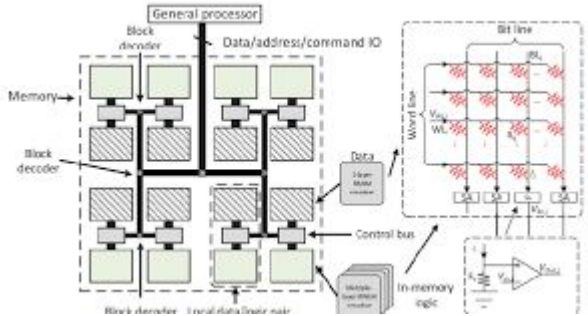
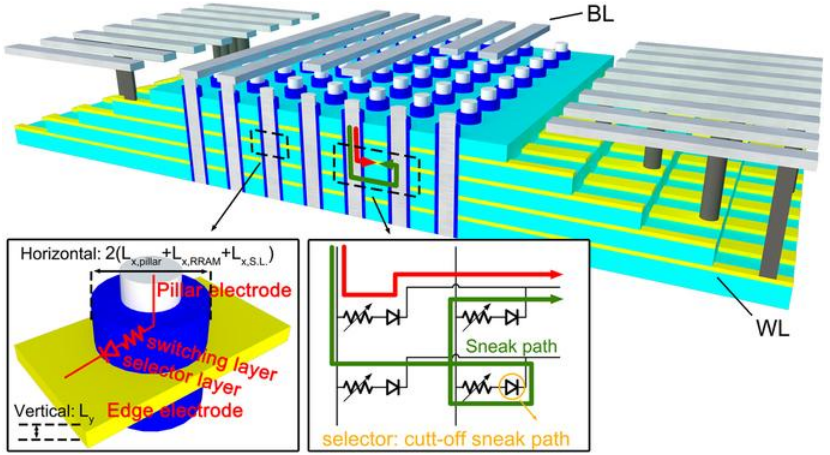
Prototyping

By: Noah Mack, Olivia Price, Travis Jakl, Sam Burns
Advisor: Professor Henry Duwe III
Client: Professor Cheng Wang
SDMay25-19

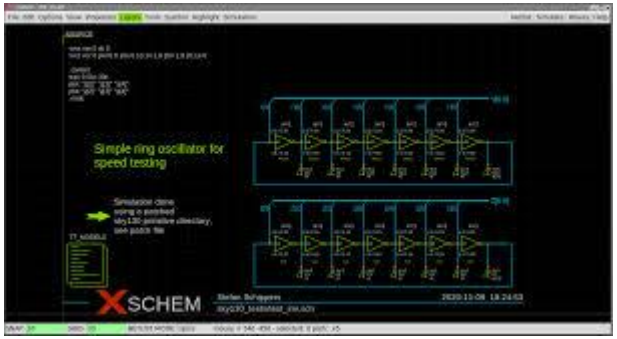
Project Overview

Our project focuses on creating a ReRAM-based compute-in-memory (CIM) test chip to improve matrix-vector multiplication efficiency in machine learning. Traditional CPUs struggle with the data bottleneck and energy cost of constant data movement, so our design incorporates multiple ReRAM architectures to enable parallel computation directly in memory. We'll fabricate the chip using the Skywater 130nm process, allowing ISU researchers and ChipForge club members to test and analyze different ReRAM designs. Alongside the chip, we'll provide documentation and C code for interfacing, helping research teams evaluate CIM's potential in low-power computing.



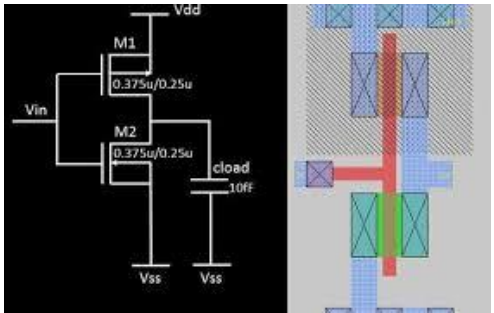


Software/Tools



- Xschem (schematic capture)
- Magic (Layout)
- Netgen (LVS)
- Slack (online community for help with open source toolset)

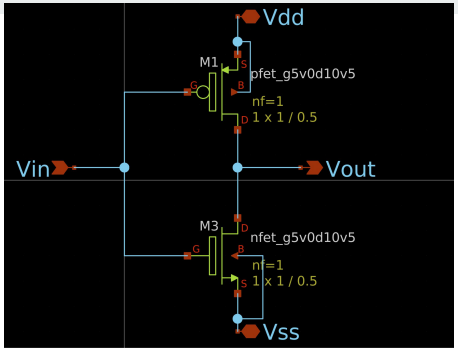
The tools shown above are open sourced meaning that there is little to no documentation on how to use. Precisely on how you use them together. For the first part of the semester, we have to go back through old tutorials and debug them to find out how to use the toolsets.



Inverter

Used to teach us how to use the tool sets, and to build for future applications in the design process.

Schematic Capture (Xschem)

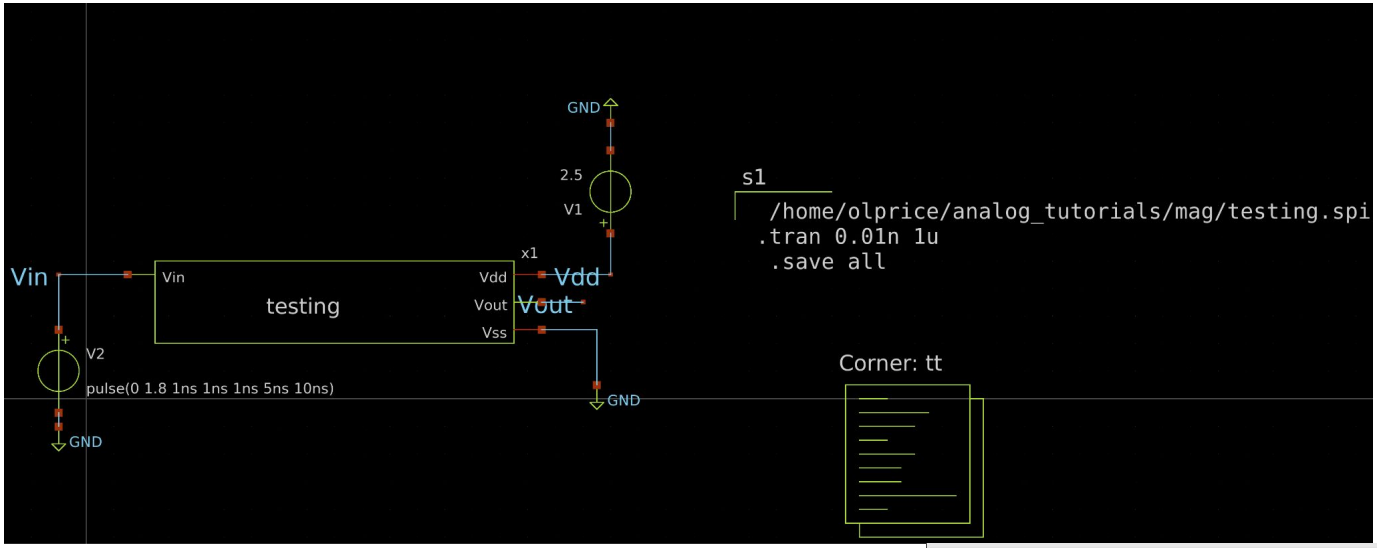


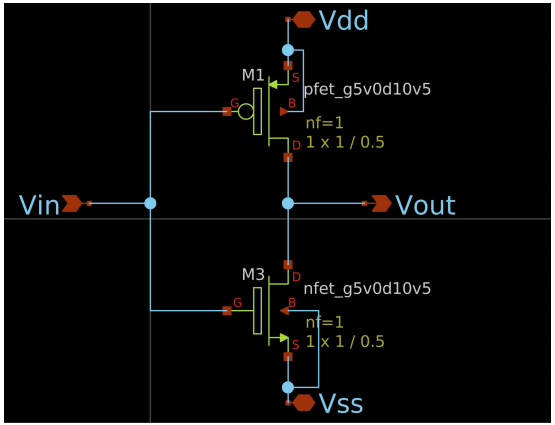
The first part of building a component that was needed for our architectures was to start our with a schematic.

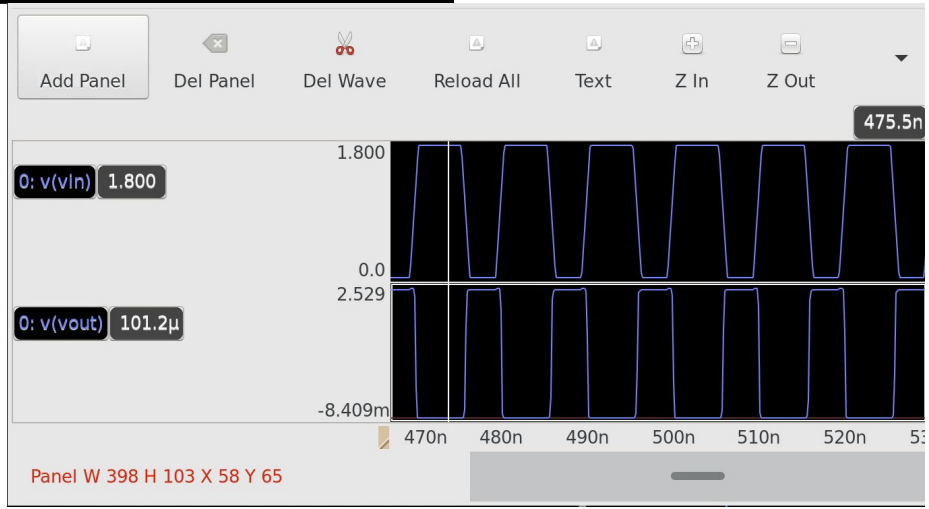
The schematic included:

- pMOS
- nMOS
- 4 pins (Vin, Vout, Vdd, Vss)

Then it was time to see if our schematic worked by creating a testbench for it. The testbench and results are on the next slide


GND
2.5
V1
x1
Vin
Vin
testing
Vdd
Vout
Vss
Vdd
Vout
V2
pulse(0 1.8 1ns 1ns 1ns 5ns 10ns)
GND
GND
s1
/home/olprice/analog_tutorials/mag/testing.spi
.tran 0.01n 1u
.save all
Corner: tt


Vdd
M1
pfet_g5v0d10v5
nf=1
1 x 1 / 0.5
Vin
Vout
M3
nfet_g5v0d10v5
nf=1
1 x 1 / 0.5
Vss


Add Panel
Del Panel
Del Wave
Reload All
Text
Z In
Z Out
475.5n
0: v(vin)
1.800
1.800
0.0
0: v(vout)
101.2µ
2.529
-8.409m
470n
480n
490n
500n
510n
520n
Panel W 398 H 103 X 58 Y 65

Layout (Magic)

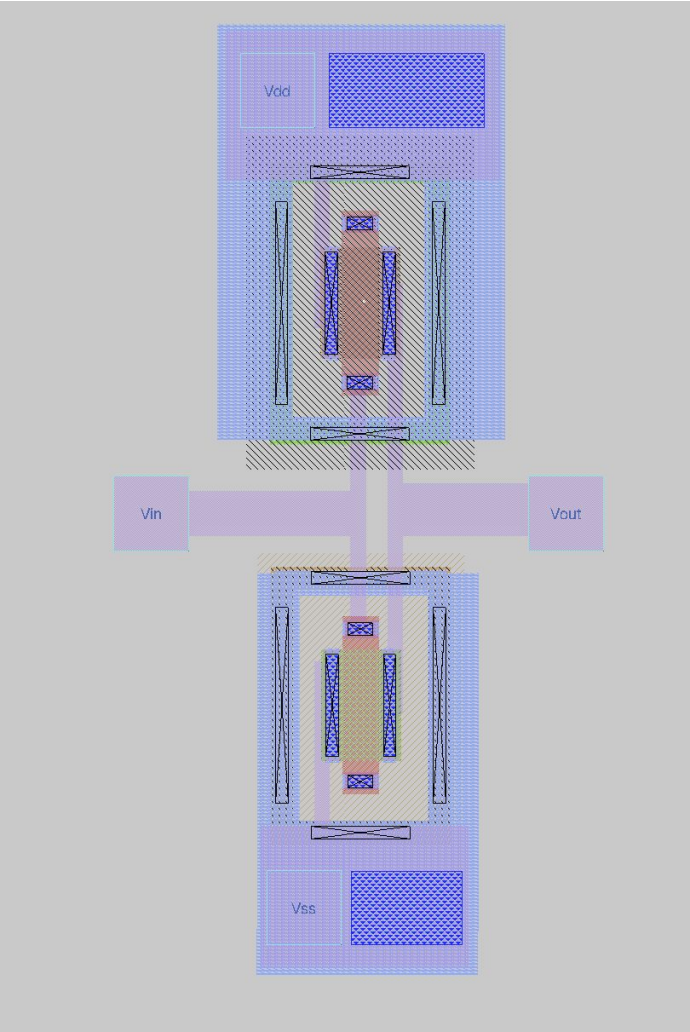
Once the testing and the verification of the schematic was made, we then need to make a layout. The layout shows the different types of metal that goes in the chip.

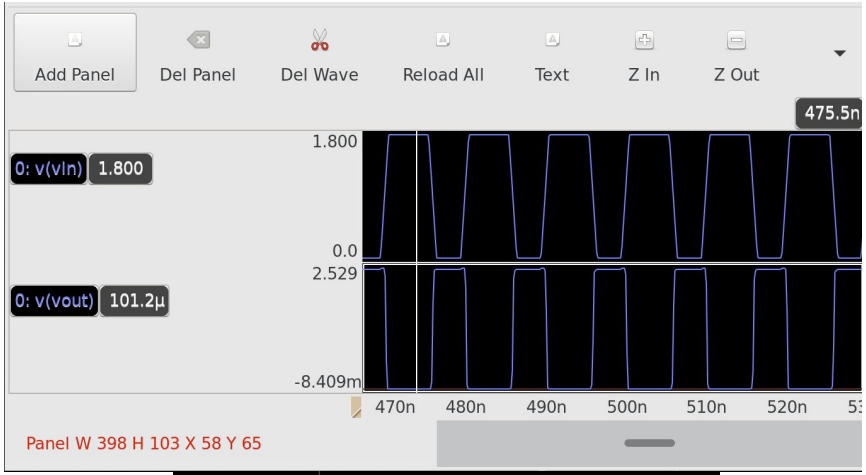
The layout includes:

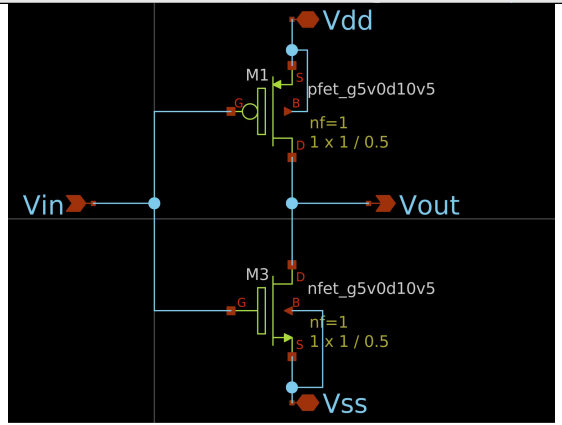
- pMOS
- nMOS
- 4 pins (Vin, Vout, Vdd, Vss)

The layout was also tested in order to make sure that the functionality was kept throughout the process. While also performing layout, we needed to make sure that there are no DRC (Design Rule Check) or LVS (Layout Vs. Schematic) errors.

The next slide shows, the layout and the testbench.








```
Scaled magic input cell sky130_fd_pr__nfet_g5v0d10v5_N5FAYL geometry by factor of 2
Cell sky130_fd_pr__pfet_g5v0d10v5_7EBZY6 read from path /home/olprice/analog_tutorials/mag
Loading DRC CIF style.
No errors found.
```

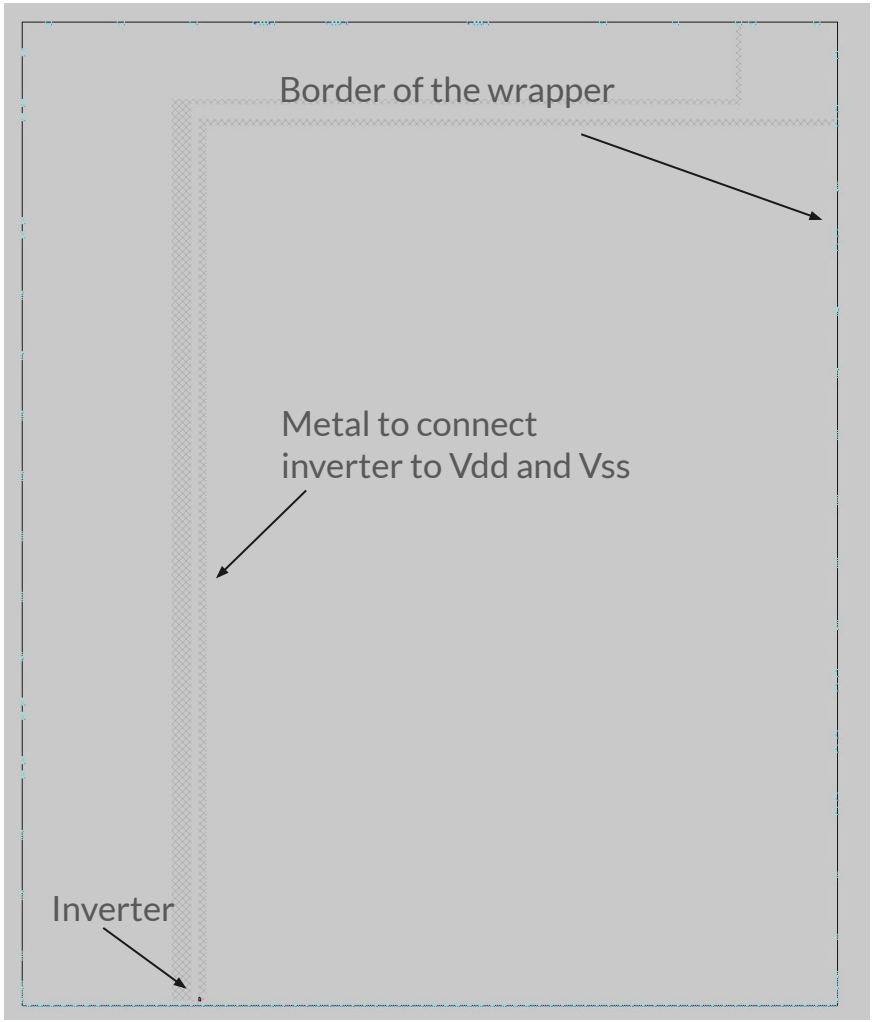
Placing the Design into the Wrapper

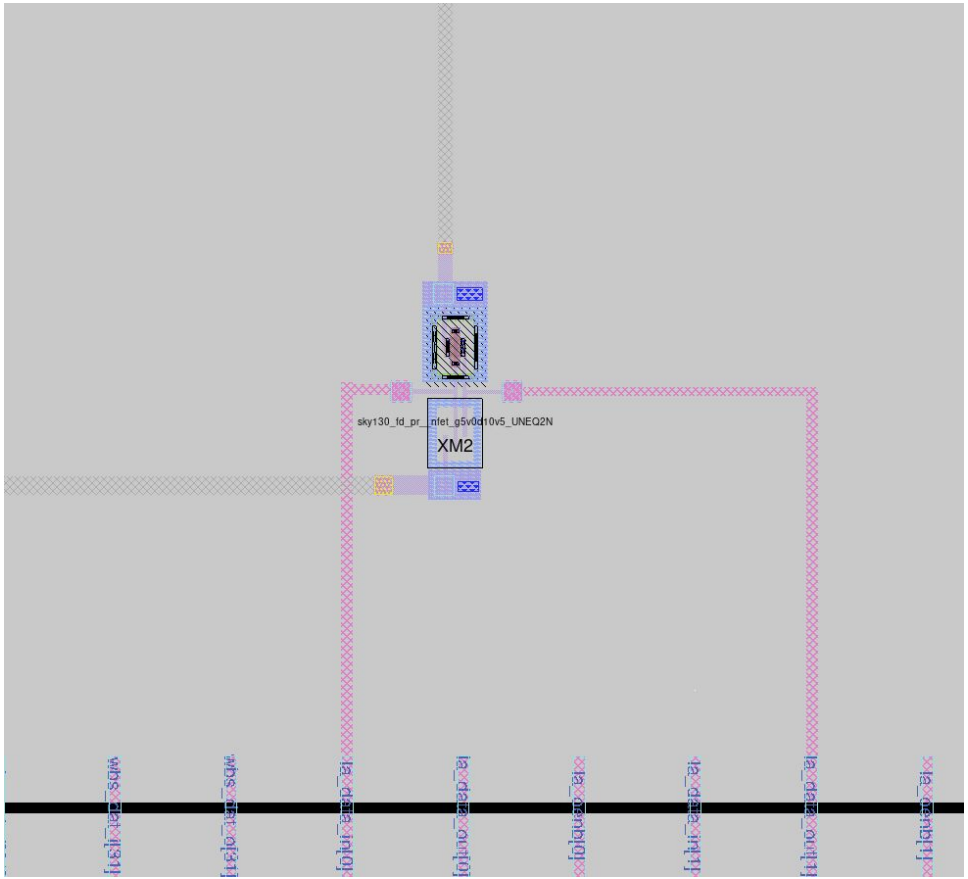
The wapper layout is where we will place our design which will be sent to efabless on Apri 21st.

The wrapper layout in the end will include:

- 4 ReRAM cells (8x8)
- 4 different architectures

The images on the next slide will show the concepts of placing a component on the wrapper, and the size dimensions of the wrapper.


Border of the wrapper
Metal to connect
inverter to Vdd and Vss
Inverter


sky130_fd_pr__nfet_g5v0d10v5_UNEQ2N
XM2

Reflect

- What we learned?
  - From the prototype, we learned that is takes a while to design a component with these tools. It takes us almost twice as long compared to other chip design softwares like cadence.
- What worked?
  - The design of the inverter worked and the behavior was verified by a testing software.
- What didn't work?
  - For the most part, we kept running into LVS troubles, but slowly figure them out.
  - We could not get design check or the LVS to work on some of our teams inverters, so we are still trying to troubleshoot that dilemma.

Implications and Next Steps

Next steps include:

- Fixing the tutorial documentation for ChipForge
- Create a troubleshooting document
- Verifying past teams components /design and making sure they work properly
- Creating our own components for our two new architecture designs

Conclusion

Journey to creating a component:

- Schematic
- Verification
- Layout
- Verification
- DRC and LVS checks
- Place in wapper (Connect pins to corresponding planes)
- DRC and LVS checks

With every component in our design, we will need to follow these steps in order to successfully create them.