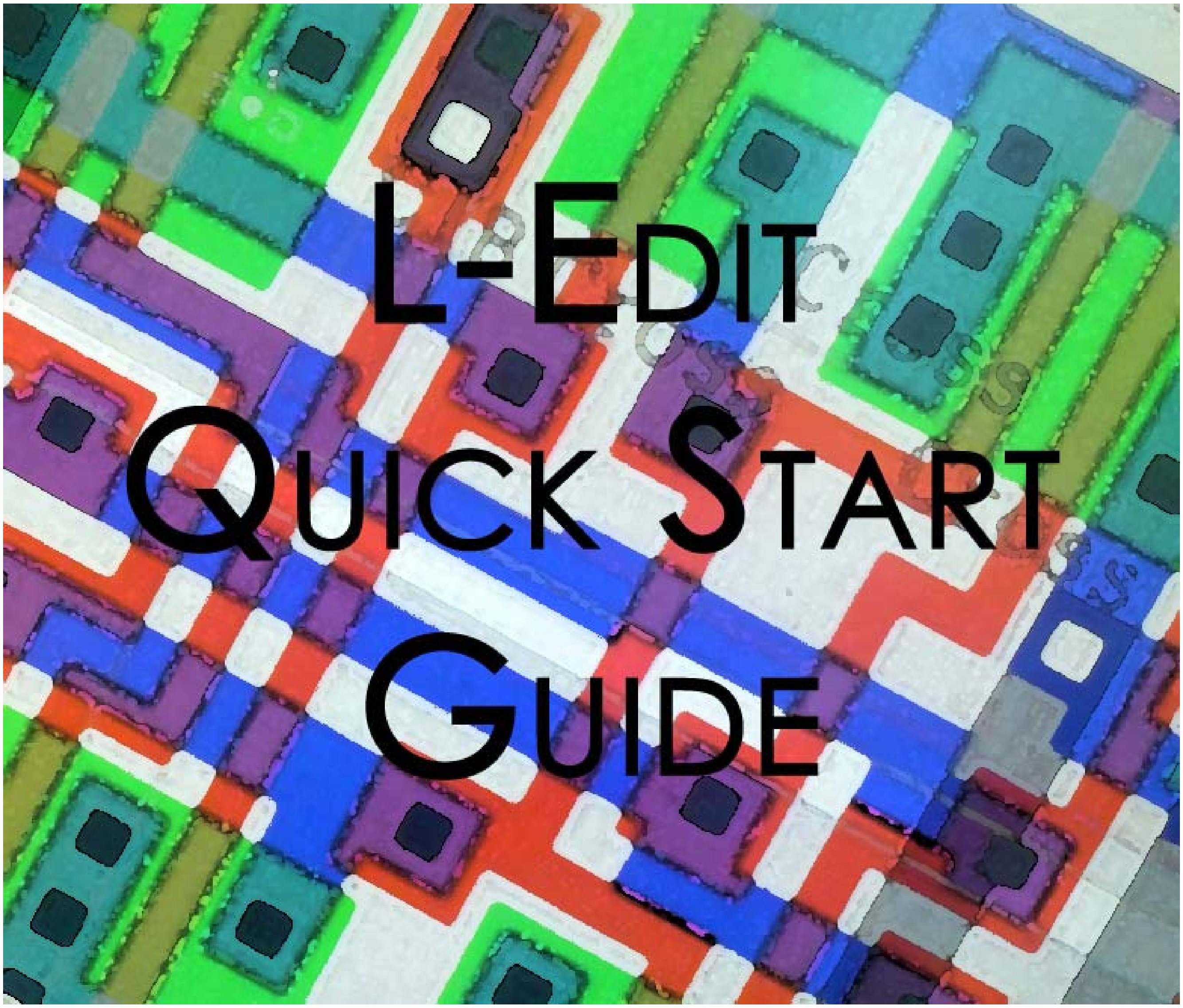


L-Edit Quick Start Guide

Nathan Kupp
Princeton University
PRISM REU Program 2006

1. Introduction

This guide is intended to supply the L-Edit user with a tutorial and reference for quickly gaining proficiency with the L-Edit software layout tool. By reading this guide and doing the provided tutorials in L-Edit, the user should be able to become relatively familiar with the program within a few hours time. This guide does not provide an exhaustive reference to all the features of L-Edit, as a complete documentation of these features is provided in the L-Edit User's Guide. The L-Edit User's Guide is provided in Adobe PDF form, accessible from the Help menu within the L-Edit program.

2. User Interface

The user interface provided by L-Edit is similar to that of many other CAD and layout design software packages. There are three important sections of the L-Edit User Interface:

- Toolbars
- Sidebar
- Drawing Area

The collection of toolbars on the top of the screen allows the user to select what type of objects to draw, as well as manipulate previously created objects. Figure 1 displays the toolbar and sidebar with the most frequently used buttons labeled. Most of the buttons here are self-explanatory.

L-Edit also has a sidebar which contains the layers specified under "Setup → Layers...". A default collection of most commonly used layers is provided by the **`Generic_025um.tdb`** file, provided in the root L-Edit directory. Since this file provides all of the common layers such as N-well, Active, N-select, P-select, Metal1, etc. it will not likely be necessary to modify the default layers for this tutorial.

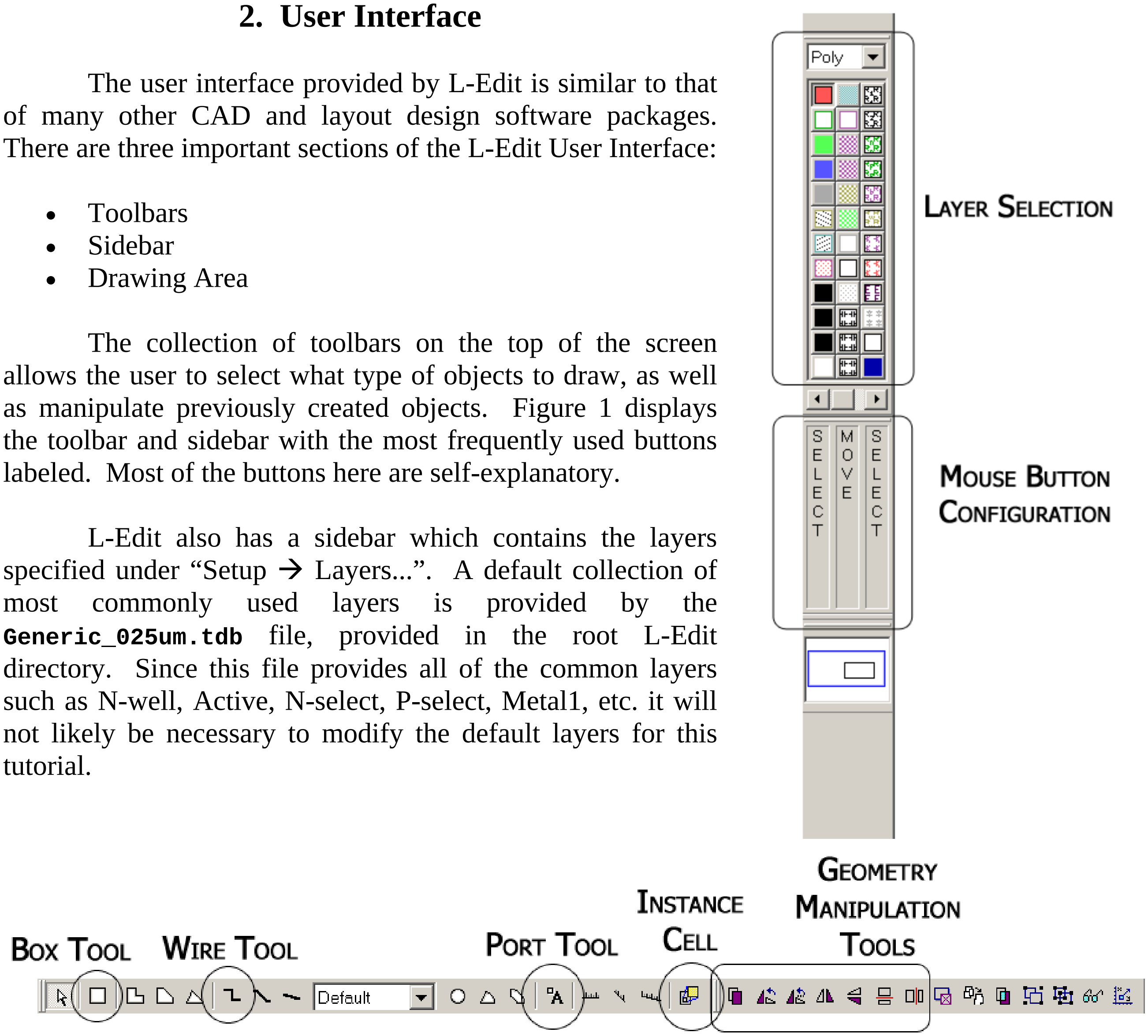


Figure 1. L-Edit Toolbar/ Sidebar

For custom layers, a blank setup can be used without any default layers, as described in Tutorial #1: NMOS Design & Layout. Layers can be selected on the sidebar with either their representative icon or from the list above the icons. Note that the names of layers are functionally irrelevant, and serve only to clarify the design – in reality, all layers are simply translated to geometry and the layer names discarded when the file is exported to GDSII.

With the creation of a new file, the user is presented with a layout area entitled Cell0, representing the first "cell" within his or her design (Cells will be explained in further detail in section 4). Figure 2 shows the layout area of this Cell0. The layout area is where all of the design takes place – the user selects appropriate geometry from the toolbar, such as a rectangle; a layer, from the sidebar, and draws it on the layout area. As the object is drawn, the status bar at the bottom left of the screen will display the drawn width and height, and area of the object. The drawn object can subsequently be resized by left-clicking on the object to select it, and then clicking down with the mouse scroll wheel on the corner of the object and dragging it. The status bar will update to reflect the change in size. The object can be repositioned by clicking near its center with the mouse scroll wheel and dragging it with the mouse. Finer movement can be achieved with the Draw → Nudge dialog, which allows movement in any direction with three decimal places.

Use the arrow keys to display different parts of the design in the viewing window. Also, the view can be zoomed in/out by clicking anywhere outside the drawing with the left mouse button, and then rotating the scroll wheel on the mouse while holding Ctrl on the keyboard. The display of the grid can be toggled within this viewing window by selecting Setup → Design and selecting the Grid tab. Note that under the Technology tab there are settings for the units used in the design. The settings under Grid control the distance between grid points as well as whether the mouse snaps to this grid. If the Suppress major/minor grid is set too large in this dialog, the grid will likely not appear unless the view is zoomed fully in (again to scroll, use the middle mouse button). Finally, note that the origin (x,y = 0) appears as a cross symbol on the drawing area (see figure below).

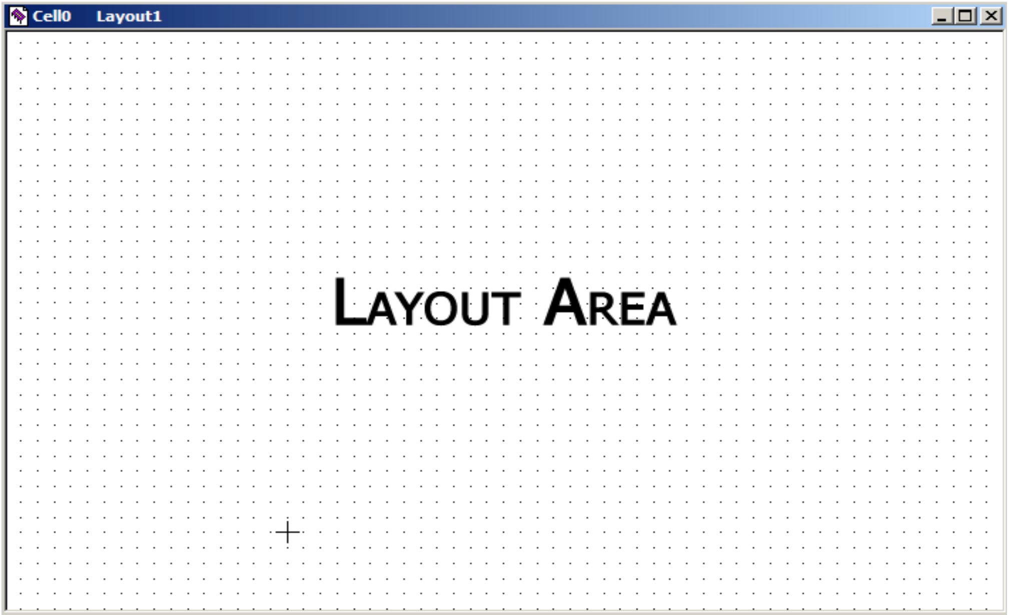


Figure 2. Cell0 Window

What's Important:

- ***Toolbar: Select, Rectangle, Wire, Port, Instance, and Rotate/Mirror icons***
- ***Layers icons in Sidebar***
- ***Layout Area***

3. Tutorial #1: NMOS Design & Layout

In the first tutorial, the design of an n-type MOSFET on a p-type substrate is presented. An NMOS transistor consists of an n-type source and drain regions, a gate terminal, and a substrate terminal. Remember, a MOSFET is a four terminal device, requiring not only connections to the Source, Drain, and Gate, but the Substrate as well.

Process:

1. Select File → New to create a new layout. A dialog will appear asking for a layout name as well as a location to copy a TDB setup from. The file you choose for the TDB setup governs the layer setup in the new design – therefore, if you wish to create a design with entirely customized layers, you can select **<empty>** and a layout will appear with an empty Layers toolbar. Copy the TDB setup from the file C:\Tanner\LEdit102\Samples\tech\Generic0_25um\Generic_025.tdb. Since the transistor is NMOS, the source and drain are n-doped and thus must be formed directly on a p-type material (in this case, the substrate).

2. Draw two Active regions in the design. The Active layer denotes an opening in the field oxide through which n-type and p-type depositions can be made – we need an area for the source and drain of the transistor, as well as a connection for the substrate.

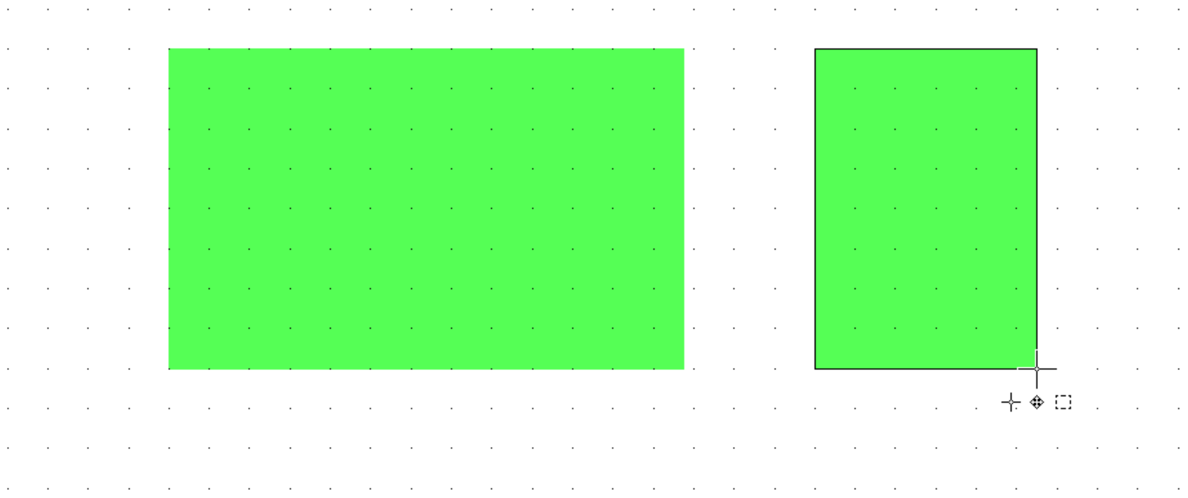


Figure 3. Active Area Layer

Suggestion: *The purpose of this tutorial is to teach you how to use L-Edit. To save time, you do not need to understand the names and functions of different layers. Just draw objects the way you see them in this tutorial.*

3. Draw N-select/P-select regions over the source/drain and substrate Active areas, respectively. The substrate can never be directly connected to metal layers, so we need to create doped select layers where we want to make connections. These layers denote areas that will be implanted with n/p-type material. Note that the N-select can be drawn right across the polysilicon gate, as this gate will prevent n-type deposition directly beneath the gate when the chip is bombarded with n-type atoms.

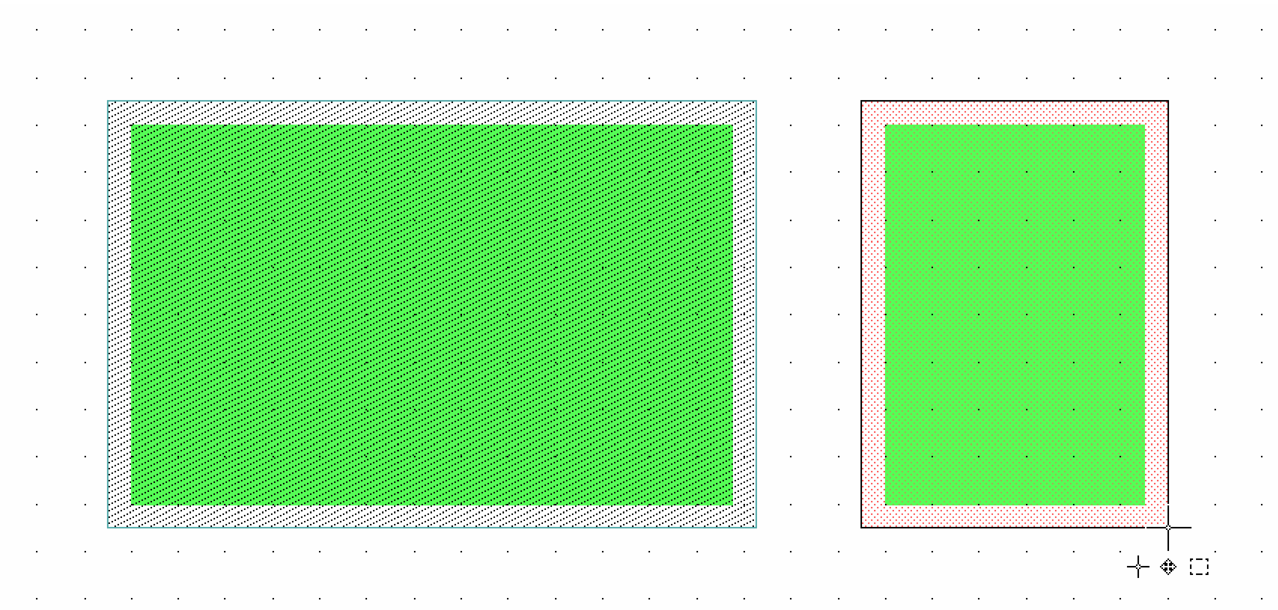


Figure 4. N-Select and P-Select Layers

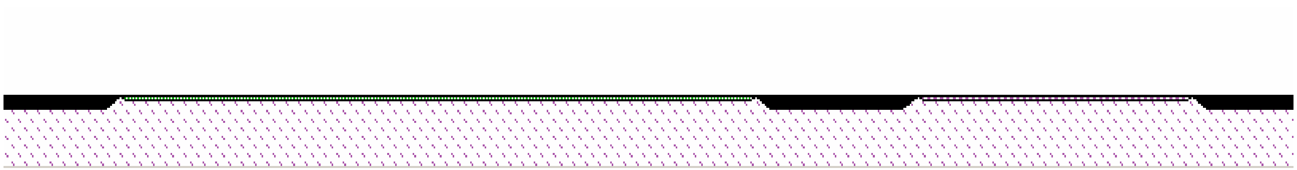


Figure 5. N-Select and P-Select Cross-Sectional View

4. Draw Poly layer. The gate of a MOSFET is typically created out of polysilicon, represented by the Poly layer. To create a gate for our NMOS transistor, we need to draw a Poly layer rectangle over the active region of the source/drain. Note that in the cross-sectional view, the large polysilicon pad is not shown, because the cross-section was taken across the transistor itself. The Poly gate shown below can be generated by drawing two separate, overlapping rectangles. These rectangles can then be merged into a single object by selecting Draw → Merge.

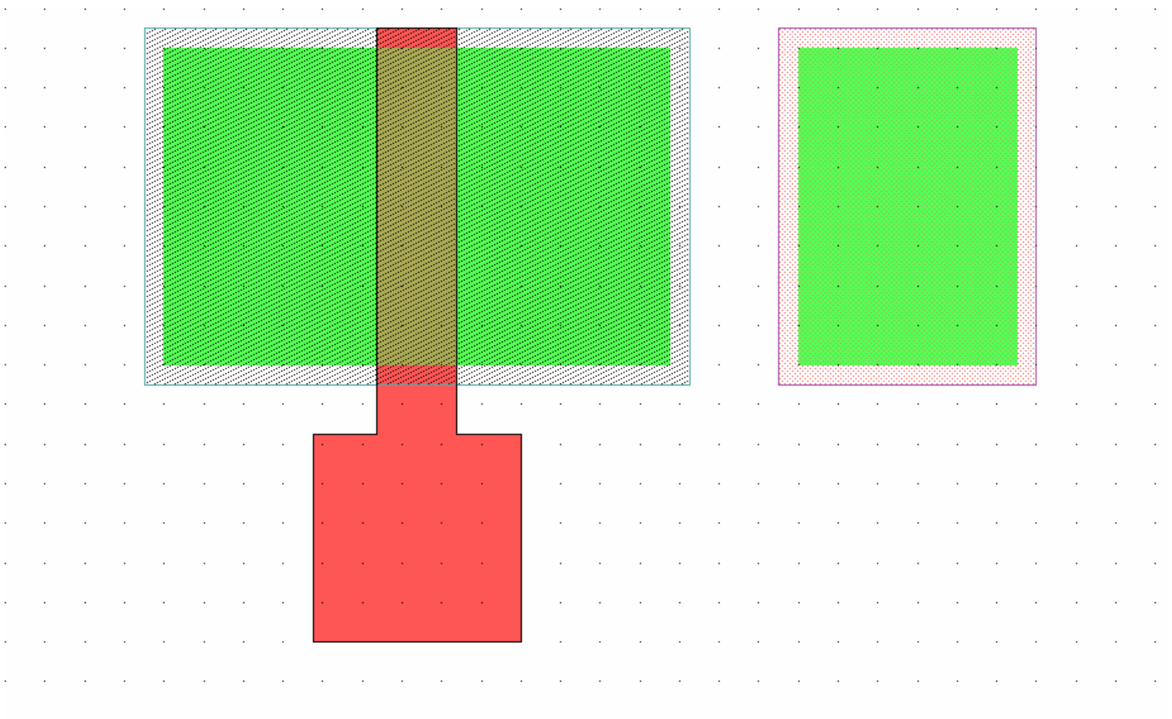


Figure 6. Poly Layer

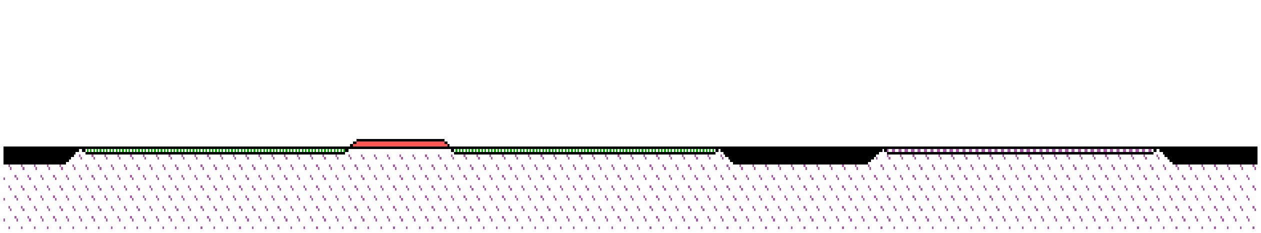


Figure 7. Poly Layer Cross-Sectional View

5. Draw Active contacts at the source drain and substrate, and a Poly contact on the gate. We now need to create an opening in the gate oxide to wire up our transistor. This is denoted by the Active Contact layer and the Poly Contact layer – the Active Contact layer makes connections to the Active layer, and the Poly Contact layer makes connections to the Poly layer. Again note that in the cross-sectional view, the Poly contact is not visible because the cross-section was taken across the transistor itself.

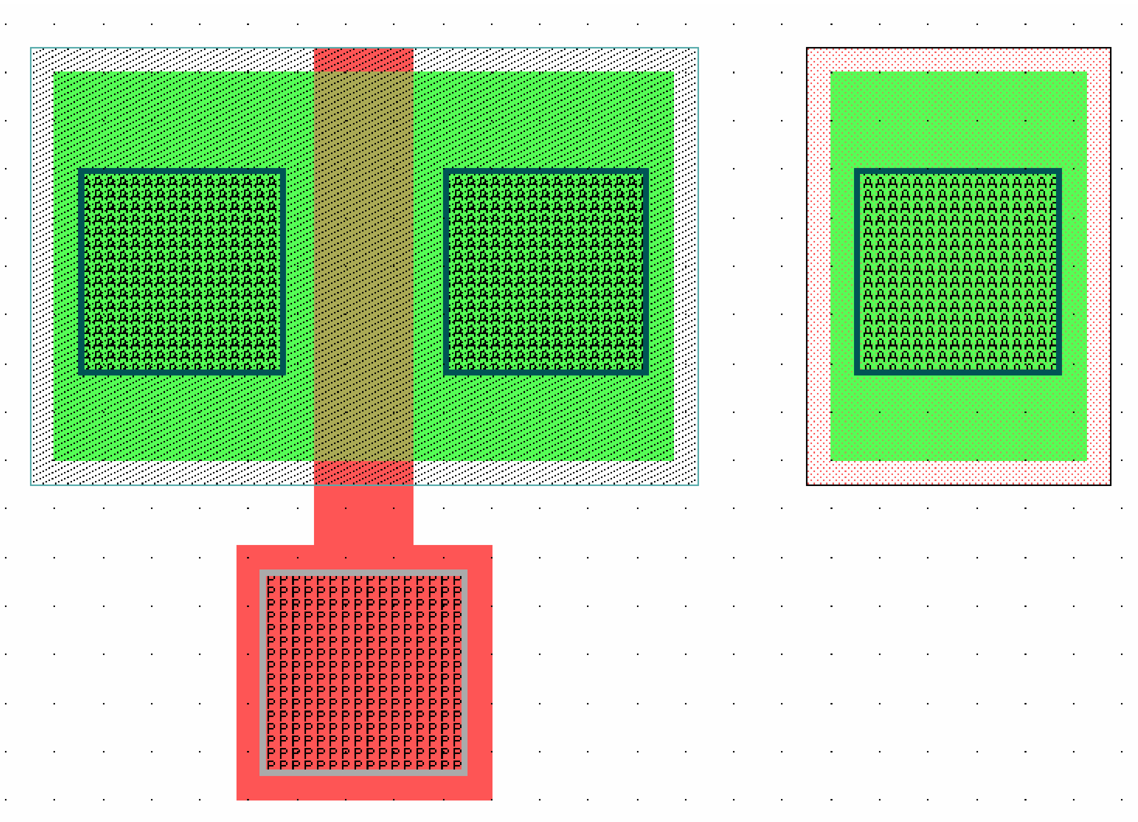


Figure 8. Active and Poly Contact Layers

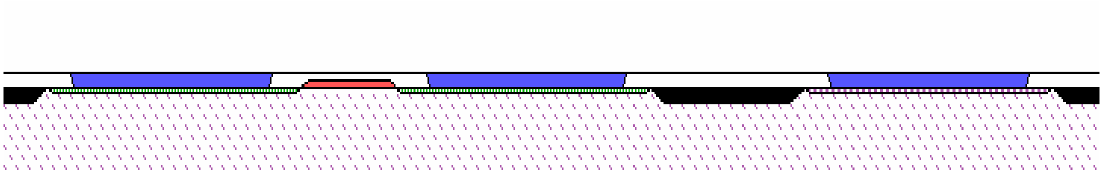


Figure 9. Active and Poly Contact Layers Cross-Sectional View

6. We now need wires for the transistor to be useful. Draw Metal1 layer connections to each of the four contact points. These can be created by either selecting the wire tool and drawing a wire, or using the rectangle tool to create a rectangular wire segment. Again, we cannot see the Metal1 wire on the gate in the cross-sectional view because of the location of the cross-section. Note that it is common to connect one terminal of the N-type MOSFET to the substrate and then to ground.

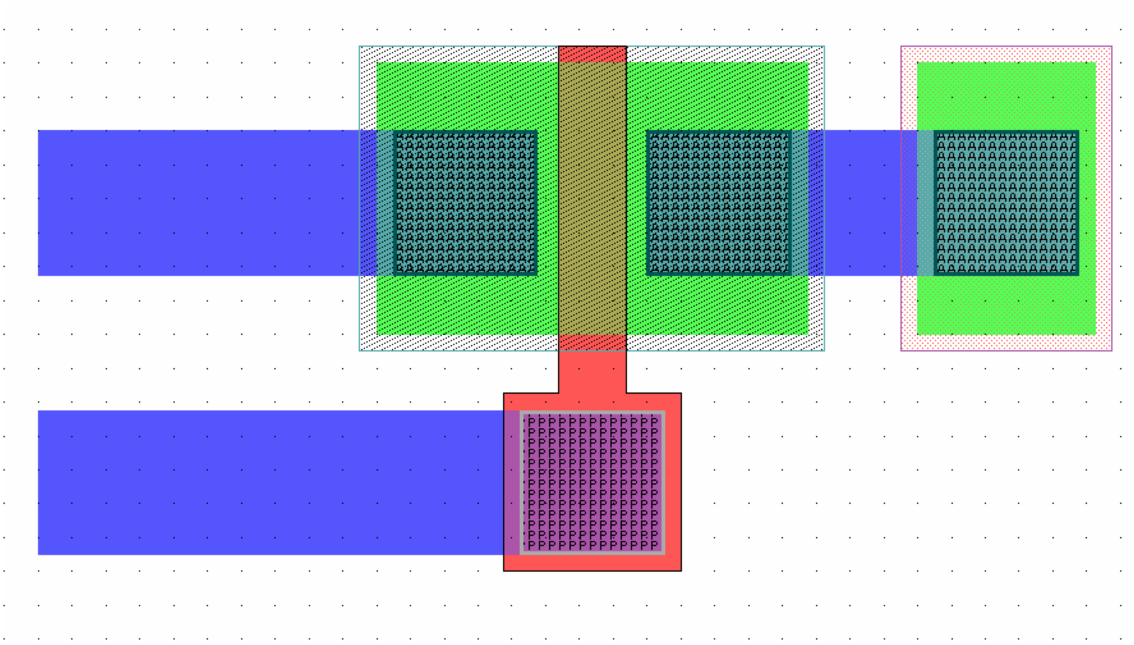


Figure 10. Metal1 Layer

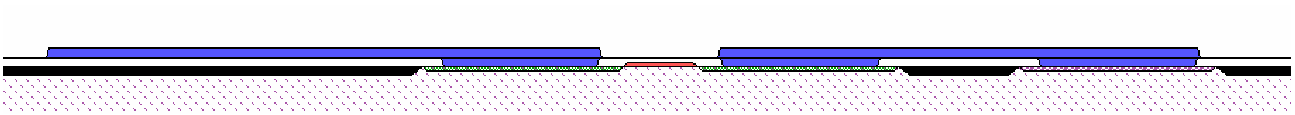


Figure 11. Metal1 Layer Cross-Sectional View

7. Now we have a working NMOS transistor! Do a Cell → Rename… to name this cell "**NMOS**" and a File → Save As... to save this file as tutorial.tdb. You can now create a PMOS device as well by simply forming it entirely within an n-well, with the substrate connection now being to an N-Select layer within the n-well. Remember, the substrate connection of a PMOS device is connected to VDD, not GND, producing a reverse biased pn-junction at the substrate connection. Similarly our NMOS device had a substrate connection to ground, also producing a reverse biased pn-junction between the substrate and the more positively charged transistor source and draing. A layout and cross sectional view of the PMOS transistor is shown below. Create a new cell (Cell → New...), and name it "**PMOS**". Draw this transistor, and then click File → Save.

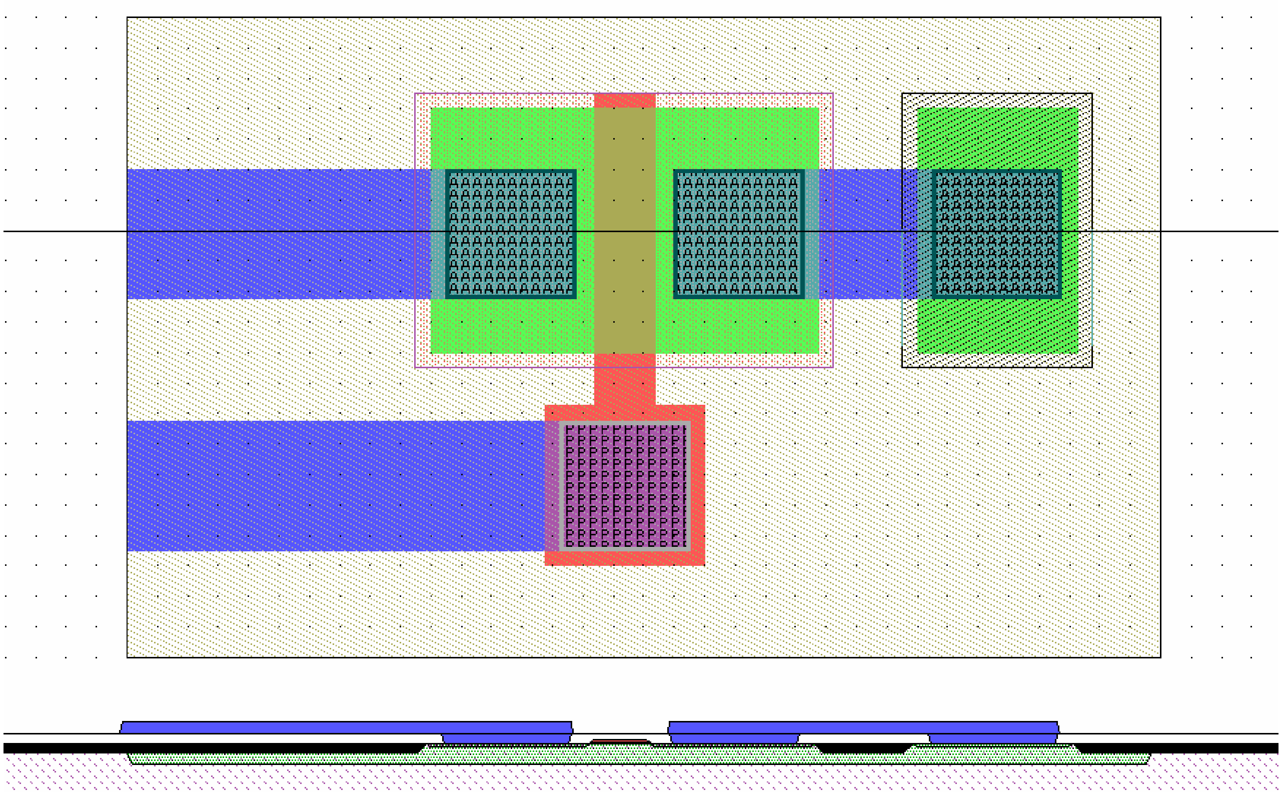


Figure 12. PMOS

Cross-Sectional View Note: *The cross sectional views used in this tutorial can be created using L-Edit's built in Cross-Sectional View Generator. This is accessed from* **Tools → Cross-Section...** *as shown below in Figure 13. Use the settings as shown in Figure 14, and select a vertical coordinate using* **Pick***. Place the horizontal black line across the design, and then click* **OK***. The cross-sectional view will be generated as shown in Figure 15. Note that these cross-sections represent an oversimplification of the fabrication process, and are merely intended to provide the user with an understanding of the interconnections between the layers drawn.*

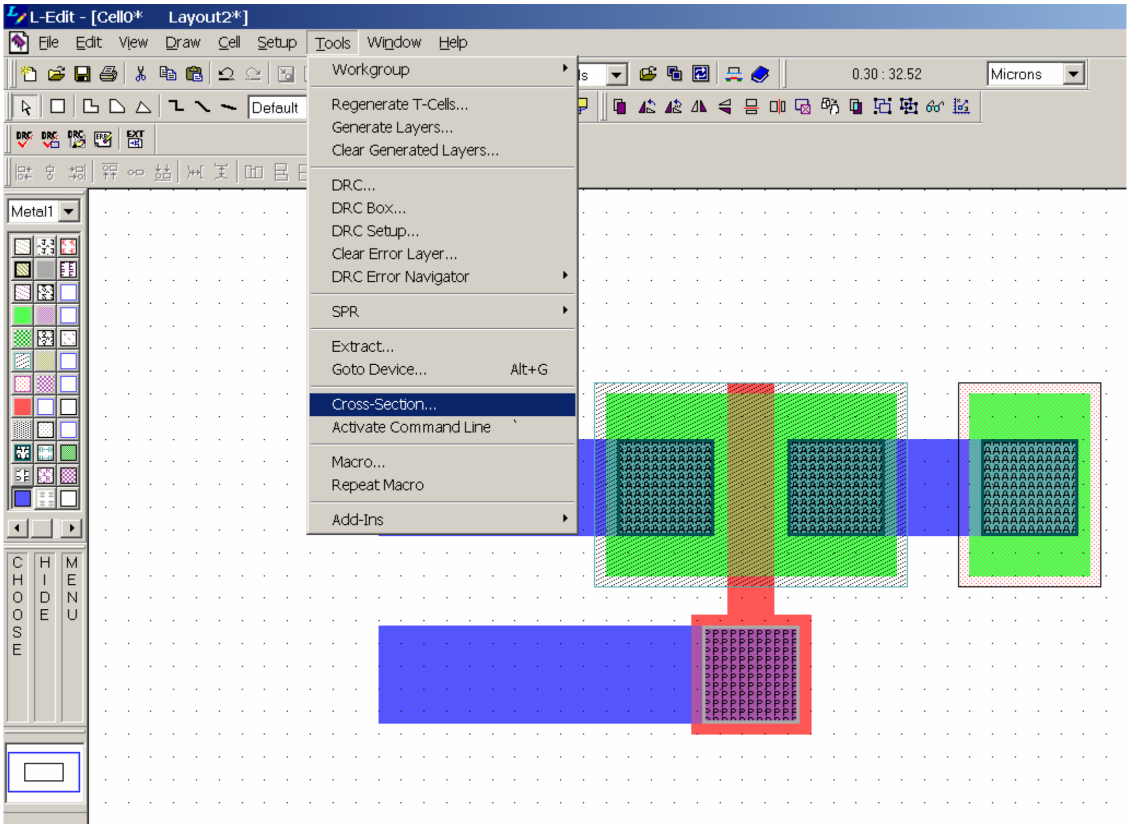
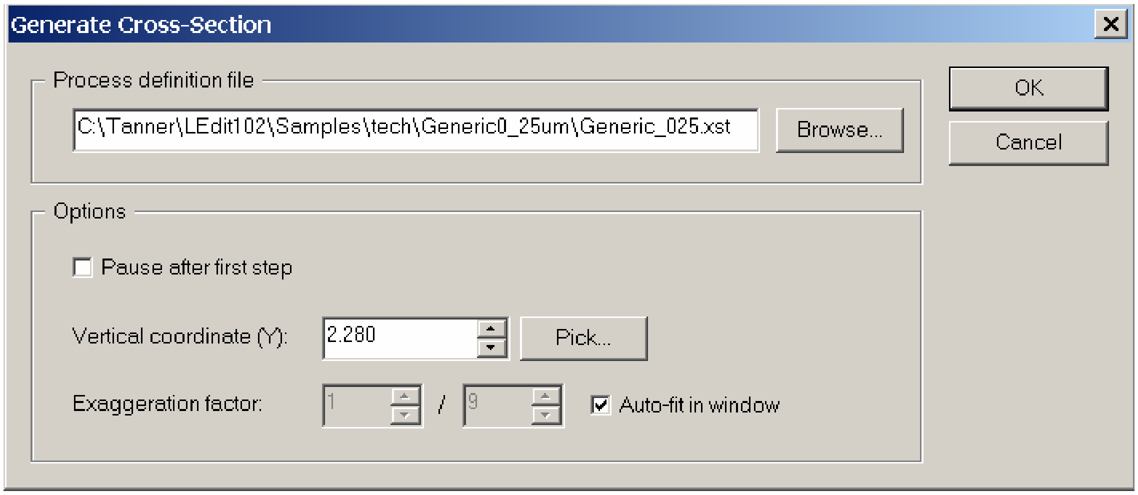


Figure 13. Cross-Sectional View Creation



Generate Cross-Section

Process definition file

C:\Tanner\LEdit102\Samples\tech\Generic0_25um\Generic_025.xst

Options

☐ Pause after first step

Vertical coordinate (Y): 2.280

Exaggeration factor: 1 / 9 ☑ Auto-fit in window

Figure 14. Generate Cross-Section Dialog

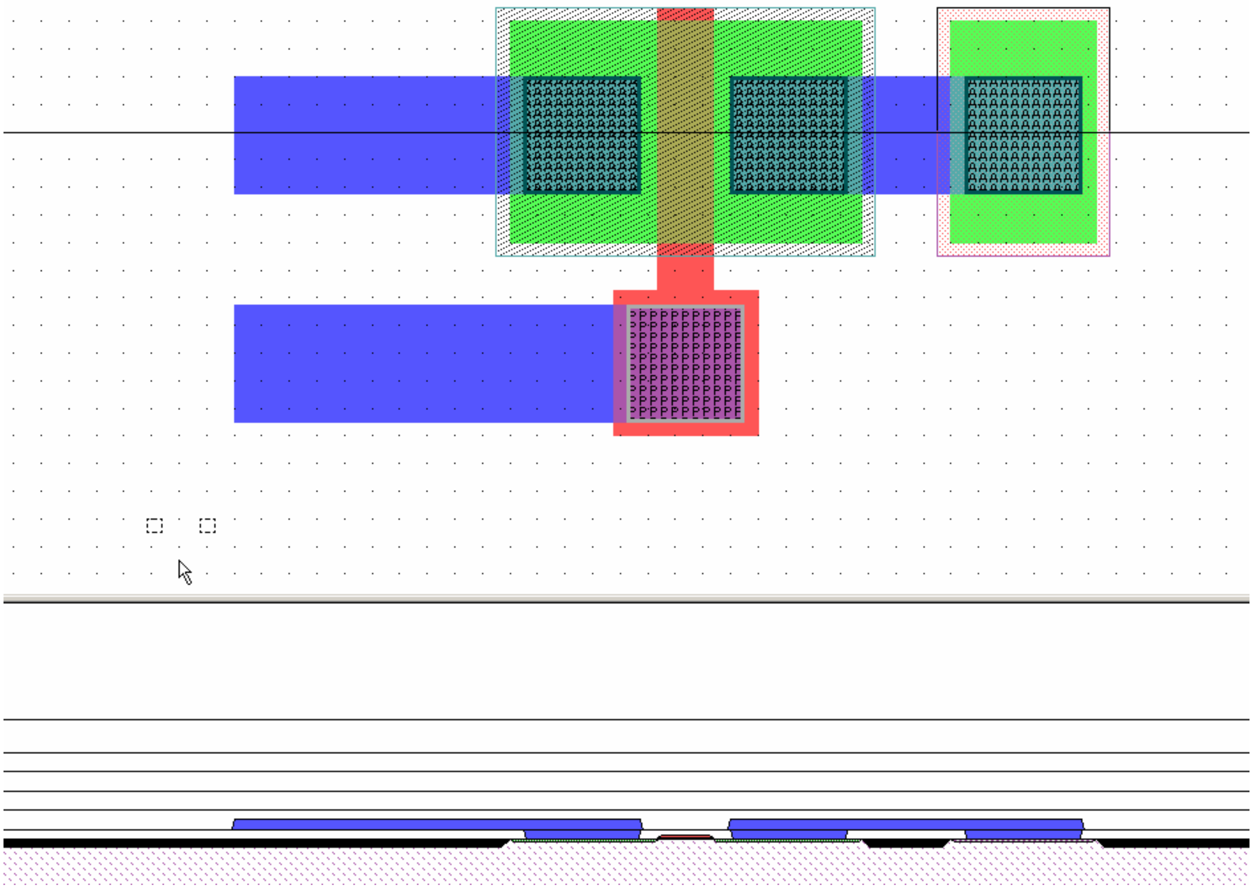


Figure 15. Cross-Sectional View

4. Cells

The design process in L-Edit is centered on cells, which are the fundamental building blocks at all levels of hierarchy – from a single MOSFET, to a logic gate, to an entire design. The figure below shows a cell hierarchy including MOSFETs as well as digital logic assembled on several levels to create a complete layout. The TDB files that L-Edit works with are essentially a collection of all cells related to a single design.

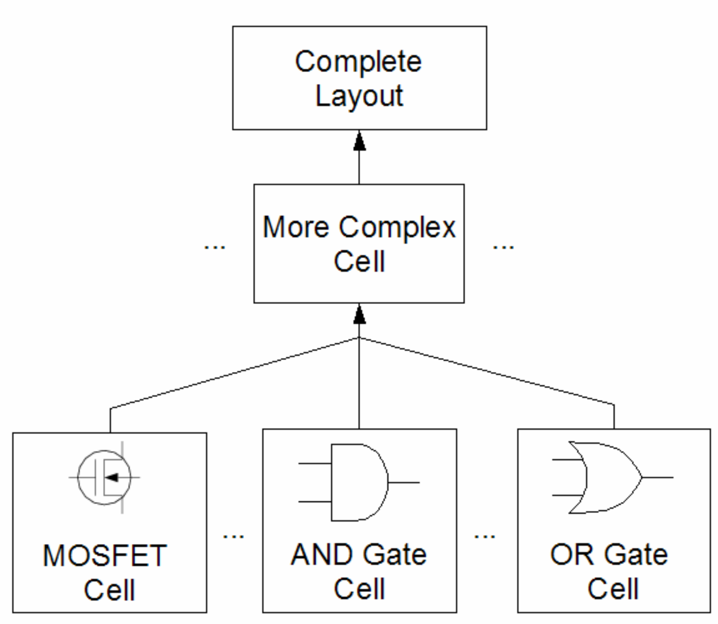


Figure 16. Cell Hierarchy

What's Important:

- ***Cell Menu: New, Open, Copy, Rename, Delete, Instance***

5. Tutorial #2: Cell Hierarchy Based Inverter Design

In the first tutorial we created a **tutorial.tdb** file containing the cells **NMOS** and **PMOS**. Because of the hierarchical design structure of L-Edit, we can now use these two cells together in a third cell to create an inverter, as shown in Figure 20.

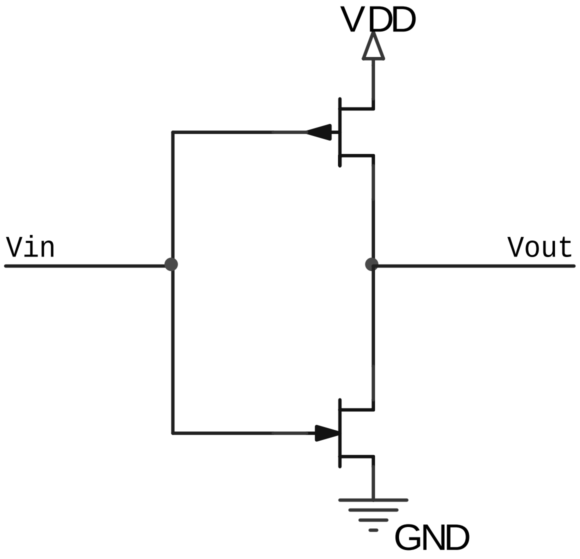
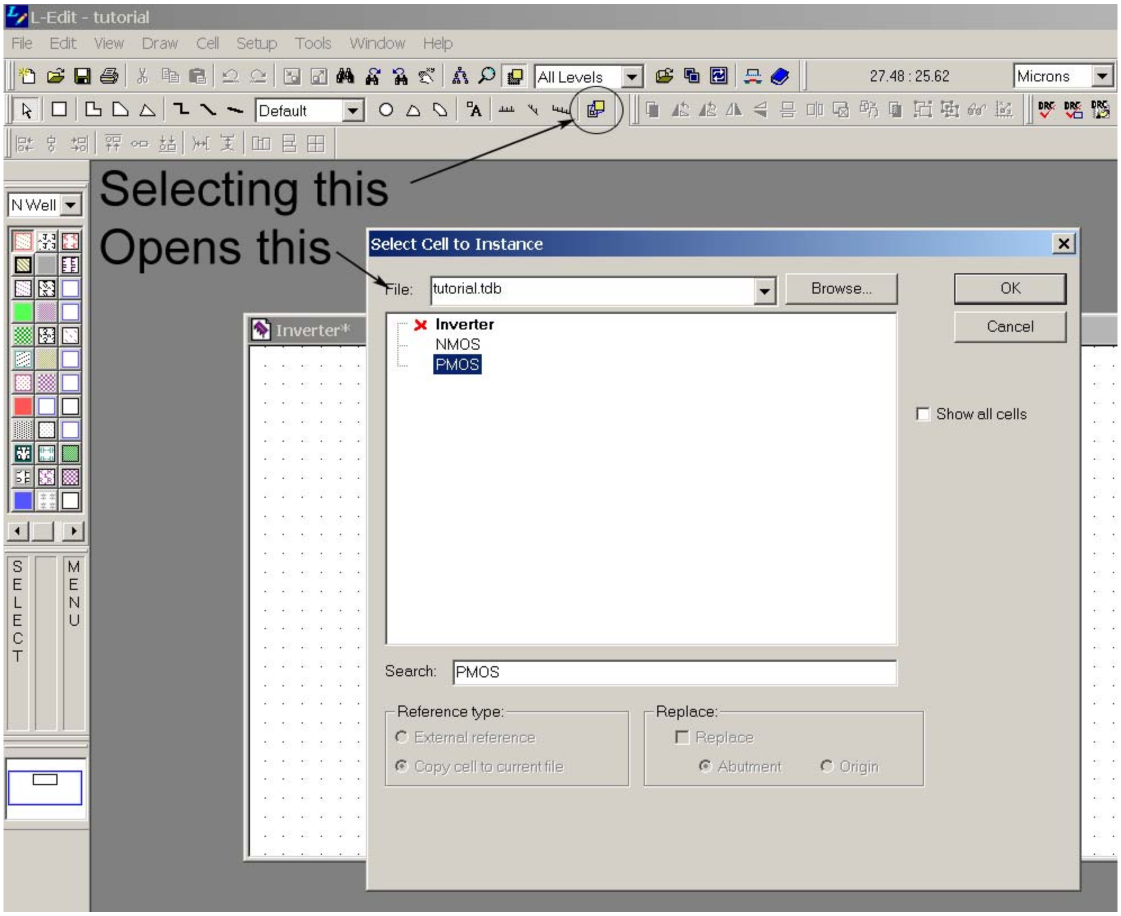


Figure 17. Inverter Schematic

When the input of this device is high, the lower (NMOS) transistor is on and the upper transistor is off – causing the output to be pulled low. When the input is low, the upper (PMOS) transistor is on and the lower transistor is off – causing the output to be pulled high. Although using the two NMOS and PMOS transistors we drew to create an inverter will work, we will see that using two separate components can be messy and that a better approach is to use two transistors connected directly together within a standard cell frame. Standard cell frames will be discussed in more detail in the section on Standard Place and Route. The layout process for an inverter using two transistors is as follows:

Process:

1. Open the original **tutorial.tdb** file we created by doing File → Open... and selecting the file. Create a new cell within **tutorial.tdb** entitled **Inverter** by selecting Cell→New... and entitling it "**Inverter**". Now, click on the Instance Cell button on the toolbar, or go to Cell → Instance and instantiate both the **NMOS** and **PMOS** cells within the **Inverter** cell by selecting each one and placing it on the **Inverter** cell layout area.



Select Cell to Instance

File: tutorial.tdb

Inverter
NMOS
PMOS

□ Show all cells

Search: PMOS

Reference type: ○ External reference ● Copy cell to current file

Replace: □ Replace ● Abutment ○ Origin

Figure 18. Cell Instance Dialog

2. After instantiating both cells in the **Inverter** cell, we will have something similar to the layout view shown in Figure 19. Obviously it would be difficult to wire the cells up in this orientation, so we need to rotate and mirror the cells until we have them in an arrangement more conducive to orderly wiring. Figure 20 displays the **PMOS** and **NMOS** cells rotated and connected together with Metal1 wires to match the schematic of Figure 17. If you now make changes to **PMOS** or **NMOS** cells, these changes will be automatically reflected in the **Inverter** cell.

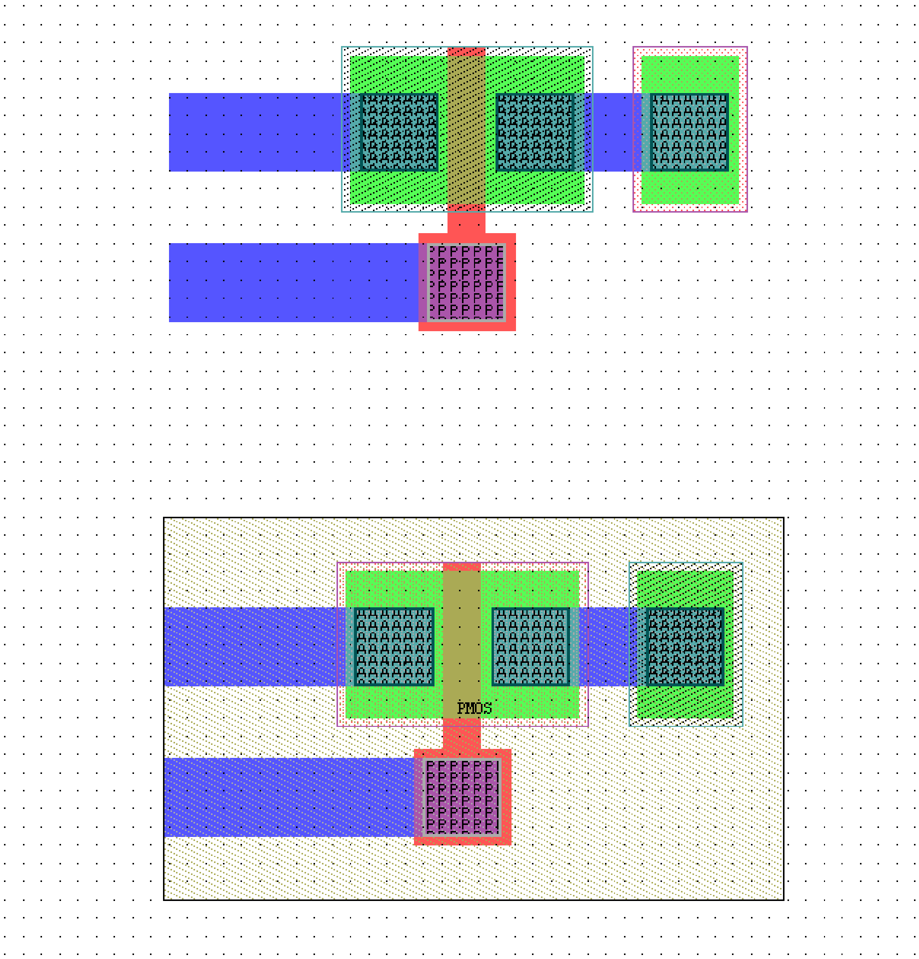


Figure 19. NMOS and PMOS Cells Placed

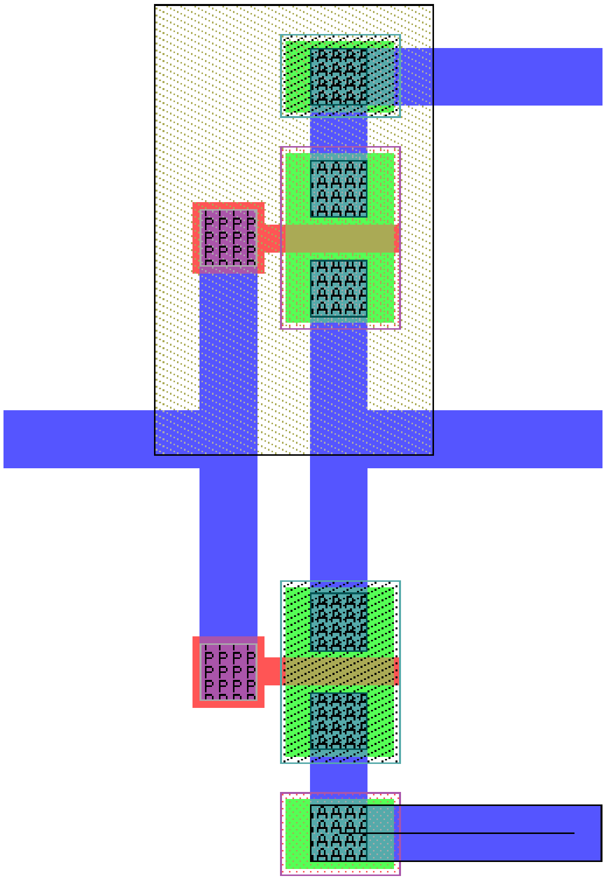


Figure 20. After Rotation and Metal1 Wire Connections

3. Finally, we need to label the inputs and outputs of our inverter using the Draw Ports button on the toolbar. By labeling the ports, we give the design the ability to be used within other designs easily. To do a Standard Place and Route, we need to label all the ports within a Standard Cell Frame to use that cell within our design. Again, this will be discussed in more detail in the section following.

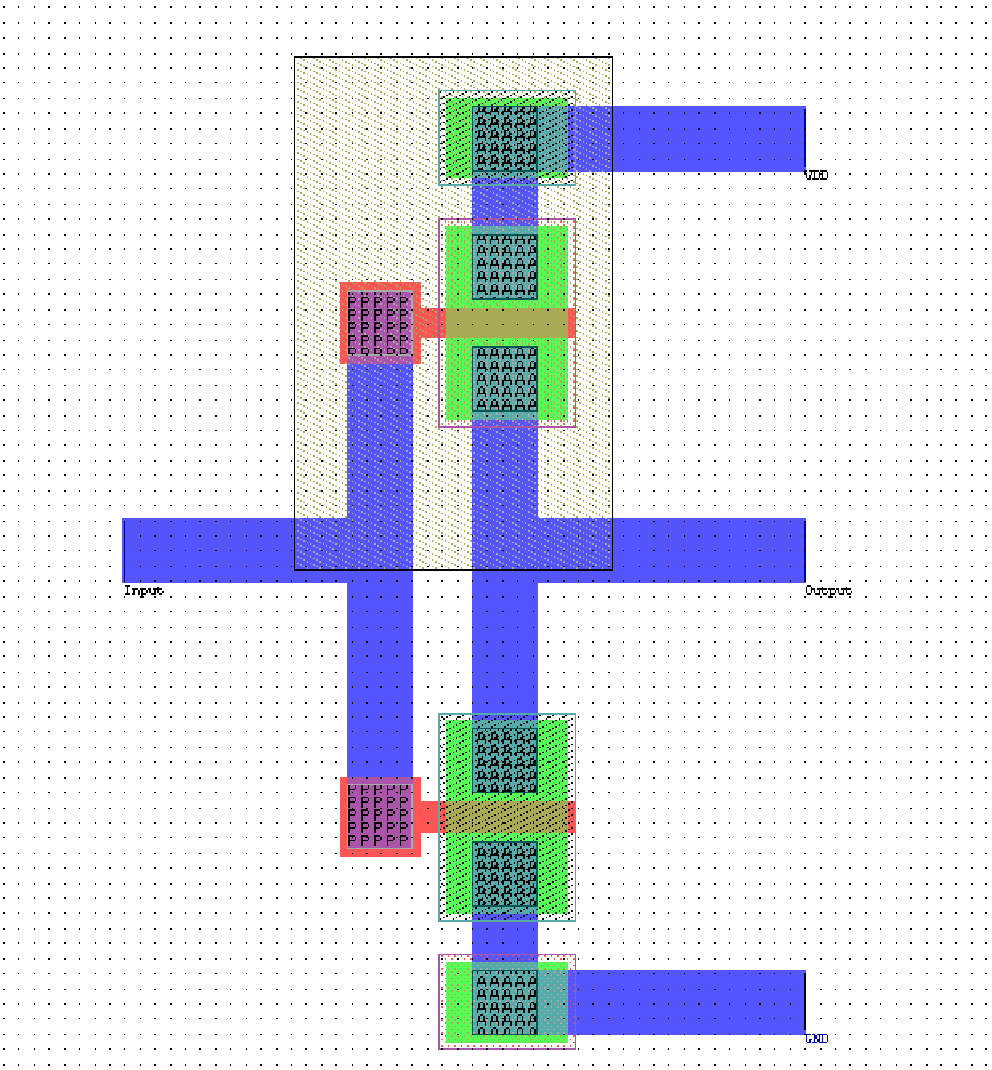


Figure 21. Ports Labeled

6. Standard Place and Route (SPR)

With the cell-based design method provided by L-Edit, there are essentially two approaches to design:

- Full Custom
- Standard Place and Route (SPR)

The first method provides much higher transistor density and more efficient operation – but can be very tedious and takes significantly more time to design. The second design method does not result in as high of a transistor density, but can be created from a schematic in a matter of minutes. Rapid automatic placement and routing this way requires cells to meet specific requirements, defined by the **Standard Cell Frame,** shown in Figure 22. This cell frame has a fixed height, VDD/GND connections at the top/bottom respectively on both sides of the frame on the Metal1 layer, and signal connections on the Metal2 layer. The width of the frame is unspecified, and is controlled by the complexity of the cell unit the user is designing – the higher the complexity, the more width is necessary to accommodate the design.

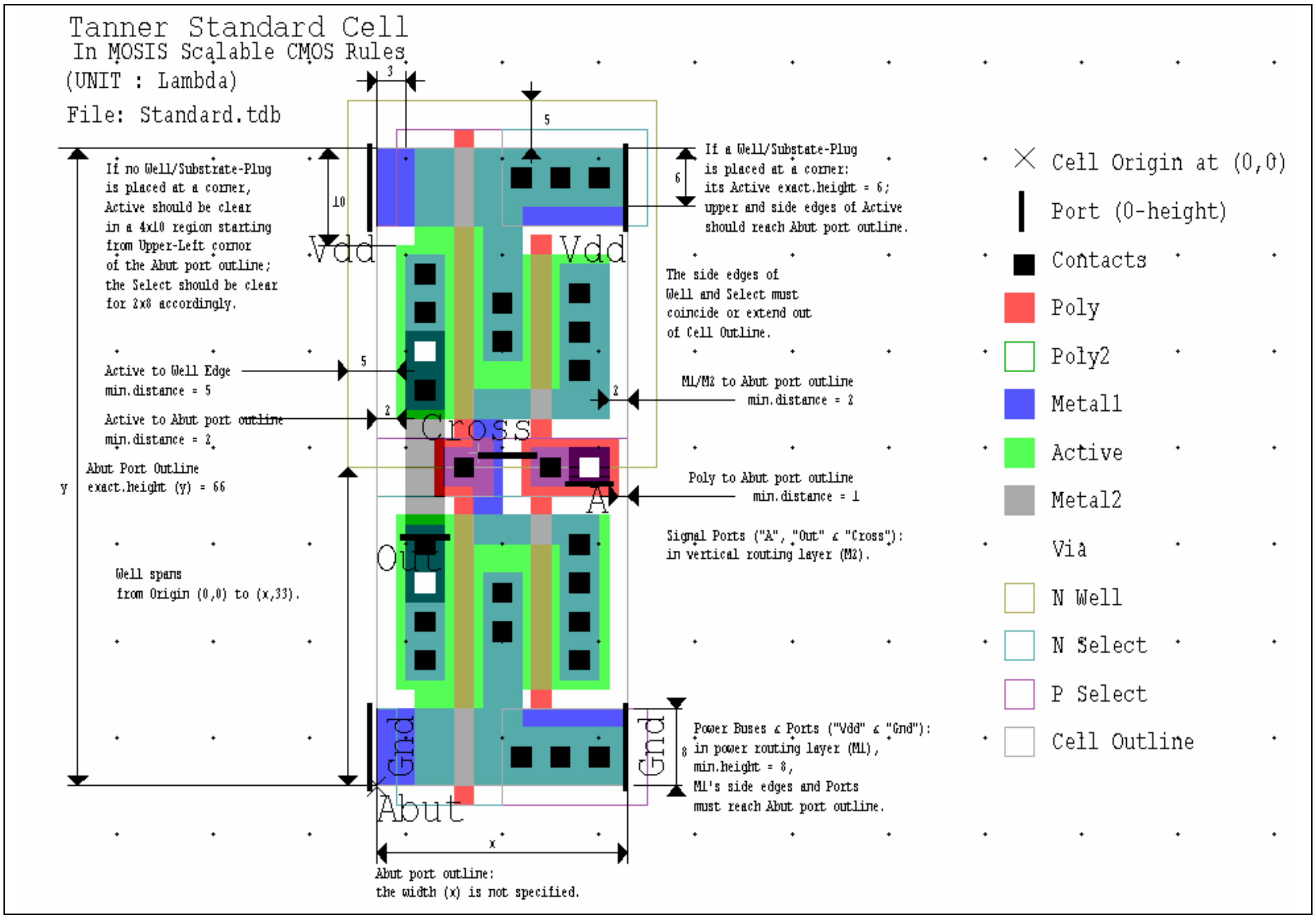

Tanner Standard Cell
In MOSIS Scalable CMOS Rules
(UNIT : Lambda)
File: Standard.tdb
If no Well/Substrate-Plug is placed at a corner, Active should be clear in a 4x10 region starting from Upper-Left cornor of the Abut port outline; the Select should be clear for 2x8 accordingly.
10
3
5
Vdd
Vdd
If a Well/Substate-Plug is placed at a corner: its Active exact.height = 6; upper and side edges of Active should reach Abut port outline.
6
The side edges of Well and Select must coincide or extend out of Cell Outline.
Active to Well Edge min.distance = 5
5
Active to Abut port outline min.distance = 2
2
M1/M2 to Abut port outline min.distance = 2
2
Cross
Abut Port Outline exact.height (y) = 66
y
A
Poly to Abut port outline min.distance = 1
Signal Ports ("A", "Out" & "Cross"): in vertical routing layer (M2).
Out
Well spans from Origin (0,0) to (x,33).
Gnd
Gnd
Power Buses & Ports ("Vdd" & "Gnd"): in power routing layer (M1), min.height = 8, M1's side edges and Ports must reach Abut port outline.
8
Abut
x
Abut port outline: the width (x) is not specified.
Cell Origin at (0,0)
Port (0-height)
Contacts
Poly
Poly2
Metal1
Active
Metal2
Via
N Well
N Select
P Select
Cell Outline

Figure 22. Standard Cell Frame

To use SPR, the user must create a library of standard cell frames, as well as certain other cell frames such as crossing frames, etc. With these in hand, the user can open S-Edit and create a library of corresponding schematics. Using both libraries together, the user can create a schematic in S-Edit, export it to L-Edit, do an automatic place and route, and obtain a final layout within a matter of minutes.

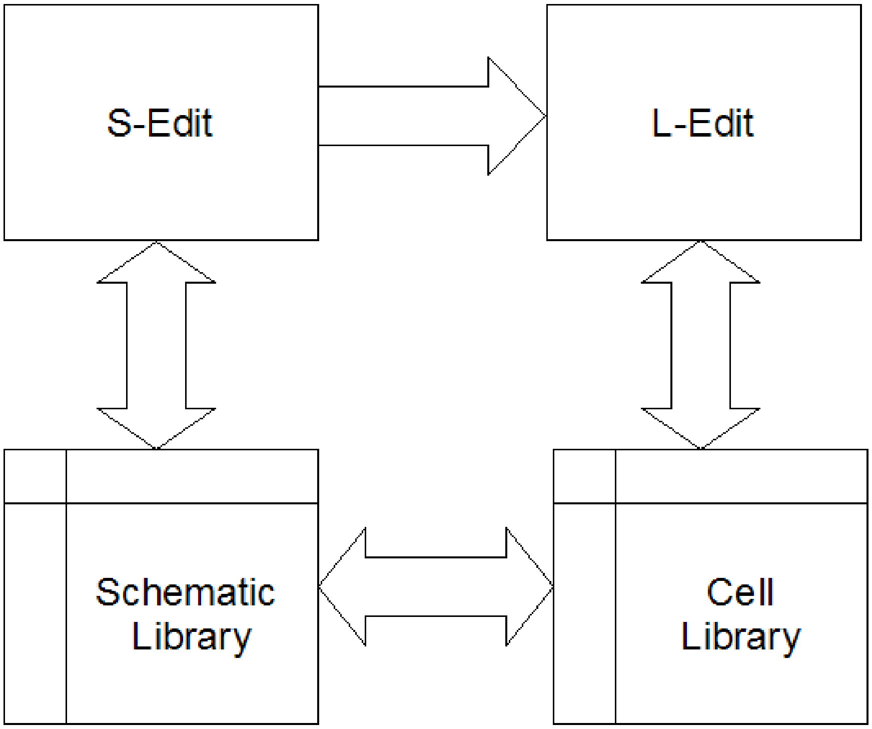
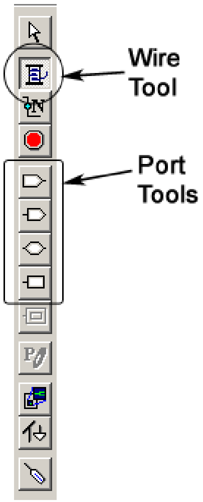
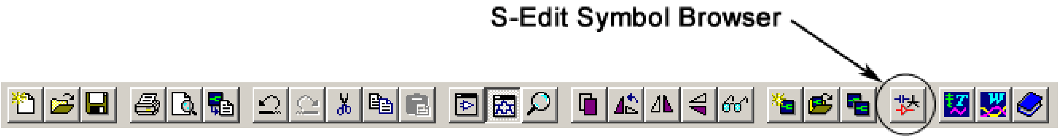


Figure 23. SPR Diagram

7. Standard Place and Route Tutorial: Ring Oscillator

The Standard Place and Route method of design begins in S-Edit. To create schematic driven designs, we must create two libraries with one-to-one associativity between schematics and cells. One library must be in L-Edit TDB format, while the other must be in S-Edit SDB format. Because creating original libraries from scratch would be time consuming, two libraries are already provided under **C:\SPRTutorial**, as **library.tdb** and **library.sdb**. The important icons on the toolbars of S-Edit that we need are shown in the figures to the right and below.





Process:

1. Select File → Open to open the schematic **SPRTutorial.sdb**, located under the **C:\SPRTutorial** directory. Click the Symbol Browser icon as shown in the toolbar above, and the dialog shown below should appear. Note that no schematic symbols are available because we have not added a library yet. To add the library we will use for this tutorial, click Add Library... and add the file **C:\SDBTutorial\library.sdb**.

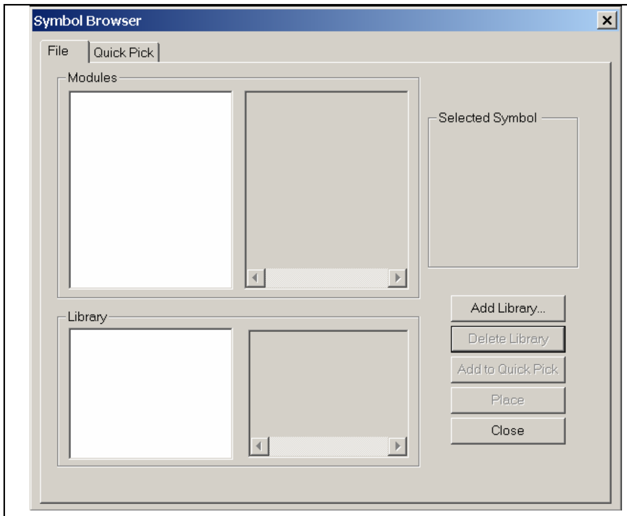
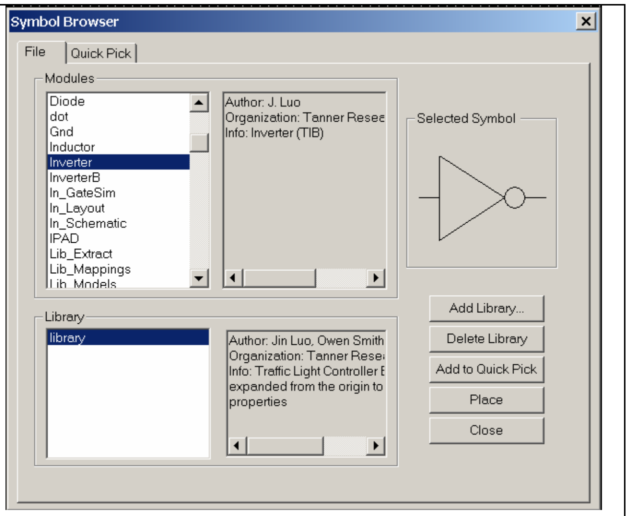


Figure 24. Symbol Browser Dialog



Diodedot
Gnd
Inductor
Inverter
InverterB
In_GateSim
In_Layout
In_Schematic
IPAD
Lib_Extract
Lib_Mappings
Lib_Models

Author: J. Luo
Organization: Tanner Resea
Info: Inverter (TIB)

library

Author: Jin Luo, Owen Smith
Organization: Tanner Resea
Info: Traffic Light Controller E
expanded from the origin to
properties

Figure 25. Symbol Browser with Library Added

2. In the list on the left, select Inverter and click the Place button, then click Close. An inverter will now be placed on your schematic, as shown in Figure 29. Since we are making a ring oscillator, we need five inverters – so using either Ctrl+C or Edit → Copy, copy the inverter, and then paste it four times using either Ctrl+V or Edit → Paste.

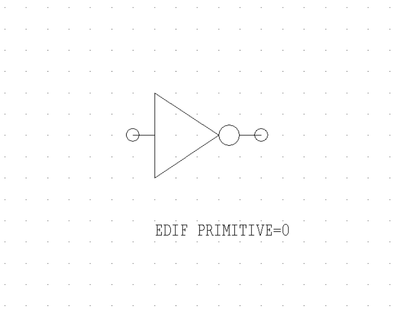


Figure 26. Inverter

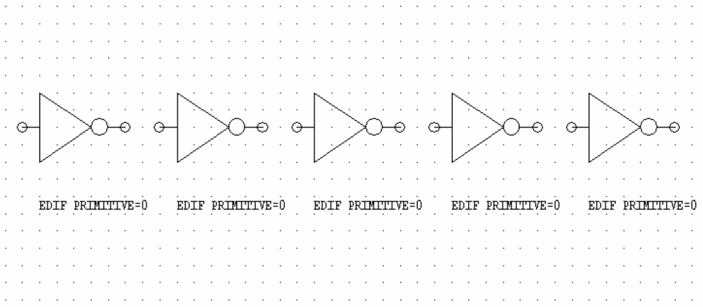


Figure 26. Five Inverters

3. Next, we need to use the wire tool in the left toolbar to connect the oscillators in a ring, as shown in Figure 28 below.

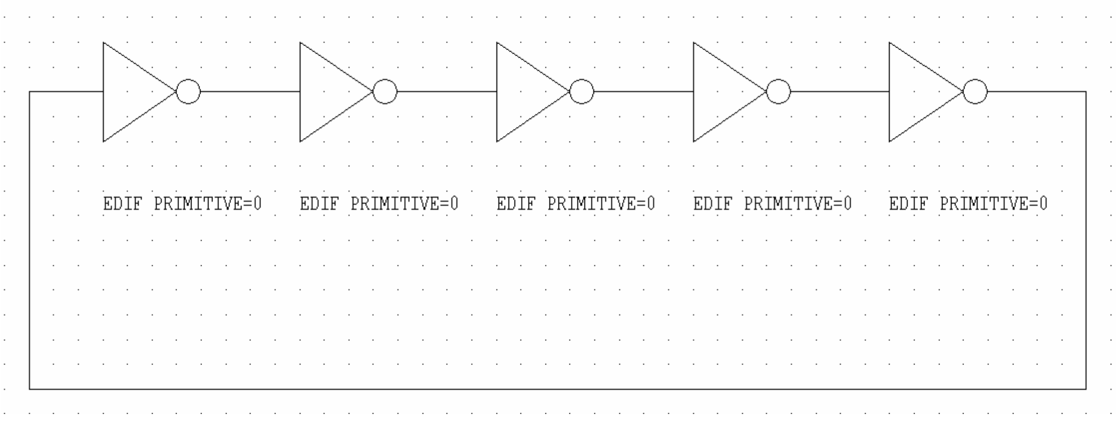
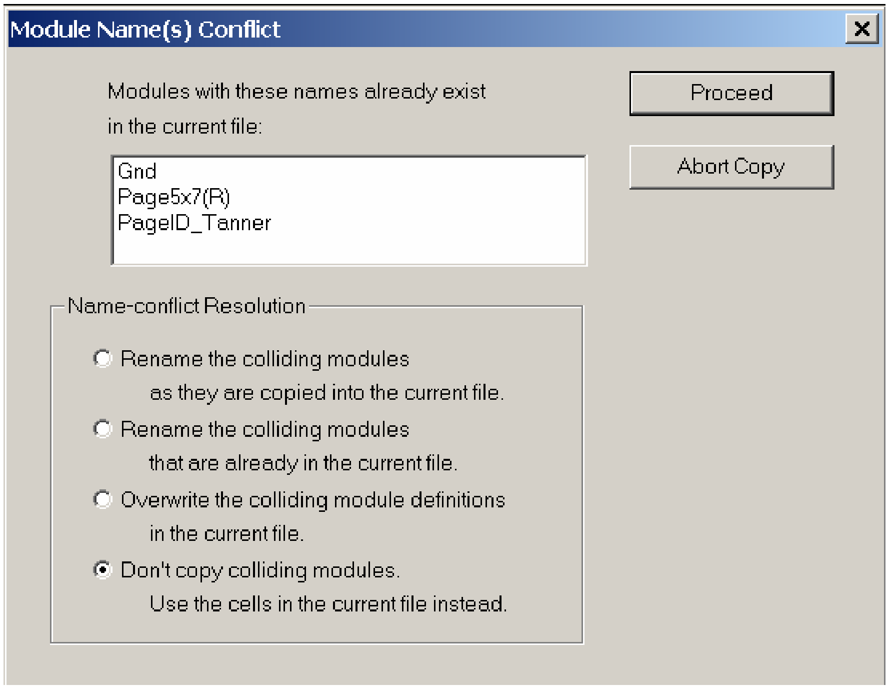


Figure 27. Ring Oscillators Connected

4. We also need connections for Vdd and Ground, as well as an output port to monitor the ring oscillator. To create these, reopen the Symbol Browser and place the parts PadVdd, PadGnd, and PadO. Note that when you try to place these parts on the schematic, S-Edit will complain that there is a name conflict – just select "Don't copy colliding modules" and click Proceed. After placing these parts, the schematic should look like Figure 30.



Modules with these names already exist
in the current file:

Gnd
Page5x7(R)
PageID_Tanner

Name-conflict Resolution

○ Rename the colliding modules
as they are copied into the current file.

○ Rename the colliding modules
that are already in the current file.

○ Overwrite the colliding module definitions
in the current file.

● Don't copy colliding modules.
Use the cells in the current file instead.

Figure 28. Module Name(s) Conflict Dialog

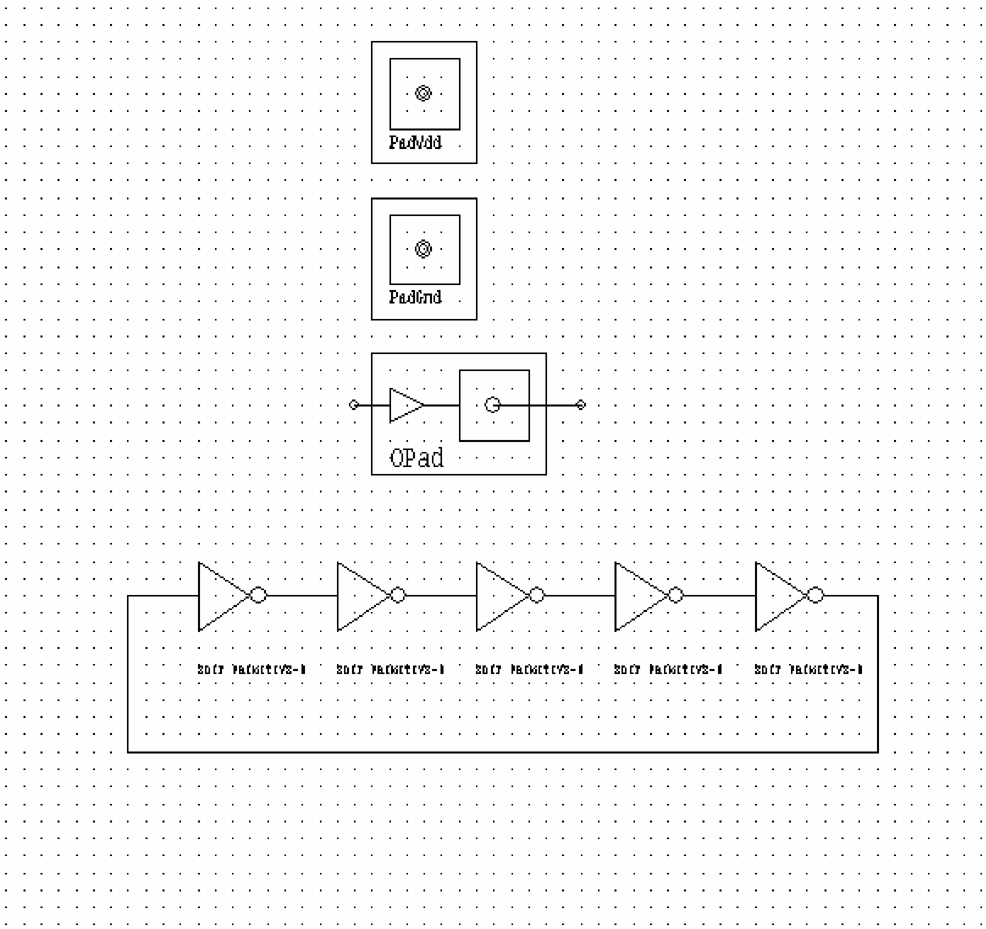


Figure 29. Power and Output Pads Placed

5. After placing these pads, they also need to be connected to S-Edit's ports, so that S-Edit knows where to connect the padframe pads to the core circuitry. This is done by placing bidirectional port connections to both PadVdd and PadGnd, and an output port on OPad. Note that the names of the ports are used to indicate to S-Edit which side of the padframe that connection will be, in the format PAD_XY, where X is either T, B, L, R, representing the Top, Bottom, Left, and Right, of the padframe. Y is a number between 1 and the maximum number of pads per side that your design will support. To name the ports, select Edit → Object Properties. For this tutorial, name the pads PAD_R1, PAD_L1, PAD_R2. Note that S-Edit requires that Vdd and Ground must always be on opposite sides of the padframe.

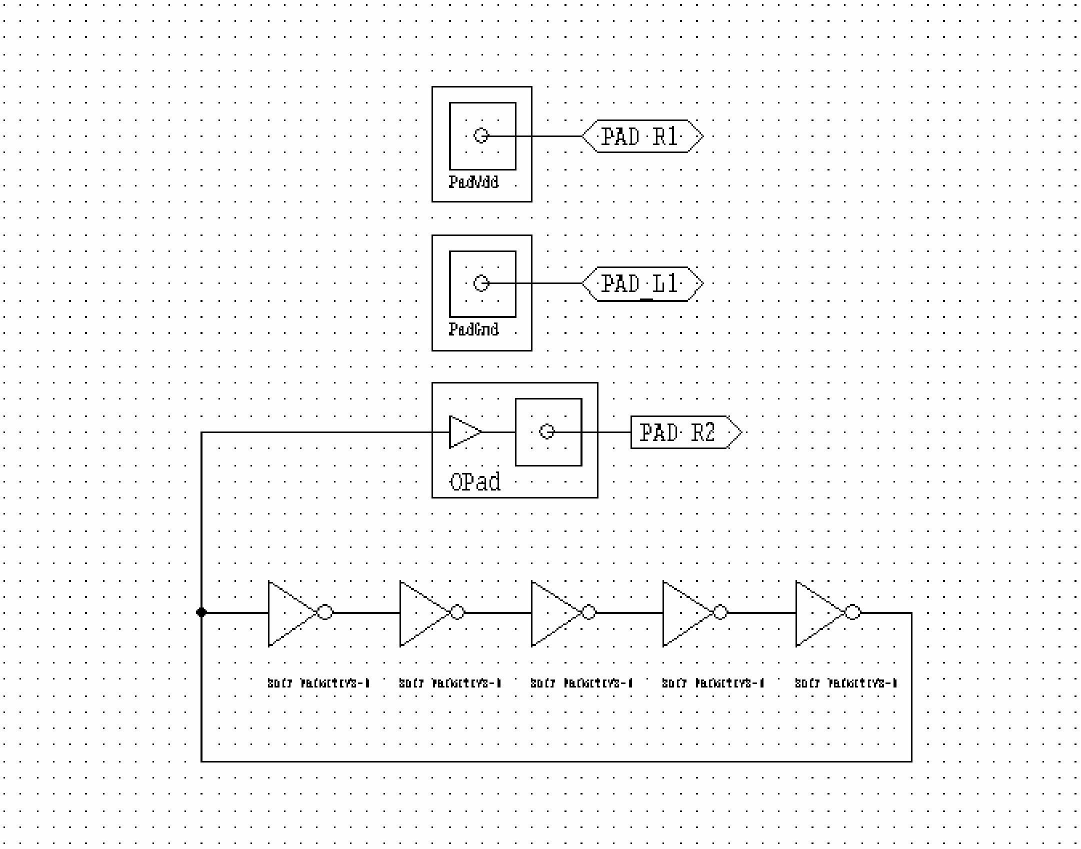
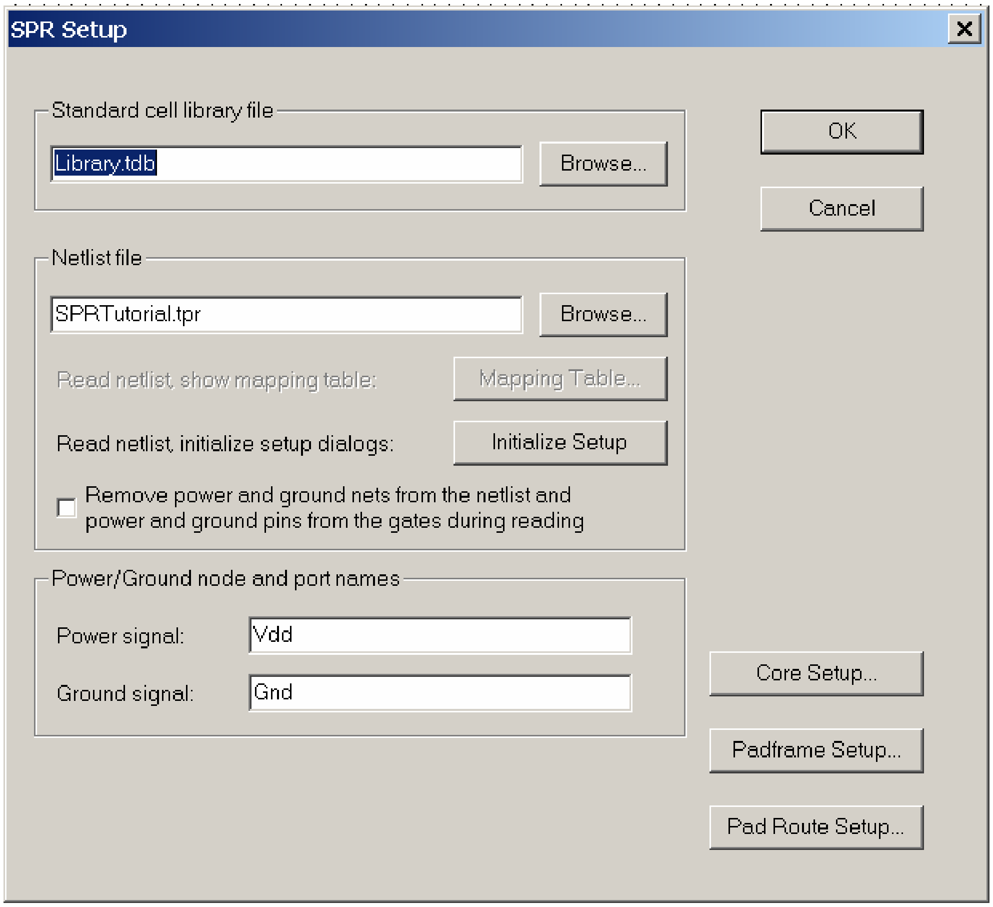


Figure 30. Ports Placed

6. The design is now ready to export to L-Edit. Select File → Export, and export the file as the netlist **SPRtutorial.tpr**. We can now open L-Edit, and open the blank **SPRMain.tdb** located in the **C:\SPRTutorial** folder. To generate our SPR layout, we first need to tell L-Edit some details about what sort of SPR we want. Open Tools → SPR → Setup, as shown in the figure below. In the SPR Setup dialog, we need to select the standard cell library, **C:\SPRTutorial\Library.tdb**, and the netlist file we exported from S-Edit, **C:\SPRTutorial\SPRTutorial.tpr**. After selecting these files, click Initialize Setup to have L-Edit load the files selected. The Core Setup, Padframe Setup, and Pad Route Setup buttons provide more advanced settings that we do not need here. Click OK to continue.



SPR Setup

Standard cell library file: Library.tdb

Netlist file: SPRTutorial.tpr

Read netlist, show mapping table:

Read netlist, initialize setup dialogs:

☐ Remove power and ground nets from the netlist and power and ground pins from the gates during reading

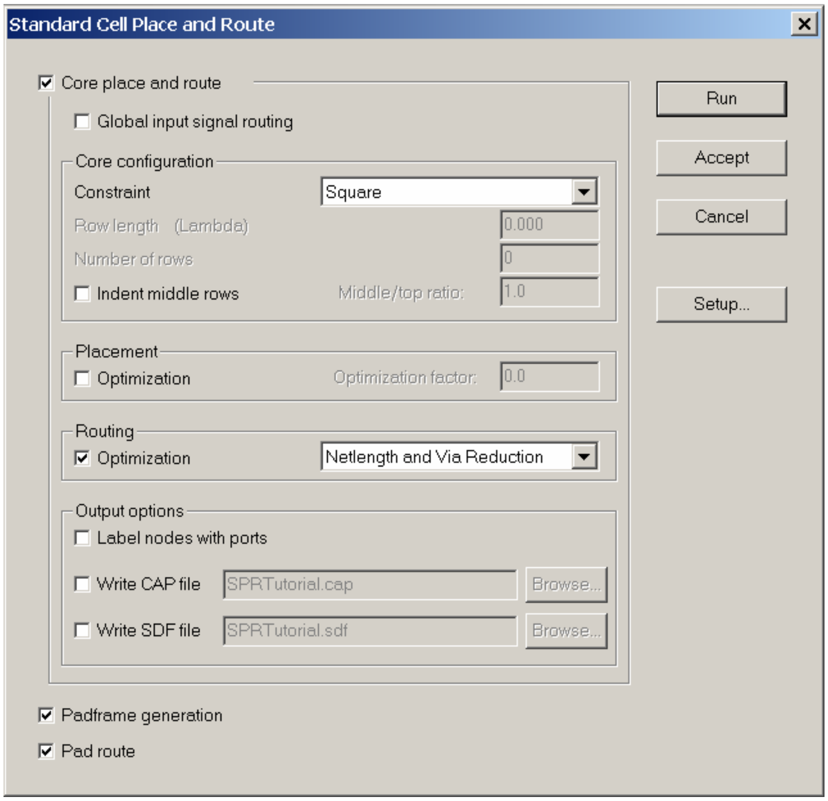
Power/Ground node and port names

Power signal: Vdd

Ground signal: Gnd

Figure 31. SPR Setup Dialog

7. Now we are ready to do the place and route. Select Tools → SPR → Place and Route, and the following dialog should appear. Make sure that the settings you have match the figure shown, and then select Run.



Standard Cell Place and Route

☑ Core place and route

□ Global input signal routing

Core configuration

Constraint: Square

Row length (Lambda): 0.000

Number of rows: 0

□ Indent middle rows   Middle/top ratio: 1.0

Placement

□ Optimization   Optimization factor: 0.0

Routing

☑ Optimization   Netlength and Via Reduction

Output options

□ Label nodes with ports

□ Write CAP file   SPRTutorial.cap

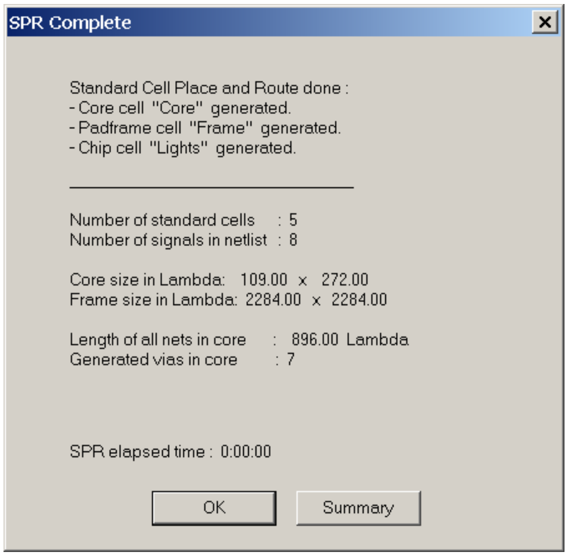
□ Write SDF file   SPRTutorial.sdf

☑ Padframe generation

☑ Pad route

Figure 32. Standard Cell Place and Route Dialog

8. When the SPR is complete, a SPR Complete dialog will appear giving you information about the size of your design, the number of cells/signals, and various other information. Clicking OK will return to the layout, displaying something similar to what is shown in Figure 35. Inspection will reveal that our five inverters have been automatically generated, placed, and connected to each other and the padframe.



```
Standard Cell Place and Route done :
- Core cell "Core" generated.
- Padframe cell "Frame" generated.
- Chip cell "Lights" generated.

Number of standard cells    : 5
Number of signals in netlist  : 8

Core size in Lambda:   109.00  x   272.00
Frame size in Lambda:  2284.00  x  2284.00

Length of all nets in core    :   896.00  Lambda
Generated vias in core        : 7

SPR elapsed time :  0:00:00
```

Figure 33. SPR Complete Dialog

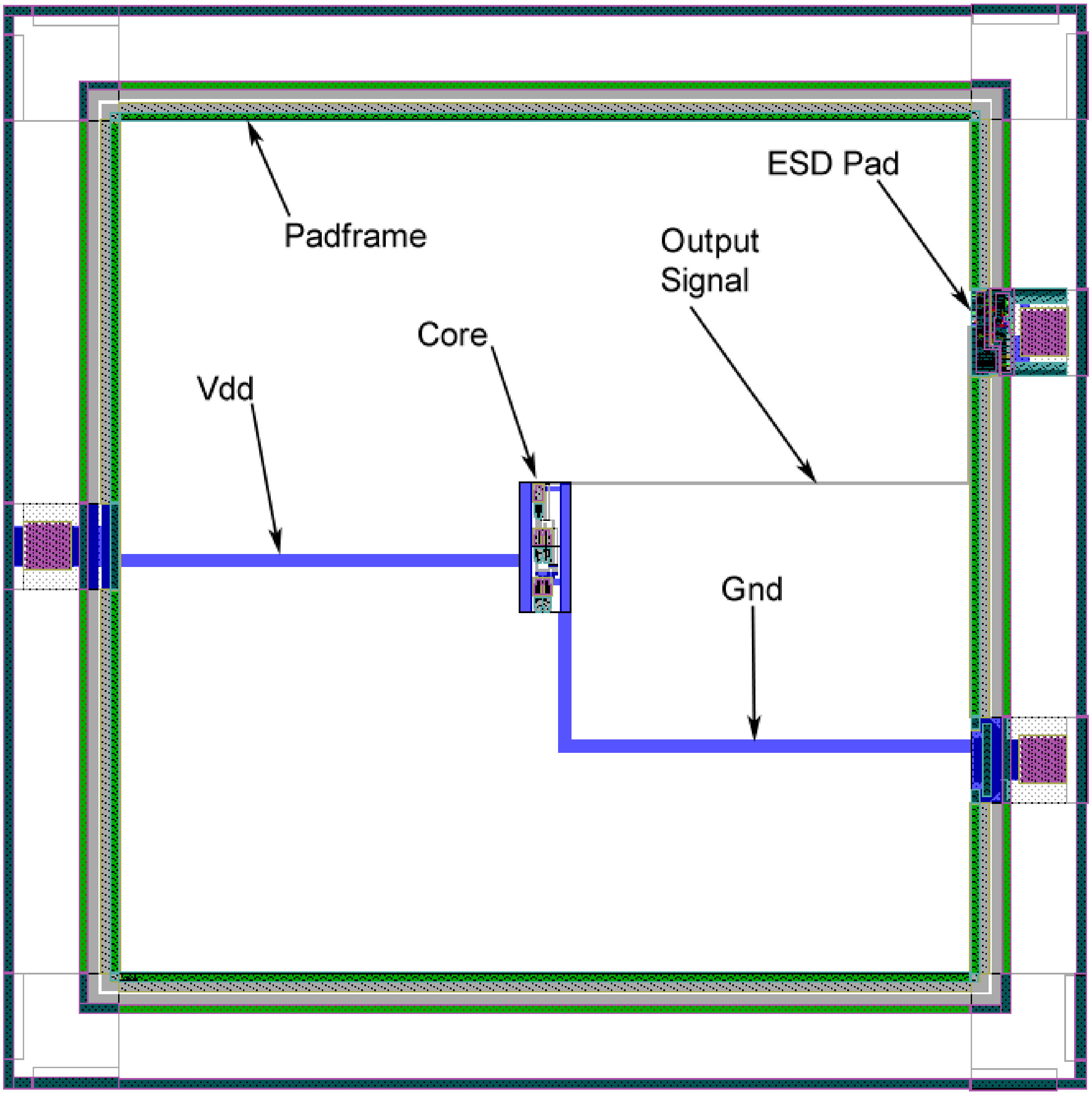


Figure 34. SPR-Generated Layout

Rescaling and Dimensions Note: *Note that the dimensions of this padframe can be adjusted.* **Setup → Design** *shows the mapping of units of lambda (L-Edit's built in unit) to microns. This can be adjusted to fit the design to a specific size. Using the ruler tool on the L-Edit toolbar will give the dimensions of each portion of the design. A more advanced method of rescaling is to use the built-in rescaling wizard. This is found under* **Setup → Rescale Wizard**. *This wizard allows the selective scaling of design elements by either a percentage or a ratio.*