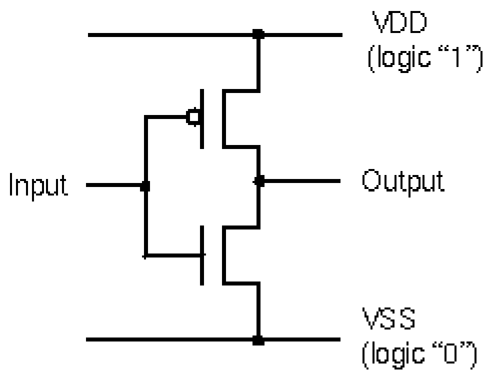
A Step-by-Step Example: Layout of a CMOS Inverter Using SkyWater Technology (SKW) 130nm Process

Ryan Ridley, Teo Ene, and James E. Stine
Oklahoma State University
VLSI Computer Architecture Research Group

This document attempts to give a quick breakdown of how to design an inverter in SkyWater Technology (SKW) 130nm process using the Magic VLSI layout editor. Although this tutorial really breaks down the process, it is important to stress the importance of floorplanning before embarking on this process. Floorplanning makes custom logic far simpler and also allows for better ideas to be integrated into the design. Haphazard planning when creating one's layout can result in layout bloat and possibly incredibly poor performance.



The inverter consists of an NMOS transistor and a PMOS transistor. The channel length, L, of the two devices will be set at the minimum allowed by the technology: 0.15um. The channel width, W, of the devices will be set to 0.42um and 0.68um, respectively. The sizings of each transistor allow designers to custom the current-drive for each gate based on fan-in, fan-out, and other influential factors. Note that the source and the body (p-substrate) of the NMOS are connected to ground (GND node), while the source and the body (n-well) of the PMOS are connected to the positive supply (Vdd node) even though they are not shown. This is because for most introductory circuit classes, the three port model is used (shown), however, when we are dealing with VLSI design, we must utilize the four-port model which accounts for substrate connections which are vital to preventing the body-effect. We assume that the target fabrication technology is SkyWater Technology (SKW) 130nm CMOS process. The technology name is **sky130**. As this is not as scalable CMOS technology, there is no lambda, and we work directly with physical units.

In this example, we use the basic Magic drawing command to lay out the inverter. Remember that you can undo just about everything by typing :undo or u.

Workable Sizes within SKW 130nm

The SkyWater Technology has a vast myriad of elements that allow creation of digital and analog designs. However, the sizing within the transistors may cause some issues related to its simulation due to the SPICE deck being created differently. Therefore, choosing the correct sizes to allow good simulation is important. We have created some excellent results in a spreadsheet that showcase the allowable transistor sizes that one can use.

- These two csv files show the list of allowable sizes for NMOS and PMOS transistors within the Skywater Technology (SKW) 130nm binned SPICE deck:
  - https://foss-eda-tools.googlesource.com/skywater-pdk/libs/sky130_fd_pr/+/refs/heads/main/cells/nfet_01v8/sky130_fd_pr__nfet_01v8.bins.csv
  - https://foss-eda-tools.googlesource.com/skywater-pdk/libs/sky130_fd_pr/+/refs/heads/main/cells/pfet_01v8/sky130_fd_pr__pfet_01v8.bins.csv
- The following Google Sheets spreadsheet presents an analysis of circuit behavior for a spectrum of inverters with different NMOS and PMOS widths. It also includes many parameters found through SPICE simulation:
  - https://docs.google.com/spreadsheets/d/1WSAnwUpeRn5gqAJrMtAzc1bWt9NWs4_AfDjbFqdMZFA
  -
  - --
  - James E. Stine, Jr., Ph.D.
  - Edward Joullian Endowed Chair and Professor in Engineering
  - Oklahoma State University
  - Department of Electrical and Computer Engineering
  - VLSI Computer Architecture Research Group
  - 215 General Academic Building
  - Stillwater, OK 74078 USA
  - E-mail: james.stine@okstate.edu
  - Phone: (405) 744-9244
  - Research URL: http://vlsiarch.ecen.okstate.edu
  - BouldinList URL: http://bouldinlist.ecen.okstate.edu
  - --
  -
  - /edit#gid=598635113
- The next Google Sheets spreadsheet has some sizes in it for a small number of standard-cells just to give an idea of the overall speed one could obtain. This data is subjective as speed is based on its loading/interconnection network and would require multiple simulations to obtain a good feeling of the sizing required for your application. The main goal is to get a general sense of how things are for the 130nm process and not to really give you all the possible combinations:
  - https://docs.google.com/spreadsheets/d/1BTvnNFhkFqT4uqZs7uIK6D6QAJ4tRYkpU44OZFgme9Q/edit#gid=1572495657.

CAUTION: Use of any non-allowable sizes could result in your simulation not working!

Starting Magic (layout) design

Start Magic from the Unix prompt:

```
prompt> magic inverter
```

Expand the graphics window over the screen, but so that you can still see the text window prompt. Remember that the graphics window must be active and the cursor must always point to the graphics window. Zoom out and show the lambda grid so that it is easier to see what you are doing.

ctrl z or **:zoom 0.5**

Something that is useful to do at the start is set the grid to be 30. Since the scale is currently set to 0.05um, setting the grid to 30 will show boxes of size 0.15um x 0.15um which will help when painting transistors.

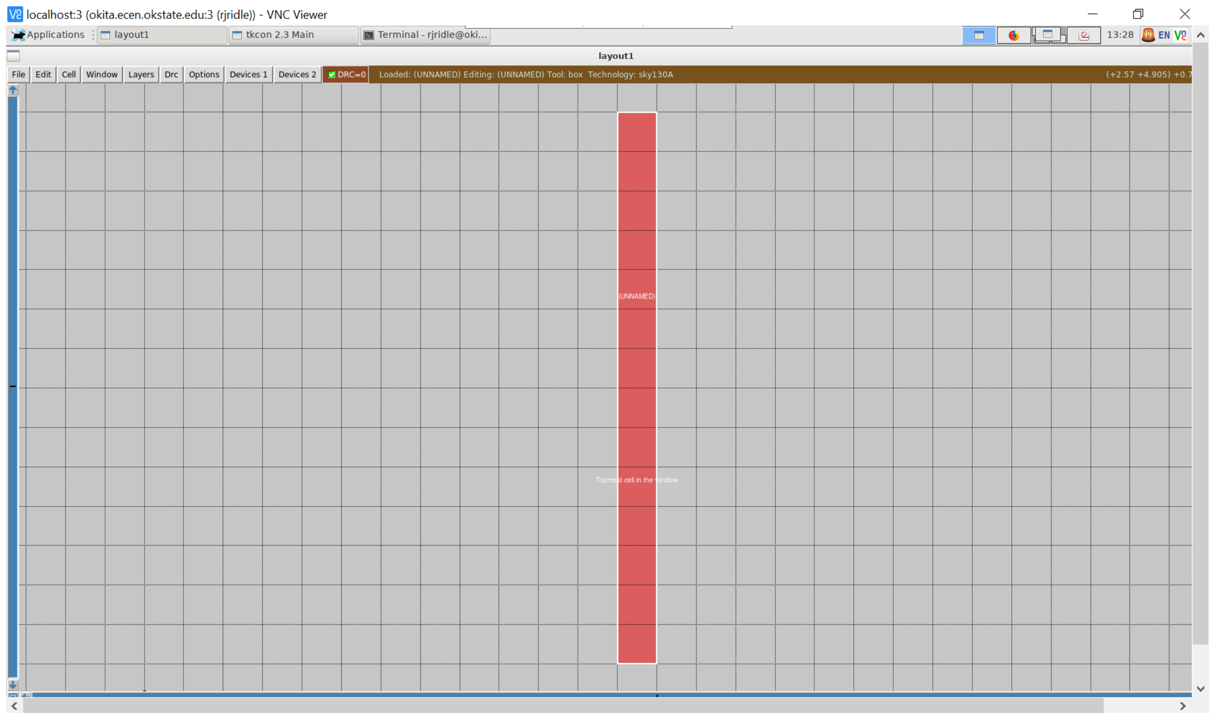
```
:grid 30
```

Laying down Polysilicon for the Gate of a Transistor

The order in which layers are placed in the graphics window is not important, and so the layout steps described here are by no means unique. Since the device channel width W and length L are specified, we can start by painting the **poly** of each device. Start by using the mouse to create a box that is 0.15um x 1um. The length of the box is not important at this stage since it will be adjusted later. Once you have a correct sized box in the window use the command:

```
:paint poly
```

You should have a layout that resembles this:


localhost:3 (okita.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
Applications
layout1
tkcon 2.3 Main
Terminal - rjridle@oki...
13:28
EN
layout1
File
Edit
Cell
Window
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A
(+2.57 +4.905) +0.7
(UNNAMED)
Topmost cell in the window

Note: links to high resolution images are provided to help clarify the image, if needed.

https://drive.google.com/file/d/1B6F1DR_F3VM7tlgO_5IkjrnkfltCIZ60/view?usp=sharing

Creating diffusion for the n/p active regions

Next, we can paint the diffusion layer of both the NMOS and PMOS. The width and length of these diffusion layers can change depending on what drive strength you want these transistors to have. Note that throughout this document, width refers to the vertical dimension, while length refers to the horizontal dimension.

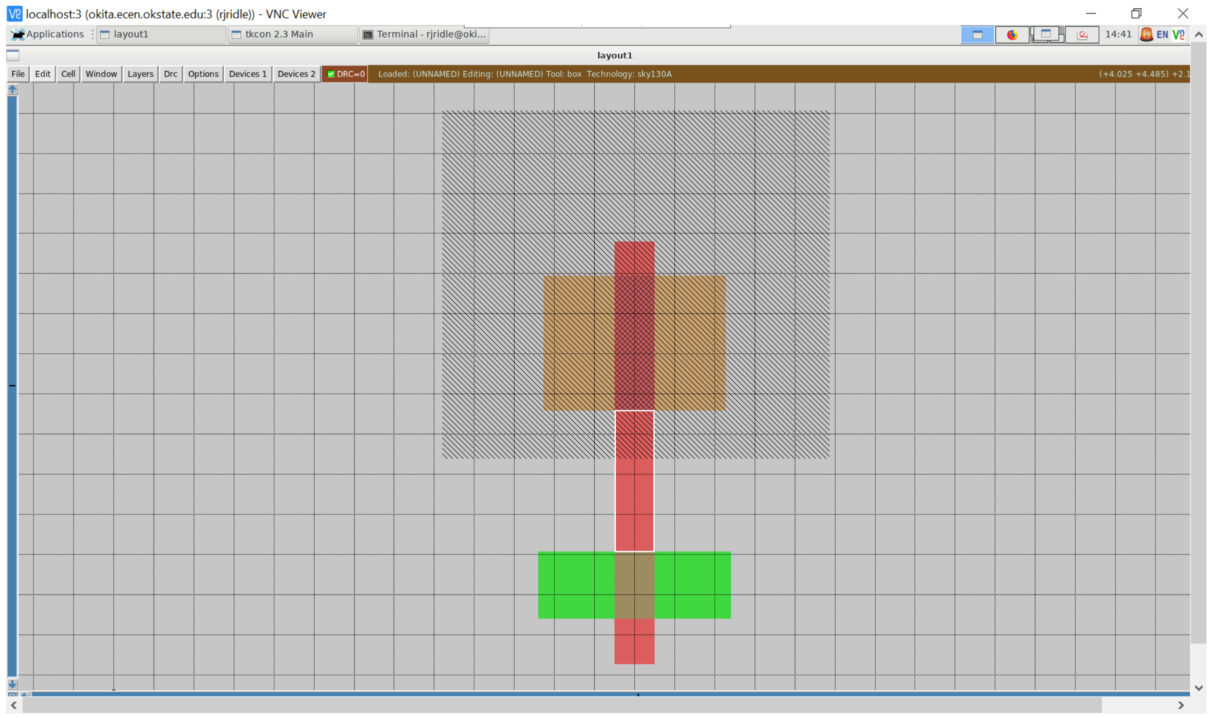
The wider the diffusion layer is, the more drive strength the transistor will have. This also affects the size of the diffusion contacts that connect the local interconnect metal layer to the diffusion layer. Larger contacts will be needed to drive transistors with larger widths. In this tutorial, we will be using the minimum sizes for everything. The P-type transistor is usually 1.5 to 2 times the size of the N-type transistor, and we will be following that here. Paint a P diffusion layer of 0.68um x 1.00um near the top of the poly.

:paint pdiff

and an N diffusion layer of 0.68um x 0.42um near the bottom of the poly.

:paint ndiff

A layer of **N-well** surrounding the **pdiff** layer is also necessary to get rid of the DRC's that will pop up due to placing **pdiff**. The size of the **N-well** does not matter as long as it is large enough to get rid of the DRC's. Also, make sure that the poly hangs over the diffusion by at least 0.13um or else it will cause a DRC.


localhost:3 (okita.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
Applications
layout1
tkcon 2.3 Main
Terminal - rjridle@oki...
14:41
layout1
File
Edit
Cell
Window
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A
(+4.025 +4.485) +2.1

https://drive.google.com/file/d/1oXZnWxswACDDfz9OvlDviUBo5bFDSbdx/view?usp=sharing

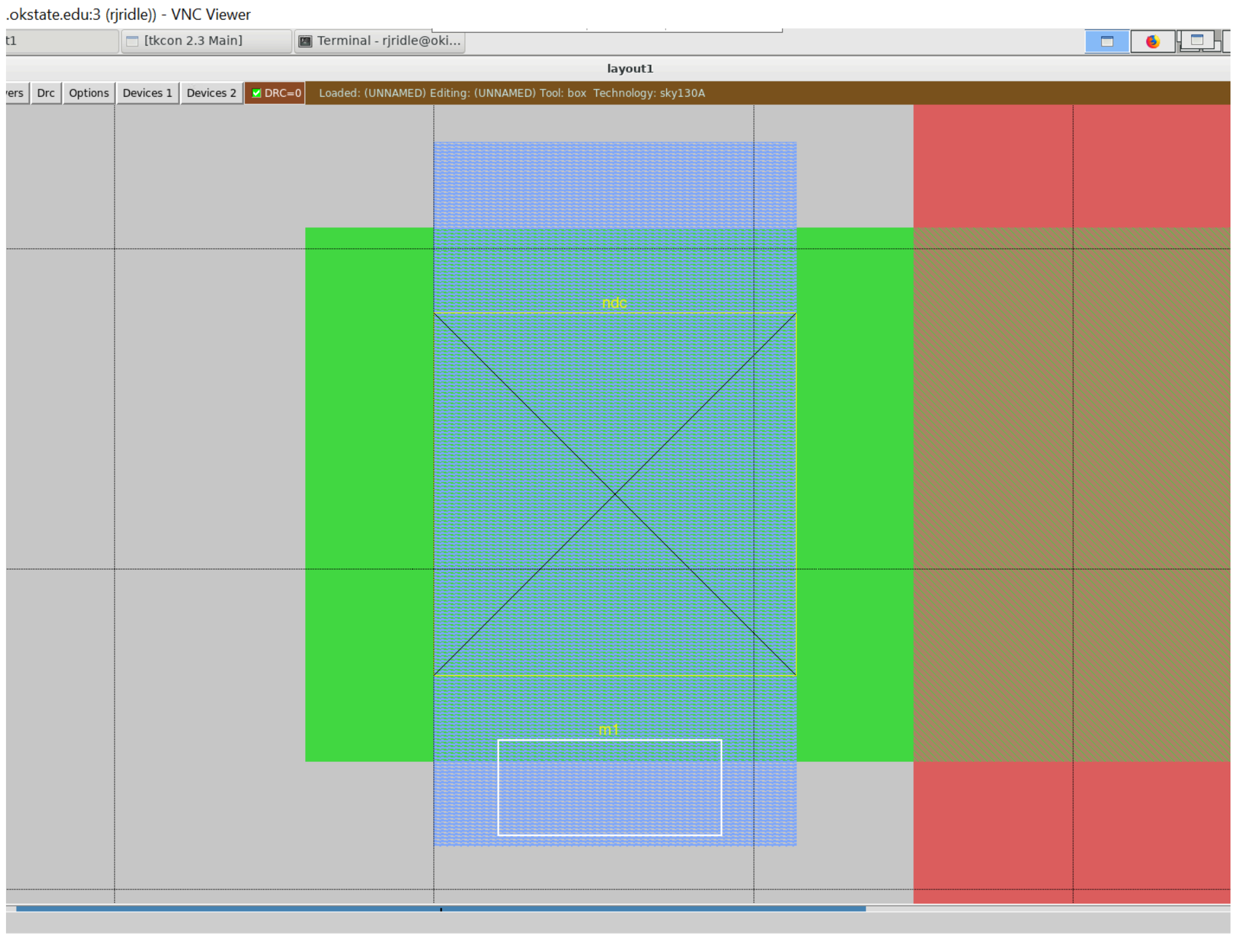
Creating contacts (cuts) within the diffusion

Next, the diffusion contacts can be placed that will connect the diffusion layers to the local interconnect layer. The smallest size allowable for contacts is 0.17um x 0.17um. Contacts can be made larger if needed, but this will increase the width of the diffusion layer since there is a minimum amount of diffusion overlap that is needed. A certain amount of local interconnect metal will also need to be placed on the top and bottom of each contact to satisfy DRC. Create a 0.17um x 0.17um box over the diffusion layer on either side of the poly, and paint a contact. If you are making contacts for N diffusion, the command is:

:paint ndc

Then paint a local interconnect layer on the top and bottom, that touches the contact, and extends out at least 0.08um.

:paint li


.okstate.edu:3 (rjridle)) - VNC Viewer
t1
[tkcon 2.3 Main]
Terminal - rjridle@oki...
layout1
ers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A
ndc
m1

https://drive.google.com/file/d/1vv5bXL1JVrgY7rBjUfL2YM2P2z_xrNh6/view?usp=sharing

If making contacts for P diffusion, the command is:

:paint pdc

Since the size of the PMOS is usually bigger than the NMOS, the size of the P-diffusion contacts should reflect that. Contacts can be any size as long as they are bigger than the minimum size (0.17um x 0.17um) and they don't overlap the diffusion. After placing the contact, make sure to paint local interconnect around it to get rid of the DRC's:

:paint li

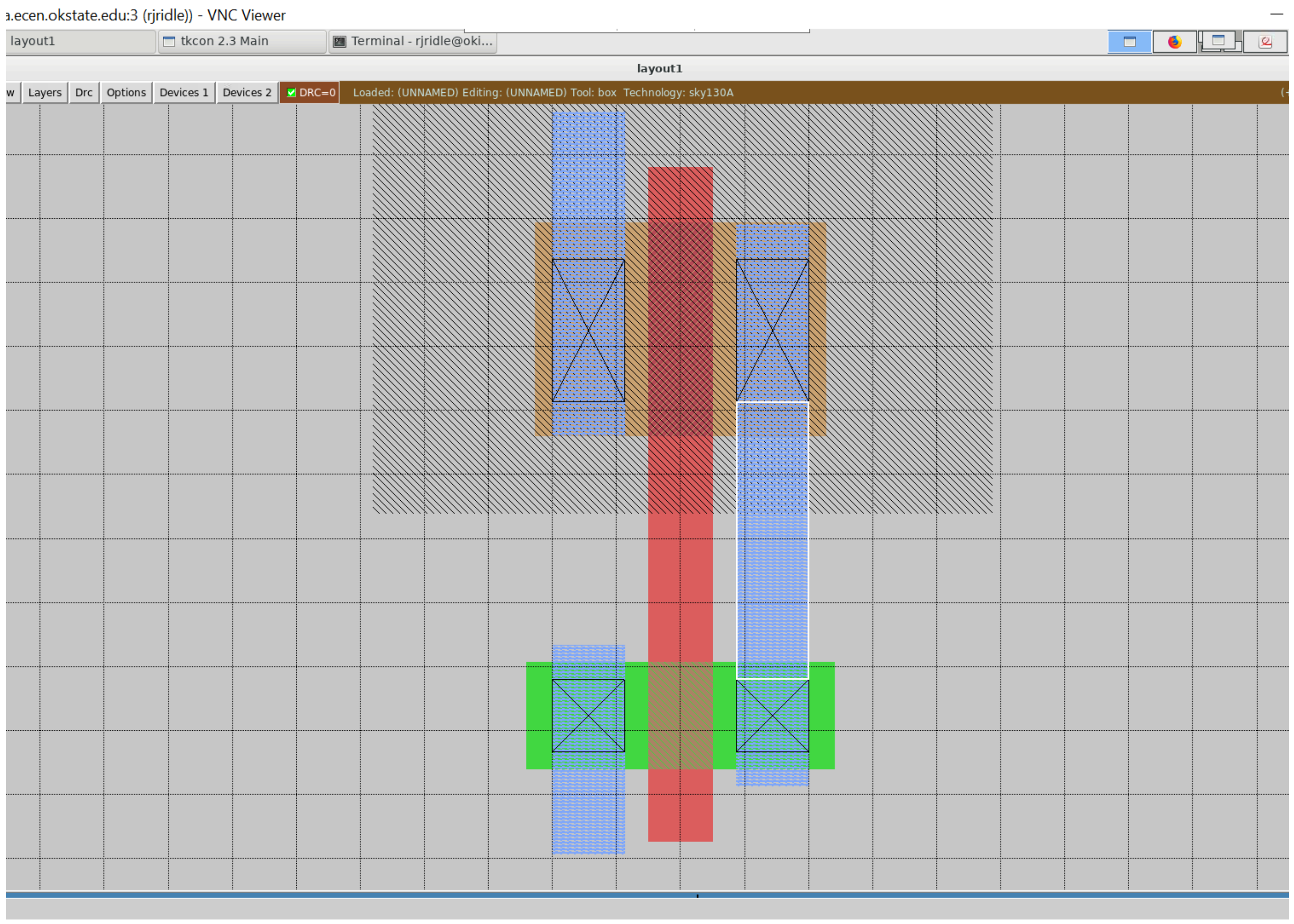
Possible DRC encounters

Here are some common DRC errors that might come up when placing contacts:

DRC	What the DRC means
(N/P)-diffusion overlap of (N/P)-diffusion < 0.06um in one direction	*The diffusion extending out from the contact is < 0.06. Try increasing diffusion area around contact*
Diffusion contact to gate < 0.055um	*The contact is too close to the poly. Move the contact farther away*
Local interconnect overlap of (N/P)-diffusion < 0.06um in one direction	*There is not enough local interconnect metal extending from the top or bottom of the contact. Increase the length of the local interconnect metal on the top or bottom*
(N/P)-diffusion contact width < 0.17um	*The contact is too small. The minimum size for contacts is 0.17um x 0.17um. Increase the size of the contact*
Diffusion width < 0.15um	*The diffusion that was painted is too small. This can happen if a smaller rectangle of diffusion that is less than 0.15um x 0.15um is connected to a larger one. Magic will see the smaller rectangle as being not big enough. Remove the irregular square to make a perfect rectangle*
N-diffusion spacing to N-well < 0.34um	*The N-diffusion needs to be moved further away from the N-well. An easy way to do this is to select the whole bottom half of the design using* ***a*** *on the keyboard, then while holding shift, press* ***w****. This will move the selected area down while extending non-selected things that are connected to it.*
Local interconnect spacing < 0.17um	*Two separate layers of local interconnect are too close to each other*
Poly overhang of transistor < 0.13um	*There must be at least 0.13um of poly that extends away from the diffusion on the top and bottom. Try extending the poly on the side that has the DRC*

This list is by no means comprehensive. Determining what a DRC means and how to fix it will take trial and error. Refer back to this list to help fix unknown DRC's since a lot of them are similar, but in different areas and in different layers.

After placing contacts for both N-diffusion and P-diffusion, we can then connect the source of the PMOS to the drain of the NMOS. This is the output of the inverter. Using local interconnect, paint a rectangle that connects the right PMOS contact to the right NMOS contact. Should resemble this:

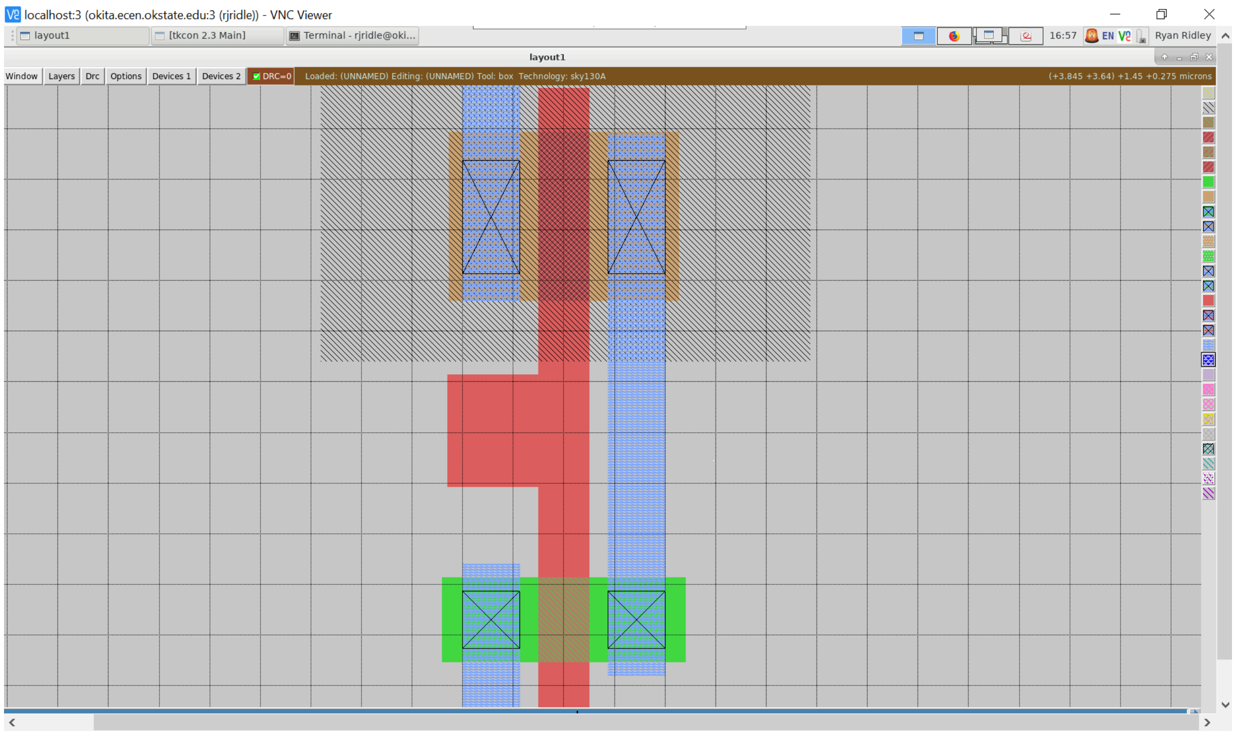

a.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
layout1
tkcon 2.3 Main
Terminal - rjridle@oki...
layout1
w
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A

https://drive.google.com/file/d/1CnPXOsTLdA1ghqFQgcKhK3BbnwfxNLOV/view?usp=sharing

Which contacts are chosen to be connected does not matter as long as the rails are connected properly in a way that reflects the inverter. This means that either side of the transistor can act as the source or drain. It's up to the user to determine which way to orient the transistor. Choosing to connect the contacts on one side will make the layout much cleaner and less complicated. We can check to make sure that this metal is connected to both the NMOS and PMOS. Place your cursor on the layer you want to know the connections to. Press **s** on your keyboard outlines that layer. By hitting **s** multiple times on that layer, magic will outline things it is connected to. Keep hitting **s** until you don't see anything new outlined, or if the same outlines repeat over and over again. By hitting **s** on the metal layer we just placed, Magic should outline the local interconnect along with the contacts and diffusion on the right side. If you don't see this, then the local interconnect is not connected to the diffusion contacts. Check and make sure the correct diffusion contacts were used and if the local interconnect is actually touching the contacts.

Polycontacts

Next, a contact that connects local interconnect to poly can be placed. The poly contacts follow similar rules to the diffusion contacts. There must be enough surrounding poly for the contact, as well as local interconnect on opposite sides. First, place a 0.27um x 0.33um rectangle on the left side of the poly.



Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A

(+3.845 +3.64) +1.45 +0.275 microns

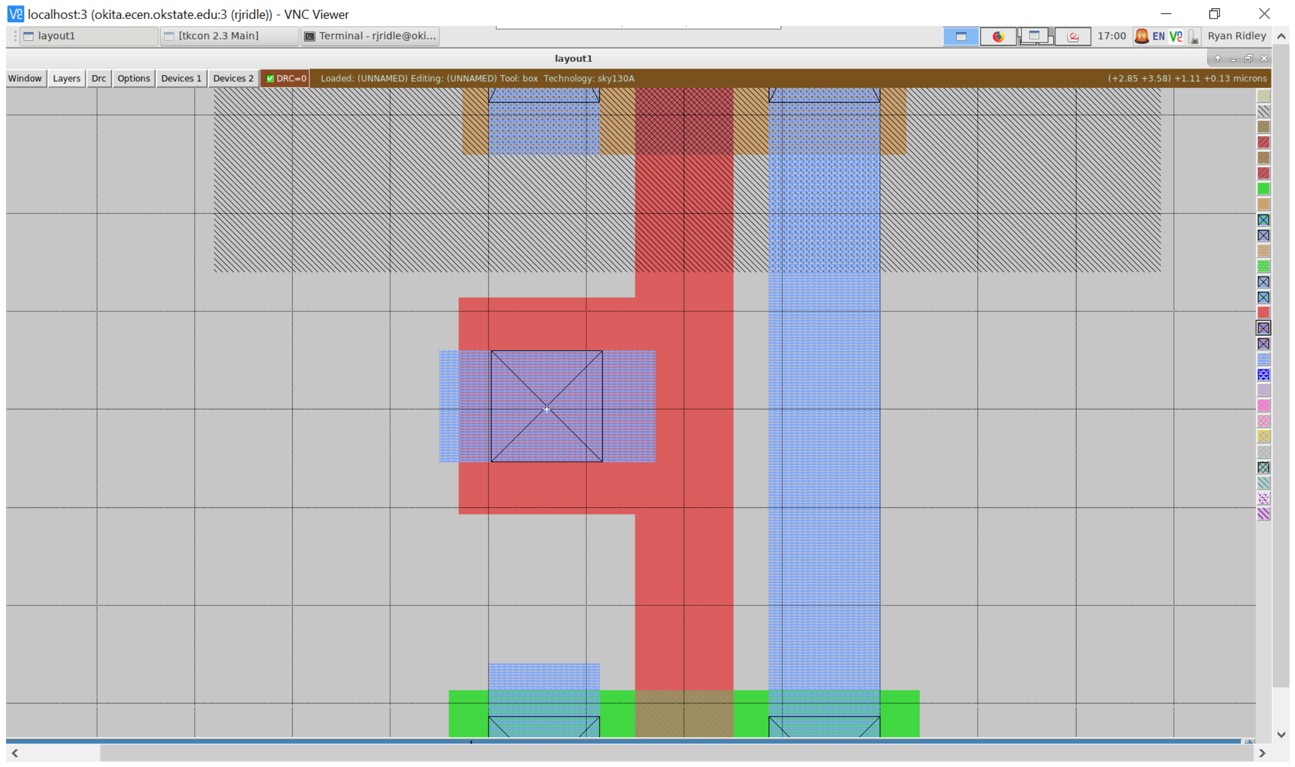
https://drive.google.com/file/d/1kXL2sCabFIcO4YEk2CwlY3SqX9dJ_Pr2/view?usp=sharing

:paint poly

Then place a 0.17um x 0.17um poly contact in the middle of the new rectangle.

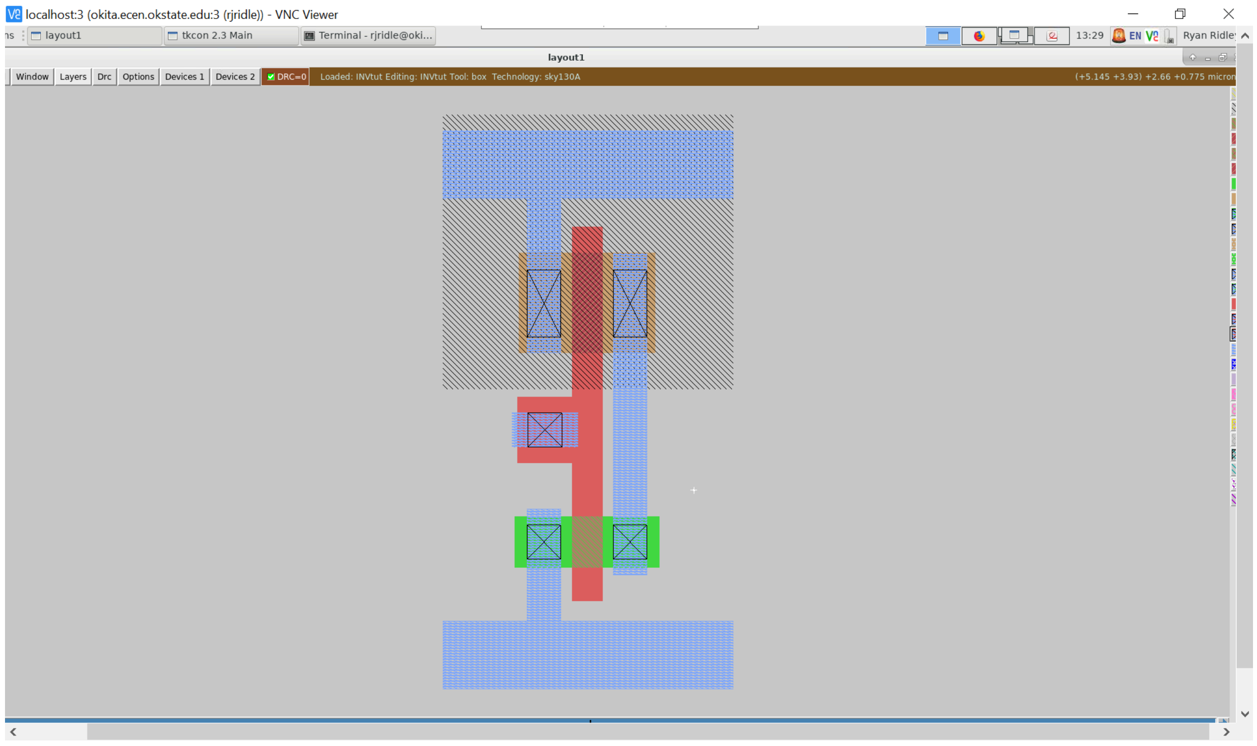
:paint pc

Remember to add local interconnect on the left and right side until there are no more DRCs.


localhost:3 (okita.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
layout1
[tkcon 2.3 Main]
Terminal - rjridle@oki...
17:00
EN
Ryan Ridley
layout1
Window
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A
(+2.85 +3.58) +1.11 +0.13 microns

https://drive.google.com/file/d/1QmA9fENcNIYHU0PxhvbUjuethg9TLuEf/view?usp=sharing

Next, the power and ground rails can be placed. Local interconnect placed above the PMOS and below the NMOS will act as the power and ground rails, respectively. Paint some local interconnect above the PMOS and below the NMOS. A rule of thumb to follow when determining the size of the rails is to double the minimum height of the metal that is being used for the rails. Since we are using local interconnect, the minimum height is 0.17um. Doubling that gives us 0.34um as the height of the rails. This is just a suggestion to follow and not the gold standard. When developing a standard cell kit, the size of every cell needs to be the same. This is because cells (AND gates, OR gates, XOR gates, etc.) need to be stitched together when the design is being placed and routed. If cells are different heights, it will cause DRC errors when trying to place down the design on the chip. Once the height of every cell is decided on, use any extra height possible to fill with the rails. For this example, we'll just use 0.34um for the height of the rails. Your design should resemble something like this:


localhost:3 (okita.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
layout1
tkcon 2.3 Main
Terminal - rjridle@oki...
13:29
Ryan Ridle
layout1
Window
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: INVtut Editing: INVtut Tool: box Technology: sky130A
(+5.145 +3.93) +2.66 +0.775 micron

https://drive.google.com/file/d/11gGrqiKRrOYHvxK_VNYR8FdNqceuUi4c/view?usp=sharing

Substrate Contacts for Avoiding Unwanted Body-Effect Voltages

Next, substrate contacts need to be placed. N-substrate contacts need to be placed on the power/vdd rail while P-substrate contacts need to be placed on the ground rail. When placing these substrate contacts, a layer of substrate diffusion needs to be placed along with them. Starting with the power rail, paint a layer of N-substrate diffusion (nsd) that covers the area where the contact will be placed. The amount of the diffusion placed does not matter as long as there is enough surrounding the contact to not cause a DRC error.

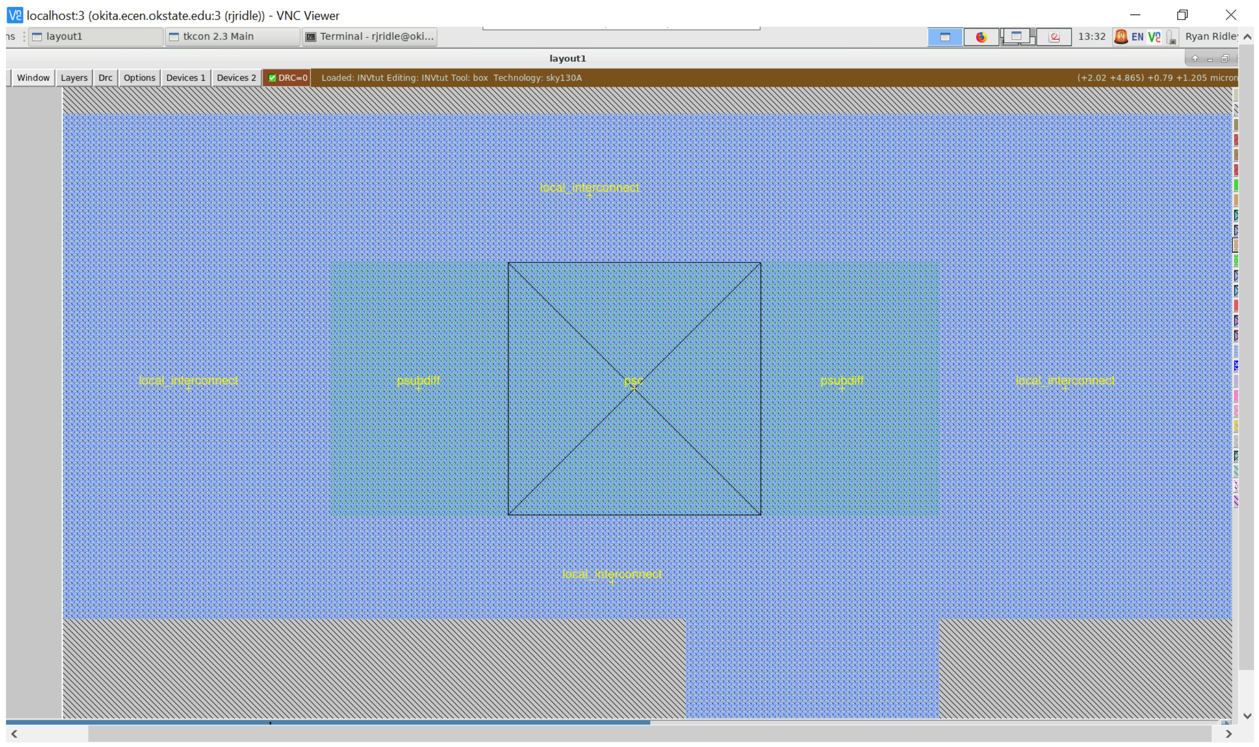
:paint nsd

Next, paint an N-substrate contact over the substrate diffusion.

:paint nsc

The amount of substrate contacts that are placed on the rails does not matter DRC-wise, but it is recommended to put as many as there are transistors in the layout plus one. For this example, since we have one NMOS transistor, two contacts were placed on the ground rail. Since there is one PMOS, two were also placed on the power rail. Here is a close up of an N substrate contact on the vdd rail.

Labels were placed on the different layers to differentiate between them. These labels do not need to be placed in the final design.

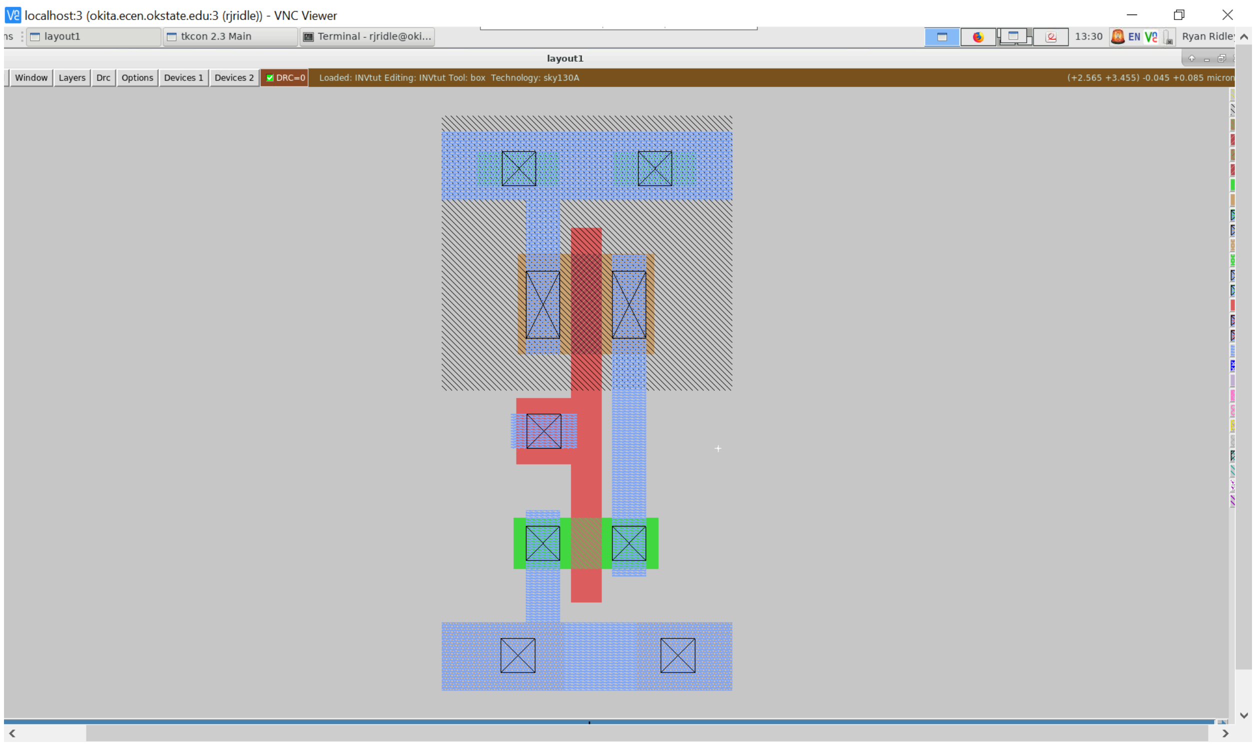


Loaded: INVtut Editing: INVtut Tool: box Technology: sky130A (+2.02 +4.865) +0.79 +1.205 micron

https://drive.google.com/file/d/17iRIx7yQOtRFNAfF7ZT6pV6J08D3PObJ/view?usp=sharing

Remember that substrate diffusion needs to be placed UNDER the contact and not just surrounding it. After placing the contacts on the power rail, move to the ground rail and place the P-substrate diffusion and P-substrate contacts.

:paint pds :paint psc


localhost:3 (okita.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
layout1
tkcon 2.3 Main
Terminal - rjridle@oki...
13:30
Ryan Ridle
layout1
Window
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: INVtut Editing: INVtut Tool: box Technology: sky130A
(+2.565 +3.455) -0.045 +0.085 micron

https://drive.google.com/file/d/1vP-TLjporoiJ8VL_vAc7VV0qKXwXu19F/view?usp=sharing

Creating Labels

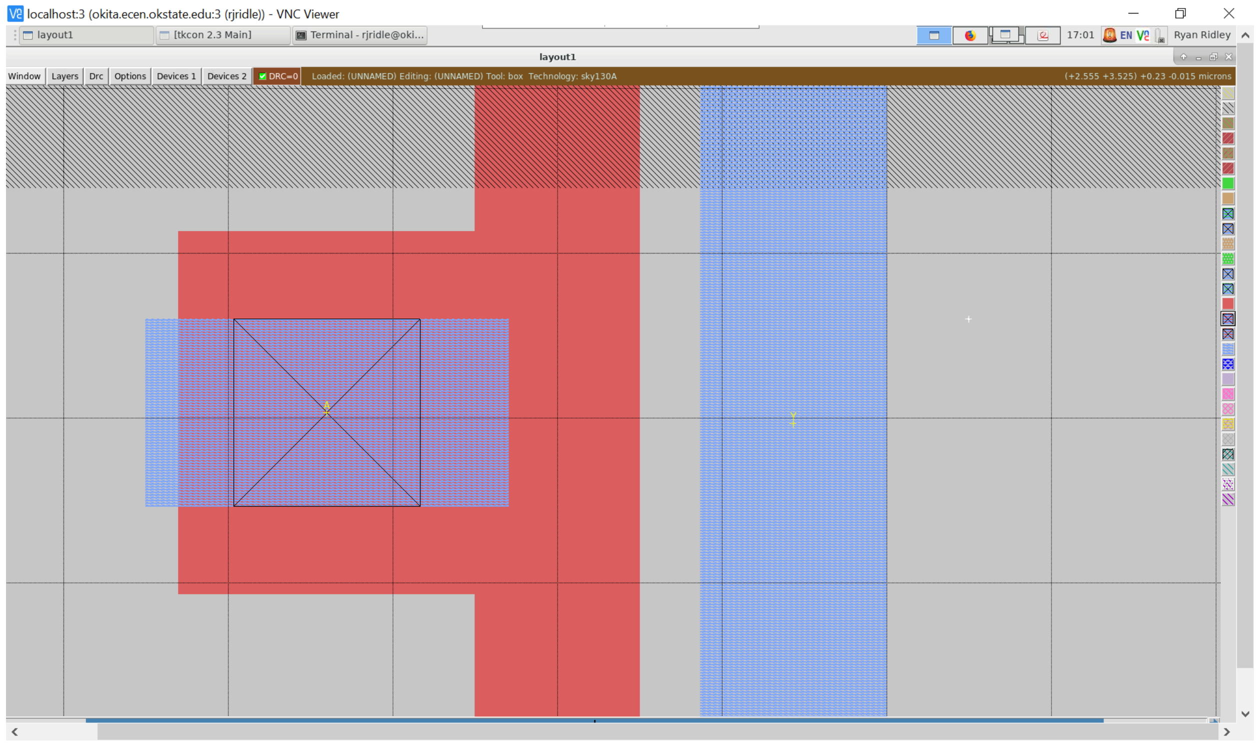
Here is the overall layout with substrate contacts on both rails.

Next, the inputs and outputs need to be labeled so that LVS can be run on the design. On the poly, place a label called “A”. Typically, we teach students to create a label by clicking the left-mouse button and then the right-mouse button in succession to create a + symbol. Then, the label can be created using the commands listed below. However, other students sometimes like creating rectangles for a label - the option is left to the user’s preference. Some labels are also challenging, because the label gets associated with a specific layer. Therefore, when creating a layer on an area that has multiple layers in its vicinity, it is a good idea to watch the magic command window to make sure the label gets placed on the proper layer. More information on the label command can be found here:

:label A

On the local interconnect that connects the transistors, place a label called “Y”.

:label Y


localhost:3 (okita.ecen.okstate.edu:3 (rjridle)) - VNC Viewer
layout1
[tkcon 2.3 Main]
Terminal - rjridle@oki...
17:01
Ryan Ridley
layout1
Window
Layers
Drc
Options
Devices 1
Devices 2
DRC=0
Loaded: (UNNAMED) Editing: (UNNAMED) Tool: box Technology: sky130A
(+2.555 +3.525) +0.23 -0.015 microns
Y

https://drive.google.com/file/d/1BIUIlfrLp80Rc3_B2BrO3Tko2C-1e0KP/view?usp=sharing

Saving your work and quitting

Make sure the names of the labels are the same as the names in your drawn schematic. Save the layout you created:

:save

To exit Magic, type:

:quit

`ext4mag` : The power of scripting

The next step in the design flow process will be to use **ext4mag** on the .mag file (this is an Oklahoma State University script we created to allow easier extraction for magic). This script extracts ".gds", ".ext", ".sim", ".sp", and ".spice" files. Scripting is very important to Electronic Design Automation Tools. It not only simplifies things, but teaches the importance of how scripts can save time for our students.

We also believe it stresses the importance of automating the process as well as how computers can be used to create chips as well as saving time which is what computers are meant to do.

The link for the ext4mag script is found: here

This ".sim" will typically have the same name as the one your schematic extracts. Make sure to keep magic and schematic files inside different folders so that you do not overwrite your files.

Final Design and Download of Inverter

The final design is now complete. Layout is fun but a time sink. It is always a good idea to have a plan (stick diagram) to attack the problem. Thoughtful techniques and tricks in creating the layout are best found by planning a design instead of an impromptu layout design. We have provided the inverter so you can download if you want to see any of these items inside magic. Theoretically, you could also delete any of the layers to recreate the images within this tutorial.

https://drive.google.com/file/d/1mFYrHuHVGZdcY62p8SamjvsIOWk3tRTJ/view?usp=sharing

Cell Hierarchies

The concept of cells is essential for efficient and structured layout of VLSI circuits. The basic idea is quite simple: if you created a layout for a building block (such as an inverter) that can be used in another design or is going to be used many times in the current design, then it is useful to simply include this layout of the building block as a cell. Every time you create and save a circuit in Magic, you have effectively created a cell. Each cell can be included in the hierarchy of a new circuit. The cells you created or the cells created by others in a library can then be put and connected together in a new circuit. Here are basic commands related to working with cells.

Start a new circuit. Then,

:getcell name – includes a cell that you want in the new circuit.

The included cells are shown as "black boxes" with only the instance names showing. No details of the included cell are shown. This is known as the unexpanded form.

You can select and manipulate each cell as you could a box of paint. In order to "connect" cells together, you need to know where the "connection points" of a cell are. Thus, expand the cell and paint wires between appropriate points of the cells. The following macros are used to expand or unexpand cells:

x (lower-case x) : expands a selected cell, all details are visible.

X (upper-case X) : un-expands a selected cell, all details are hidden.

If you include a number of pre-designed identical cells and you realize that you need to make one small change to all of them, you can edit just one cell by itself and the changes will be reflected in all the other cells. To edit the cell, type:

:load cell-name

Magic will load that cell (and all of its sub-cells) into the window. Once all changes have been made, :save your changes and :load the circuit you were working on. All changes will be reflected in your circuit's cells.

Magic Tutorials, Summary of Commands, and Complete Manual Pages

Magic comes with a set of tutorials that go through commands, macros, and mouse functions. However, many of these tutorials are quite old. This tutorial is probably a good start. There are also many good tutorials at http://www.opencircuitdesign.com/magic.