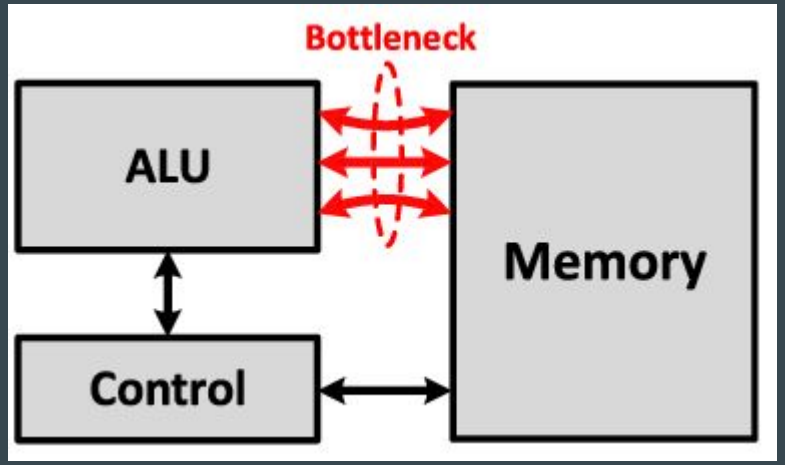
ReRAM Compute ASIC Fabrication

Team: sddec23-08
Team Members: Joshua Thater, Matthew Ottersen, Aiden Petersen, Regassa Dukele
Client: Dr. Henry Duwe
Advisor: Dr. Cheng Wang

Digital RAM

Most memory storages are in the digital domain

- Data is stored in the memory
- Data is transferred over to the ALU
- Computations are done
- Results are sent back to memory



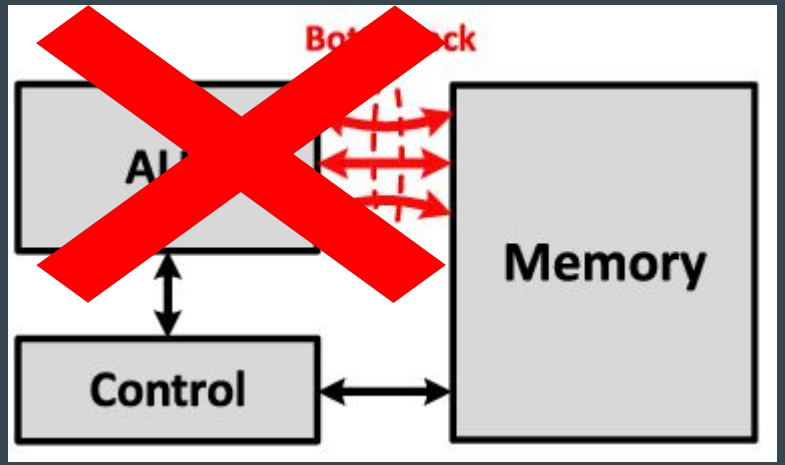
Source: [11]

Is there any way to push pass this bottleneck?

New Technology - ReRAM

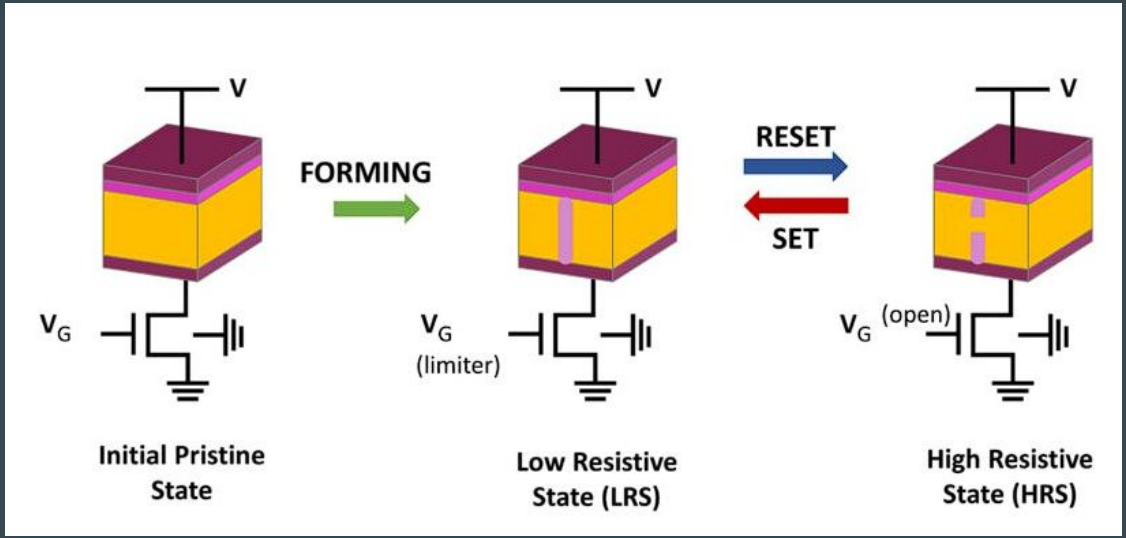
ReRAM (resistive RAM) is a type of non-volatile memory that does its computations in the analog domain instead of the digital domain

- Lower latency
- More power efficient
- More area efficient



Source: [11]

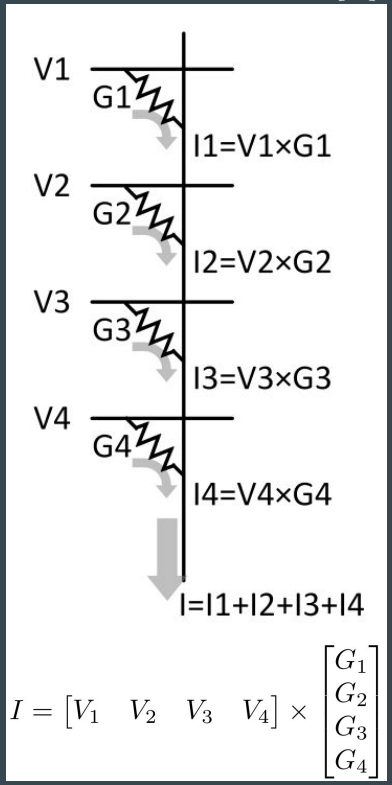
ReRAM



How ReRAM works

Source: [1]

Source: [12]



How ReRAM does computations

Problem Statement

Problem:

- Our client/advisor is interested in ReRAM and its computational potential
- Design a ReRAM compute crossbar ASIC using open-source tools

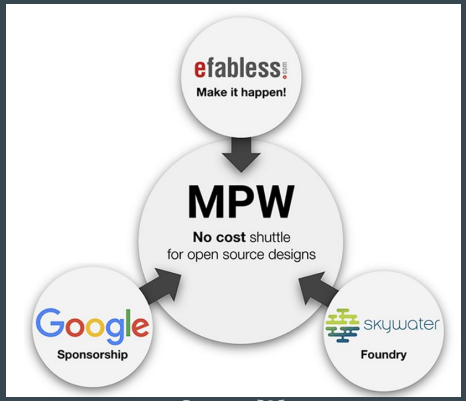
Goals:

- Use Efabless to submit a ReRAM chip proposal using the Skywater 130 nm process
    - Silicon prove design
- Document the open-source analog/mixed-signal design flow
    - Previous senior design teams have done digital designs
    - Our design is almost entirely analog

Source: [3]

ASIC Fabrication Through Efabless

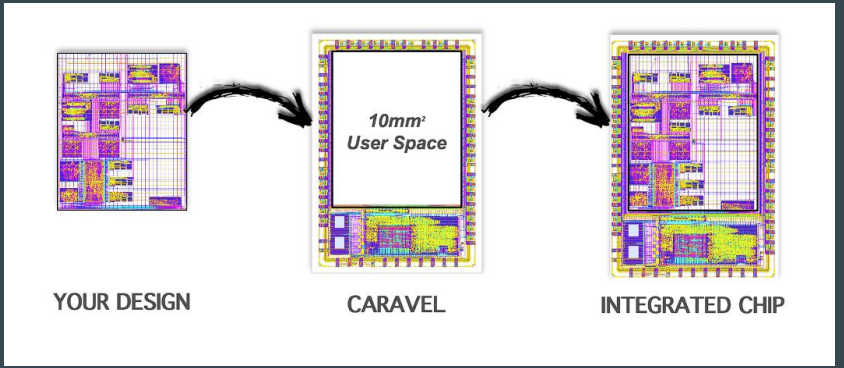
- Efabless hosts shuttles roughly every 3 months
    - Sponsored by Google
        - It's free
    - Using Skywater 130 nm process
    - Completely open-source



Source: [2]

Process:

- Design our circuit using open-source tools
- Integrate our circuit with Caravel harness
- Pass prechecks
- Submit design



Source: [2]

Requirements

Functional requirements:

- Design a ReRAM crossbar that can perform multiply and accumulate operations
- Design and verify functionality of analog circuitry
- Ensure design is able to interact with the Caravel harness
- Submission must pass Efabless precheck

$$I = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 \end{bmatrix} \times \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}$$

Non-functional requirements:

- Create detailed bring-up plan
- Create documentation on all parts of design flow
  - Setting up the environment
  - Tutorials on tool usage
  - Integration with Sky130 nm process

Users

1 - Dr. Duwe

- Create co-curricular for chip fabrication
- Analog design flow outlined

2 - Dr. Wang

- See how ReRAM is fabricated
- Help with his research

3 - Future students

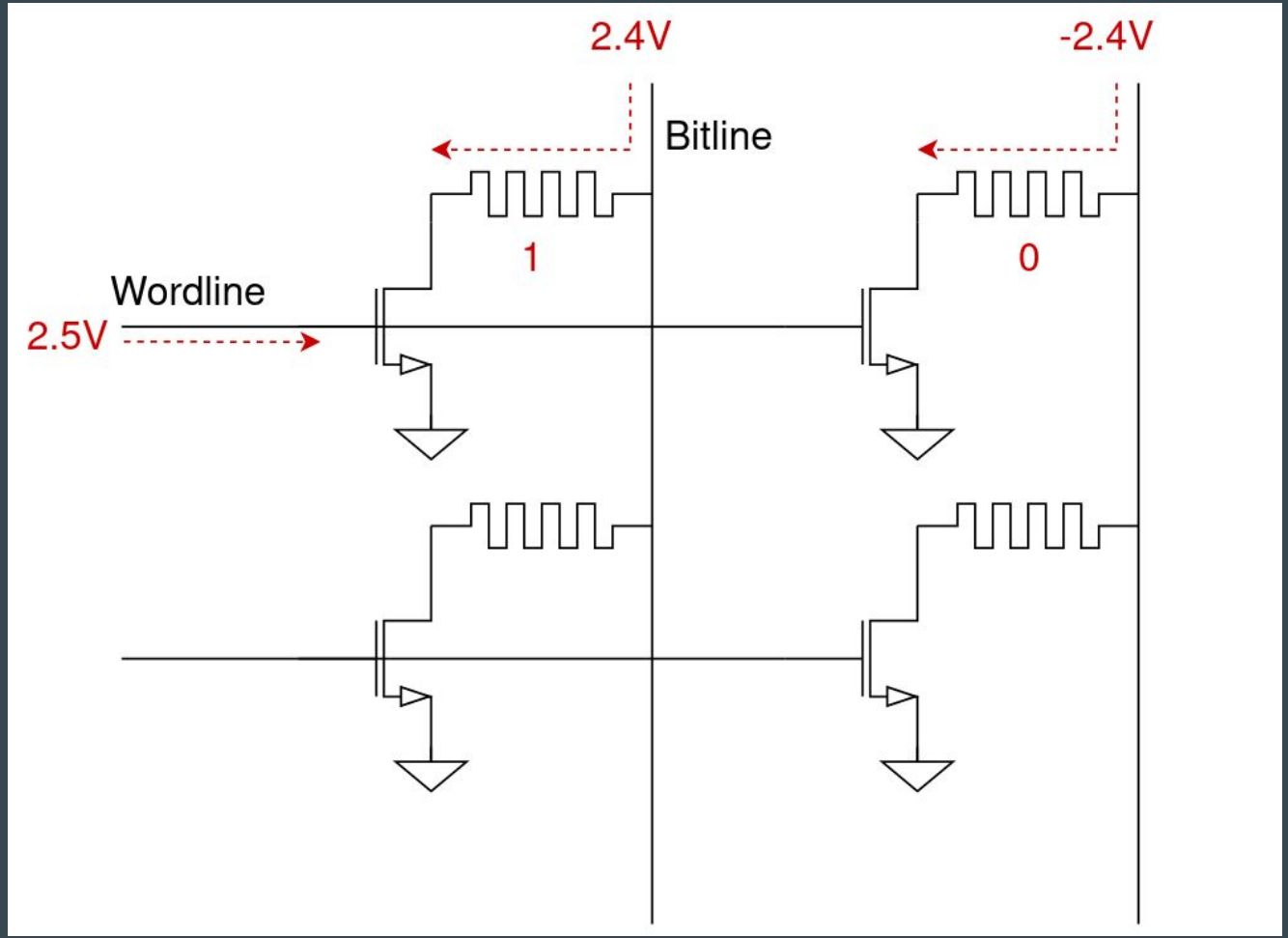
- Interested in chip fabrication
- Want examples and tutorials

Design Operations/User interface

- Write
- Multiply and Accumulate

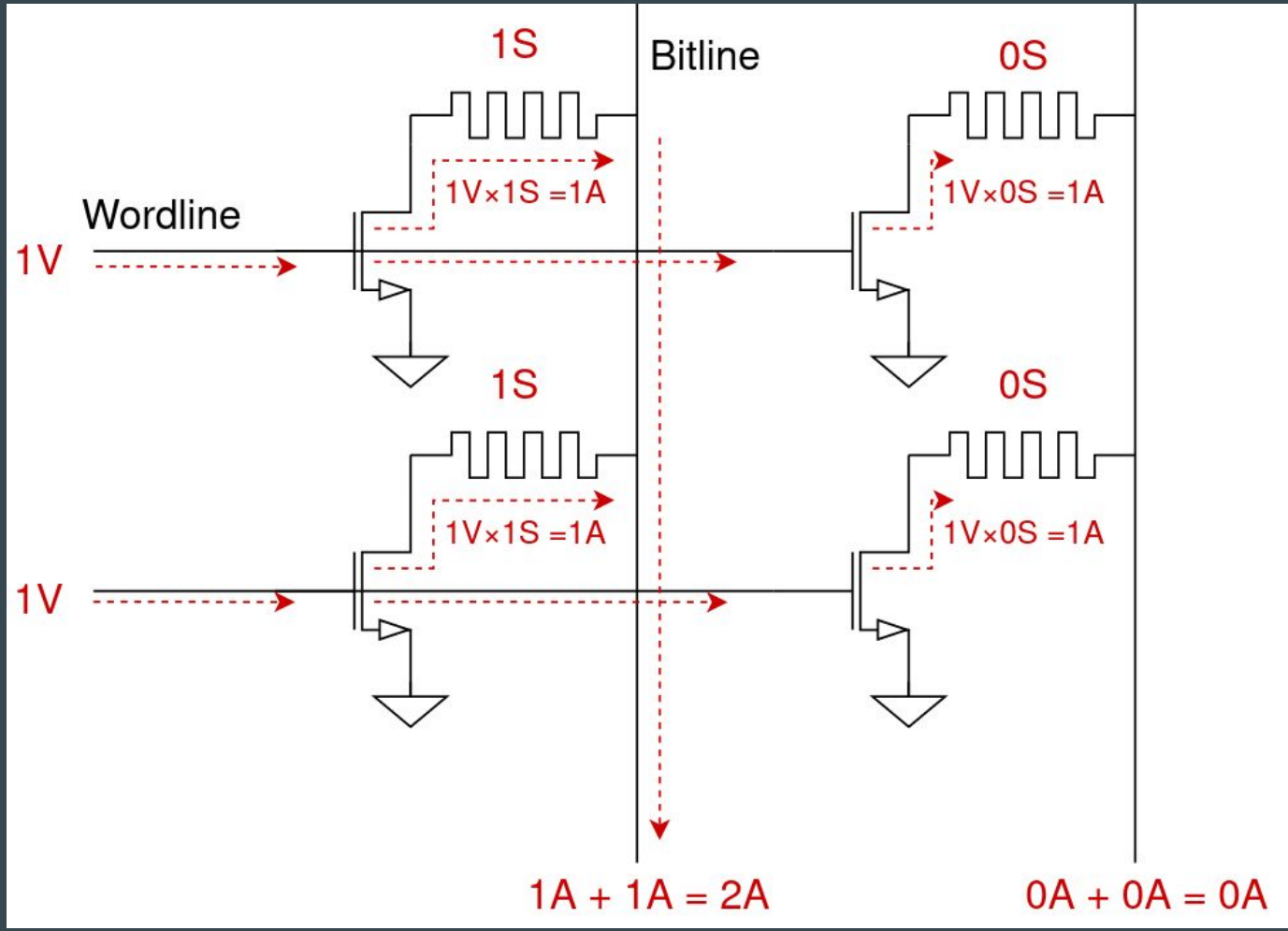
Write

- Bitline
  - Written values
- Wordline
  - Row written to



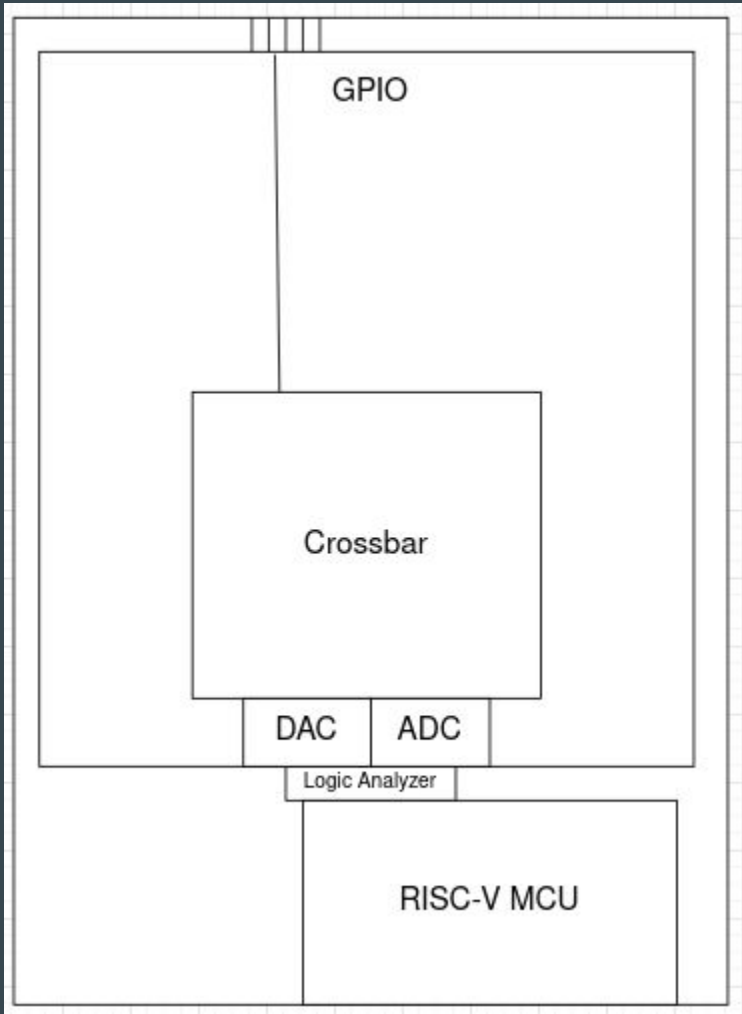
MAC

- Currents add
- Resistor multiplies
- I = VG
- G = 1/R

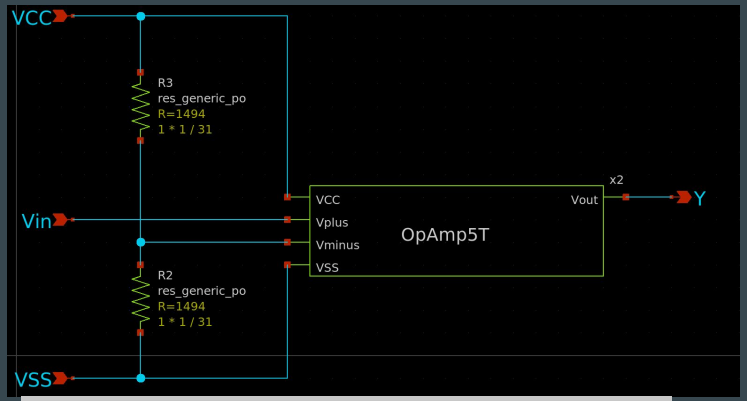


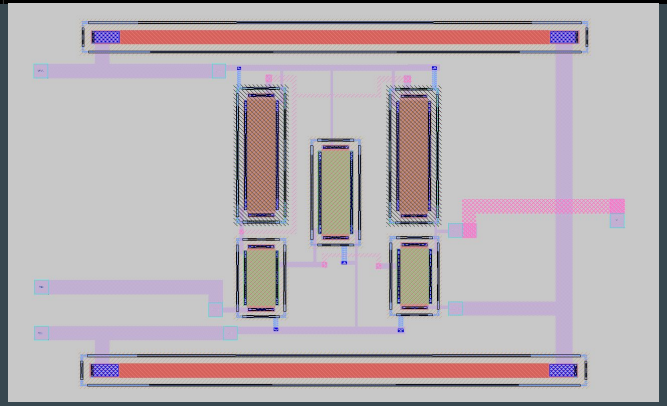
Abstract Top Level Design and I/O

- 128-bit Logic Analyzer
  - Configurable for In/Out
  - Retrieves computed values
- GPIO
  - Reads from crossbar for analog debugging.

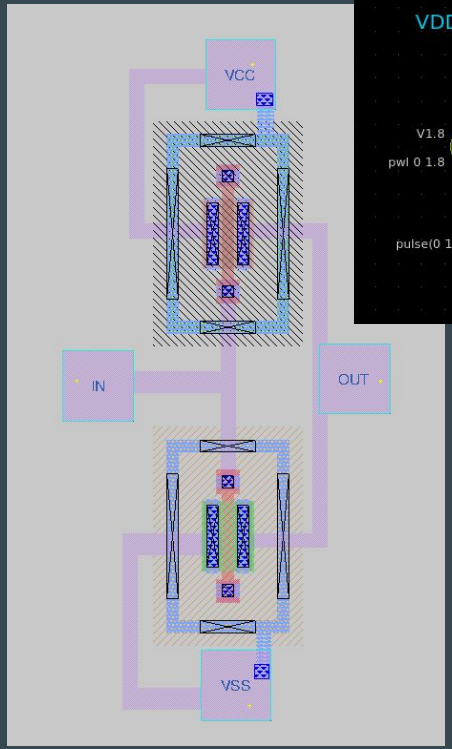


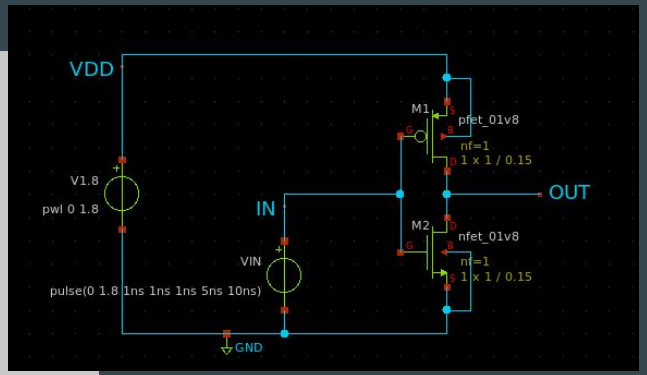
DAC and ADC Designs





ADC schematic and layout





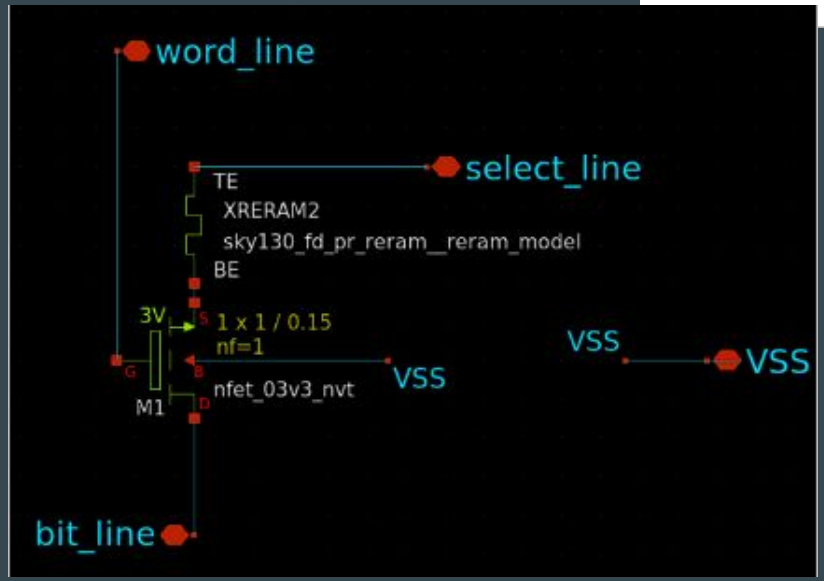
DAC layout and schematic

ReRAM 1T1R Cell

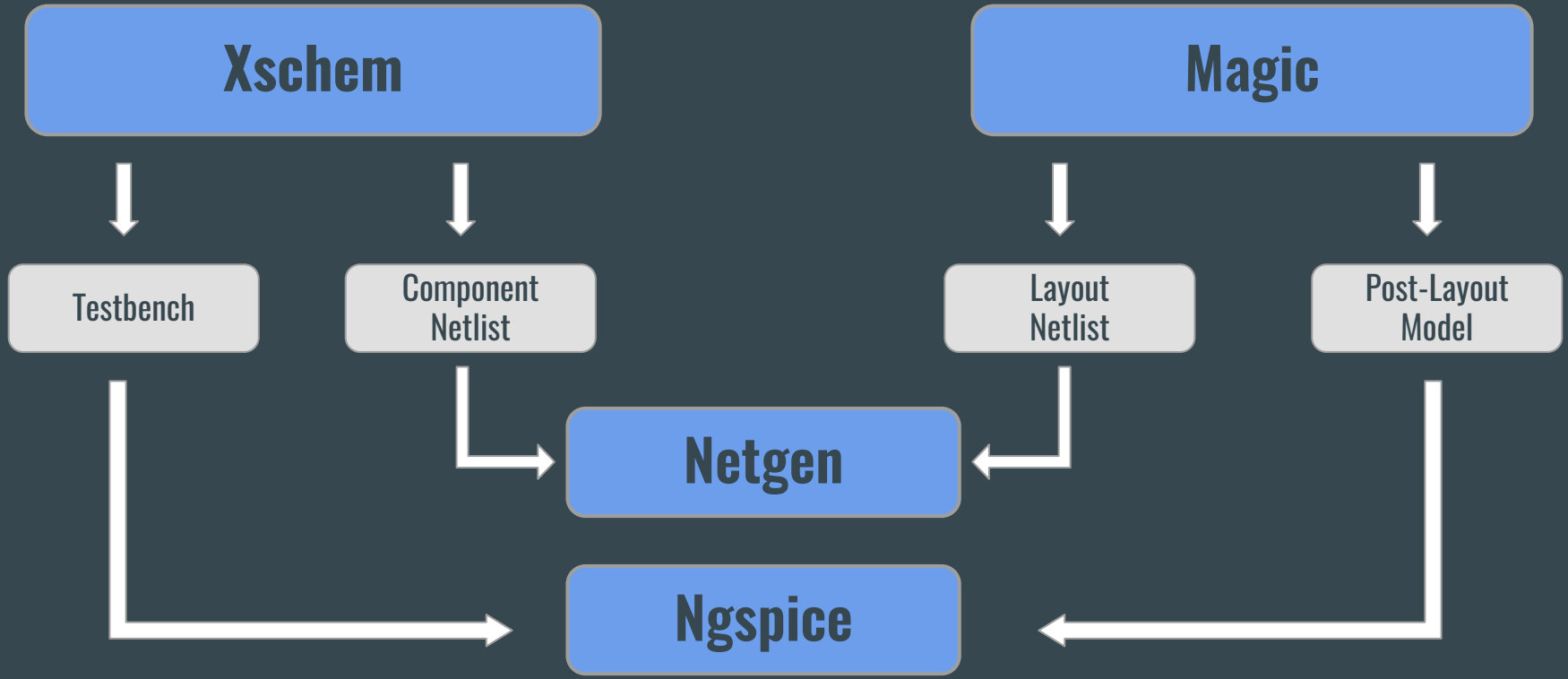
Table 1 1T1R #5

1T1R #5	WL (V)	BL (V)	SL (V)	PW (ns)	Yield (%)
Pristine					99.90
Form	1.4 - 2.0 (0.1 step)	2.6 - 3.1 (0.1 step)	0	1000	92.73
Reset	2.5	0	2.6	1000	90.83
Set	1.7	2.4	0	1000	97.45

Source: [5]



Design Process

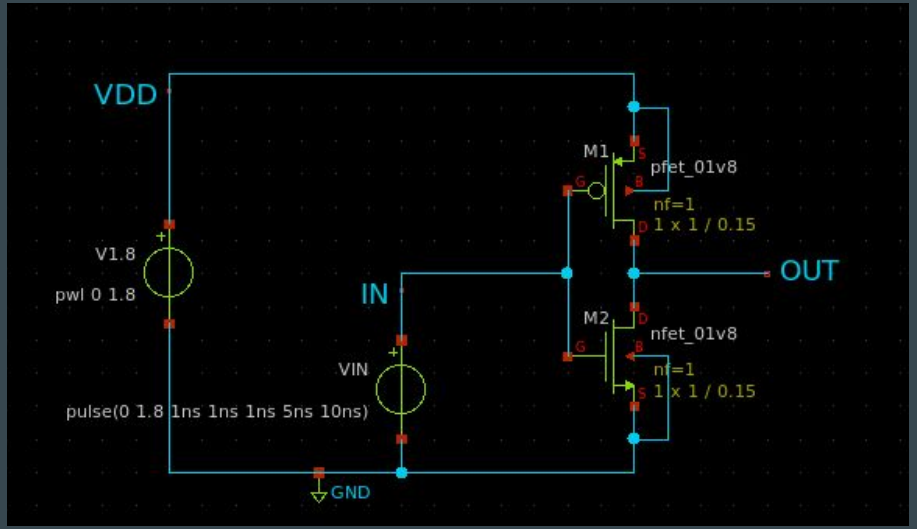


Design Tools

Xschem - schematic editor

- Create component models
- Create testbenches
- Exports Spice netlists

Source: [10]



Design Tools

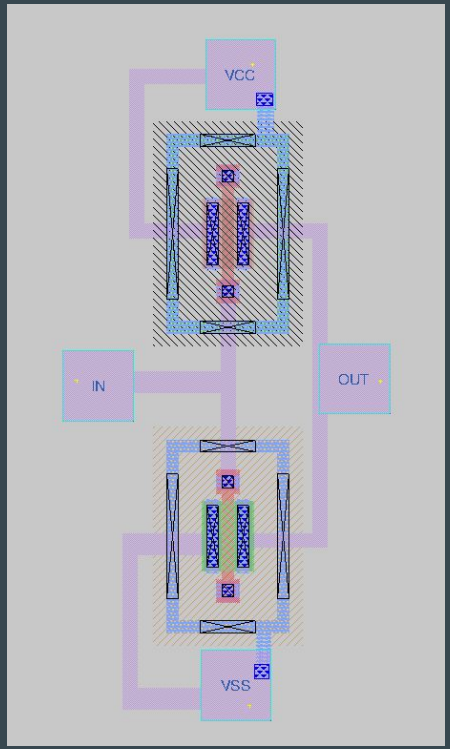
Magic - layout editor

- Creates component layout
- Runs design rule check (DRC)
- Exports Spice netlists
- Exports post-layout model
  - Includes parasitic capacitances

Netgen - LVS

- Runs layout vs schematic (LVS) check
  - Compares component netlist to layout netlist

Source: [8]



Simulation Tools

Ngspice

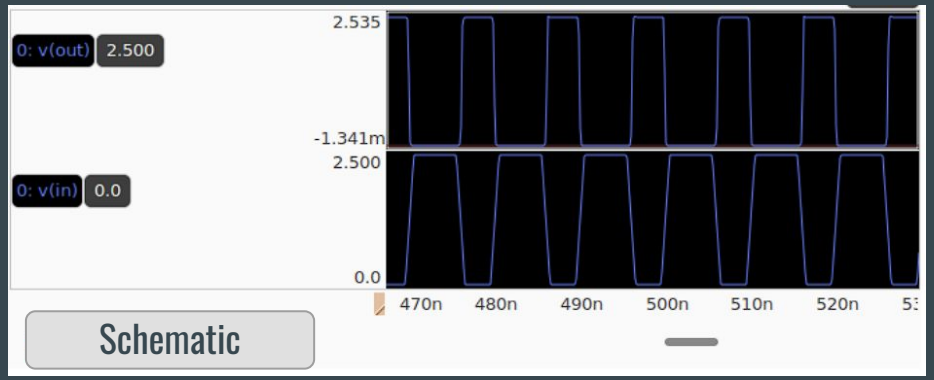
Source: [9]

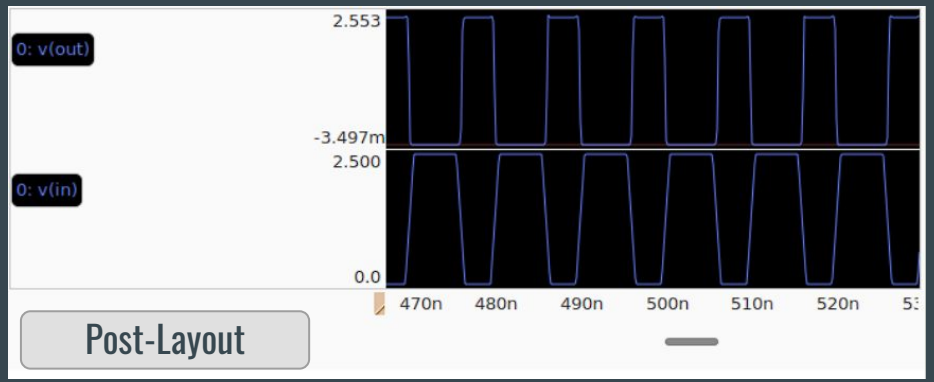
- Simulates Spice netlists

Xyce

Source: [7]

- Simulates both Spice and Verilog-A netlists

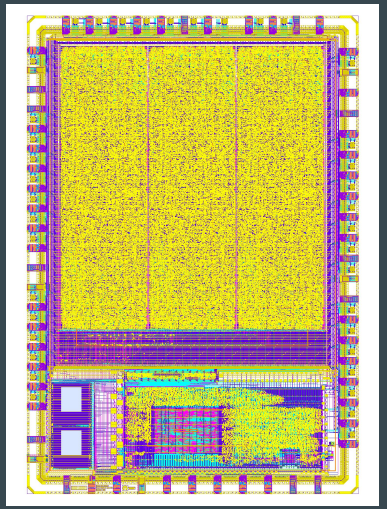




Integration Testing

The ReRAM will be integrated into the user project area of the Caravel harness.

- Digital simulation on the ReRAM crossbar in the Caravel harness
- Analog simulation of the ReRAM crossbar
- LVS and DRC check on the combined layout



Source: [4]

Implementation

Project Plan

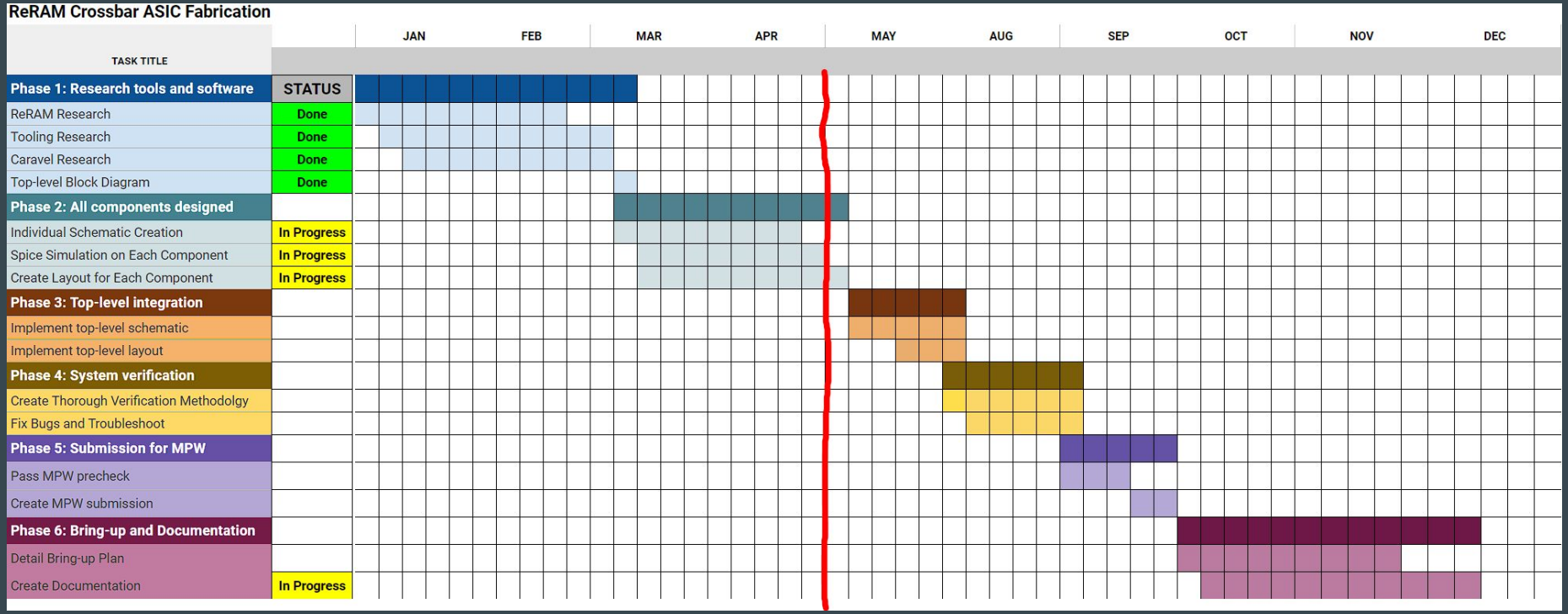
- Research tools and software
- Design components
- Integration of top-level components
- System verification
- Submission for MPW
- Documentation

Through each phase, we will adjust as necessary

- We start with the design functionality
- Different design considerations

Tasks
Phase 1: Research tools and software Research all the open-source software that will be required, along with tools provided by Efabless. Also, will need to research other ReRAM crossbar designs.
Phase 2: All components designed. We implement all components in XSchem, simulate with Ngspice and create layouts using Magic. These layouts must pass LVS checks.
Phase 3: All components in the top-level integrated into one-part Implement all the components into the top level. This top-level must include numerous spice simulations to test functionality and a top-level layout that passes LVS checks.
Phase 4: The system is fully verified. Create detailed simulations for the top level to ensure the crossbar works properly within the Caravel harness.
Phase 5: Submission for MPW Must pass MPW precheck and submit it to potentially be fabricated
Phase 6: Bring-up plan & documentation. We must devise a plan to test our design once it comes back from fabrication. Also, create documentation on our experiences so that future teams will have an easier time.

Project schedule



Project Status

Ongoing component design/simulation

- Completed the implementation of 1-bit DAC
- Currently working on the implementation of 1-bit ADC
- TIA (transimpedance amplifier) and sample and hold components pending implementation
- Schematic design and digital behavioral implementation completed for testing

Our plan

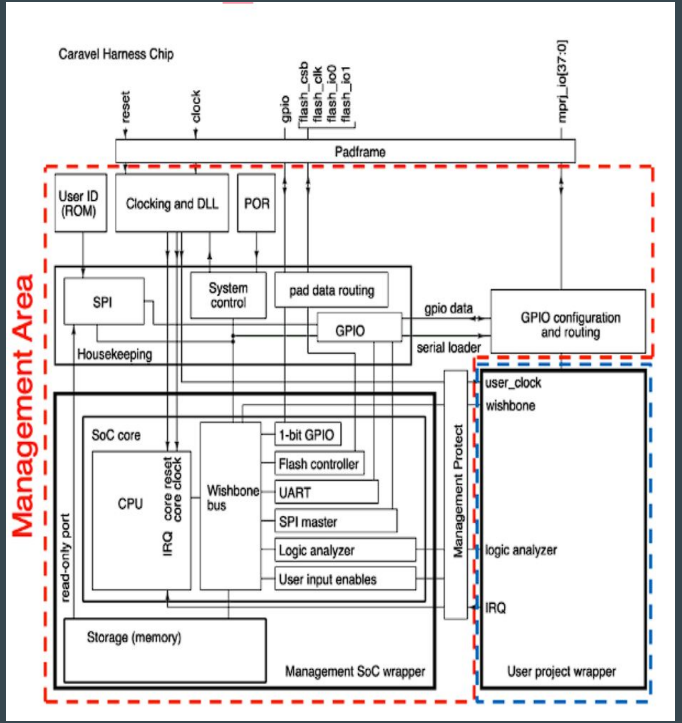
- Start with 1-bit ADC
- 3-bit ADC for design option

Next Steps/Conclusion

Next steps

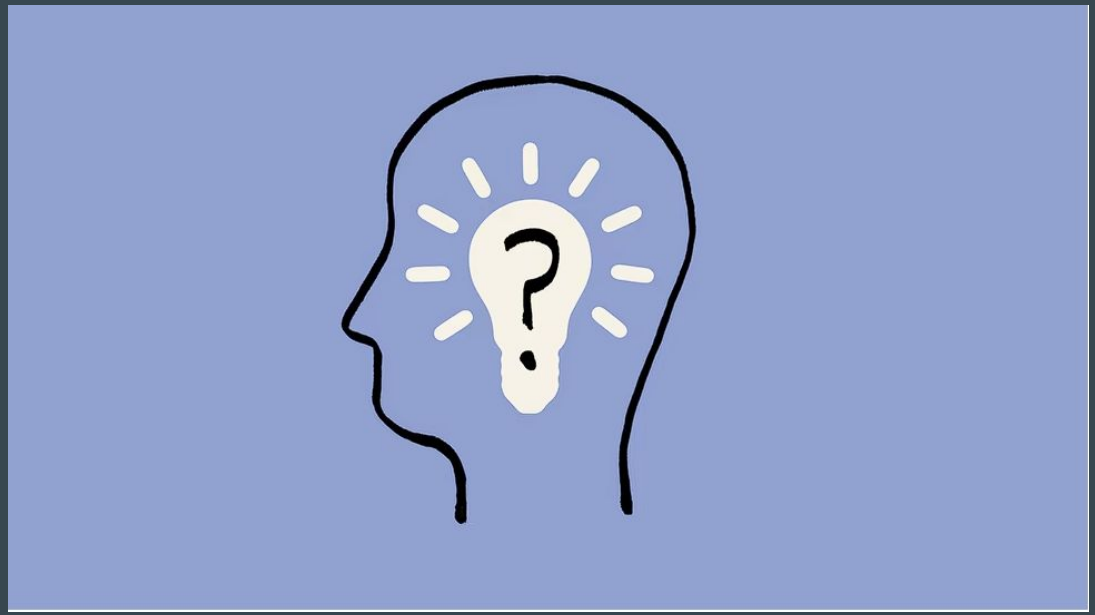
- Complete the implementation of ADC, TIA, and sample and hold
- Integrate the ReRAM crossbar into the design
- Connect all components to form the user area

- Good progress made
- DAC implemented, ADC ongoing
- Our ultimate goal is to fully implement all of the components and fully functional user area interfaced through the Caravel harness



Source: [13]

Questions?



Source: [6]

Supplementary Slides

Hi Dr. Jones!

Work Breakdown

Aiden

- Caravel Harness -> User Area Interaction
- Digital Behavioral Model
- Top Level Design

Joshua

- Created tool environment with correct dependencies/configurations
- Learned and documented entire open-source analog process flow
- Pushed 1-bit DAC through analog process flow

Matthew

- Learned Analog process flow
- Designed Op-Amp, 1-bit ADC, and Sample-and-hold

Regassa

- Analog process flow
- Designing Second stage OP-Amp, 3-bit ADC

Definitions

FOSS - Free and open source software

ASIC - Application-specific integrated circuit

MPW - Multi project wafer

ReRAM - Resistive RAM

PDK - Process design kit

Efabless - Chip design company

Design flow - Steps taken for ASIC design to tape-out

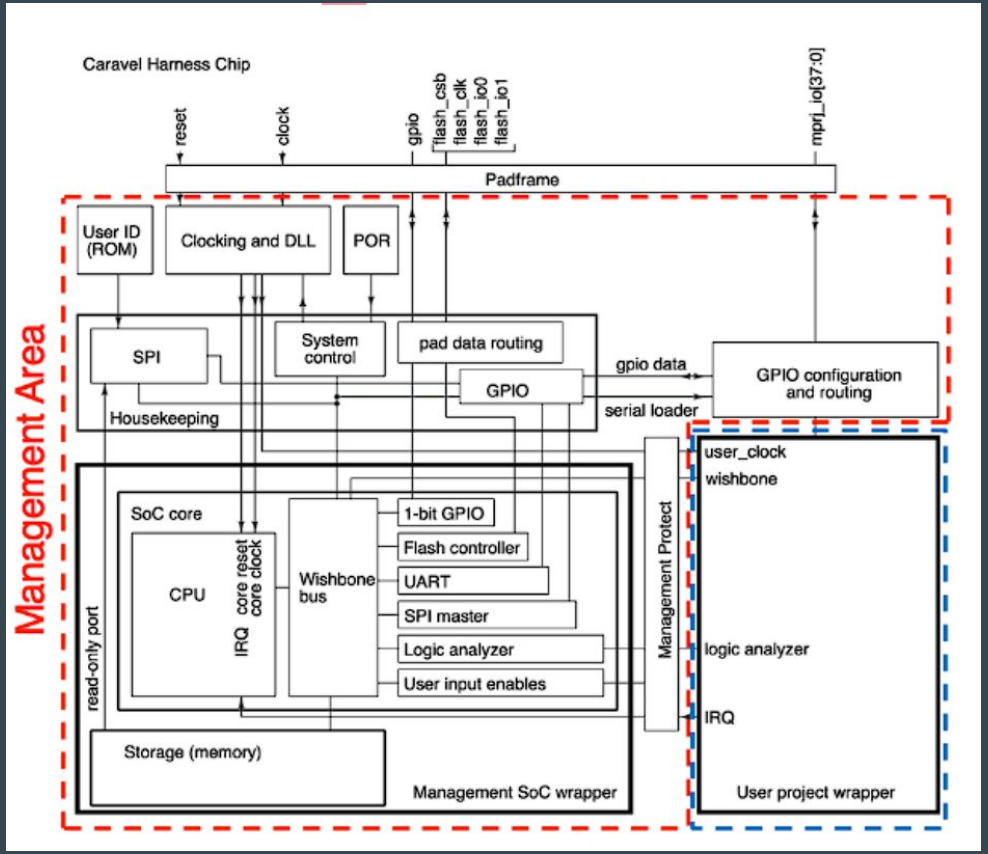
Caravel Harness - PCB circuit that will house our chip

Shuttle - Efabless wafer fabrication

Wafer - Slice of silicon for fabrication of integrated circuits

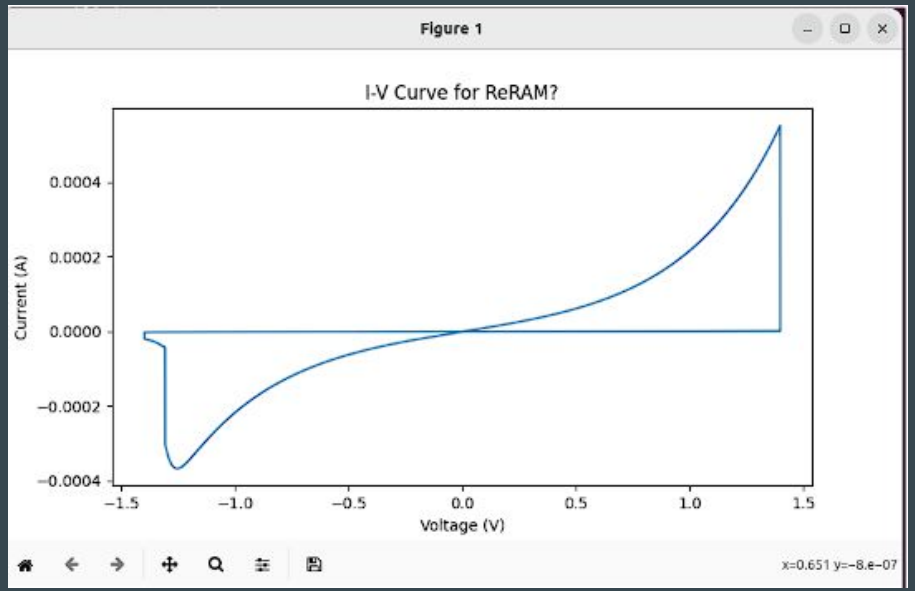
TIA - Transimpedance Amplifier

Caravel Harness Diagram

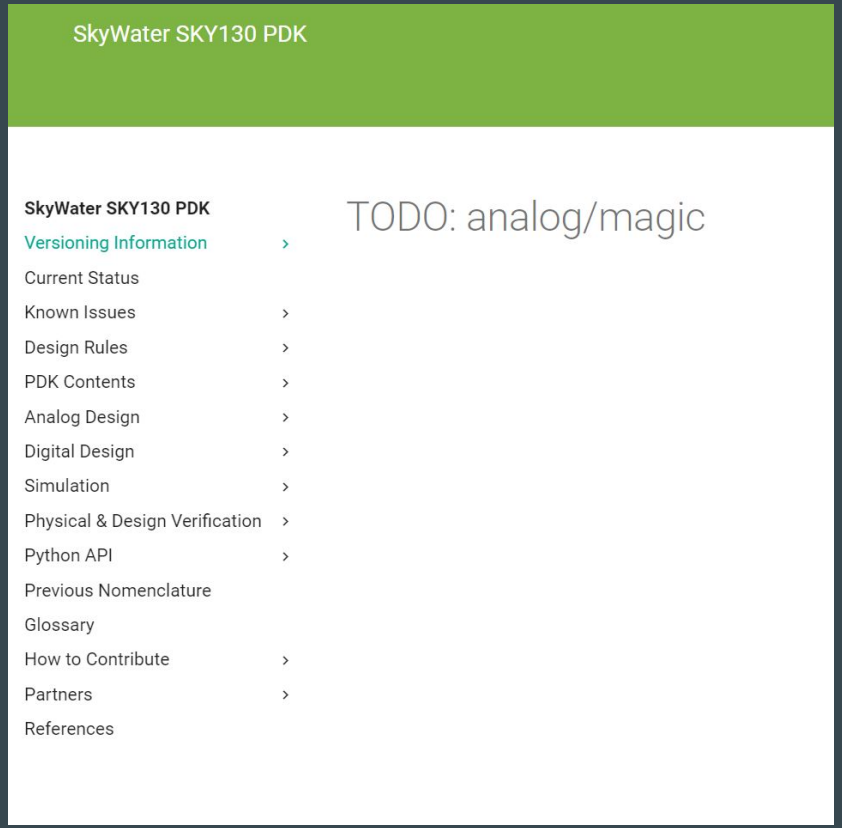


Source: [13]

Skywater 130 nm ReRAM Simulations

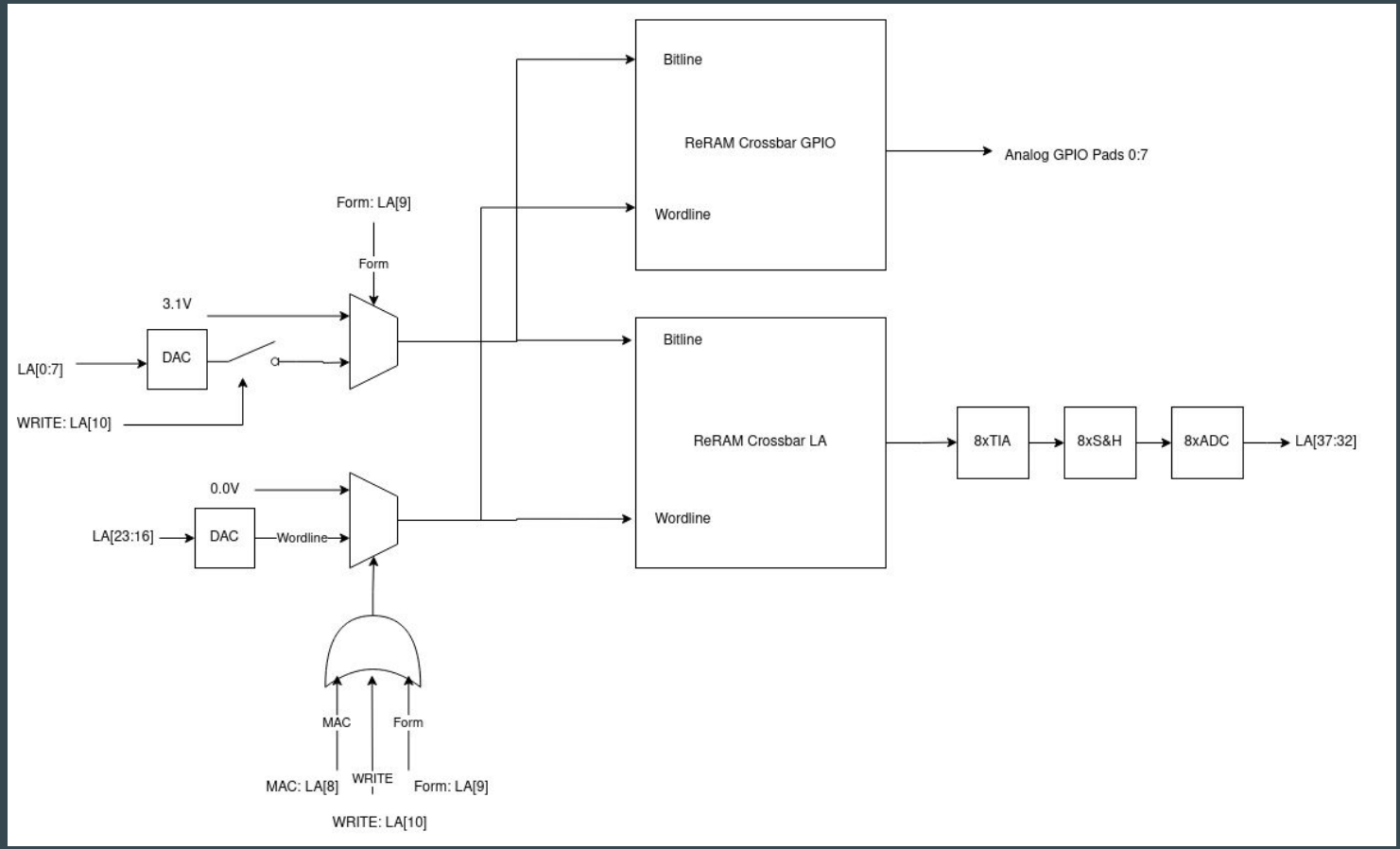


Skywater 130 nm Analog Design Flow Documentation

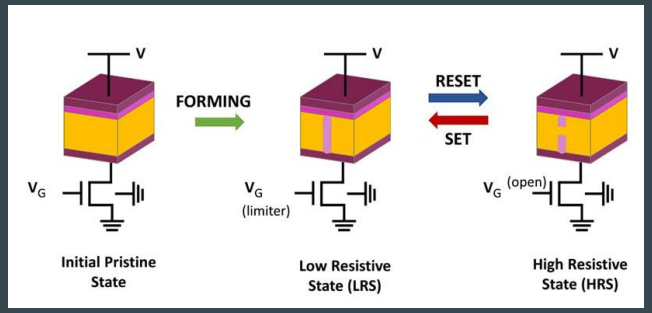


Source: [3]

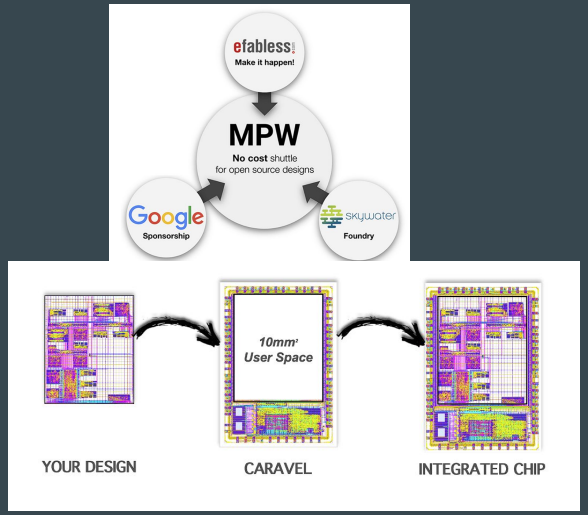
High Level Sketch



References



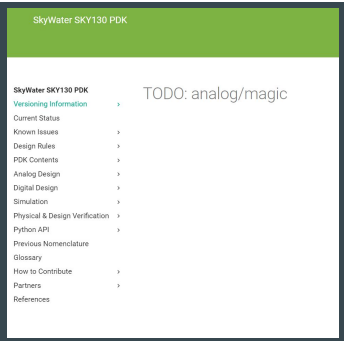
[1] https://www.yolegroup.com/player-interviews/spotlight-on-resistive-ram-reram-an-interview-with-weebit-nano/



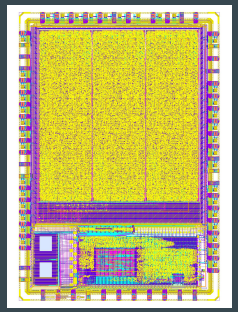
[2] https://efabless.com/open_shuttle_program

Google + skywater TECHNOLOGY

FOSS 130nm Production PDK



[3] https://skywater-pdk.readthedocs.io/en/main/

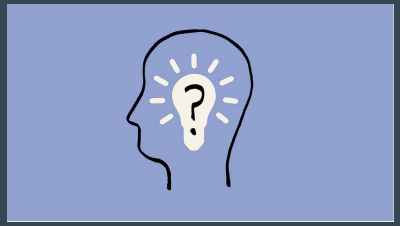


[4] https://platform.efabless.com/projects/9

Table 1 1T1R #5

1T1R #5	WL (V)	BL (V)	SL (V)	PW (ns)	Yield (%)
Pristine					99.90
Form	1.4 - 2.0 (0.1 step)	2.6 - 3.1 (0.1 step)	0	1000	92.73
Reset	2.5	0	2.6	1000	90.83
Set	1.7	2.4	0	1000	97.45

[5] https://sky130-fd-pr-reram.readthedocs.io/en/latest/index.html

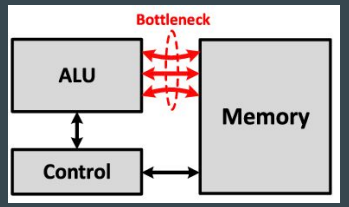


[6] https://www.theatlantic.com/newsletters/archive/2023/01/journalist-interview-asking-questions-techniques/672755/

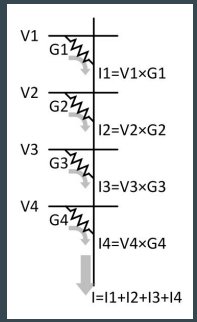
References

[7] https://www.osti.gov/servlets/purl/1333487

[8] http://opencircuitdesign.com/magic/giffiles/magic_title8_2.png



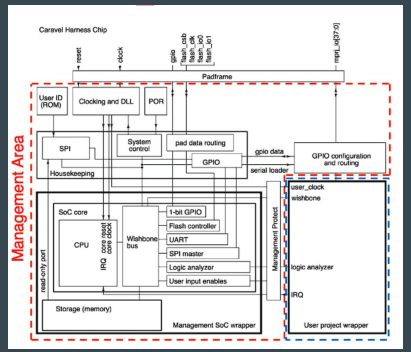
[11] https://www.semanticscholar.org/paper/A-Reconfigurable-4T2R-ReRAM-Computing-In-Memory-for-Chen-Lu/c640e83c98b0c55d54a755e69b57e6281e8b5eec



[12] https://web.eecs.umich.edu/~zhengya/papers/chou_micro19.pdf

[9] https://mueller-semi.xyz/project/ngspice/

[10] https://sourceforge.net/p/ngspice/discussion/133842/thread/711f9043d7/c0e9/attachment/schematic_sky130.PNG



[13] https://github.com/efabless/caravel