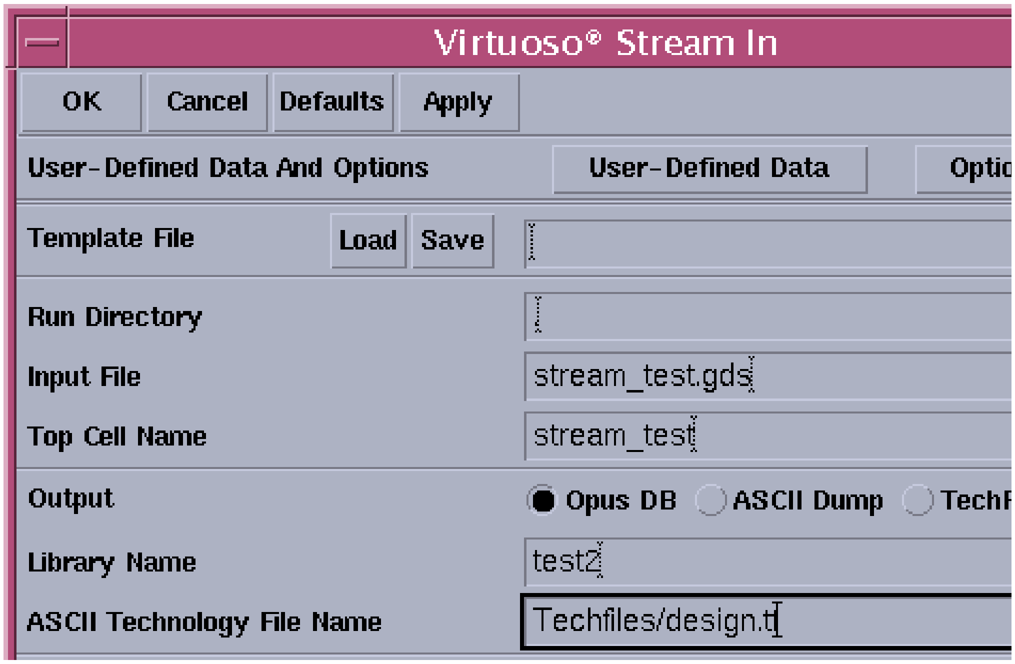
```
PIPO STRMOUT is running (PID = ipc:-1)...
PIPO STRMOUT (PID = ipc:-1) completed
successfully.:
```

Note: *If you did not get the successful completed dialog box, check the strmout.log file to determine what is wrong. Then rerun the translator with corrected input.*

5. In the STRMOUT PopUp Message dialog box click **OK**.

Streaming In *stream_test.gds*

1. Select **File—Import—Stream** from the CIW menus and change the following:



Run Directory
Input File stream_test.gds
Top Cell Name stream_test
Output ● Opus DB ○ ASCII Dump ○ TechF
Library Name test2
ASCII Technology File Name Techfiles/design.t

Notice that you have specified the *design.tf* file. This is the entire technology file that has been dumped from the design library.

2. Add the *layer.map* in the User-Defined Data form.

3. Click **OK** to start the stream translation.

Verifying the Translation

You should see these messages in the CIW.

```
PIPO STRMIN is running (PID = ipc:-1)...
PIPO STRMIN (PID = ipc:-1) completed
successfully.:
```

1. From the Library Manager form, select **View—Refresh**.

2. Open the *stream_test* cellview from the *test2* library.

   Do you notice anything different about this design from the original?

   All layers translated over except for the *contact dg* layer. This is because the layer is not defined in the layer mapping file.

Adding a Layer to the Mapping File and Streaming Out *stream3.gds*

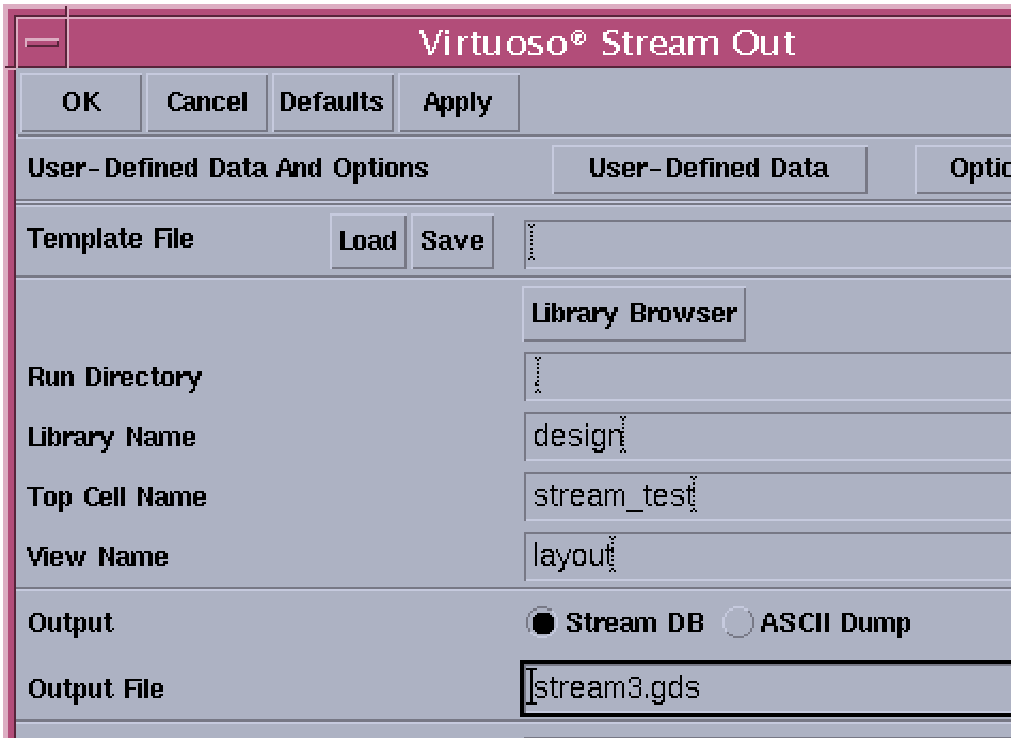
1. In a terminal window, type:

   ```
   vi ~/Layout/layer.map
   ```

2. Add the following to the mapping file:

   ```
   contact    drawing 55 0
   ```

3. Select **File—Export—Stream** and change the following:



Virtuoso® Stream Out

User-Defined Data And Options

Template File

Run Directory

Library Name: design

Top Cell Name: stream_test

View Name: layout

Output: ● Stream DB ○ ASCII Dump

Output File: stream3.gds

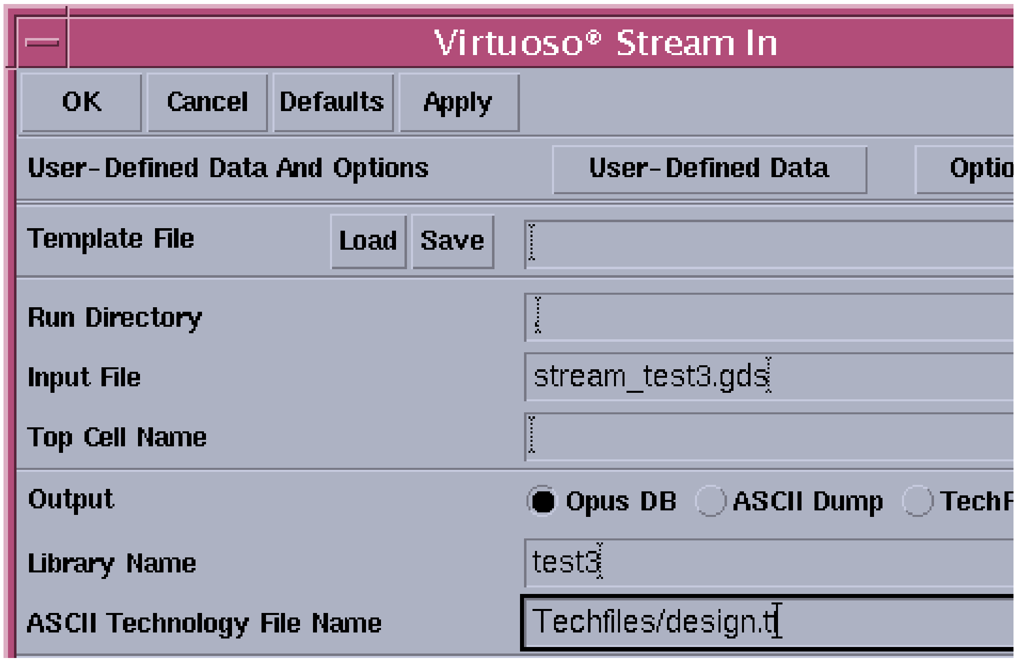
4. Make sure the *layer.map* file is entered in the Layer Map Table field of the User-Defined Data form.

5. Click **OK** to start the stream translator.

6. In the STRMOUT PopUp Message dialog box click **OK**.

Streaming In *stream3.strm*

1. Select **File—Import—Stream** from the CIW menus and change the following:



Virtuoso® Stream In

Template File

Run Directory

Input File stream_test3.gds

Top Cell Name

Output ● Opus DB ○ ASCII Dump ○ TechF

Library Name test3

ASCII Technology File Name Techfiles/design.tf

2. Click **OK** to start the stream translator.

3. From the Library Manager form, select **View—Refresh**.

4. Open the *stream_test* cellview from the *test3* library.

   All layers translated over because the layer *contact* is now defined in the layer mapping file.

5. Close the all design windows.

6. Quit out of the Layout Editor software by selecting, **File—Exit** from the CIW.

Summary

In this lab, you learned the following:

- Learned how to export and import your design.
- Learned how to export and import your design using a technology file and layer mapping file.

Lab 4-4 Plotting Your Design

Objective: Plot the *pk44chip* design using the plot command.

You need to plot the *pk44chip* design now that the layout is complete. Use the **Submit Plot** command to set the options for plotting.

Detailed Steps for Creating a Custom Plot

1. Open the *pk44chip* cellview.
2. Select **Design—Plot—Submit**.
3. In the Submit Plot form change Plot Area to **Select**, and draw a selection box in the design window of the area to be plotted.
4. Turn on **Notes** and type your name in the text field.
5. Select **Display Options**, and turn off **Axes**.
6. Set the grid to **Dots**, and click **OK** in the Display Options form.
7. In the Submit Plot form change the *Template File* name to *plot.template* and click **Save**.
8. Click **OK** in the Submit Plot form to start the plot.

Summary

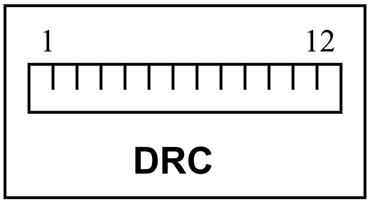
In this lab, you did the following:

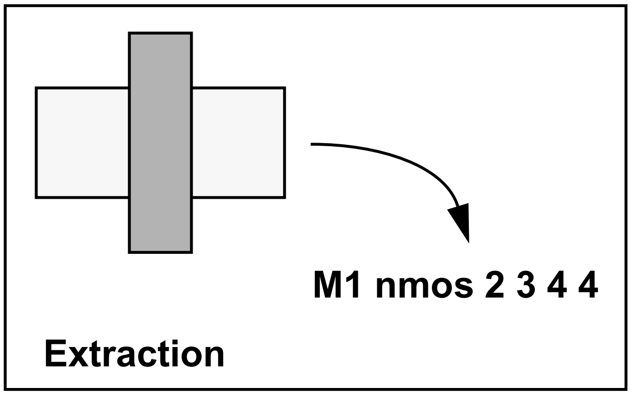
- Learned how to plot using different options.

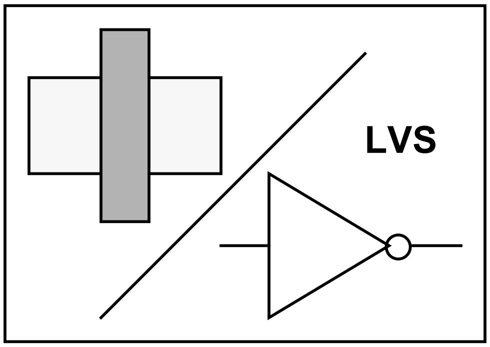
End of Lab

Labs for Module 5

Using Assura







Lab 5-1 Running Design Rule Checks from the User Interface

Objective: Become familiar with running a DRC, analyze the results, and get familiar with the run-specific file.

Logging In

1. Log in to your workstation. Your instructor will give the login name and the password.

   Note: *If you log out of your workstation at the end of the day, you must log in again the following day before performing the labs.*

Starting Assura

1. Find the Assura executable location by typing in the command line:

   ```
   cds_root assura
   ```

   The path to the Assura executables is displayed.

2. If necessary, set the ASSURAHOME environmental variable to point to the path of the Assura executables.

   ```
   setenv ASSURAHOME <path_returned_by_cds_root>
   ```

3. Change into the course directory:

   ```
     cd Layout/av
   ```

4. Start the Virtuoso® layout editor. It contains the user interface to Assura™ verification.

   In an xterm enter:

   ```
   layoutPlus&
   ```

   When *layoutPlus* has finished loading, the Command Interpreter Window (CIW) appears.

   You might see a *What's New in 4.4.x* window appear. To close the window for this session, select **File— Close**. To prevent the window from appearing during future sessions, select **Edit—Off at Startup.**

Running a DRC

1. Bring up the form to open the design:

   Select **File — Open** in the CIW.

2. Complete the form as shown below and click on **OK**.

Library Name	`avclass`
Cell Name	`adder4bit_drc`
View Name	`layout`

Note: *Although you open the design in edit mode, the Assura software does not require write access to the cellview. The error data is not stored within the layout.*

The design window containing the layout appears. A small narrow window (the Layer Selection Window, or LSW) containing your working layers appears next to the layout window.

3. Press **Shift-f** in the layout window (*display all levels*).

   All levels of data appear in the design window.

4. Select **Assura — Run DRC**.

   A pop-up window may appear if a default *DIVA.rul* file exists. Click **No**. You will enter a rules file name later.

   The DRC form appears. The library name and cell name are taken from the current design window but the rules file name may be missing.

5. Complete these fields in the DRC form:

Run Directory	`./adder`
Rules File	`./RULES/process_a.test_rules`
Include File	`./include/included.rsf`

6. Do not modify any other part of the form.

7. Click **OK** to start the DRC.

   If the Overwrite Existing Data form appears, click **Yes**. DRC starts and a progress form appears.

   When DRC finishes, a dialog box appears asking you if you want to view your DRC results.

8. Click **Yes** in the dialog box.

   The View Layer Window (VLW) and Error Layer Window (ELW) appear. The VLW is iconified.

   The VLW controls the verification colors you display on your screen. This window displays layers used by Assura. The colors in the VLW are assigned by Assura and might not be the same as the original layout.

Viewing DRC Errors from the Top-Level Cell

The Error Layer Window contains a list of errors. You can use the arrow buttons in the ELW to see each type of error at its level of the design hierarchy.

The following errors appear in the ELW.

```
[1] poly1 spacing lt 1
[6] poly1/diffspacing lt 1
[2] metal1spacing lt 1
```

1. Click **View — Summary** in the Error Layer Window.

   The DRC summary file created during the DRC run appears.

   The summary contains the contents of the run-specific file, memory usage, time summary for various internal tasks performed by Assura, time and memory used for each rule, and the total time for the job.

   Notice the time taken for each check. (Your times may vary from the ones in the example.)

```
;# 1
  pdiff = geomOr( pdiff )
  cpu: 0.0   elapsed: 1   virtual: 30M task sum: 124
;# 2
  ndiff = geomOr( ndiff )
  cpu: 0.0  elapsed: 0   virtual: 30M task sum: 112
```

   cpu: 0.0 indicates the time was less than .1 second.

   The *task sum* value indicates the number of shapes created by the rule.

   This report is useful in identifying the steps that take a long time and then optimizing the rules to reduce the time.

2. Look at the end of the file.

```
Total cells checked = 14
Total rules checked = 19

Run time = 4.00 seconds
CPU time = .39 seconds
```

   The summary does not give the number of errors.

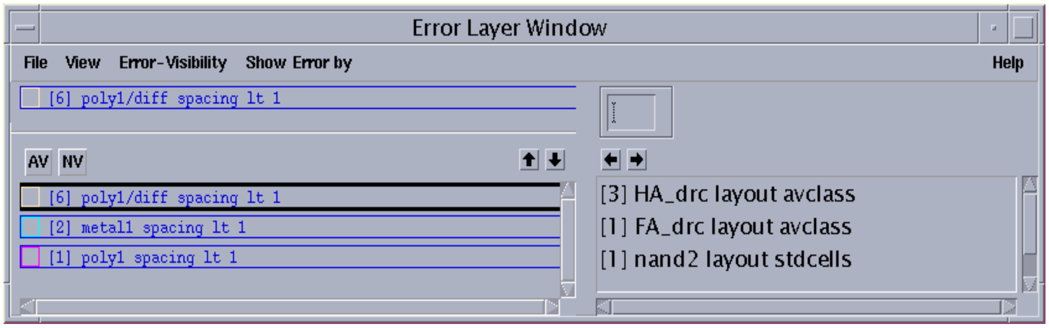
   Your run and CPU times may vary from the numbers shown.

3. In the Summary window select **File — Close Window**.

Using the ELW to Find DRC Errors

1. Click the *poly1 spacing* error (not *poly1/diff spacing*) in the Error Layer Window.

   The right column shows the list of cells in which this error occurs. The only cell shown containing this error is *HA_drc layout avclass*. Assura picks one of 10 color swatches for the error type.



```
[6] poly1/diff spacing lt 1

[6] poly1/diff spacing lt 1
[2] metal1 spacing lt 1
[1] poly1 spacing lt 1
```

```
[3] HA_drc layout avclass
[1] FA_drc layout avclass
[1] nand2 layout stdcells
```

2. View all the errors.

   a. Click on the right-facing arrow above the list of cells.

   If you receive a dialog box stating errors exist at other levels of hierarchy, display all levels with **Shift-f**.

   The layout display zooms in on the first *poly1/diff spacing* error. The error flag uses the same color as the swatch next to the error message. You can use the right arrow on the Error Layer Window to view all remaining errors when there are more than one.

   The Error Layer Window shows: *1 of 3*

   b. Select **View — Explain** in the ELW and click an error marker.

   The marker text window appears with the cell hierarchy noted. These statistics are also reported in the CIW. Close the marker text window.

   c. In the ELW, you have selected *poly1/diff spacing* and *HA_drc layout avclass*.

   Move your cursor into the layout window and press **Esc**. This brings focus to the window area. Move the cursor around to see which shapes are editable at this level. You see the pre-select dotted white line over the shape that is selectable.

d. Continue using the right arrow to sequence through the *poly1/diff* error flags. When you pass the last error in the cell (3 of 3), the ELW highlight moves from *HA_drc* to *FA_drc*. The last error in *HA_drc* remains in the cell view window. Click the right arrow again to highlight the first error in *FA_drc*.

e. Continue to use the right arrow. This will take you through each of the cellviews and each of their errors. When you have made your way through all the errors, a dialog box reports you there are no more errors.

f. Close the dialog box and the marker text window.

3. In the cellview use **Shift-b** until you return to the top-level design.

Using the ELW Error Filter

You can control the display of errors in the ELW by making choices on the ELW Preferences form.

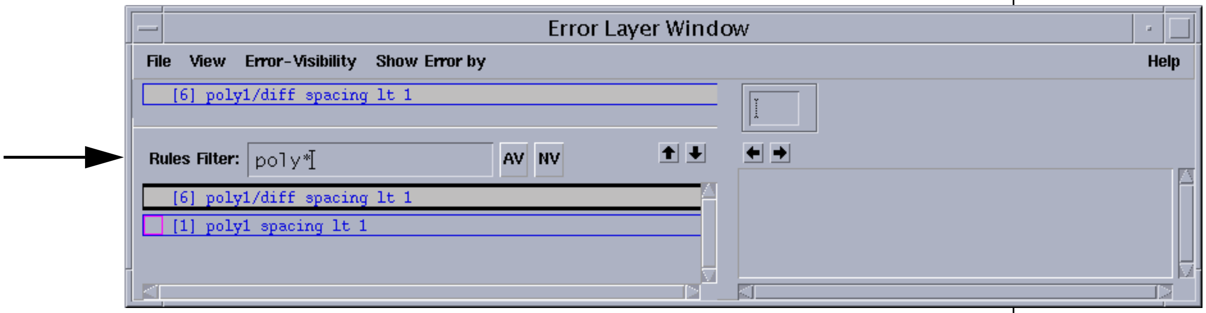
1. In the ELW select **File — Preferences**.

   The ELW Preferences form appears.

2. In the ELW Preferences form select **Enable: Rules Filter**.

   The filter field appears above the error list in the ELW.

3. In the filter field enter *poly** and press **Return**.



[6] poly1/diff spacing lt 1

Rules Filter: poly*

[6] poly1/diff spacing lt 1
[1] poly1 spacing lt 1

   Only error messages containing the string *poly* appear.

4. Click **Defaults**.

5. In the ELW select **poly1/diff spacing**.

6. In the Rule Order section of the ELW Preferences form, cycle the choices.

   The listing of the rules in the ELW changes.

   Leave *Error Count* selected.

7. In the Cell Order section, cycle the choices.

   The listing of cell names in the ELW changes between an alphabetical listing and a listing by error count.

8. In the Step through section, cycle the choices.

   When you choose *Selected Rule*, one rule type is visited in each cell where the error occurs. When you choose *Selected Cell*, all rules are cycled for the selected cell.

9. Click **Add Comments during Sign Off**.

   This choice allows you to enter comments about errors that have been signed off.

10. In the Enable section, click each choice to see its effect.

    a. *Bubble Help*: Displays the function of an arrow or button which has been made visible by turning on the ELW Preferences *Enable* buttons. Place your cursor over the arrow or button. Pause for a moment. The bubble help will show text describing the function.

    b. *Rules Filter*: Opens an entry window above the rules in the ELW. You use this to filter rules available in the ELW.

    c. *Cells Filter*: Opens an entry window above the list of cells in the ELW. You use this to filter the list of available cell names.

    d. *Preview Buttons*: Enables a group of arrows above the list of rules. The arrows let you cycle through the highlighted error in the current cell only. They do not continue to the next cell in the list.

    e. *Sign Off Feature*: Displays a button and text box above the list of cells. You use this to learn many errors were signed off.

    f. *Fixed Feature*: Displays a text box above the list of cells in the ELW. You use this to learn how many errors were fixed.

g. *Cell Selectors*: Enables the up and down arrows at the far right of the ELW. They move the highlight up or down in the cell names area of the ELW.

11. Click **Defaults** to return the Preferences form to its original state.

12. Select **File — Close ELW** in the ELW.

    The ELW and ELW Preferences form close.

13. Return to the top level of the layout design.

14. Iconify the layout window.

End of Lab

Lab 5-2 Running Design Rule Checks from UNIX

Objective: Become Familiar with the Run-Specific File.

Getting Familiar with the Contents of a DRC Run

1. Make sure you're in the course directory (*~/av*).

   Note: *Various UNIX commands are required in subsequent portions of the lab. All these commands must be executed in an xterm window. Examples of these command are: cp, mv, more, ls, pwd, and cd.*

   Additionally, you will perform some editing tasks. The Cadence classrooms provide vi, emacs, and textedit.

2. Change into the directory where the Assura files are stored. Enter:

   ```
   cd adder
   ```

   Files created by Assura were stored in the directory. They are named after the cell name since no run name was specified.

3. Look at the run-specific file.

   ```
   more adder4bit_drc.rsf
   ```

   Assura names the run-specific file *<runName>.rsf*. This file contains basic commands needed to run Assura.

   Notice near the top of the file in the *avParameters* section the line *?modifyHierarchy nil*.

   This is used when you are running medium to small cells. It allows you to present the error data in the hierarchy where the error occurred. This is important when you are working at lower levels in the hierarchy. This allows the error flag to post in the hierarchy where the error occurred.

   Top-level designs will be checked with *?modifyHiererchy t,* which is the default setting. The error markers will then be placed at higher levels in the design.

4. Look at the log file and find the steps listed below that were performed by Assura DRC.

   ```
   more adder4bit_drc.log
   ```

   Building the VDB part 2 and Part 3 in background mode — finalizes the conversion of the Cadence® Design Framework II layout data to the Assura database.

   Running the Task Processor — performs the design rule checks.

   Creating Error Database — lists the number of different types of errors.

   RealCount — gives the total number of errors you have to correct. The error will appear in the cell in which it occurs.

   FlatCount — gives the total number of errors in the design including replicated errors due to the design's hierarchy.

5. Look at the DRC summary. Enter:

   ```
   more adder4bit_drc.sum
   ```

   This is the file that you see when you click **Summary** in the Error Layer Window.

6. Look at the error file. Enter:

   ```
   more adder4bit_drc.err
   ```

   This file lists the error flag locations for each cell. The errors are listed with their coordinates for each error type. Use these coordinates to find your errors.

   These files will help you debug a problem if Assura does not complete successfully or you do not see some expected errors.

   If you expect some DRC violations but the summary file (*.sum*) failed to show the expected DRC rule, Assura did not execute the DRC command. You should examine your rules files at this point to find missing checks.

7. Close the DFII session.

Running Assura Sequentially in UNIX

This part of the lab shows you how to create an RSF for sequential DRC. The application is common to verification of cell libraries. You will create RSFs for running DRC on two cells, *nand2* and *nor2*. You will use a custom sequential verification script written in UNIX to start Assura.

1. Make a copy of the run-specific file for this lab:

   ```
   cp adder4bit_drc.rsf ../nand2.rsf
   cp adder4bit_drc.rsf ../nor2.rsf
   ```

   These new RSF files are now in directory *av*.

2. Change to *av*:

   ```
   cd ..
   ```

3. Edit *nand2.rsf.* In the *avParmeters* procedure modify the following lines. (Make sure to enter ../ for the rules file path.)

   ```
   ?inputLayout ( "df2" "stdcells" )
   ?cellName "nand2"
   ?rulesFile "../RULES/process_a.test_rules"
   ?runName "nand2"
   ?workingDirectory "nand2"
   ?avrpt
   ```

4. Save the file.

5. Now perform these modifications to *nor2.rsf.* (Make sure to enter ../ for the rules file path.)

   ```
   ?inputLayout (“df2” “stdcells” )
   ?cellName "nor2"
   ?rulesFile “../RULES/process_a.test_rules”
   ?runName "nor2"
   ?workingDirectory "nor2"
   ?avrpt t
   ```

   You now have two functional RSFs for running sequential verification.

6. Run the sequential design rule checks from an xterm window with this custom verification UNIX shell script:

```
./av_seq stdcells
```

7. Look at the log file as the run progresses. Enter:

```
tail -f stdcells/stdcells.log
```

   All the processing steps appear on your screen.

   Assura completes the job with the following message:

```
***** Assura terminated normally  *****
```

8. Press **Control-c** to exit *tail*.

9. As time permits, look at the script you used to run sequential verification:

```
vi av_seq
```

Opening and Viewing the Results of an Assura Run

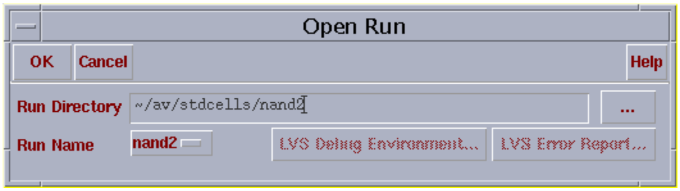
1. Start the layout tool:

```
layoutPlus&
```

2. Open *stdcells nand2 layout*.

3. From the *nand2 layout* window, select **Assura** — **Open Run**.

   The Open Assura Run form appears.



Run Directory ~/av/stdcells/nand2

Run Name nand2

4. Enter the path to the *nand2* run directory.

5. Make sure *nand2* is visible in the Run Name field.

6. Click **OK**.

   When the run opens, you will see errors from the *nand2* DRC run posted in the ELW.

7. Investigate the error.

8. Select **Assura — Close Run**.

9. Select **Window — Close** in the *nand2* cell.

10. Repeat steps 1-4 for *nor2*.

11. Close the DFII session.

The steps in the last two sections showed you how to run an Assura job from a shell window and how to open the run data from the graphical user interface to view the results.

Learning More About Using Assura Verification

1. Start the Cadence online documentation (CDSDoc):

   ```
   cdsdoc &
   ```

2. In the Active Document Hierarchy section select **Assura 3.0.**

   The window lists all the verification documents by topic.

   Note: *The path for cdsdoc is set by the file .cdsdoc/cdsdoc.ini in your home directory. Assura documentation has a separate path from DFII documentation.*

3. Open the Assura DRC/LVS folder.

4. Select and open the *Assura Physical Verification User Guide*.

   The table of contents of the Assura manual appears.

5. Look in Chapter 3, Using DRC. You will see the topic Viewing DRC Errors. You will recognize links into information about the Error Layer Window. Explore them if you wish.

6. Iconify the CDSDoc windows when you finish.

Lab Summary

In this module

- You learned to run a design rule check using Assura.
- You familiarized yourself with the contents of the run-specific file.
- You familiarized yourself with the output files created by Assura.
- You learned how to execute Assura using a batch script.
- You learned how to open an existing Assura run.

End of Lab

Lab 5-3 Checking a GDSII Database and Viewing the Errors

Objective: Learn how to check a non-Design Framework design and view the errors

Starting the DRC from Assura

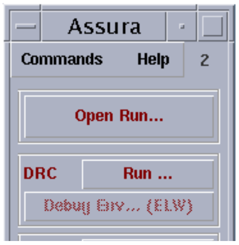
1. Change to the directory for this lab. Enter:

   ```
   cd ~/av/gds2
   ```

2. Start the Assura user interface. Enter:

   ```
   avview &
   ```

   The CIW appears and disappears (it’s iconified by *avview*). The *avview* console form appears. It has the title *Assura* and has different command boxes for the various checking operations.

3. Click **Run** in the **DRC** box.

4. Complete the DRC form.

Layout Design Source	`Stream`
Input File	`./adder4bit.db`
Cell	`adder4bit_drc`
Run Name	`adder4bit_gds2`
Rules File Name	`~/av/RULES/process_a.test_rules`
RSF Include	`~/av/include/included.rsf`

5. Click **OK**.

   The DRC starts.

   When the DRC finishes, a dialog box appears asking you if you want to see your DRC errors.

6. Click **Yes** in the dialog box.

   The Open Run form appears.

7. Click **Open Run**.

   The cellname *adder4bit_drc* appears in the form. All the cellnames appear that were checked during the run.

8. Double-click on the name **adder4bit_drc** or select **adder4bit_drc** and click **OK**.

   The View Layer Window (VLW), the Error Layer Window (ELW), and the layout window appear.

   The errors are the same errors that appeared when you checked the corresponding design in DFII.

   In the VLW the layer visibility for all layout layers is off.

   The layout window shows the top cell with four placements of the cell *FA_drc*. It does not show any mask layers.

Viewing the Errors

1. In the VLW click on **Original AV** to display the original layers.

2. Press **Control-r** in the layout window to redraw the window.

   The window redraws to show the *FA_drc* cell outlines and the polygon shapes that connect the cells.

3. Press **Shift-f** in the layout window.

   All levels of data appear in the design window.

   The colors of the layers may be different from the colors seen when you opened the corresponding DFII design.

   The colors for the layers in the LSW are controlled by the user section of the technology file. Assura colors in the VLW use 10 system layers from the techfile.

   Apart from the difference in color, you debug the errors in the same way as you did when you used *layoutPlus*.

4. Select **Commandsr— Quit** from the Assura console to exit *avview.*

Getting Familiar with the Run-Specific Options and the Rules File

1. From a xterm window, view the run-specific file. Enter:

   ```
   more gdsrsf
   ```

2. Notice the *inputLayout* line:

   ```
   ?inputLayout ( "gds2” "adder4bit.db” )
   ```

   This line instructs Assura to read a Stream (*gds2*) input file and specifies the name of the file. You can give a filename relative to the working directory or give the full path to the file.

3. Look at the rules file. Enter:

   ```
   more ../RULES/process_a.test_rules
   ```

   You used the same rules file, *process_a.test_rules*, for layout data in Stream format and DFII layout format.

   You see two separate *layerDefs* sections, one for Stream format and one for Design Framework II format.

   Being able to specify input data in different formats and keep the entirety of the rules common is a useful feature in the Assura rules file.

Lab Summary

In this module

- You learned to use *avview* to run a DRC and view errors.
- You learned how to check a Stream format database containing layout data.
- You became familiar with Assura’s run-specific commands to read Stream format.

At your site you use *layoutPlus* (including other DFII executables) or *avview* for executing checks on DFII layouts. Use *avview* to check and debug data in Stream format.

End of Lab

Lab 5-4 Running a Layout Versus Schematic Check

Objective: Get familiar with running LVS and the output files from LVS.

In this lab you become familiar with LVS by performing an LVS check on the *FA* (Full Adder) design and then viewing the output files. In a later lab you will run LVS on another design that has the *FA* design placed as an instance.

The next two sections show you how to run an LVS and look at the error reports from the graphic user interface.

Running an LVS

1. Change to the LVS directory:

   ```
   cd ~/av/LVS
   ```

2. Start *layoutPlus*:

   ```
       layoutPlus &
   ```

   The CIW appears.

3. Open the following design:

Library Name	`avclass`
Cell Name	`FA`
View Name	`layout`

4. Select **Assura — Run LVS**.

   The Assura Run LVS form appears. The layout name is already in the form. Assura fills in the layout name from the cellview in the layout editor window. Other information in the form can be populated by the *state* file.

5. In the Run LVS form click **Load State**.

   The LVS State form appears.

   The state file contains the form field data defining the initial setup of the Assura interface. (The name of the state file is arbitrary.)

6. Select the state named **fa** and click **OK**.

   This loads a state file named *.lvs.fa.state*. All values were taken from this file. The state file was created for the course. This is a convenient mechanism to save the form settings.

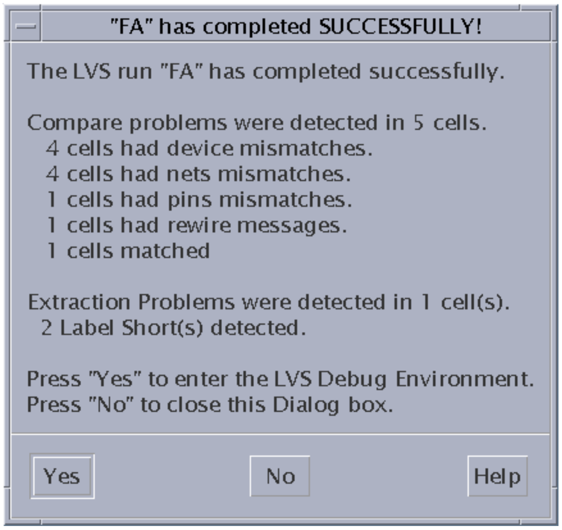
   The LVS form is populated with all necessary field entries.

7. Click **OK** in the form.

   The LVS begins.

   A dialog box informs you that the LVS completed successfully and asks if you want to view the results. You also see a quick summary in a pop-up form.

.

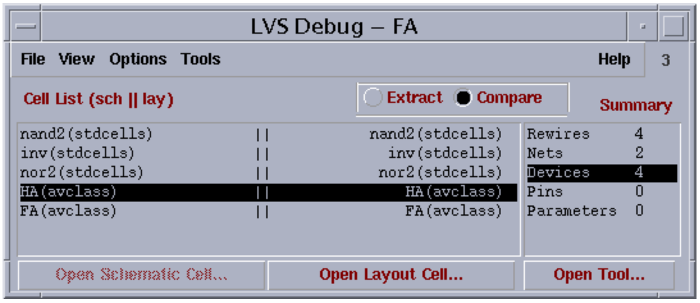


"FA" has completed SUCCESSFULLY!

```
The LVS run "FA" has completed successfully.

Compare problems were detected in 5 cells.
  4 cells had device mismatches.
  4 cells had nets mismatches.
  1 cells had pins mismatches.
  1 cells had rewire messages.
  1 cells matched

Extraction Problems were detected in 1 cell(s).
 2 Label Short(s) detected.

Press "Yes" to enter the LVS Debug Environment.
Press "No" to close this Dialog box.
```

8. Click **Yes** in the dialog box.

   You are directed into the LVS Debug Environment. The LVS Debug form comes up.



```
nand2(stdcells)    ||    nand2(stdcells)
inv(stdcells)      ||      inv(stdcells)
nor2(stdcells)     ||     nor2(stdcells)
HA(avclass)        ||        HA(avclass)
FA(avclass)        ||        FA(avclass)
```

```
Rewires      4
Nets         2
Devices      4
Pins         0
Parameters   0
```

Viewing Errors

In this section you will look at errors caused by mismatched devices and nets.

1. In the LVS Debug form, select **HA(avclass)**, select **Devices**, and click **Open Tool**.

   The Devices Mismatch Tool Form opens.

   The viewPort descends into the *HA* layout cellview.

2. In the Device Mismatches Tool window, select **I1 (inv)** in the Schematic side of the form and click **Probe**.

   The *HA* schematic opens. You can now see the *inv* cell highlighted in schematic and layout.

   The schematic window opens. *I1 (inv)* is highlighted in both *schematic(I1)* and *layout(I8)* views.

3. Position the schematic and layout views at the top half of the screen and leave the bottom half for the Devices Mismatch and Nets Mismatch forms.

   (If the highlight does not remain visible on the schematic, select and probe the *I1 inv* instance again from the Devices Mismatch Form.)

4. In the LVS Debug form, *HA(avclass)* is already highlighted. Now select **Nets** and click **Open Tool**.

5. Position the Devices Mismatch and Nets Mismatch tool windows so you have easy viewing of the schematic and layout views. Fit the layout window so you can see the entire layout.

6. Notice in the schematic that *I1 inv* is driven by net B. Select **B** in the Nets Mismatch tool form and click **Probe**.

7. You can now see the *I1 inv* instance and net B highlighted. This is an excellent way to see the mismatches.

8. Repeat the process with net A and the other devices in the Devices Mismatch tool form.

   Experiment with adding and removing probes.

   Note: *The command to remove all probes is located under Tools — Probe in the Mismatch Tool form.*

9. Remove the highlights and close the tool forms, the LVS Debug form, and the schematic window.

10. The layout cellview is probably still open to *HA* layout. Press **Shift-b** to return to the top cellview *FA* layout.

11. If you have any probes still highlighted , remove them by choosing **Assura— Probe — Remove All**.

12. Close the run.

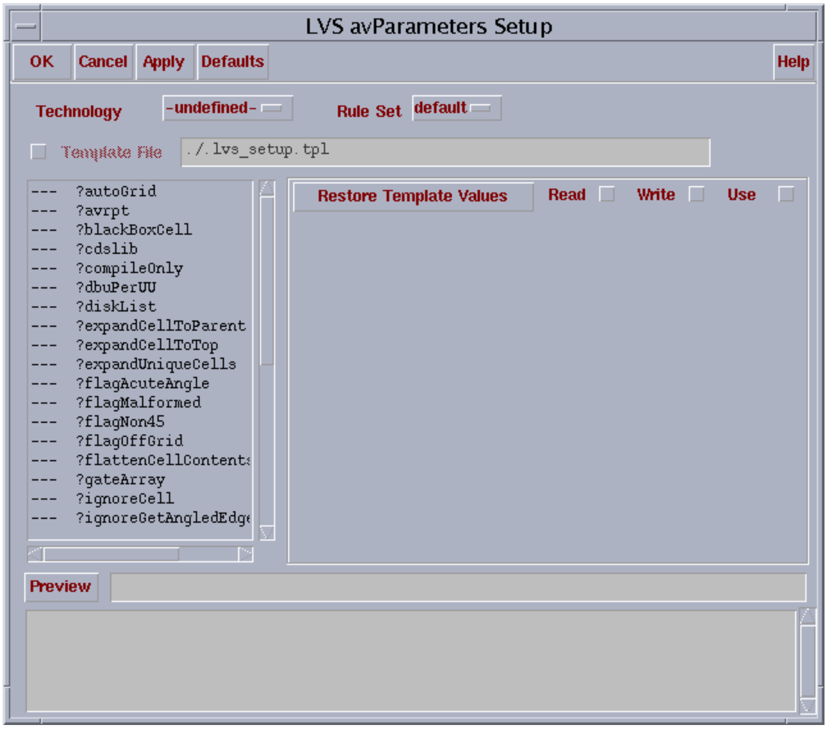
Using LVS Options

Assura checks that the top-level layout and schematic input and output names match. You can instruct Assura to ignore top-level pin errors. This is useful when you are checking a partially completed layout and you want to know if functionally the layout matches the schematic.

1. From the *FA layout* window click on **Assura — Setup — LVS avParameters**

   The LVS avParameters Setup form appears. There are several Assura options that you can choose from. You will be shown how to add other options which are not available on the form. For example, to ignore top-level pin errors you must define an option in a user-defined run-specific file and include the file when you run LVS.

2. Scroll down the options list. When you select an option, such as *?ignoreCell*, you will see the explanation displayed in the Preview window at the bottom of the form.



LVS avParameters Setup

Technology -undefined- Rule Set default

□ Template File ./lvs_setup.tpl

```
---   ?autoGrid
---   ?avrpt
---   ?blackBoxCell
---   ?cdslib
---   ?compileOnly
---   ?dbuPerUU
---   ?diskList
---   ?expandCellToParent
---   ?expandCellToTop
---   ?expandUniqueCells
---   ?flagAcuteAngle
---   ?flagMalformed
---   ?flagNon45
---   ?flagOffGrid
---   ?flattenCellContent:
---   ?gateArray
---   ?ignoreCell
---   ?ignoreGetAngledEdge
```

3. Cancel the form.

Using an Included Run-Specific File

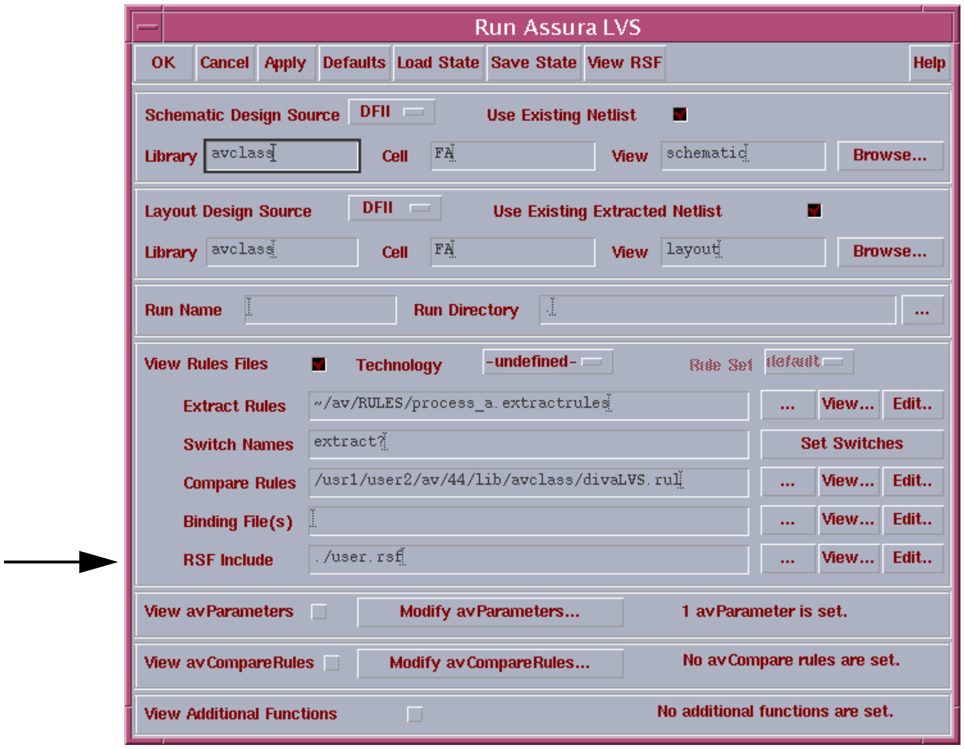
1. From the *FA* layout window select **Assura** — **Run LVS**.

   Some run options are not yet available in the options form. The suggested method of introducing variant run options is by listing them in a run-specific include file. These files are then listed in the *RSF Include* field.

   An example RSF in the course directory is *user.rsf*. This file already has the following command to turn off checking for the existence of all layout pins at the top level:

   ```
   avCompareRules(
     verifyTopSchPins(nil)
   )
   ```

2. Enter *user.rsf* in the RSF Include field.



Run Assura LVS

Schematic Design Source DFII — Use Existing Netlist ☒
Library avclass — Cell FA — View schematic

Layout Design Source DFII — Use Existing Extracted Netlist ☒
Library avclass — Cell FA — View layout

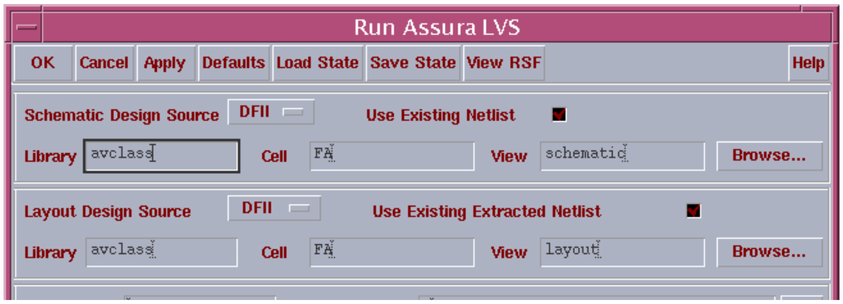
Run Name — Run Directory .

View Rules Files ☒ — Technology -undefined- — Rule Set default

Extract Rules ~/av/RULES/process_a.extractrules
Switch Names extract?
Compare Rules /usr1/user2/av/44/lib/avclass/divaLVS.rul
Binding File(s)
RSF Include ./user.rsf

View avParameters □ — 1 avParameter is set.
View avCompareRules □ — No avCompare rules are set.
View Additional Functions □ — No additional functions are set.

3. Turn on the toggle buttons to use the existing netlists for both the layout and the schematic.



Schematic Design Source DFII Use Existing Netlist ☒

Library avclass Cell FA View schematic

Layout Design Source DFII Use Existing Extracted Netlist ☒

Library avclass Cell FA View layout

4. Click **OK**. When Assura asks if you want to overwrite the existing file, click **OK** again.

5. The Completion form appears and shows a listing of compare errors and a message *No extraction Problems Were detected*.

6. Click **Yes** to enter LVS debug environment.

7. Notice that the LVS Debug form comes up in *compare* mode. If you switch back to *extract* mode, you will see there are no errors posted. This is because we included the *user.rsf* file which directs LVS not to try to match the schematic pins in the layout.

8. Close the LVS Debug form. (This quits the debug environment.)

9. Select **Assura** — **Open Cell**.

10. In the Open Cell form select **LVS Cells** — **matched**.

    In this second run, ignoring top-level pins, the top cell *FA* matches with no pin errors.

11. Cancel the Open Cell form.

12. Edit *user.rsf* to force LVS to print the number of matched devices even when the layout matches the schematic. This provides documentation of a clean run.

    a. Choose **Assura** — **Run LVS** — **RSF Include** — **Edit**.

       The form appears with *user.rsf* already in the field.

b. Click **Edit** to open the default editor.

c. Remove the comment character (the semicolon) from the following line:

```
;filterReduceStatistics()
```

d. Save the file.

13. Rerun LVS.

14. In the LVS Debug form, select **View — LVS Error Report**.

    The error report that appears. You see the device statistics for the top-level cell and the statement that *Network Matching was OK*.

15. Close the report window.

Viewing the Extracted Devices and Nets

1. Click on the VLW icon to open the VLW window.

2. In the LSW, select **text drawing** and click **NV** to turn visibility off for all other layers.

3. In the VLW, click on **Other** — **AV**.

   This sets the device recognition shape layers to *all visible*.

4. Press **Shift-f** if necessary to display all levels. Press **Control-r** to redraw the screen.

   You see the rectangles making up the *pgate* and *ngate* regions.

5. In the VLW, click on **Connect** — **AV** and redraw the screen.

   This shows all the layers connecting the devices. One of the layers is the *pwell* substrate layer which covers the entire design. Another layer is the *nwell* layer which covers all the *pgates*.

6. Click with the middle mouse button on layers *pwell* and *nwell* to turn the display off for these layers.

7. Press **Control-r** to redraw the layout window.

   Now you can see the other connected layers better.

8. Press **Control-f** to show only the top level.

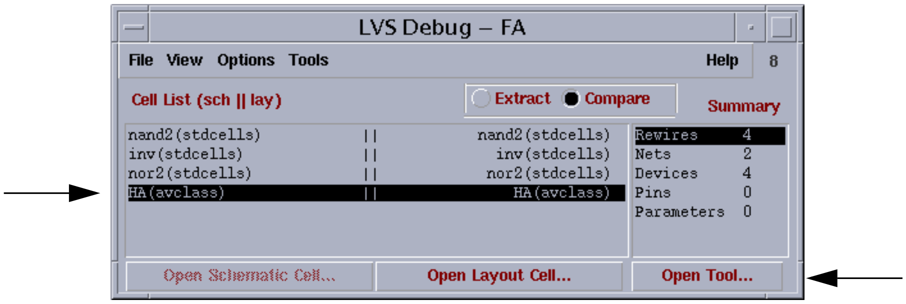
   Notice only the mask layers at the top level are visible. There are no devices at the top cell.

9. Press **Shift-f** to display the entire hierarchy again.

10. Iconify the VLW and LSW.

Using the Rewire Tool

1. Select **HA(avclass) Rewires** in the LVS Debug form and click **Open Tool**.

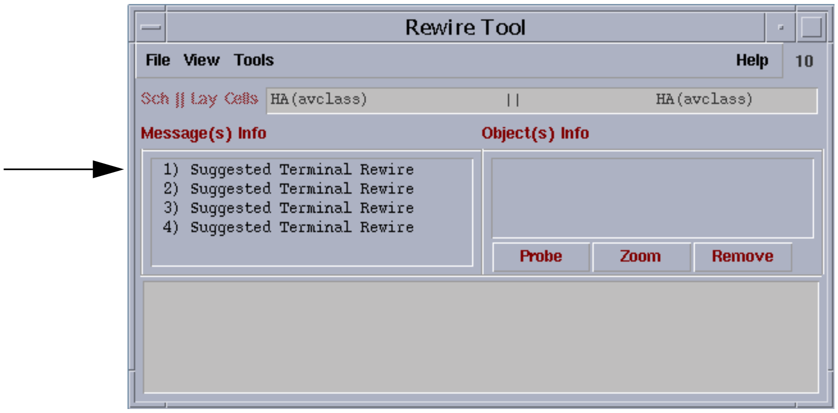


```
nand2(stdcells)     ||     nand2(stdcells)    Rewires     4
inv(stdcells)       ||       inv(stdcells)    Nets        2
nor2(stdcells)      ||      nor2(stdcells)    Devices     4
HA(avclass)         ||         HA(avclass)    Pins        0
                                              Parameters  0
```

   The Rewire Tool form comes up.

2. Reduce the size of the layout and schematic windows and position them so you can see both of them.

3. Select the first message in the window. You will be given suggestions about how to fix the crossed nets.



Sch || Lay Cells HA(avclass) || HA(avclass)

```
1) Suggested Terminal Rewire
2) Suggested Terminal Rewire
3) Suggested Terminal Rewire
4) Suggested Terminal Rewire
```

4. Select **B(lay terminal)** and click **Probe**.

   The net highlights in schematic and layout views. The comment window states the layout terminal B should connect to net B.

5. Select **I26 (lay device)** and click **Probe**.

   You can now see the *nand2* instance. The error has been made easily visible. The net is not connected to the proper *nand2* device.

Probing the Nets

1. Select **Assura — Open Schematic Cell**.

2. Resize the schematic and layout to half a screen each and position them side-by-side so you can see the cross-probes.

Verifying Connectivity

1. From the Rewire Tool form, select **Tools — Probe Net**.

2. Select a net in the schematic.

   You see it cross-probed in the layout.

3. Select **Tools — Remove All Probes**.

4. Select **Tools — Probe Device**.

5. Select one of the symbols in the schematic.

   The *stdcell* level is highlighted.

6. In the layout window press **Shift-f** to view all levels.

7. Select **Tools — Probe Device** and select one of the *layout P transistor* recognition layers.

   At this moment in this cell, the transistors are unmatched because we miswired the *A* and *B* nets. The CIW states *no mapping to schematic device*.

   Note: *We will provide more LVS debugging in a later module.*

8. Exit the DFII session.

Running an LVS Check without the Graphic User Interface

1. Look at the files in the *LVS* directory and familiarize yourself with them. The files present are:

   ```
   FA_vldbrules
   .lvs.fa.state
   cds.lib
   .cdsinit
   display.drf
   user.rsf
   lvsrsf
   ```

   The *FA_vldbrules* file contains the instructions to create a netlist from a DFII schematic. The *lvsrsf* file is an ASCII file and contains the commands for Assura.

2. Create the schematic netlist. In an xterm enter:

   ```
   dfIIToVldb FA_vldbrules
   ```

   This command creates the *FA.vldb* file as specified in the *FA_vldbrules* file. This is a binary file.

3. Start the LVS. In an xterm enter:

```
assura lvsrsf >& lvs.log
```

Getting Familiar with the LVS Steps

1. When the LVS has completed view the log file *lvs.log*.

2. Notice the steps.

   These are the DRC steps where Assura reads the layout database and processes the data according to the rules in the rules file.

   Assura names the internal layout database *<runName>.dat*. This database contains the original layout data and the extracted data.

3. Notice the lines starting with:

   ```
   Executing: extractMOS("nfet"...
   ```

   Assura is extracting the devices. The log file lists the number of devices extracted by Assura in each cell.

4. Notice the line:

   ```
   Starting ... avnx lvsrsf -exec1 -LVS
   ```

   Here Assura is converting the extracted layout to a netlist format. The extracted layout is part of the *<runName>.dat* file. The binary netlist file name is *<runName>.ldb.*

5. Notice the lines:

   ```
   Starting ... nvn lvsrsf -exec1
   ...
   Reading schematic network
   ```

   Here the program *nvn* is comparing the layout netlist to the schematic netlist.

   Notice the input to *nvn*. It is your Assura command file *lvsrsf*.

   The *nvn* program gets the names of the layout netlist from the value of the keyword ?*runName* and the name of the schematic netlist from the *avCompareRules* section.

   The run name is *lvs*. Assura created the netlist *lvs.ldb* from the extracted database *lvs.dat* during the *avnx* step.

   The *avCompareRules* section defines the schematic netlist as:

   ```
   (vldb "FA.vldb")
   ```

   Later modules in this course will cover more detail about the netlister.

6. Exit your viewer.

Getting Familiar with the Output Files

1. In an xterm enter:

   ```
   ls lvs.*
   ```

   You see all the output files created by Assura. Some are created during the extraction step and some are created during the comparison step.

2. Look at the LVS report file containing the matched cell statistics. Enter:

   ```
   more lvs.csm
   ```

   This file contains the following lines:

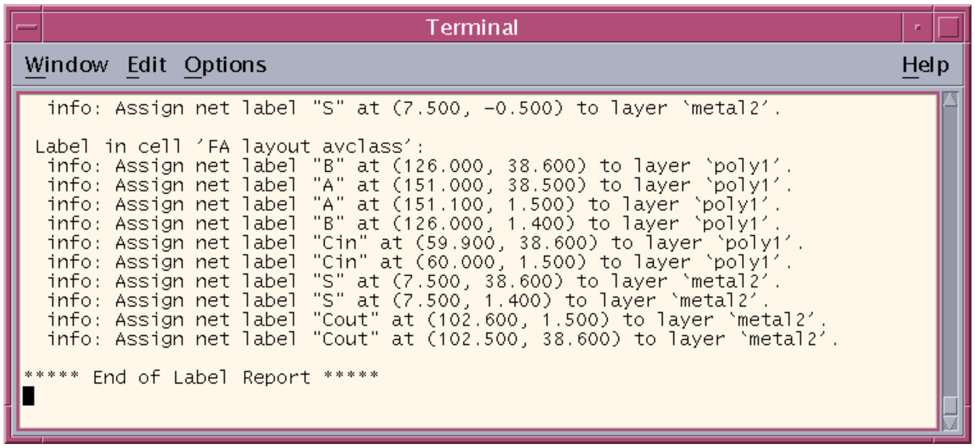
```
Schematic          | Layout                         | Status
------------------------------------------------------------------------
nand2              | nand2 layout stdcells          | matched
inv                | inv layout stdcells            | matched
nor2               | nor2 layout stdcells           | matched
HA                 | HA layout avclass              | matched
FA                 | FA layout avclass              | matched,pin errs *

Mismatch between Schematic and Layout
   1 cell matched with pin errors
   4 cells matched
```

The file shows that all the cells matched but the *FA* cell has pin errors. This means that some of the input and output signals names on the layout were different than those on the schematic or were missing on the layout.

One reason for pin errors is that the layout does not have labels for all the pins for the top block. This is common when a layout is only partially complete. In this case the *FA* layout labels are not considered pins. Texting layout will be discussed in a later module.

3. Look at the label report file *lvs.erc*:

   ```
   tail lvs.erc
   ```

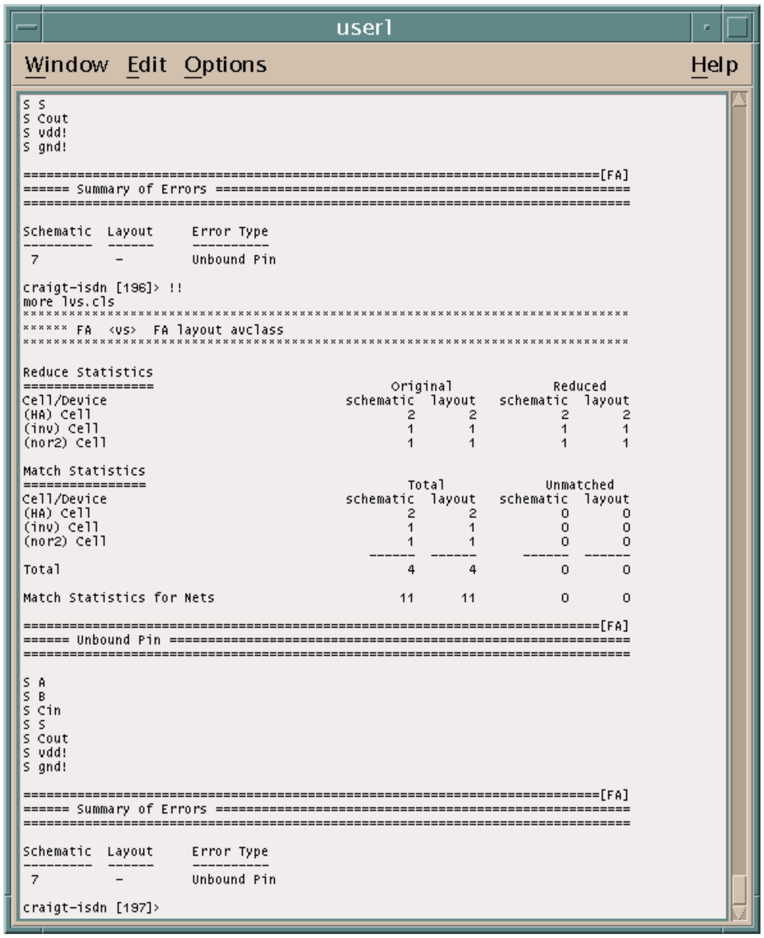


```
  info: Assign net label "S" at (7.500, -0.500) to layer `metal2'.

 Label in cell 'FA layout avclass':
  info: Assign net label "B" at (126.000, 38.600) to layer `poly1'.
  info: Assign net label "A" at (151.000, 38.500) to layer `poly1'.
  info: Assign net label "A" at (151.100, 1.500) to layer `poly1'.
  info: Assign net label "B" at (126.000, 1.400) to layer `poly1'.
  info: Assign net label "Cin" at (59.900, 38.600) to layer `poly1'.
  info: Assign net label "Cin" at (60.000, 1.500) to layer `poly1'.
  info: Assign net label "S" at (7.500, 38.600) to layer `metal2'.
  info: Assign net label "S" at (7.500, 1.400) to layer `metal2'.
  info: Assign net label "Cout" at (102.600, 1.500) to layer `metal2'.
  info: Assign net label "Cout" at (102.500, 38.600) to layer `metal2'.

***** End of Label Report *****
```

Notice that there are no warning messages. This informs you that there are no shorts or opens between labelled nets.

4. Look at the *lvs.cls* file.

```
more lvs.cls
```



```
S S
S Cout
S vdd!
S gnd!

==========================================================================[FA]
====== Summary of Errors =====================================================
==============================================================================

Schematic  Layout     Error Type
---------  ------     ----------
 7          -         Unbound Pin

craigt-isdn [196]> !!
more lvs.cls
******************************************************************************
****** FA  <vs>  FA layout avclass
******************************************************************************

Reduce Statistics
=================
                                               Original             Reduced
Cell/Device                               schematic  layout   schematic  layout
(HA) Cell                                         2       2           2       2
(inv) Cell                                        1       1           1       1
(nor2) Cell                                       1       1           1       1

Match Statistics
================
                                                 Total             Unmatched
Cell/Device                               schematic  layout   schematic  layout
(HA) Cell                                         2       2           0       0
(inv) Cell                                        1       1           0       0
(nor2) Cell                                       1       1           0       0
                                             ------  ------      ------  ------
Total                                             4       4           0       0

Match Statistics for Nets                        11      11           0       0

==========================================================================[FA]
====== Unbound Pin ===========================================================
==============================================================================


S A
S B
S Cin
S S
S Cout
S vdd!
S gnd!

==========================================================================[FA]
====== Summary of Errors =====================================================
==============================================================================

Schematic  Layout     Error Type
---------  ------     ----------
 7          -         Unbound Pin

craigt-isdn [197]>
```

Look at the *Original* and *Reduced* device counts for each cell.

The number of devices in the schematic and layout are the same. When you have a cell mismatch your first debugging step is to look at the original and reduced device counts and see if they match.

Lab Summary

You will be debugging LVS reports in later labs. This lab introduced you to the LVS procedure, the report files, and the use of the graphic user interface to identify the unmatched cells and view the reports.

Always remember to check the following files before you start to debug your LVS errors:

- The *<runName>.csm* summary file.

  Correct the lower level cells first. (The *.csm* file orders the cells from bottom to top.)

- The *<runName>.erc* label report file.

  Correct any open or short circuits (WARNINGS in the *.erc* file).

- The *<runName>.cls* detailed error report file.

This lab also showed you how to probe for devices and nets using two methods:

- By probing a shape

- By giving the name of the net or device.

When you have an LVS error involving a device or a net, you can use these probing mechanisms to identify the device or net and begin your investigation into the problem.

End of Lab

Appendix A

Using Diva to Verify Your Designs

Lab A-1 Design Rule Checking

Objective: Run a Flat DRC and Analyze Errors.

You will run a design rule check of *pk445chip* to check the layout for physical design constraints and to correct construction flaws.

1. Change to the working directory in an xterm window. Enter:

   ```
   cd Diva
   ```

2. Enter in the xterm window:

   ```
   setenv CDS_Netlisting_Mode Analog
   ```

   Note: *The environment variable CDS_Netlisting_Mode tells the Cadence software which netlist format should be used - either Analog or Digital.*

3. In an xterm window, enter:

   ```
   layoutPlus &
   ```

4. From the CIW, select **File** — **Open**, then enter the following values in the form:

Library Name	`DIVA`
Cell Name	`pk445chip`
View Name	`layout`

Using Flat DRC

1. Stretch the CIW top edge up two inches.

2. Select **Verify** — **DRC**.

3. Set the DRC form:

Checking Method	`flat`
Checking Limit	`full`

   a. Click the **Set Switches** button.

   b. Select *poly* in the list box and click **OK**.

4. Start the DRC. Click **OK**.

   Commands from the DRC procedure scroll in the CIW. Errors flash on the screen and are listed in the CIW:

   Note: *DRC Errors in Diva appear on the marker layer, error purpose.*

   Note: *The statements under "Violated Rules" are taken from the message field of the* ***drc*** *command.*

```
********* Summary of rule violations*****
# errors Violated Rules
  12  3b: minimum poly spacing = 0.6
   6  3a: minimum poly width = 0.6
  18  Total errors found
```

Analyzing DRC Errors with Find

1. Select **Verify** — **Markers** — **Find**.

   If no errors are found in the design, a dialog box appears. Click on **Close**, check the form contents described above and run DRC again.

2. Select **Zoom To Markers** in the form.

3. Click **Next**.

4. If you cannot see shapes related to the error, press **Shift [f]**. This zooms in on errors sequentially and reports error information to the marker text window. You can use **Previous** to search backwards through the list.

   When you get to the last error, a dialog box appears. Click on **Close**.

5. Click **Delete** in the Find Marker form.

   The marker is deleted and the window zooms in on the previous error region (unless the error you deleted was the first one found).

6. Click on **Cancel** the Find Marker form.

Running DRC on All Interconnect

1. Select **Verify** — **DRC**.

2. Click the **Set Switches** button.

3. Select **allInterconnect** in the list box.

4. Click **OK** in the list box.

5. Start the DRC. Click on **OK**.

   Wait until DRC finishes.

6. Select **Verify** — **Markers** — **Explain**.

7. Click on any flashing marker.

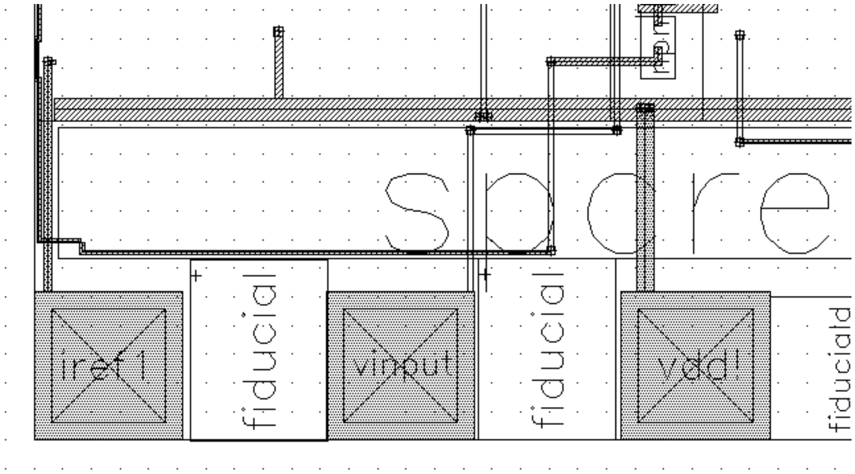
   The Info box appears describing the error that pertains to the marker.

Excluding Cells from DRC or Extraction

Note: *The fiducial cell is a test cell that contains data that deliberately fails the design rules. This type of cell is often used to monitor the manufacturing process. This task prevents it from being checked.*

1. Select **Window** — **Fit All [f]** and press **Control [f]**.

2. There are two placements of the *fiducial* cell. Select one of *fiducial* cells on the bottom-left side of the design.



3. Select **Design** — **Hierarchy** — **Descend [X]**.

   The window banner should read, *Virtuoso Layout Editing: DIVA fiducial layout.*

Add the *ivIncludeValue* Property

You will add the *ivIncludeValue* property to prevent DRC from checking shapes in the *fiducial* cell:

1. Select **Design** — **Properties [Q]**.

2. Click on the **Property** button.

3. Click the **Add** button in the Edit Cellview Properties form.

4. Set up the Add Property form:

Name	`ivIncludeValue`
Type	`Int`
Value	`2`

Caution

Set the type of a property before entering its value. If you do not, the tool may reset the value after you set the type.

5. Click **OK** in the Add Property form.

6. Click **OK** in the Edit Cellview Properties form.

7. Select **Design** — **Hierarchy** — **Return [B]**, to return to the top-level of the design.

8. Click **Yes** in the dialog box to save the changes you just made.

Run DRC Again

1. Set the **Inclusion Limit** to **0** in the DRC form.

2. Run DRC with the switch set to *allInterconnect*.

   No errors appear in the *fiducial* cell.

Deleting DRC Errors

1. Select **Verify** — **Markers** — **Delete All**.

   The Delete All form appears.

2. Click **OK**.

   All error markers are removed from the window.

Lab A-2 Area DRC

Objective: Run area DRC.

Using Area DRC with *drcZeroHalo*

1. The *pk445chip layout* from the *DIVA* library should be open in edit mode.

2. Select **Window** — **Fit All [f]**.

3. Press **Control [f]** to view top-level data only.

4. Select **Verify** — **DRC**.

   Note: *Area can also be specified by typing coordinates or by clicking* ***Sel by Cursor****, then entering a box in the window.*

5. Click on the **by area** button.

6. Turn on **Run-Specific Command File**.

7. In the Run-Specific Command File text field, enter:

   ```
   zeroHalo
   ```

   The Diva verification tool calculates a "halo" distance around the area to capture all interactions that might impact the correctness of the processing. This halo is based on the maximum distance that can cause interactions or relationships as defined by the dimensions in the design rules.

8. Click the **Set Switches** button.

9. Select *all* in the list box and click **OK**.

10. Click **OK** in the DRC form.

11. Follow the prompts in the *layout* window to enter a box over the left *amplifier* cell area with the left mouse button.

    Note: *Only the area specified in the window is checked.*

12. Follow the prompts:

    ```
    enter first point of box
    enter second point of box
    ```

    Notice that errors are found only in the area specified.

13. Zoom in around the *metal2* path connected to the *vinput* pad. This path is too narrow and when you ran full DRC, the entire shape was covered by a marker.

    Why is the marker not covering the entire shape now?

14. Try this several times in other areas of the design.

    Note that previous errors remain displayed.

15. Select **Window** — **Fit all [f]** to display all levels of data.

16. Use **Verify** — **Markers** — **Find** to look at the errors you found.

17. Select **Verify** — **Markers** — **Delete All**.

    The Delete All form appears.

18. Click **OK**.

    All errors are removed from the window.

Lab A-3 Hierarchical DRC

Objective: Run hierarchical DRC.

Using Hierarchical DRC

1. In *pk445chip layout* from the *DIVA* library, press **Control [f]**.

   Only top-level data and instances are visible.

2. Select **Verify** — **DRC**.

3. In the DRC form, select **hierarchical** as the Checking Method.

4. Select **full** as the Checking Limit.

5. Turn **off** the boolean button for Run Specific Command file.

6. Click the **Set Switches** button.

7. Select *all* in the list box and click **OK**.

8. Click on **OK** to run the checks.

   Errors appear in the design.

Hierarchical Error Analysis

1. Stretch the CIW to see the DRC log.

   Note: *Each cell in the hierarchy is checked. Pcells are flattened before checking and are not listed.*

   ```
   Running drclayout analysis
   ...
   Maximum halo distance is 20
   Doing: DIVA/npn/layout
   Doing: DIVA/capacitor/layout
   ```

```
Doing: DIVA/pnp/layout
Doing: DIVA/amplifier/layout
Doing: DIVA/nand2/layout
Doing: DIVA/control/layout
Doing: DIVA/pk445chip/layout
```

Analyze Hierarchical DRC Errors with Explain

1. Click left on one of the *control* instances.

2. Select **Design** — **Hierarchy** — **Edit In Place[x]**.

   The *control layout* is now being edited in the window with the hierarchical DRC error stored in the cell. While you could see those markers when viewing the top-level design, they are actually stored in the lower-level cell.

3. Select **Verify** — **Markers** — **Find**.

   Note: *If no errors are found in the design, a dialog box appears. Click on* ***Close****.*

4. Click **Next** to see the first error.

   ```
   location: ("DIVA" "control" "layout")
   reason: 3a: minimum poly width = 0.6
   ```

5. Click **Next** to see the other errors and then cancel the find markers command.

Fix the Errors and Re-run DRC

1. Select the *poly* path connected to the A pin of the nand2.

2. Select **Edit** — **Properties [q]**.

3. Change the **Width** of the path to **0.6** and click **OK**.

4. Select **Design** — **Hierarchy** — **Return [B]**.

   Click **Yes** in the dialog box to save the design. The window returns to *pk445chip layout*.

5. Run hierarchical DRC again.

   The *poly* errors in *control layout* disappear.

6. When you are done, select **Verify** — **Markers** — **Delete All**, and in the form that appears, change **Search Scope** to **hierarchy starting from top cellview**.

7. Click **OK**.

End of Lab

Lab A-4 Extraction

Objective: Run extraction.

Running Extraction

1. Select **Verify** — **Extract** in *pk445chip layout* cellview from the *DIVA* library.

2. Set the Extractor form:

Extract Method	flat

Note: *If the extracted design is not open, the results are written to disk. The resultant extracted view is not displayed. It must be opened as a standard cellview. If an extracted view is currently open it will be updated when extraction finishes*

3. Click **OK**.

   Extraction begins. As it runs, the commands executed are echoed in the CIW. When extraction is done, you see a final message in the CIW:

```
    Total errors found: 0
```

4. Iconify the *layout* view for future use.

Exploring the Extracted View

1. Open *pk445chip extracted* cellview from the *DIVA* library.

2. Zoom in around the control section between the *in2* and *vcap* pads at the top of the design.

   Shift [f] to view the symbols for the *nmos* and *pmos* devices indicating that the designed devices have been extracted.

3. Select **Options** — **Display [e]**.

4. In the Display Options form, turn on **Nets** in the Display Controls section and click **Apply**.

   Notice the flightlines from the pins on the *nmos* symbol indicating that the connectivity to this device has been extracted.

5. When you are done looking at the flightlines, turn off **Nets** in the Display Controls section and click **OK**.

6. In the *extracted* view, select **Verify — Probe**.

   Note: *Only extracted or schematic views can be probed.*

   The Probe form appears.

7. To probe nets by cursor:

   a. Select **Add Net**.

   b. Click on the *vinput* bonding pad.

   The entire *vinput* net is highlighted by a yellow outline.

   Note: *Probing uses hilite layers.*

8. To probe nets by **name**:

   a. Select **Add Net**.

   b. Enter the net name in the CIW (include double quotes):

   ```
   “vdd!”
   ```

   The entire *vdd!* net is highlighted.

9. To explain probes:

   a. Select the **on CIW** option for **Explain**.

   Note: *All explanations of the probe appear in the CIW.*

   b. Click **Explain**.

   c. Click on the *vdd!* bonding pad.

   The CIW displays the name of the net:

   ```
   Object Type: net
   Object Name: /vdd!
   ```

10. To remove probes:

    a. Select **Remove Net**.

    b. Point to the *vdd!* bonding pad.

    Only the highlighting of *vdd!* is removed.

    c. Select **Remove All**.

    The hilighting of all the nets is removed.

11. Cancel the **Probing** form.

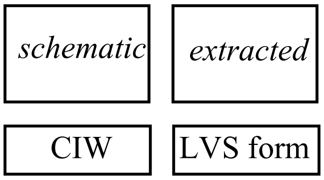
End of Lab

Lab A-5 Layout Versus Schematic

Objective: Run LVS and find errors.

Running LVS

1. Open *pk445chip extracted* and *pk445chip schematic* cellviews from the *DIVA* library.



Set your windows up as shown.

If you already have the *extracted* view open and iconified, de-iconify the *extracted* view.

2. In the *extracted* view, select **Verify** — **LVS**.

   You may get a dialog box that LVS is not a directory. If you do, click **Close** in the dialog box.

3. Set up the LVS form.

   a. Use the default Run Directory Name in the LVS form:

   `LVS`

   b. Make sure the designs you opened appear in the LVS form. If they do not, select the designs for LVS by clicking on the **Sel by Cursor** buttons. Follow the prompts in the CIW and point to the *extracted* and *schematic* views.

   Note: *The LVS form contains the name of the extracted cellview it was called from and the corresponding schematic cellview by default.*

4. Make sure the *Rules File* section specifies the following:

   `divaLVS.rul`

5. In the *Rules Library field*, select **design**.

6. Click **Run** in the LVS form.

   A dialog box may ask you to save cellviews that were previously edited. Click **OK** in the dialog box.

Caution

If you do not ***OK*** *the dialog box, the LVS program does not start.*

7. The LVS program begins. In the CIW, you see:

```
    LVS job is now started.
```

Watching the LVS Run

1. Click **Info** in the LVS form.

   The Display Run Information form appears.

2. Click on **Log File** for **Run Info**.

   A text window appears and updates as the LVS program runs. When the job is done, scroll to the bottom of the form. Notice that the netlists fail to match and that you have mismatched terminals and nets.

3. Click **OK** in the **Analysis Job Succeeded** dialog box.

4. Close the log file window.

Finding Unmatched Device and Terminals Errors

1. Move your cursor into the extracted design window and press the **[Esc] key**.

   This makes the extracted layout window current.

2. Click **Error Display** in the LVS form.

3. In the LVS Error Display form, click on the **Display All** button.

   Note: *The displayed errors are highlighted on the hilite layer. All errors are displayed by default.*

   The *vss!*, *gnd!* and *vdd!* pads are highlighted in yellow. These are the unmatched terminals.

   The nmos and pmos devices in the device *fiduciald* cell placed between the *gnd!* pad and the *vdd!* pad at the bottom of the design are highlighted in yellow. These are the unmatched instances.

4. In the LVS Error Display form, click on **Explain** and then click on either of the highlighted devices.

   Explanation of Terminal failure appears in the LVS Error Display form. The explanation is cut off in the form.

5. To read the complete explanation, click on **Info** in the LVS form.

   The Display Run Information form appears.

6. In the **Extracted** section of the form, click on **Bad Terminals**.

   A text file appears with bad terminal information.

7. Select **File** — **Close Window**.

8. In the **Extracted** section of the form, click on **Bad Devices**.

   A text file appears with bad device information.

   /+42 and /+16 are the internal device names.

9. Select **File** — **Close Window** when you are done reading.

10. In the LVS Error Display form, click on **Clear Display** and cancel the LVS Error Display form.

Fix the Unmatched Errors by Excluding the Device fiduciald Cell

None of the fiducial cells have any corresponding information in the schematic. These cells should not be included when running LVS.

1. Open *fiduciald layout* for edit.

2. Select **Design** — **Properties**.

3. Click on the **Property** button and verify that the *ivIncludeValue* property is 5.

4. Close *fiduciald layout*.

Extract and Run LVS Again

1. Close the *pk445chip extracted* cellview and open the *pk445chip layout* cellview.

2. Extract *pk445chip layout* in **flat** mode with the Inclusion Limit set to **0**.

3. Open the *pk445chip extracted* and check to make sure the design has been correctly extracted.

   The shapes in either *fiducial* or *fiduciald* should not have appeared in the *extracted* design.

4. In the LVS form, turn off **Create Netlist** for **schematic**.

   The schematic netlist is correct and does not need to be created again.

5. Click **Run** in the LVS form.

   A dialog box appears because the extracted view has changed and needs to be saved. Click **OK**.

   LVS begins.

6. Click **Info** in the LVS form.

   The Display Run Information form appears.

7. Click on **Log File** in the Run Info section.

   A text window appears. When the job is done, notice the errors that were found:

```
The net-lists failed to match.
      ....
           layoutschematic
               instances
un-matched     00
rewired        70
size errors    00
pruned         00
active         6145
total          6145
     ....
               nets
un-matched     00
merged         01
pruned         00
active         2930
total          2930
      ....
               terminals
un-matched     00
matched but
different type 00
total          99
```

   Notice that there are no unmatched instances or nets now. There is only a merged net error.

8. Click **OK** in the **Analysis Job Succeeded** dialog box.

9. Close the log file window.

Finding the Schematic Merged Net Error

A merged net in the schematic typically means you have two nets shorted in the layout. To find the nets involved, you need to display the errors in the schematic.

1. Click **Error Display** in the LVS form.

2. Move to the schematic window working area and press **[Esc]**.

   This makes the schematic window current. Because the error was reported in the schematic portion of the LVS output, it can only be displayed in the schematic.

3. In the form, turn all buttons off, except **Merged nets**. Then in the *Display* section click on **First**.

   The explanation is shown in the LVS Error Display form (the net names may vary):

   ```
   Net /net065 merged with /net44
   ```

   Net *net065* and *net44* in the schematic are both highlighted.

4. Cancel the **LVS Error Display** form.

Cross-Probing between Schematic and Extracted Views

The merged net cannot be cross probed because it did not match. The net it was merged with can be cross probed.

1. Display the entire *extracted* view in the window.

2. Select **Verify** — **Probe** from the *extracted* view.

3. Move to the schematic window and press **[Esc]**.

   This makes the schematic window active.

4. Cross-probe using the following steps:

   a. Click **cross probe matched** for the *Probing Method.*

   b. Click **Add Net**.

   c. Click on *net44* in the schematic.

      The corresponding net is highlighted in the *extracted* view. The error is on this net. *Do you see it?*

      Probably not, because this is a relatively large net.

5. Click on **Remove All** in the Probing form.

6. To find the error use the following steps:

   a. In the Probe form, click **Add Dev**.

   b. Click on the npn in the schematic that is connected to *net065*.

   c. Repeat steps a and b and cross probe the other npn devices that are to be connected to *net065*.

   d. Look in the extracted layout and notice that the *net065* should only be connected to the two npn's you probed. It is also connected to another net.

   e. If you still don't see the error, look at the end of the lab for the answer.

7. Click **Remove All** and cancel the Probing form.

Fix the Short

1. Open the *pk445chip* layout view and press **Control [f]**.

2. In *pk445chip layout*, delete the rectangle piece of *metal1* that is on top of the two *npn* instances, causing the short.

3. Rerun extraction and LVS.

4. Look at the log.

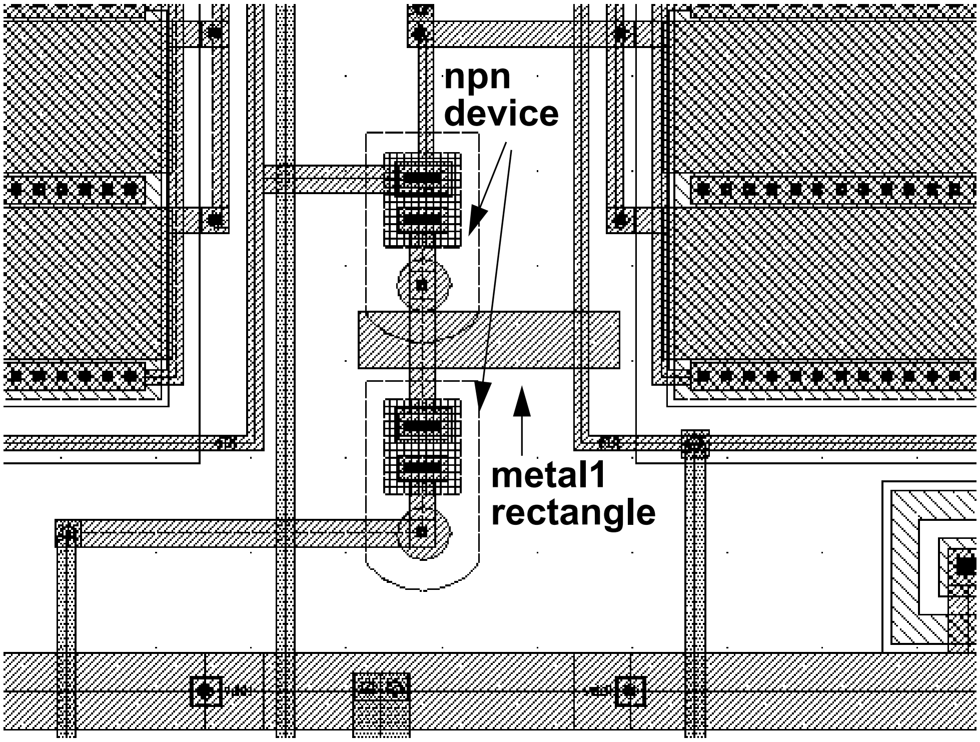
   When the job is done, notice the result:

```
The net-lists match.
      ....
           layoutschematic
               instances
un-matched     00
rewired        70
size errors    00
pruned         00
active         6145
total          6145
     ....
               nets
un-matched     00
merged         00
pruned         00
active         3030
total          3030
      ....
               terminals
un-matched     00
matched but
different type 00
total          99
      ....
```

5. Close the LVS form and the schematic, layout and extracted windows.

Answers - Finding the Schematic Merged Net Error

The short is caused by a rectangular piece of *metal1* that was accidentally placed over the *npn* instances. It causes a short between *net065* and *net44*



It may be easier to find if you press **Control [f]** to view only shapes at the top level of the design and instances.

End of Lab

Appendix B

Introduction to Relative Object Design

Lab B-1 Creating Aligned Rectangles

Objective: This exercise demonstrates how to create simple rectangles and enforce alignment between them using Relative Object Design constructs.

In this exercise, you will use Relative Object Design (ROD) constructs to create several rectangles and enforce alignment between them. This will demonstrate the application of two key ROD functions, *rodCreateRect* and *rodAlign*. The calls to these functions have been captured in files, and an automated mechanism has been provided to load this SKILL code into the design framework session. Each step of the activity displays the SKILL code involved in the step in a text viewer window. You need only select items from a special menu created for this lab, called **ROD Introduction** in the layout editor window.

Start a Design Framework II Session

1. Enter these commands into a terminal window:

   ```
   cd rodIntro
   layout &
   ```

   After a few moments the Command Interpreter Window (CIW) appears followed by a new (empty) layout editor window for the cellview, *examples rectangles layout*.

2. You will see a menu item in the banner of this window called **ROD Introduction**. Click on this menu item and choose **Begin activity**.

Create your First ROD Rectangle

1. A window appears containing SKILL code that you will execute for this step. To execute the code, select the menu item **ROD Introduction** — **Execute code**.

   This code assigns the current cellview object to a SKILL variable called *CV*. This variable is used by the ROD functions called in this exercise.

   After you select the **Execute code** menu item, the source code displayed in the window will be printed into the CIW indicating that this step is complete.

2. To verify this step, enter the following into the CIW:

   ```
   cv
   ```

   You see the printed representation of the cellview returned as the value for the variable. It will look similar to this:

   ```
   db:17099820
   ```

   Note: *The number following the db: prefix will be different with each DF session.*

3. To proceed to the next step, choose the menu item **ROD Introduction — Go to next step**.

   Notice that another menu item is now enabled under the **ROD Introduction** menu. You can choose **ROD Introduction — Go to step** to select any step in the activity.

4. After you select the **Go to next step** command, a new window appears with the SKILL code for this step.

   Examine the code which creates a new ROD object, a simple rectangle, in the cellview window. Try to answer these questions as you read the code:

   a. Which ROD function is used here?

   b. How does the function know in which cellview to add the rectangle?

   c. How are the dimensions and location for the rectangle specified?

5. How is the layer specified? Select **ROD Introduction — Execute code** to execute the ROD function shown in the code viewing window.

   A rectangle appears in the layout editor window. Congratulations! You have created your first ROD object.

6. The ROD object ID for the new rectangle has been assigned to the SKILL variable *Gate*. Enter the following into the CIW:

   ```
   Gate
   ```

   You will see the ROD object's printed representation, which looks something like this:

   ```
   rodObj:18972696
   ```

   Note: *The number following the rodObj: prefix will be different with each DF session.*

7. Enter the following into the CIW:

   ```
   Gate~>??
   ```

   Inspect the results to gain an understanding of the information contained in a ROD object.

Create Your First Alignment Between ROD Objects

1. Select **ROD Introduction** — **Go to next step**.

   A new code viewing window appears.

2. Examine the code and try to answer these questions:

   a. What type of ROD objects will be created?

   b. Where will they appear and on which layer?

3. Select **ROD Introduction** — **Execute code** and observe what happens in the layout editor window. Are the results as you expected?

4. Select one of the newly created shapes. Now, move the shape to a new location in the layout editor window. What happens to the other ROD shapes?

5. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and try to discern what this ROD function will do.

6. Select **ROD Introduction** — **Execute code** and observe what happens to the shapes in the layout editor window.

7. Select the shape that moved. As in the previous step, move this item to a new location in the layout editor window. What happens?

   The alignment created by *rodAlign* causes the two shapes to behave as a single entity. The smaller shape is now constantly aligned to a specific point on the original rectangle. Congratulations! You have created your first set of ROD-aligned objects.

Create and Align More Rectangles

1. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and try to answer these questions:

   a. How many shapes will be created?

   b. What kind of shapes will they be?

2. Select **ROD Introduction** — **Execute code** and observe what happens in the layout editor window. Did you answer the above questions correctly?

3. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and try to answer these questions:

   a. Which shapes will be affected by ROD functions in this step?

   b. What will happen to them?

   c. Which shapes will not be affected?

4. Select **ROD Introduction** — **Execute code** and observe what happens in the layout editor window. Click left on one of the shapes affected during this step. Drag it around as described in previous steps. What happens to the other affected shapes? What happens to the shapes that were not involved in this step?

5. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and try to determine what will happen to the entire collection of rectangles in the layout editor window.

6. Select **ROD Introduction** — **Execute code** and observe what happens. Click left on any rectangle in the editor window and then drag it as before. What happens? Is this what you expected?

   You now have two structures, each made of three rectangles. As any one (or more) of these individual rectangles is moved to a new location, both structures will relocate relative to the moved rectangle(s). Spacing and orientation of the structures will remain fixed as this happens.

7. You will see a message like this in the CIW:

   ```
   ***                    ***
   *** Activity Complete  ***
   ***                    ***
   ```

   This marks the end of this portion of the lab. You may use the **ROD Introduction** — **Go to step** menu item to repeat any or all of this exercise if you desire.

   When you are ready, make sure you are at the last step in the activity (look for the message in the CIW shown above), and then choose **ROD Introduction** — **Next Activity**. This will iconify the current editor window and expose an editor window for the next activity.

Important

*You must choose **ROD Introduction — Next Activity** before proceeding to the next activity.*

Important

Do not close (destroy) the editor window for any activity.

Lab B-2 Using ROD in a SKILL Procedure

Objective: This exercise demonstrates how to use ROD constructs in a procedural fashion to automatically create representations of physical structures.

In this exercise you will define a SKILL procedure that uses ROD constructs to create a transistor structure. The function will accept an argument to specify the type of transistor to create (N or P). This function also demonstrates accessing technology file data (such as layer spacing rules) and using this data in construction of the transistor device.

As in the previous exercise, you will use items under the special **ROD Introduction** menu in the current layout editor window to navigate through the constituent steps.

Define the Transistor Generation Procedure

A layout editor window will be opened for the cellview *example ptran layout*. If this is not the case, see your instructor for assistance.

1. Select **ROD Introduction** — **Begin activity**.

   A code viewing window appears containing a somewhat lengthy SKILL procedure statement for a function named *tran*.

2. Examine this code, but do not be intimidated by it -- you are not required to understand exactly how it functions. Nonetheless, see if you can determine any of the following:

   a. What arguments are there to *tran*?

   b. Which ROD functions are used?

   c. What information is obtained from the library technology data and how is the information used?

3. When you are ready, select **ROD Introduction** — **Execute code**. This loads the definition of *tran* into the current design framework session, but does not execute it.

Create a P-type Transistor Using *tran*

1. Select **ROD Introduction** — **Go to next step**. Again, as in the first activity, this step assigns the cellview object in the current window to a global variable called *CV*. This variable will be used as you call the *tran* function.

2. Select **ROD Introduction** — **Execute code**.

3. Select **ROD Introduction** — **Go to next step** and examine the code in the viewing window. The call to *tran* is quite simple. Notice the arguments passed to it.

   a. What is the first argument?

   b. What will be the transistor’s width and length?

   c. Which argument determines the type (P or N) of the transistor?

4. Select **ROD Introduction** — **Execute code** and observe the results in the layout editor window.

5. Select the transistor and move it to a new location in the layout editor window. What happens?

6. In the banner of the layout editor window, select **Edit** — **Other** — **Rotate**. You are prompted to choose a reference point for the operation.

7. Left-click in the center of the transistor. As you move the mouse pointer, a highlighted outline will show you the amount of rotation.

8. Left-click again when the rotation indicates 90 degrees. What happens to the transistor?

9. You will see a message like this in the CIW:

   ```
   ***                    ***
   *** Activity Complete  ***
   ***                    ***
   ```

   This marks the end of this portion of the lab. You may use the **ROD Introduction** — **Go to step** menu item to repeat any or all of this exercise if you desire.

   When you are ready, make sure you are at the last step in the activity, and then choose **ROD Introduction** — **Next Activity**. This will iconify the current editor window and expose an editor window for the next activity.

Important

*You must choose **ROD Introduction — Next Activity** before proceeding to the next activity.*

Important

Do not close (destroy) the editor window for any activity.

Create an N-type Transistor Using tran

A layout editor window will be opened for the cellview *example ntran layout*. If this is not the case, see your instructor for assistance.

1. Select **ROD Introduction** — **Begin activity**.

2. A familiar operation appears in the code viewing window. Again, as in the previous activity, this step assigns the cellview object in the current window to a global variable called *CV*. This variable will be used as you call the *tran* function.

3. Select **ROD Introduction** — **Execute code**.

4. Select **ROD Introduction** — **Go to next step**. Since the *tran* function was defined in the previous activity, you can simply call it. Hence, the code for this step is a call to *tran*. Examine this code and determine the following:

   a. What will be the dimensions of the N-type transistor?

b. What determines the type of the transistor?

5. Select **ROD Introduction** — **Execute code** and observe the results in the layout editor window. You may experiment with moving and rotating this transistor as in the previous activity if you so desire.

6. You will see a message like this in the CIW:

```
***                    ***
*** Activity Complete  ***
***                    ***
```

This marks the end of this portion of the lab. You may use the **ROD Introduction** — **Go to step** menu item to repeat any or all of this exercise if you desire.

When you are ready, make sure you are at the last step in the activity, and then choose **ROD Introduction** — **Next Activity**. This will iconify the current editor window and expose an editor window for the next activity.

Important

*You must choose **ROD Introduction** — **Next Activity** before proceeding to the next activity.*

Important

Do not close (destroy) the editor window for any activity.

Lab B-3 Using ROD to Create a Cell

Objective: This exercise demonstrates the use of ROD constructs in creating the physical design for a small cell.

In this exercise, you will see how to use instances of the transistors created in the previous lab to create a simple inverter. Also, you will see how you can use *rodCreatePath* to create objects with shapes on several layers.

A layout editor window should be opened on the cellview *example inv layout*. If this is not the case, see your instructor for assistance.

1. Select **ROD Introduction** — **Begin activity**.

   The code viewing window shows that you are assigning the current cellview object to the SKILL variable *CV*.

2. Select **ROD Introduction** — **Execute code**.

3. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewing window and consider these questions:

   a. What is accomplished by the first two SKILL function calls?

   b. Do the design entities used here appear familiar?

   c. What is accomplished by the final SKILL expression?

4. Select **ROD Introduction** — **Execute code** and observe the results in the layout editor window. You see an instance of each transistor structure created in the two previous activities.

5. Select **Options** — **Display** in the layout editor window. A form will appear with many settings for the layout editor display.

6. Locate the fields for **Display Levels** in the lower left portion of the form and change the **To** field from 0 to 1 (or a larger number, if you wish). Click **OK** in the upper-left corner of the form.

   Notice that the polygons created by the *tran* function in each of these sub-cells are in place as you left them in the previous activities.

7. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewing window and consider these questions:

   a. What kind of structures will be created?

   b. Are they electrically significant?

8. Select **ROD Introduction** — **Execute code** and observe the results in the layout editor window.

9. Select one of the new shapes. Choose **Edit** — **Properties** in the banner of the editor window. A property editor form appears.

10. Select the **Connectivity** option on the form, and note that the shape has a signal associated with it. Which argument to *rodCreateRect* designated this connection?

11. Select **Cancel** on the property editor form.

12. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and answer the following questions:

    a. How many alignments will take place?

    b. Which two objects will be aligned first?

    c. Is a direct alignment made between the transistor instances?

13. Select **ROD Introduction** — **Execute code** and observe the results in the editor window.

14. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and answer the following questions:

    a. On which layer will the new shape appear?

    b. Is the shape associated with any signal?

    c. To which object will it be aligned?

15. Select **ROD Introduction** — **Execute code** and observe the results in the editor window.

16. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window and answer the following questions:

    a. Which ROD function will be used to create the new object?

    b. How many shapes will the object consist of and on which layers will the shapes appear?

    c. What function will the object serve?

17. Select **ROD Introduction** — **Execute code** and observe the results in the editor window.

18. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window. Do you notice any access to information below the current level of hierarchy? What data is used here?

19. Select **ROD Introduction** — **Execute code** and observe the results in the editor window.

20. Select **ROD Introduction** — **Go to next step**. Examine the code in the viewer window. Which new object will become the output pin for the inverter?

21. Select **ROD Introduction** — **Execute code** and observe the results in the editor window. Congratulations! Your inverter cell is complete.

22. You will see a message like this in the CIW:

    ```
    ***                    ***
    *** Activity Complete  ***
    ***                    ***
    ```

    This marks the end of this portion of the lab. You may use the **ROD Introduction — Go to step** menu item to repeat any or all of this exercise if you desire.

    When you are ready, make sure you are at the last step in the activity, and then choose **ROD Introduction — Next Activity**. This will iconify the current editor window and expose an editor window for the next activity.

Important

*You must choose **ROD Introduction — Next Activity** before proceeding to the next activity.*

Important

Do not close (destroy) the editor window for any activity.

Lab B-4 Creating a Guard-ring Structure with ROD

Objective: This exercise demonstrates the use of ROD to create a guard-ring structure

In this activity, you load a simple SKILL program for creating a guard-ring structure. The function employs one call to *rodCreatePath* to perform this otherwise tedious task.

A layout editor window will be opened for the cellview *example guardring layout*. If this is not the case, see your instructor for assistance.

1. Select **ROD Introduction** — **Begin activity**.

   A code viewing window appears with the definition of a SKILL function called *digitizeGuardring*, a SKILL function called *createGuardring*, and a keybinding for the **F10** key.

2. Examine the code for *digitizeGuardring* and attempt to determine its purpose.

3. Examine the code for *createGuardring* and try to discern the following:

   a. Is technology information used?

   b. On which layer is the *master* path drawn?

   c. On which layer is the enclosed subpath drawn?

   d. Which part of the guard-ring structure does the *subRect* keyword argument specify? Does this create a single polygon?

4. Select **ROD Introduction** — **Execute code**.

5. In the layout editor window, use the **F10** key and the mouse to digitize the guard-ring structure.

6. Choose **Edit** — **Stretch** in the layout editor window.

7. Click on one end of the guard-ring and drag that end to a new location. Notice how the contact arrays and the metal and diffusion paths adjust accordingly.

8. Select the new guard-ring object by clicking on it.

9. Choose **Edit** — **Other** — **Chop**.

10. Click the mouse near a portion of the guard-ring and drag the selection rectangle through the guard-ring.

11. Release the mouse button. What happened? Look at the code for *createGuardring* and try to determine which keyword arguments control the behavior of the metal and diffusion layers.

Summary

In these exercises, you have seen how ROD shapes can be created and aligned. You also have seen how ROD can be used to create a low-level design entity (a transistor), complete cell (an inverter), and a multi-layer structure (a guard-ring).

You now have an insight to the capabilities provided by Relative Object Design.

End of Lab

Appendix C

Using the Strokes Editor

Lab C-1 Using Strokes

Objective: Learn to use strokes to enter commands.

Opening a Cellview

1. In the CIW, select **File** — **Open**, then enter the following values in the Open Design form:

Library Name	`design`
Cell Name	`tutorial`
View Name	`layout`

The *strokes layout* cellview opens for editing.

Loading the Strokes File from the *.cdsinit*

The Strokes Editor requires three files to operate: *defstrokes.il*, *stroke.il*, and *def.strokes*. The syntax for loading these files is in the *.cdsinit* file. The commands for loading the strokes files are commented out, so you must remove the comments and then reload the *.cdsinit* from the CIW.

1. In an xterm window, enter:

   ```
   vi ~/Layout/.cdsinit
   ```

   The *.cdsinit* file opens in the *vi* editor.

2. Use the **down arrow** key to scroll down to the *Load Strokes Editor* section.

   You see the following syntax for loading the three stroke files.

```
;load(prependInstallPath(“/etc/sted/stroke.il”))
;load(prependInstallPath(“/etc/sted/defstrokes.il”))
;hiLoadStrokeFile(prependInstallPath(“/etc/sted/def.strokes”) “Layout”))
```

3. You must remove the semicolon (;) from each line so these files can be loaded by the *.cdsinit* file. To do this:

a. Use the **down arrow [j]** to move to the first semicolon.

b. Press **Escape**, then press **x** to delete the semicolon.

c. Press **Escape** to get out of delete mode, then repeat the previous two steps to delete the semicolons at the beginning of all three lines.

4. To save your edits, press **Escape**, then enter:

```
:wq
```

5. To load the edited *.cdsinit* file in the CIW, enter:

```
load(".cdsinit")
```

Now the Layout Editor will recognize the default strokes.

Using Strokes to Zoom In and Out of Your Design

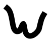
You are going to use strokes to zoom in and out on the cellview. If you enter a stroke and nothing happens, you might not have drawn the stroke correctly and you'll see a warning message in the CIW. Try drawing the stroke again.

1. To zoom in on an area, press and hold down the right mouse button and draw a "Z" on top of and approximately the size of the *2nor* instance.

   The window zooms in on the area covered by the "Z."

   strokeZoomIn

2. To fit the entire drawing in the window, hold down the right mouse button and draw a "W."

    **strokeWindowFit**

   This is the same as the **Window** — **Fit All** command.

3. To zoom in using a different stroke, hold down the right mouse button and draw a slash "/" from left to right on any area of the design.

   strokeZoomIn

   The window zooms in on the area covered by the slash.

4. To zoom out, hold down the right mouse button, and draw a backslash “\” from left to right.

   \ **strokeZoomOut**

   The window zooms out of the area covered by the backslash.

5. You can practice these **Zoom** commands until you feel comfortable with the way they work.

Panning Around Your Design Using Strokes

You are going to use strokes to pan around this design.

1. With the right mouse button, click on an instance and draw a horizontal line to the right.

   — **strokePan**

   Your line doesn’t have to be completely horizontal as long as it does not look like a 45-degree line, which the Stroke Editor will interpret as a **Zoom** command.

   When you release the mouse button, the window pans so that the instance moves from the beginning of the line you drew to the end of the line.

   You are not limited to panning to the left, right, up, or down. To pan your window more precisely, you can add left or right hooks to vertical lines, and up or down hooks to horizontal lines.

   ╮ **strokePan**

2. With the right mouse button, click on any instance and draw a vertical line upward. Do not release the mouse button!

3. Still pressing the right mouse button, draw a small hook on the vertical line to the left “⌉” and release the mouse button.

   ⌉ **strokePan**

   Don’t make the hook too big, or the system might interpret this as a separate stroke. Notice again that the instance moves to the last point entered.

4. You can practice these panning strokes until you feel comfortable with them.

   Note: If you move the drawing too far out of view, use the “W” stroke to fit the entire drawing in the window.

Eliminating Mouse Clicks

You can use the **infix** option on the User Preferences form to indicate whether the location of the cursor should be used as the first point entered for a command. When you use **Copy**, **Move**, or **Delete** strokes with **infix** on, the editor uses the last point of the stroke to select the object at that location.

1. Select **Options** — **User Preferences** in the CIW.

2. Set **Infix** on then click **OK**.

Copying and Deleting with Strokes

You are going to copy a *2nand* instance and delete it using strokes. To do this, you draw your strokes over the objects you want to copy and delete. This selects the objects and invokes the delete command when **infix** is turned on.

Note: *If you do not draw these strokes over the objects, the system prompts you to select the objects.*

1. Press and hold the right mouse button and draw a “W” to fit the entire drawing in the window.

   strokeWindowFit

2. Press and hold the right mouse button again and draw a “C” on top of any of the instances.

   strokeCopy

   This is the same as preselecting the instance and choosing **Edit** — **Copy**. The instance you indicate is selected to be copied.

   Note: *The array is flattened. You copy only one instance, not the entire array.*

3. To place the copied instance, move the cursor to the side, then click with the left mouse button.

4. The **Copy** command repeats, so press **Escape** to end this command.

5. To delete the instance you created, use the right mouse button and draw an alpha "α" on top of the copied instance.

   α **strokeDelete**

   This is the same as preselecting the instance and choosing **Edit** — **Delete**.

6. Press **Esc** to end the delete command.

Setting Infix Off

The rest of the labs are written with infix set off. To get the same results as the lab, you must set infix off before continuing.

1. Select **Options** — **User Preferences** in the CIW.

2. To set **infix** off, click the **Infix** button, then click **OK**.

In the next lab, you learn how to create your own strokes.

Lab C-2 Creating Strokes

Objective: Learn to create your own strokes using the Stroke Editor.

In this section you create a new stroke for **Create** — **Rectangle**. The SKILL function for this command is *leHiCreateRect*.

Creating New Strokes

You will not use a layout window for this lab. You start the Stroke Editor in a UNIX window and create the new stroke in the Stroke Editor window.

1. To change to the Stroke Editor directory, enter:

   ```
   cd ~/Layout/strokes
   ```

2. To start the Stroke Editor, enter:

   ```
   sted def.strokes &
   ```

   The STED window opens and displays the currently defined strokes and their associated commands. Notice that these are some of the strokes you used in the previous lab.

3. To see the rest of the defined strokes, click **NEXT PAGE**.

4. Click **NEW** to open an empty window where you can draw your new stroke.

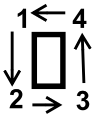
5. Move your cursor into the text window at the bottom of the Stroke Editor, then at the prompt "enter command string>" enter the following, then press **Return**.

   ```
   leHiCreateRect()
   ```

   You see *COMMAND: leHiCreateRect* displayed at the top of the window.

   Note: *Before you create a new stroke, you must know the SKILL function associated with the command. You can look up this information in the Layout Editor Reference manual.*

6. With the left mouse button, draw your stroke anywhere in the window.

   leHiCreateRect()

   Because your stroke will represent **Create** — **Rectangle**, try to make it look like a rectangle. Start in the upper left corner and draw the rectangle counterclockwise.

   The Stroke Editor records the stroke you drew and gives it a count of 1.

7. Draw the rectangle at least 10 more times so the Stroke Editor learns to recognize many variations in the stroke. Always start in the upper left corner, and draw the rectangle counterclockwise.

   If the Stroke Editor recognizes a stroke to be the same as one you previously drew, it increments the count next to that stroke. If a stroke is not similar to a previous stroke, the editor gives it a new number and a new count of 1. The more times you enter the new stroke, the more likely the editor will recognize it later.

   If you have several strokes with a count of 1 that don't look the way you want your stroke to look like, you need to delete them.

8. To delete a stroke, click **Delete**, then click on the temporary stroke you want to delete.

9. When you are satisfied that the system will recognize your new stroke (at least one of your temporary strokes has a count of 10), click **ENTER** to save the stroke.

   The stroke you created is displayed at the top of the defined strokes. You might have to click **Prev** to go to the first page of strokes.

10. Click **WRITE** to write the new stroke to the strokes file.

    The text window prompts you to:

    ```
    Enter the name for the stroke file [def.strokes]
    ```

11. To overwrite the original *def.strokes* file, press **Return**.

    Note: *If you want to continue using your own strokes, you must change your .cdsinit to load the new def.strokes file instead of the default strokes file. The def.strokes file loaded by the .cdsinit is located in the install path, not in your home directory.*

12. To close the Stroke Editor, click **EXIT**.

Testing New Strokes

1. In the CIW, load the modified *def.strokes* file by entering:

   ```
   hiLoadStrokeFile(“/Layout/strokes/def.strokes”)
   ```

2. In the layout window, draw the new stroke.

   If the system doesn’t recognize your stroke, try again. When the system recognizes your stroke, the CIW displays this prompt:

   ```
   Point at the first corner of the rectangle:
   ```

3. Use the left mouse button to draw a rectangle anywhere in the window.

4. Choose **Window** — **Close**, to close this window. You don’t need to save the changes.

Summary

In this lab, you did the following:

- You learned how to edit the *.cdsinit* file with the **vi editor**.
- You learned how to use **zoom** and **pan** using strokes.
- You learned how to **copy** and **delete** using strokes.
- You learned how to create your own custom strokes using the **Stroke Editor**.

End of Lab

Appendix D

Microwave Design

Lab D-1 Creating Transmission Lines

Objective: Use the Trl, Bend, and Taper commands to create interconnect.

In this lab, you must be running *icfb* to run microwave extractor.

Opening a Design

1. Add the *uwave* library to the Library Path editor, if it has not been added already.

2. In the CIW, select **File** — **Open** then enter the following values in the form:

Library Name	`uwave`
Cell Name	`labc1`
View Name	`layout`

The *labc1 layout* cellview appears.

3. To change to the Microwave menu, select **Tools** — **Microwave**.

There are two designs in the cellview. The design shows how the cellview needs to be connected. The design on the bottom has no interconnect. You use the **Trl**, **Bend**, and **Taper** commands to create your interconnect.

You will create the interconnect on the *Int1 nt* layer.

4. In the LSW, click on the *Int1 nt* layer.

5. To start the **Trl** command, select **Create** — **Trl**.

Change the options on the form to create the interconnect.

6. When you have completed the interconnect, select **Window — Close**.

End of Lab