EE 431 – Lab 2 Report
Inverter Circuit and Layout Design

Author: Ryan Cramer

I. Introduction

In this lab, we build the most fundamental CMOS building block, the inverter. Made out of an nMOS and pMOS, where the pMOS is twice the width of the nMOS. We use xschem to do circuit design, ngspice to do simulation, magic to do layout, and netgen to do LVS.

II. Methodology

Design Flow:

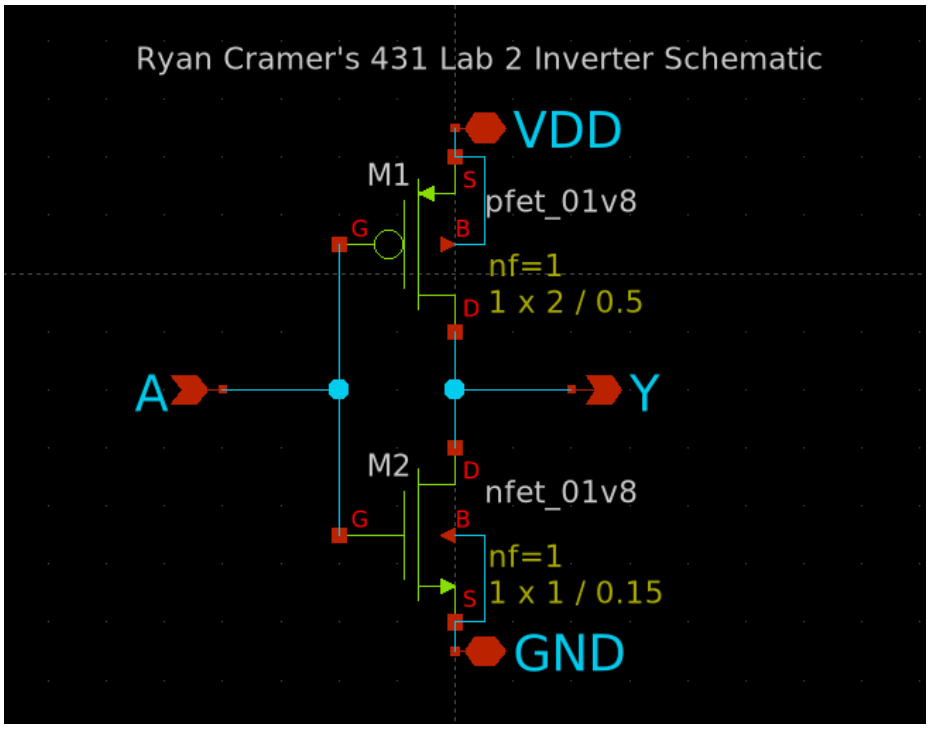
We first build the Inverter with xschem, using the sky130_fd_pr_1v8 library, which stands for the Skywater 130nm primitive 1.8V transistors. After constructing the circuit in a .sch file, we design a symbol for the inverter, using a .sym file. To test that our symbol works, we make a second .sch file which is a testbench. This testbench must reference the sky130 spice libraries, which defines the equations and constants for the simulator to use. To simulate, we use ngspice, with gaw as our waveform viewer, and we verify that the inverter works as intended. Afterwards, we design our inverter using the Magic VLSI layout tool, defining the technology as Sky130A (the open-source one). We use the "ext2spice" function to extract our layout to a spice file, and if we did everything correctly, we should see one sky130_fd_pr_nfet_1v8 and one sky_130_fd_pr_pfet_1v8 in the file. To verify that our schematic and our layout are uniquely matched, we use netgen as the LVS tool.

To build the layout, we need to actually design the geometry for the inverter. First, we need to ensure that our Polysilicon gate follows the length specified: 0.5um. Once we span the PolySilicon gate, we need to add n-diffusion and p-diffusion regions surrounding the gate to form our nMOS and pMOS, respectively. To ensure we meet the width requirements, we size the nMOS width to 1um, and the pMOS width to 2um. We then need to add local interconnect and diffusion contacts on top of our nMOS and pMOS. We also need to add substrate/well taps, so we add a n-diffusion region next to the pMOS, and a p-diffusion region next to the nMOS. Then we need to ensure we route VDD and GND through vias into a metal1 layer, which spans space to reach the taps. So, the polysilicon gate rests sandwiched in between the pMOS and nMOS layers, and that is sandwiched in between metal1 layers which have VDD routed to the pMOS source and GND routed to the nMOS source. We also need to ensure the gate has poly contact. Finally, we have to add the ports, A, Y, VDD, and GND, and ensure that they match our schematic.

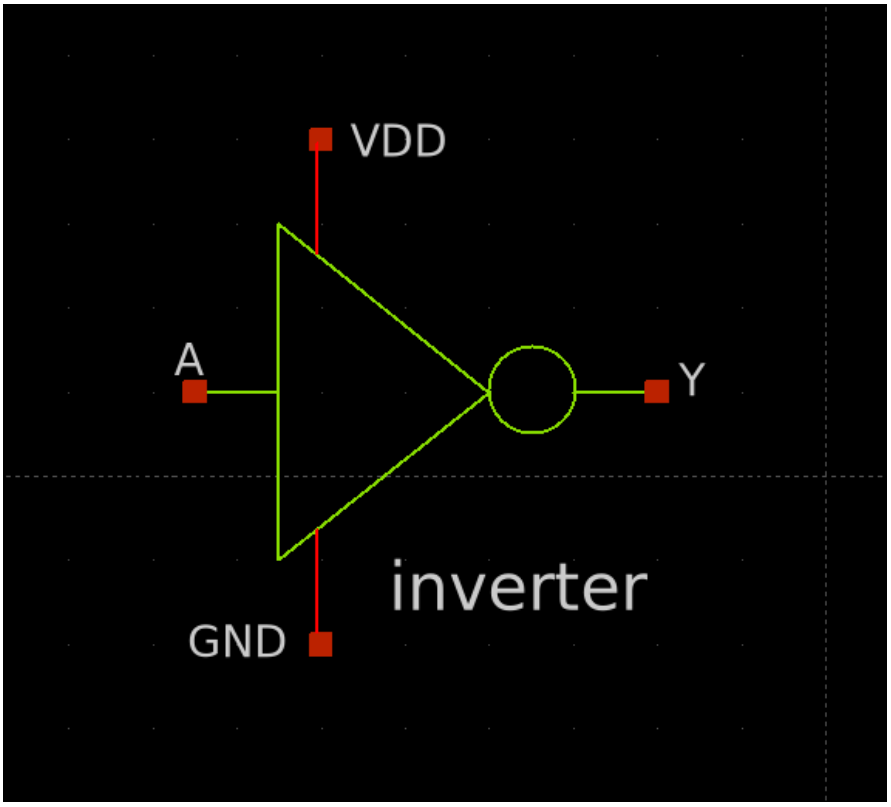
Inverter Schematic, Symbol, and Spice Simulation:

To build the schematic, I needed one pfet, and one nfet. I connected their gates and drains. I labeled these as A and Y, respectively, and also tied global VDD and GND to the pMOS and nMOS sources, respectively. Afterwards I constructed a standard inverter symbol and connected it to my schematic. I then generated a netlist, and ran simulation.

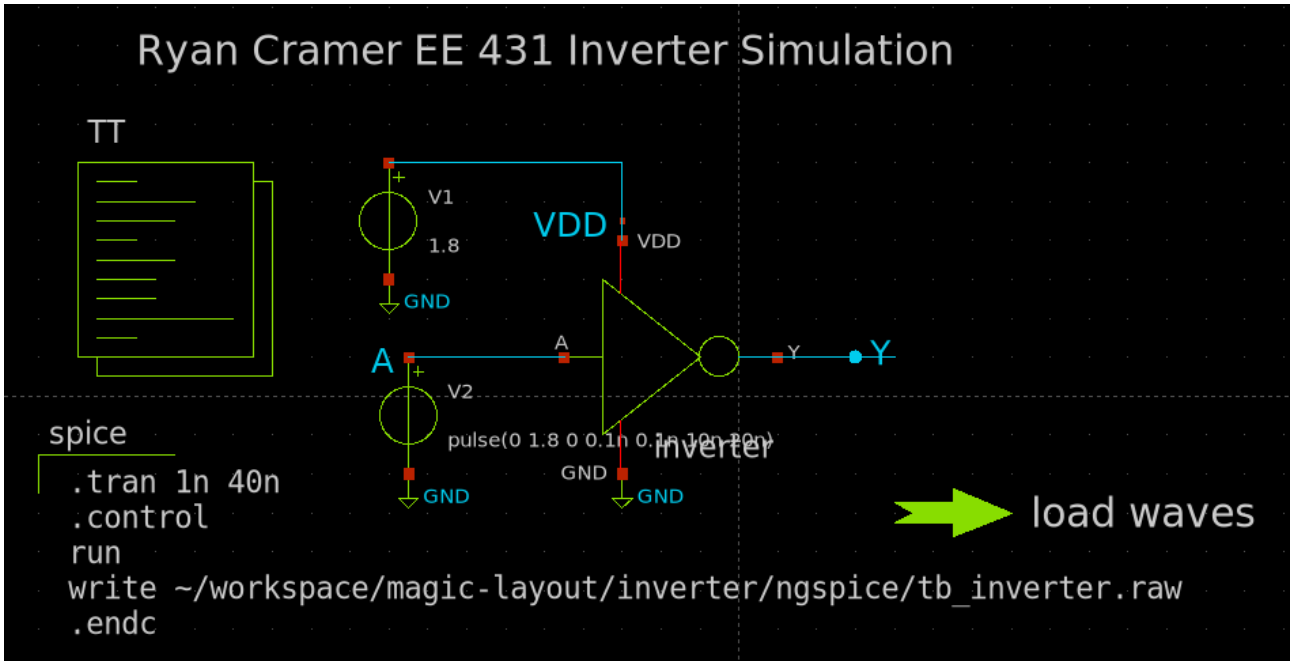
Inverter Xschem Schematic



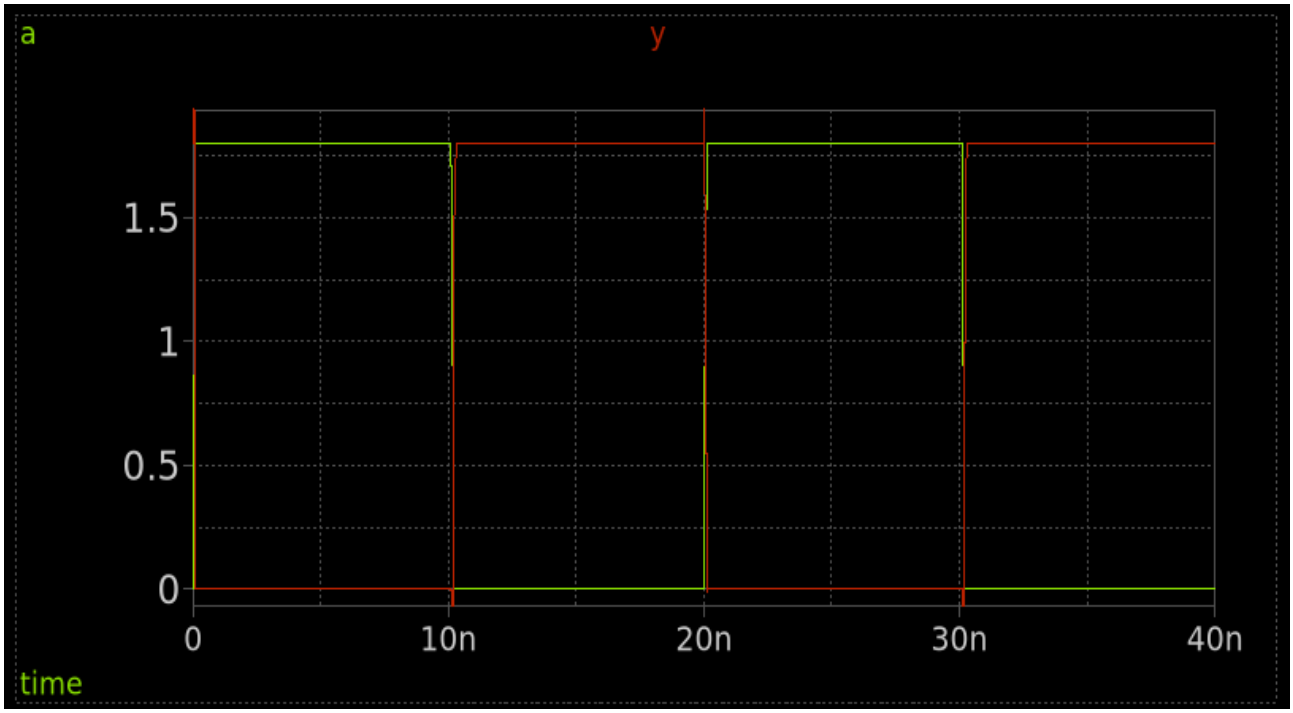
Inverter Xschem Symbol



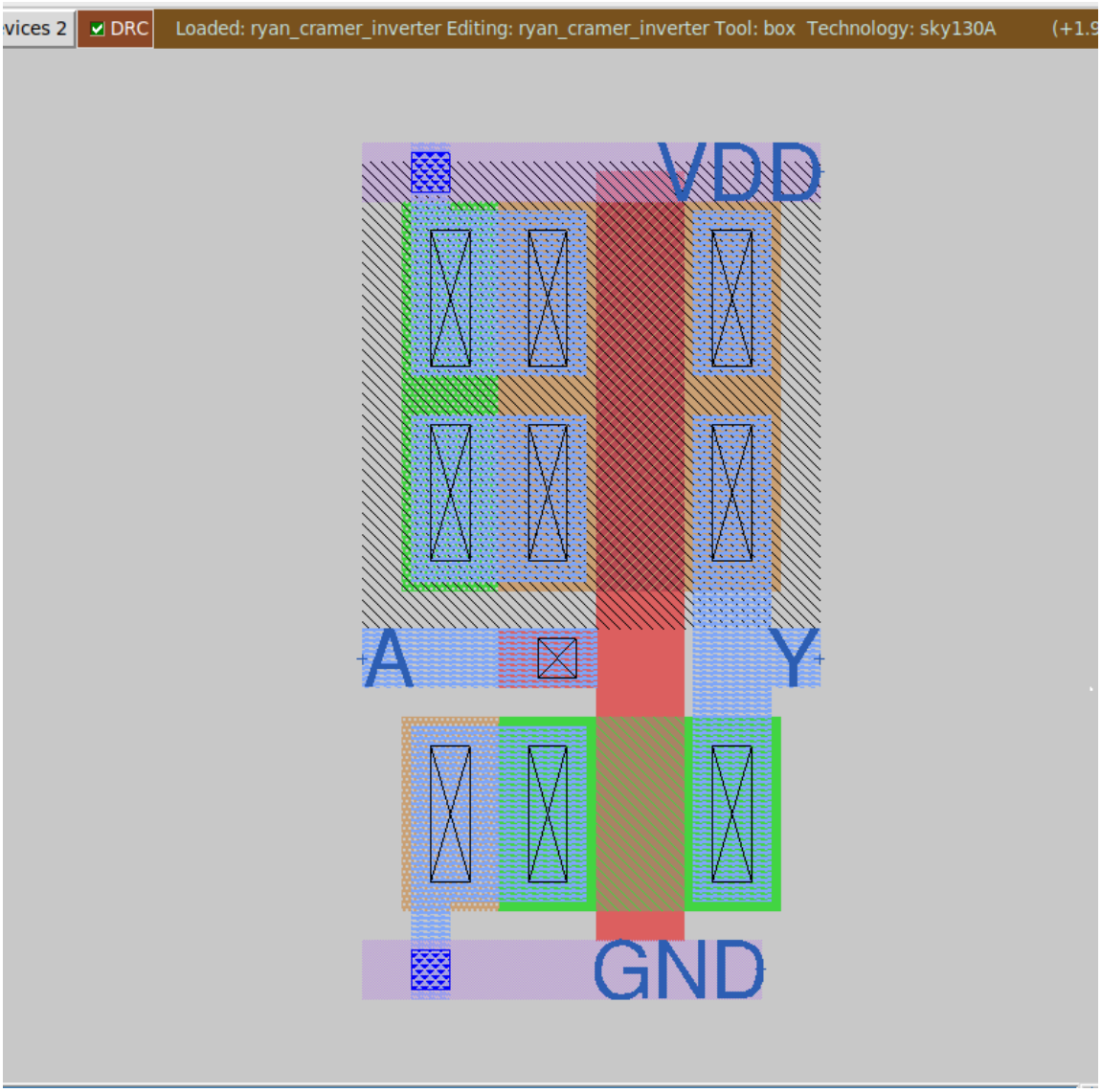
Inverter Xschem Testbench Schematic



Inverter Ngspice Simulation Results



Inverter Magic Layout (DRC Clean)



III. Result

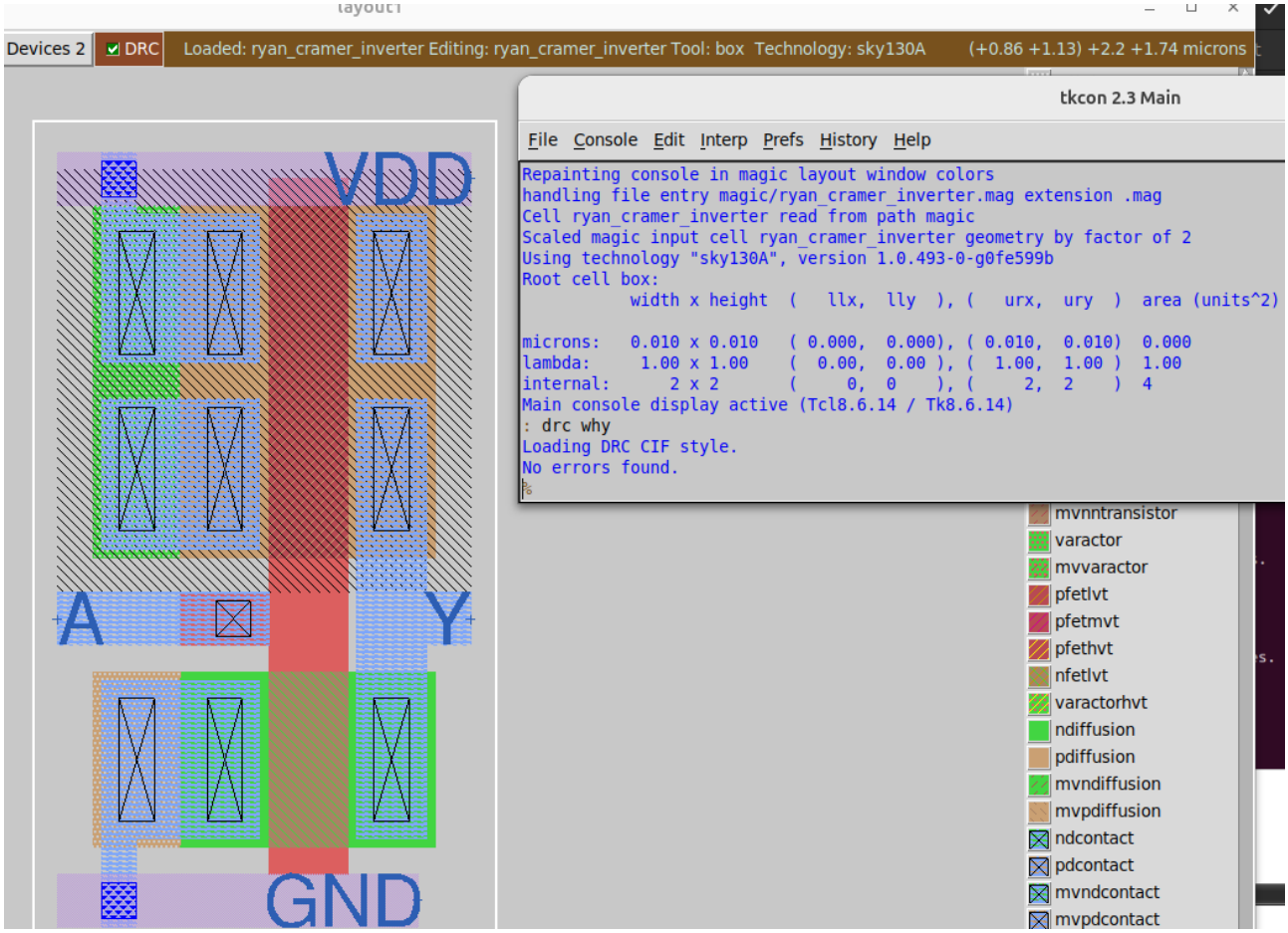
Netgen Results

```
Contents of circuit 1:  Circuit: '431_inv.spice'
Circuit 431_inv.spice contains 2 device instances.
  Class: sky130_fd_pr__nfet_01v8 instances:   1
  Class: sky130_fd_pr__pfet_01v8 instances:   1
Circuit contains 4 nets.
Contents of circuit 2:  Circuit: 'xschem_inverter.spice'
Circuit xschem_inverter.spice contains 2 device instances.
  Class: sky130_fd_pr__nfet_01v8 instances:   1
  Class: sky130_fd_pr__pfet_01v8 instances:   1
Circuit contains 4 nets.

Circuit 1 contains 2 devices, Circuit 2 contains 2 devices.
Circuit 1 contains 4 nets,    Circuit 2 contains 4 nets.


Final result:
Circuits match uniquely.
```

DRC Clean



IV. Discussion

From this lab, I learned the process of taking a circuit from schematic to verified physical layout, which helped me understand the relationship between design and fab. Creating the inverter in xschem showed the importance of proper pin labeling, transistor sizing, and understanding how the schematic symbol maps to the layout connections. Running ngspice simulations gave me a practical way to confirm the inverter's expected switching behavior before committing to layout, showing how simulation helps catch design issues early.

The layout stage in Magic was the most interesting part. It made me realize how physical geometry translates to actual transistor operation. I learned to recognize and properly place layers such as poly for gates, n-diffusion and p-diffusion for transistors, metal1 for interconnects, and n-well/p-well regions to isolate devices, as well as the various connection layers. Ensuring that design rules were met and verifying that the layout was DRC-clean helped me appreciate the precision required in VLSI design to ensure manufacturability.

One of the main challenges was aligning the layout connections correctly so that LVS matched the schematic. Small mistakes, like misaligned pins or missing contacts led to LVS mismatches, which required debugging. Going through that process gave me a better understanding of how schematic hierarchy and layout topology must agree electrically and physically.