Wyatt Tack
EE 431-01 F'24
2024-Nov-18

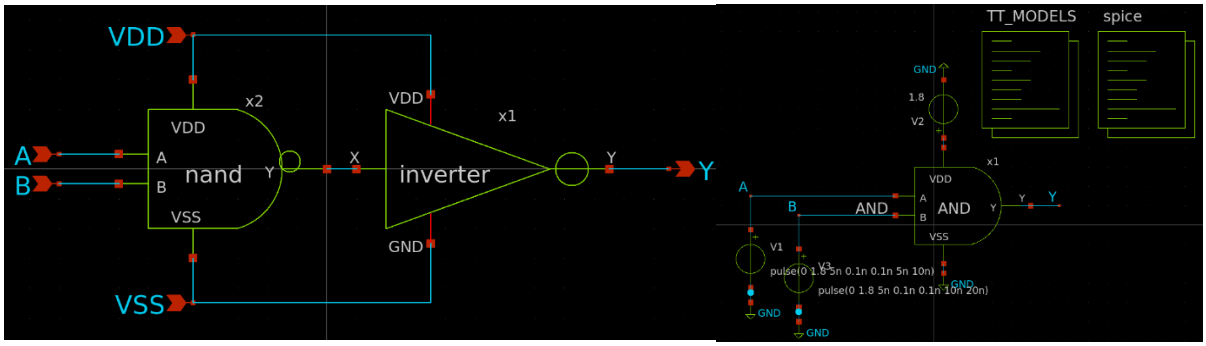
EE 431 Lab 4

Introduction:

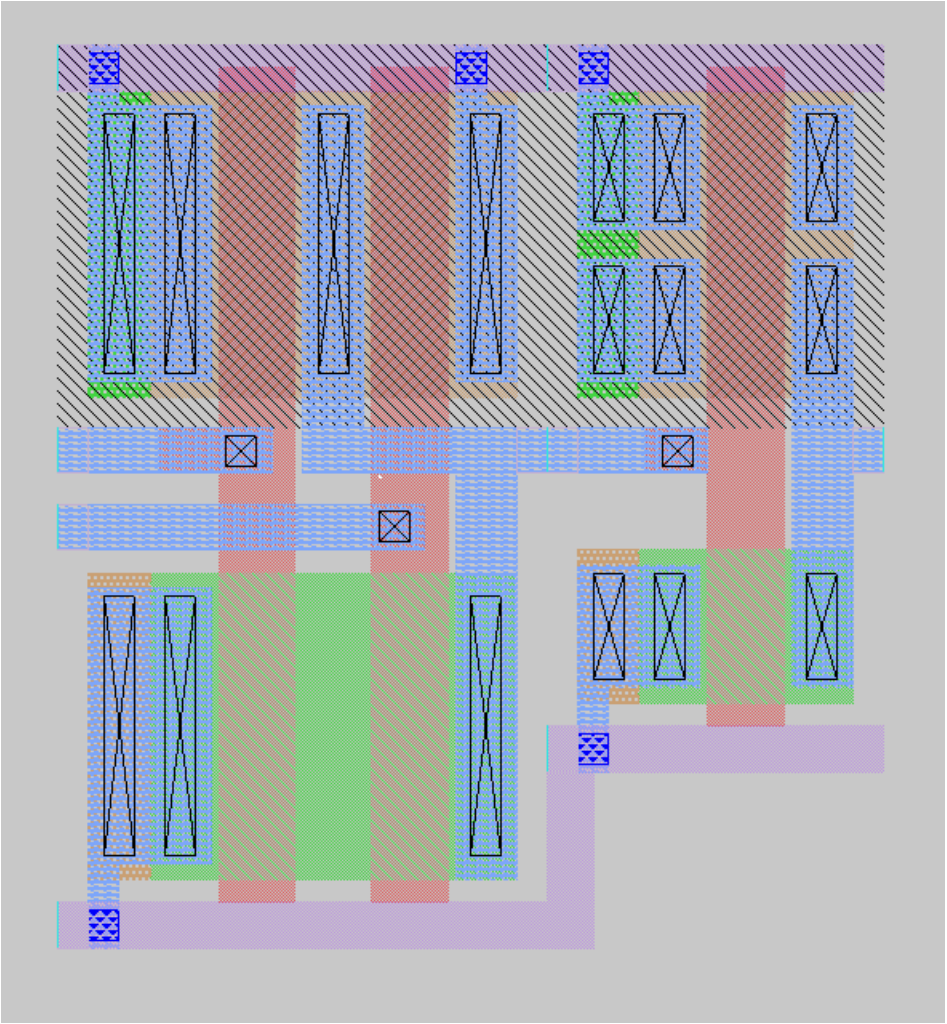
This lab was used to build off of our previous cells, being the NAND gate and the NOT gate. We used these two previous cells to create an AND gate, a much more commonly used logic statement in lower-level designs. From this we took our previous schematic symbols and placed them on a new schematic (creating a symbol to go along with it), and simulated the device. Upon verification, we used magic to import our previous 2 devices and connect them, creating a more complex layout, then exported to LVS to verify. The AND gate currently works as intended, with the open source PDK Skywater 130A used on Magic.

Methodology:

The first inverter was built as a simple CMOS AND gate with two parts, shown below. A standard 2 input NAND gate was followed by a unit inverter to create the logical operation AND. The sub devices were previously created with a 0.5um process and the Skywater 130A PDK.

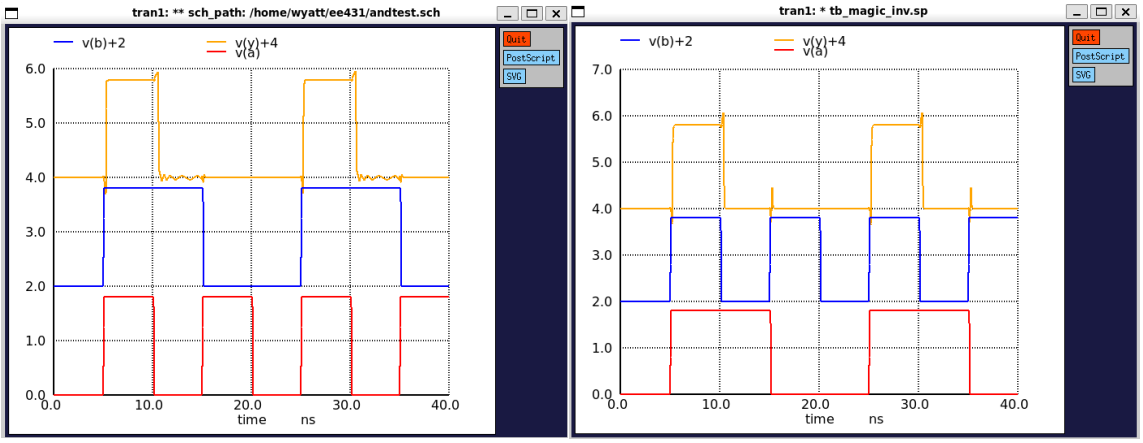


Now that our schematic does exactly what we wanted from it, we are now ready to start the layout process. Magic was opened and through using the Skywater PDK we created the logic gate. We imported the previously created standard NAND cell, and next to it placed the previously created standard NOT cell. These two cells were connected from output to input, VDD to VDD, and VSS to VSS:

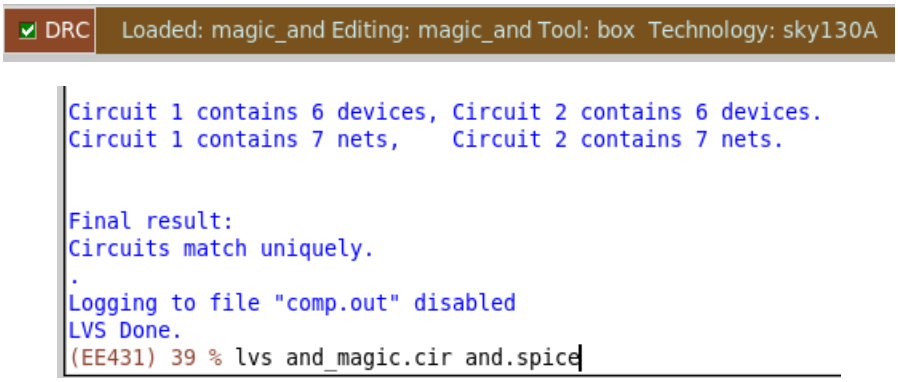


Results:

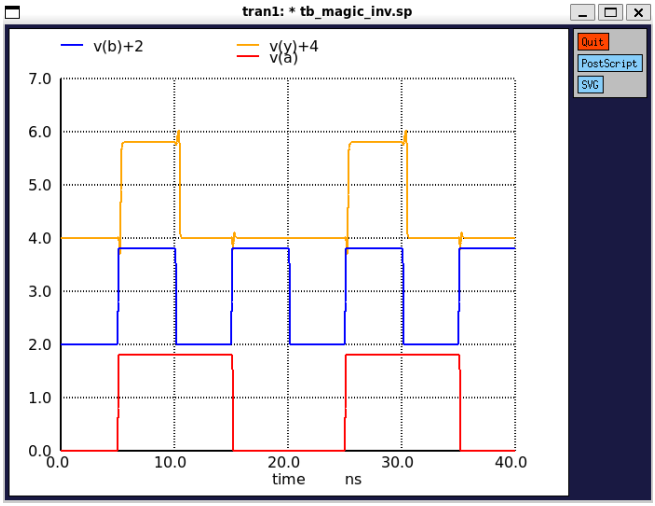
The AND gate was given nets to connect to in NGSpice, and thus simulated to recover a time domain solution. This graph is shown below. Additionally, the schematic gate was used with NGSpice to simulate according to the test bench set up above, and time domain solution graph was obtained. These graphs are presented below, schematic on the left and layout on the right.



Furthermore, LVS (layout versus schematic) was used through Netgen to compare both the exported netlist through magic of our layout, versus the exported netlist of our schematic from Xschem which we started with. Running LVS on out two netlists produced the following, along with the DRC no errors verification from Magic:



Which shows us that our layout is consistent with what we designed. Beyond this we exported the parasitic capacitances of our layout to a spice file, in which we were able to simulate this with our final AND gate design to get a more accurate plot of our transfer characteristics, with mild increase in transients during switching:



Discussion:

Both the schematic and the layout were simulated at the same voltage pulses, and when upon inspection of the time domain plots, the simulation between the schematic and the layout are indistinguishable. This lab showed us how to import standard cells to create higher level designs when imputing smaller layouts as “black boxes” into our design to connect them. This allows us to go forward and connect up these standard cells into other complicated circuits, such as latches or other.

Appendix:

Spice Files for Magic Layout:

```
* SPICE3 file created from magic_and.ext - technology: sky130A

.subckt magic_nand A B Y VDD VSS
X0 Y A VDD VDD sky130_fd_pr__pfet_01v8 ad=0.5 pd=2.5 as=0.9 ps=4.9 w=2 l=0.5
X1 Y B a_100_n230# VSS sky130_fd_pr__nfet_01v8 ad=0.9 pd=4.9 as=0.5 ps=2.5 w=2 l=0.5
X2 a_100_n230# A VSS VSS sky130_fd_pr__nfet_01v8 ad=0.5 pd=2.5 as=0.9 ps=4.9 w=2 l=0.5
X3 VDD B Y VDD sky130_fd_pr__pfet_01v8 ad=0.9 pd=4.9 as=0.5 ps=2.5 w=2 l=0.5
.ends

.subckt magic_inv X Y VDD VSS
X0 Y X VDD VDD sky130_fd_pr__pfet_01v8 ad=0.9 pd=4.9 as=0.9 ps=4.9 w=2 l=0.5
X1 Y X VSS VSS sky130_fd_pr__nfet_01v8 ad=0.45 pd=2.9 as=0.45 ps=2.9 w=1 l=0.5
.ends

.subckt magic_and A B Y VDD VSS
Xmagic_nand_0 A B magic_inv_0/X VDD VSS magic_nand
Xmagic_inv_0 magic_inv_0/X Y VDD VSS magic_inv
.ends
```

Spice Files for Xschem Schematic:

```
** sch_path: /home/wyatt/EE431/and.sch
**.subckt and VDD A Y B VSS
*.opin Y
*.ipin VSS
*.ipin VDD
*.ipin A
*.ipin B
x1 VDD net1 Y VSS inv
x2 VDD A net1 B VSS nand
**.ends

* expanding   symbol:  inv.sym # of pins=4
** sym_path: /home/wyatt/EE431/inv.sym
** sch_path: /home/wyatt/EE431/inv.sch
.subckt inv VDD X Y GND
*.ipin X
*.iopin VDD
*.iopin GND
*.opin Y
XM1 Y X GND GND sky130_fd_pr__nfet_01v8 L=0.5 W=1 nf=1 ad='int((nf+1)/2) * W/nf * 0.29'
as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 sd=0 mult=1 m=1
XM2 Y X VDD VDD sky130_fd_pr__pfet_01v8 L=0.5 W=2 nf=1 ad='int((nf+1)/2) * W/nf * 0.29'
as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 sd=0 mult=1 m=1
.ends

* expanding   symbol:  nand.sym # of pins=5
** sym_path: /home/wyatt/EE431/nand.sym
** sch_path: /home/wyatt/EE431/nand.sch
.subckt nand VDD A Y B VSS
*.ipin A
*.ipin B
*.ipin VSS
*.opin Y
*.ipin VDD
XM1 Y B net1 VSS sky130_fd_pr__nfet_01v8 L=0.5 W=2 nf=1 ad='int((nf+1)/2) * W/nf * 0.29'
as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 sd=0 mult=1 m=1
XM3 Y B VDD VDD sky130_fd_pr__pfet_01v8 L=0.5 W=2 nf=1 ad='int((nf+1)/2) * W/nf * 0.29'
as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 sd=0 mult=1 m=1
XM4 Y A VDD VDD sky130_fd_pr__pfet_01v8 L=0.5 W=2 nf=1 ad='int((nf+1)/2) * W/nf * 0.29'
as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 sd=0 mult=1 m=1
XM2 net1 A VSS VSS sky130_fd_pr__nfet_01v8 L=0.5 W=2 nf=1 ad='int((nf+1)/2) * W/nf * 0.29'
as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 sd=0 mult=1 m=1
.ends

.end
```