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
Team Website: https://sddec23-08.sd.ece.iastate.edu/

Introduction

Project Introduction

Problem:

- Our advisor & client are interested in ReRAM (resistive RAM) for it’s computational potential
- Develop in house ISU computational ASIC capabilities - specifically for the analog and mixed-signal processes utilizing open-source technology
- Want to fabricate an analog ASIC using open-source tools

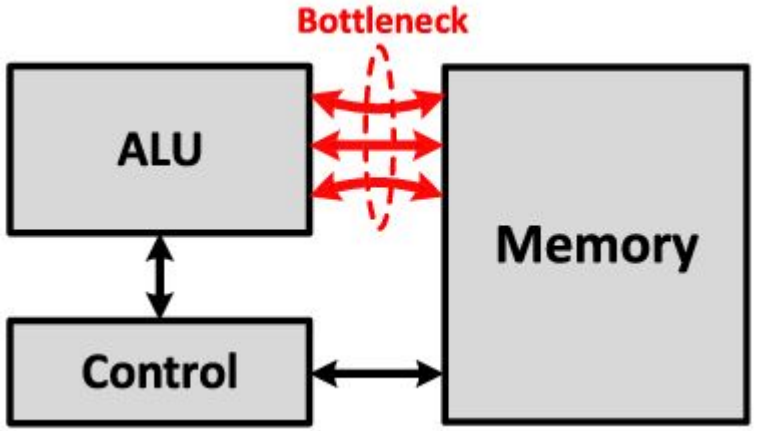
Goals:

- Silicon prove an ReRAM compute crossbar using the eFabless analog design flow
- Create detailed user documentation on the analog design flow to build up ISUs internal knowledge base of chip fabrication
- Create detailed post-fabrication bring-up plan for future students

Traditional Computing Bottleneck

Basic flow of data:

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

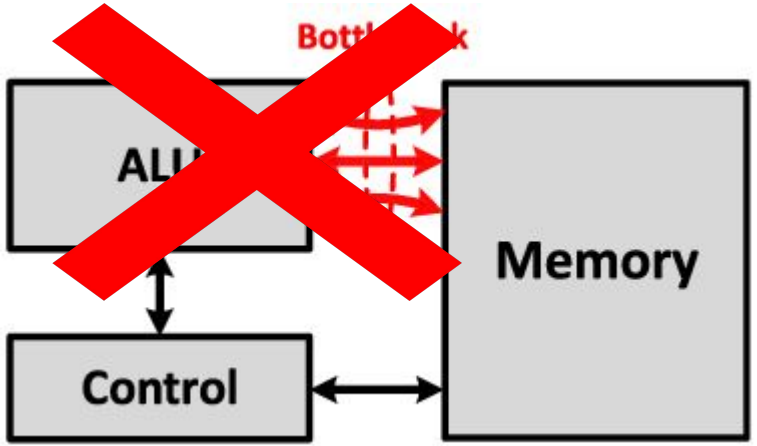


Source: [1]

Processing in Memory Architecture

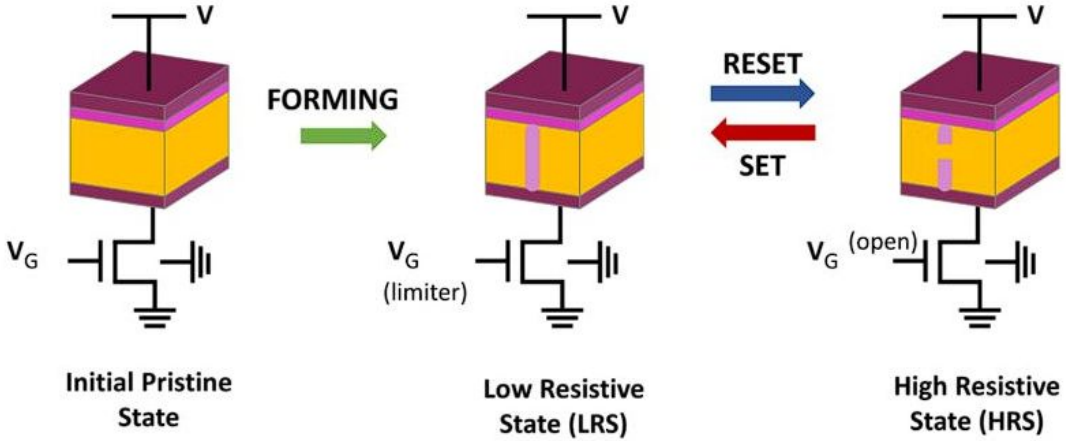
An ReRAM crossbar can do it’s computations in the analog domain inside of memory, which is more power efficient and has lower latencies.

- Less memory transfers
  - More power efficient
  - More area efficient



Source: [1]

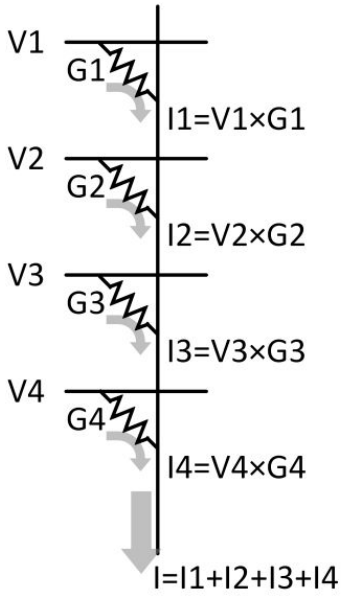
Functionality of ReRAM



Source: [2]

ReRAM States

G = Conductance



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

Source: [3]

ReRAM Multiply and Accumulate Operation

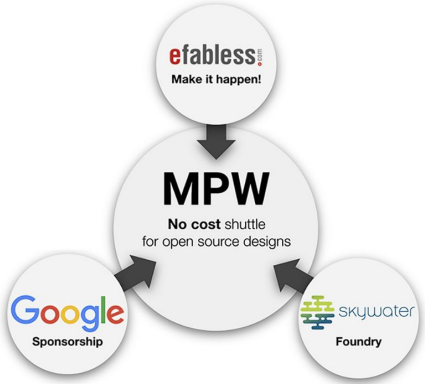
eFabless Background

eFabless hosts shuttles every few months (openMPW)

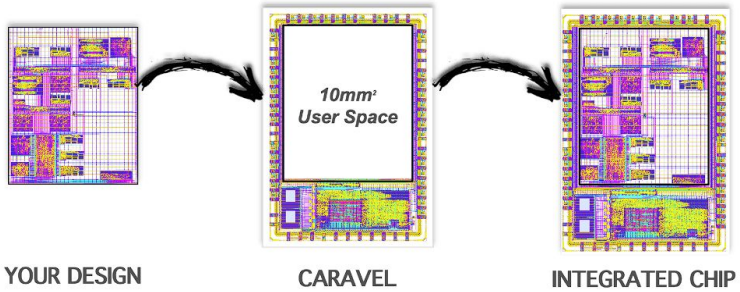
- Sponsored by Google
- Based entirely on open-source technology
- Utilizing the SkyWater 130 nm PDK

Design Process:

- Design our components with open-source tools
- Integrate and hook-up our design into the Caravel Harness
- Run precheck on design
- Submit design



Source: [4]



Source: [4]

Project Requirements

Functional requirements:

- ReRAM crossbar must be able to perform write, read, & MAC operations
- Periphery analog circuitry to support crossbar computations
  - Also to support external characterization of devices

Non-functional requirements:

- Create detailed post-fabrication bring-up plan
- Create documentation on all parts of analog design flow
  - Setting up the environment
  - Tutorials on tool usage
  - Integration of tools with SkyWater 130 nm process

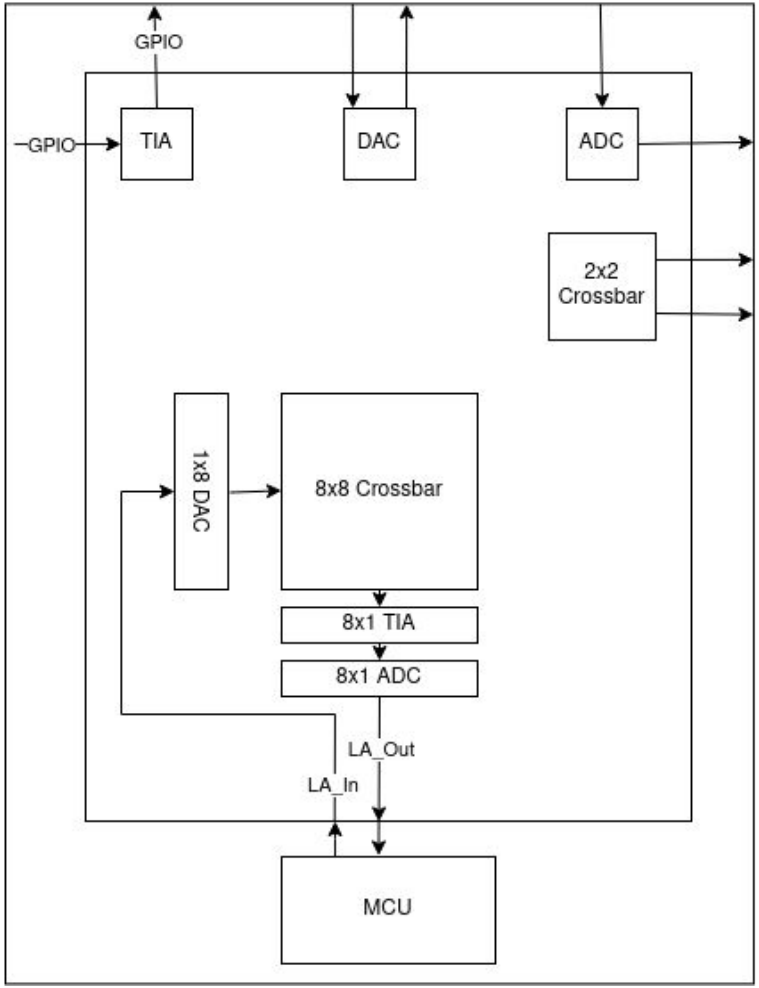
Source: [5]

Design

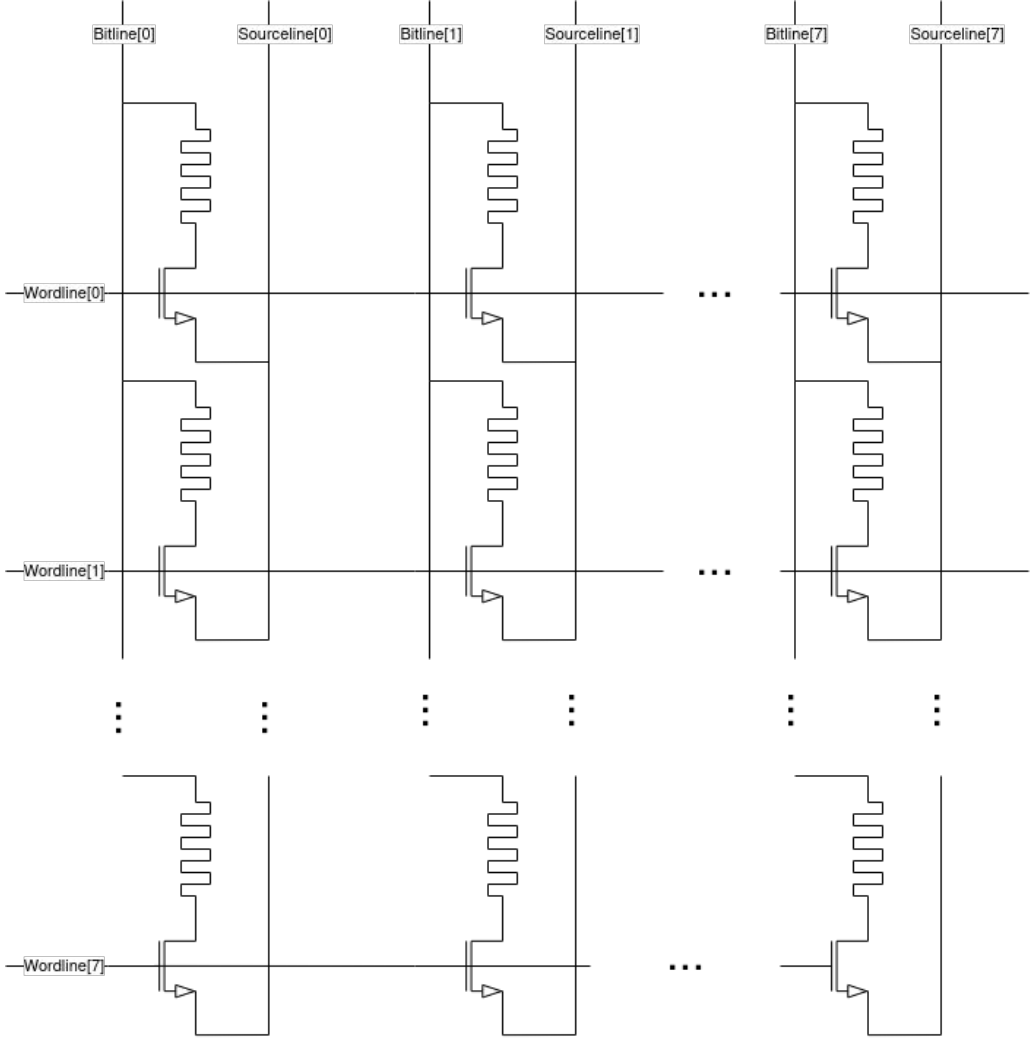
Concept Sketch

- Data is sent from MCU to the crossbar and is sent back into the MCU

- Must convert data into correct voltages when communicating with the crossbar



8x8 Crossbar



Operations

- Read line
- Write line
- MAC
- Form

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

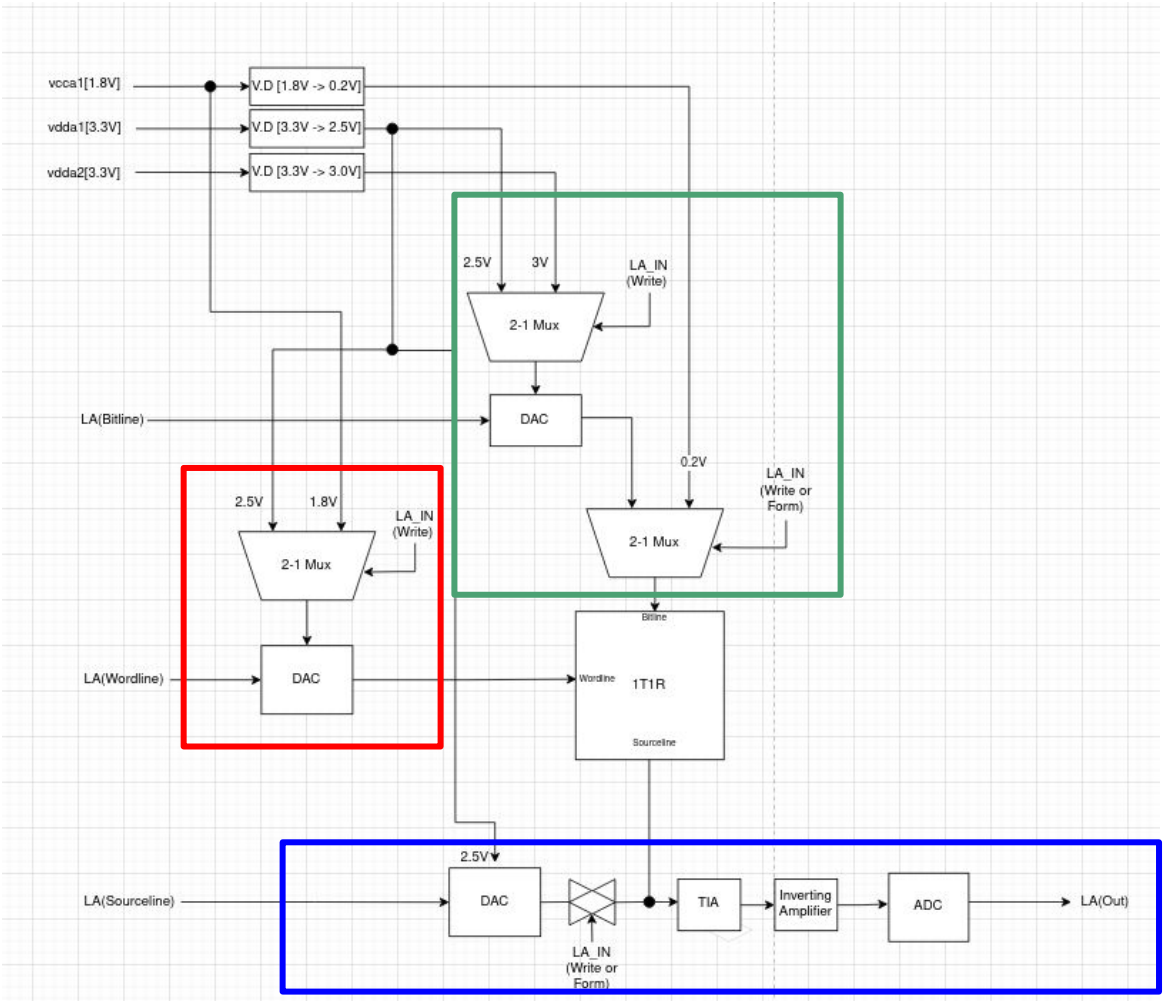
Source: [5]

Top Level 8x8 Crossbar Design

Word Line

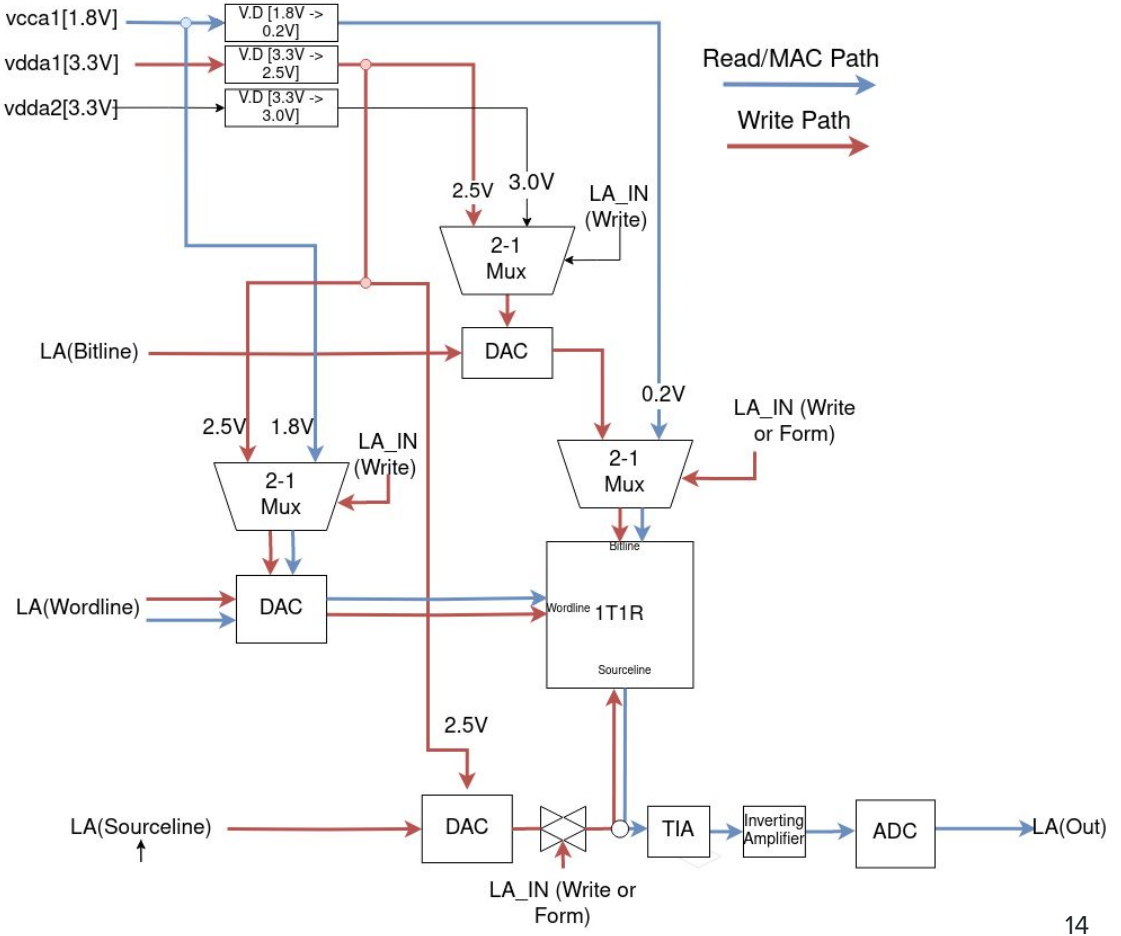
Source Line

Bit Line



High Level Operations

- Red paths are writes
- Blue paths are Read/MACs



User Area Communication & Drivers

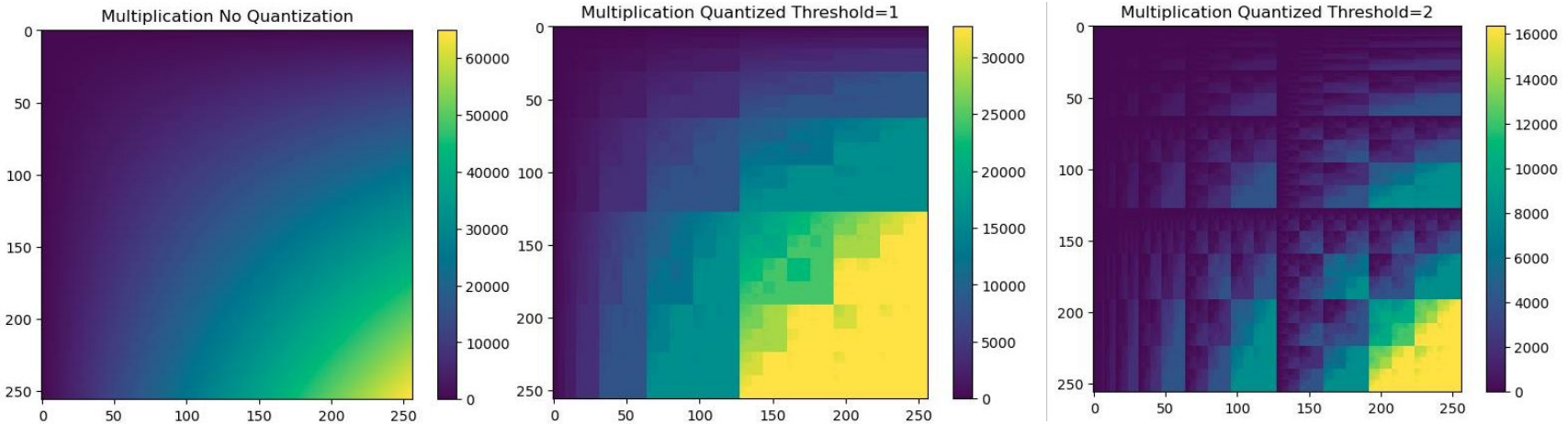
- We have created drivers to control the crossbar through the user area
- We communicate with the user area through the logic analyzer
- We have functions to perform each of the crossbar functionality

Digital Behavioral Model

- Models the expected behaviour of the ReRAM Crossbar
- Simple and single cycle, not expected to be used as real hardware
- Allows the drivers to be tested before fabrication
  - We don’t have a mechanism to run the drivers on the analog design

Usages Of ReRAM Crossbar

- Can do 1-bit vector matrix multiplication
  - This is useful for binary neural networks
- Can perform 8-bit multiplications (inaccurate)



- Our design could be expanded to allow for larger cells and bigger ADCs
  - Would allow for higher precision MACs.

Implementation

Analog Tools

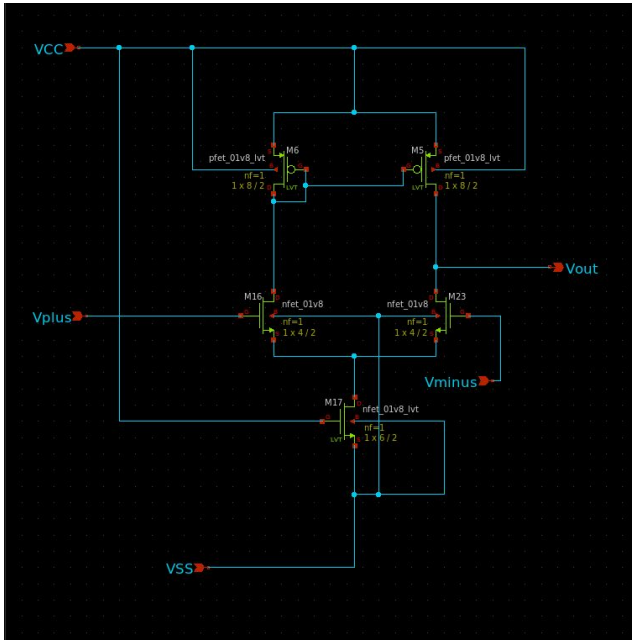
- The goal of this project is to only use open source tools for the entire design flow
- We also want to create documentation on these tools for future students and users

Tools	Function
Xschem	Schematic Editor
Ngspice	Spice Circuit Simulator
Magic	Layout Editor & DRC
Netgen	LVS

Analog Design Flow

Xschem

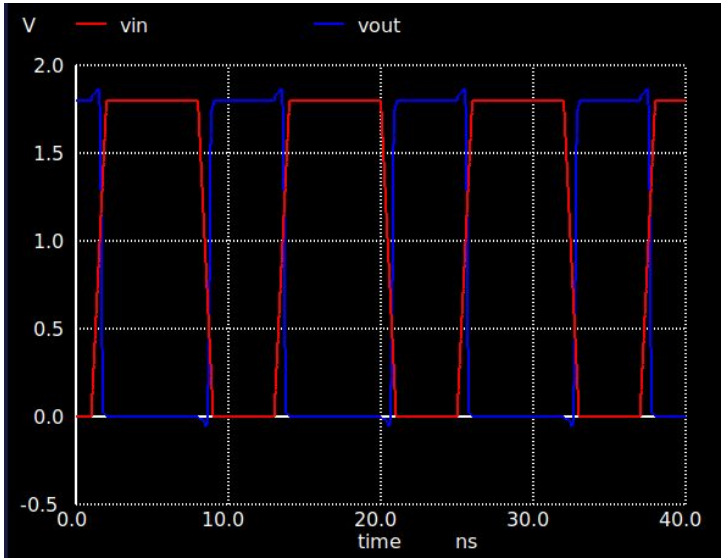
- Allows us to design schematics for analog components
- Used to create testbenches that can simulate schematics and Spice netlists
- Exports testbenches and components as Spice netlists



Analog Design Flow

Ngspice

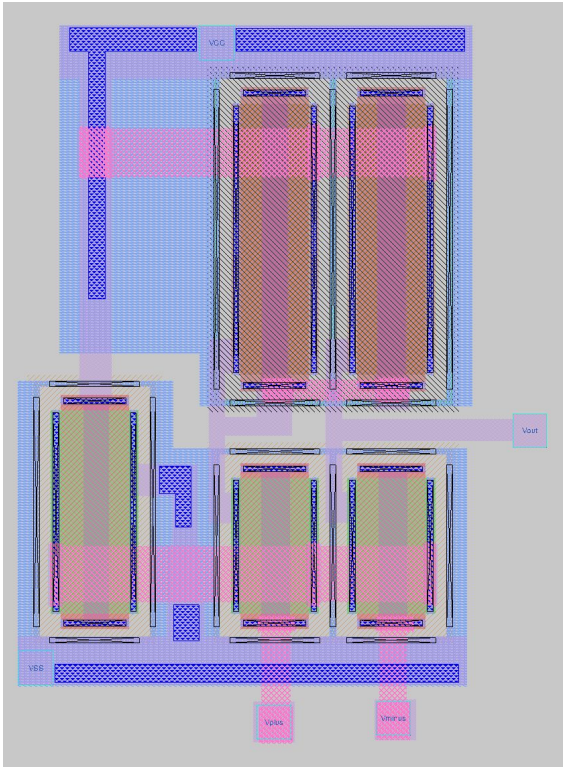
- Simulates Spice netlists
- Confirms operation of analog components
- Limited capability of simulating ReRAM



Analog Design Flow

Magic

- Used to create layout for our components
- Instiates components from Spice netlist
- Runs a design rule check (DRC)
- Exports Spice netlist of component



Analog Design Flow

Netgen

- Runs a layout vs. schematic (LVS)
- Compares netlists from Xschem and Magic

Ngspice

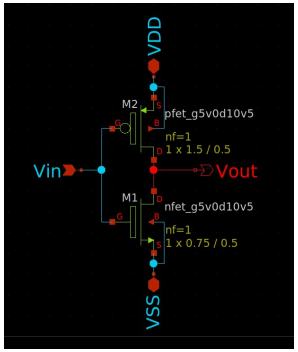
- Used to run a post layout simulation
- This simulation includes parasitic capacitances that are created in the layout

Testing

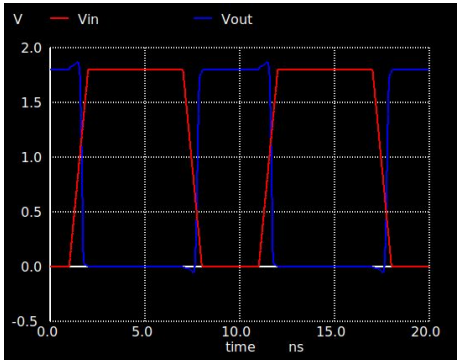
“Unit Testing”

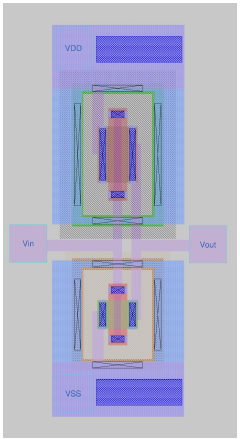
The criteria we used to verify our analog components functionality was:

- Schematic simulations prove functionality
- Layout passes LVS and DRC
- Post-layout simulations match schematic simulations

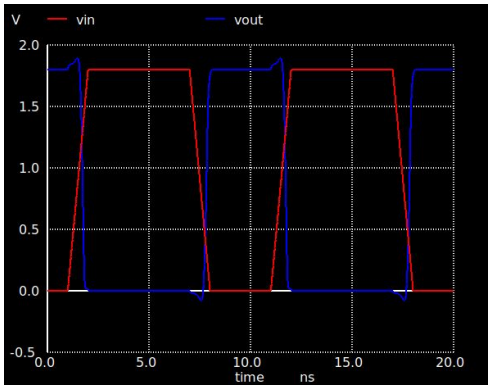


Schematic Simulation





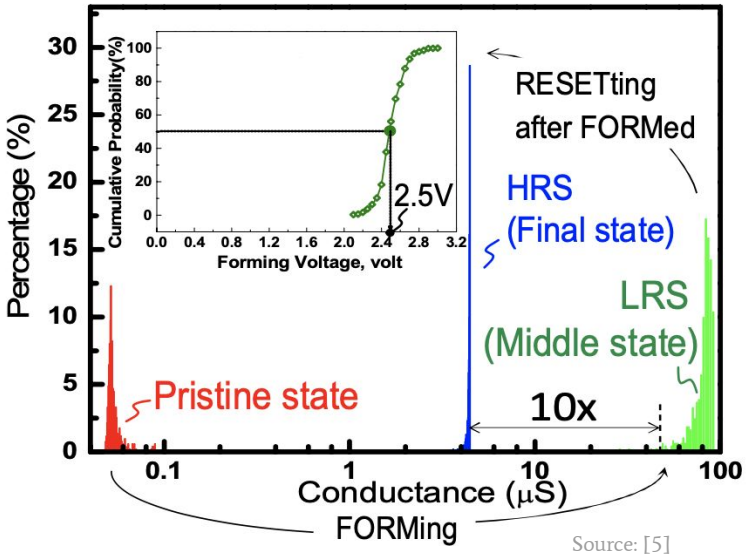
Post Layout Simulation



ReRAM Cell Challenges

One of our biggest challenges was modelling the ReRAM cell characteristics

- The Spice model provided to us in the SkyWater PDK is not robust causing convergence errors when simulated
  - We have been in contact with eFabless and this is known issue
- Our current work around to this is using resistors sized to match the conductance values in the SkyWater ReRAM documentation in place of the ReRAM



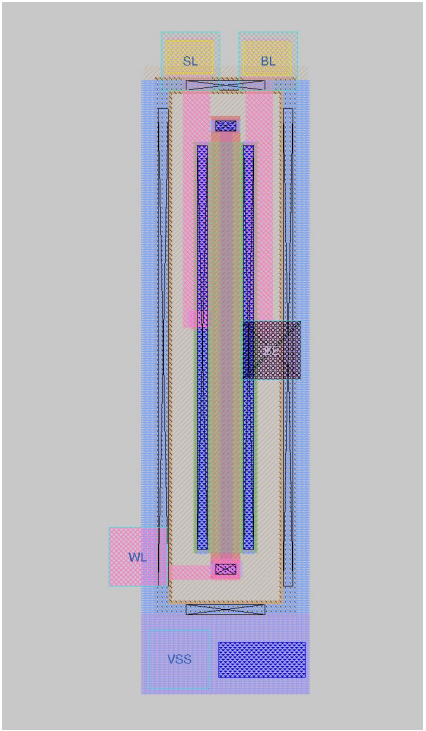
Source: [5]

ReRAM Cell Challenges Continued

The ReRAM model is unable to pass LVS checks due to recent changes to the model

- As a result we cannot pass precheck until this has been resolved
  - eFabless has been contacted and they are currently working on a fix

1T1R Layout



Risk Mitigation: 0.4V & 0.2V Crossbar

We created two crossbars, one that does read/MAC operations at 0.2V and the other at 0.4V

- We have read that 0.4V provides more accurate read/MACs but the higher voltage is more likely to disrupt the ReRAM Filament
- 0.2V is the minimum voltage needed to perform a read/MAC so we created two crossbars to compare the performance of the different voltages
- The difference between these two crossbars other than the MAC/read voltages is that the TIA is sized differently to compensate

Risk Mitigation: Analog Circuitry External Characterization

In our user project area we included multiple individual components in the user project area connected to GPIO pins so that we could externally verify the functionality of the devices we designed after fabrication

Individual devices included:

- 4 2x2 Crossbars
  - 2 “normal sized”
  - 2 “larger sized” (larger transistors)
- 2 DACs
- 2 ADCs
- 2 2-1 MUXs

Top-Level Testing

For the top-level schematic we did extensive testing

- Ran extensive tests on both the 0.2V and 0.4V crossbars
- Confirmed the sourceline, worldline, and bitline were receiving correct voltages
- Performed error testing by simulating ReRAM characteristics on edges of possible conductance ranges
- Ran corner tests to verify the designs functionality with possible variances in the MOS devices

Top-Level MAC/Read Operations Results

At 0.2 V, 80 uS

# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
0	8	TT	0.1879	0.0592	0.4692	0.0057
0	8	SS	0.1847	0.0578	0.4557	0.0039
0	8	FF	0.1901	0.0602	0.4795	0.0077
0	8	FS	0.1843	0.0595	0.4509	0.0039
0	8	SF	0.1898	0.0583	0.4775	0.0075
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
1	7	TT	0.1804	0.1203	0.7049	1.784
1	7	SS	0.1749	0.1147	0.7064	1.787
1	7	FF	0.1846	0.1245	0.7057	1.778
1	7	FS	0.1746	0.1143	0.7165	1.789
1	7	SF	0.1844	0.1247	0.6944	1.775
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
8	0	TT	0.1783	0.1666	0.772	1.789
8	0	SS	0.1715	0.1601	0.7763	1.792
8	0	FF	0.1846	0.1243	0.7057	1.778
8	0	FS	0.1746	0.1143	0.7165	1.789
8	0	SF	0.1844	0.1247	0.6944	1.775

At 0.4 V, 80 uS

# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
0	8	TT	0.3694	0.7192	0.429	0.0226
0	8	SS	0.3613	0.7045	0.4154	0.01328
0	8	FF	0.3749	0.7293	0.4394	0.03421
0	8	FS	0.3602	0.7223	0.4208	0.01415
0	8	SF	0.3754	0.7062	0.4421	0.03053
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
1	7	TT	0.3436	0.1818	0.7345	1.788
1	7	SS	0.3282	0.1684	0.7314	1.79
1	7	FF	0.3543	0.1919	0.7333	1.784
1	7	FS	0.3258	0.1668	0.735	1.792
1	7	SF	0.3555	0.1928	0.7255	1.781
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
8	0	TT	0.3315	0.2987	0.8189	1.793
8	0	SS	0.3122	0.2796	0.8225	1.796
8	0	FF	0.3447	0.3119	0.8095	1.791
8	0	FS	0.3091	0.2765	0.8297	1.796
8	0	SF	0.3462	0.3134	0.7975	1.789

At 0.2 V, 100 uS

# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
0	8	TT	0.1879	0.0592	0.4692	0.0057
0	8	SS	0.1847	0.0578	0.4557	0.0039
0	8	FF	0.1901	0.0602	0.4795	0.0077
0	8	FS	0.1843	0.0595	0.4509	0.0039
0	8	SF	0.1898	0.0583	0.4775	0.0075
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
1	7	TT	0.1799	0.128	0.7177	1.786
1	7	SS	0.1742	0.1221	0.7217	1.788
1	7	FF	0.1842	0.1323	0.07172	1.78
1	7	FS	0.1739	0.1216	0.7316	1.79
1	7	SF	0.184	0.1325	0.7058	1.778
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
8	0	TT	0.1782	0.1688	0.7741	1.79
8	0	SS	0.1714	0.1632	0.7786	1.79
8	0	FF	0.1824	0.1732	0.7636	1.78
8	0	FS	0.1704	0.1613	0.7853	1.793
8	0	SF	0.1826	0.1736	0.7508	1.784

At 0.4 V, 100 uS

# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
0	8	TT	0.3694	0.7192	0.429	0.0226
0	8	SS	0.3613	0.7045	0.4154	0.01328
0	8	FF	0.3749	0.7293	0.4394	0.03421
0	8	FS	0.3602	0.7223	0.4208	0.01415
0	8	SF	0.3754	0.7062	0.4421	0.03053
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
1	7	TT	0.3415	0.1997	0.753	1.794
1	7	SS	0.3253	0.1851	0.7502	1.796
1	7	FF	0.3526	0.2105	0.7503	1.792
1	7	FS	0.3228	0.1831	0.7569	1.796
1	7	SF	0.3538	0.2116	0.7403	1.791
# of cells in LRS	# of cells in HRS	TEST	BL_In (V)	SL_In (V)	InvOut (V)	Vout (V)
8	0	TT	0.331	0.304	0.8207	1.797
8	0	SS	0.3116	0.285	0.825	1.798
8	0	FF	0.3443	0.3176	0.8125	1.795
8	0	FS	0.3084	0.2819	0.8349	1.798
8	0	SF	0.3458	0.3192	0.8003	1.794

Status of Deliverables

Deliverable - 8x8 Compute Crossbar

- Able to push all individual components through open-source analog design flow and verify post-layout functionality
- Both crossbar designs are able to do MAC and Read operations
- All inputs to the crossbar were able to receive the correct voltages
- Overall design did not pass precheck due to outstanding LVS issues plus issues with ReRAM

Process Step					Post Layout
Component	Schematic	Simulation	Layout	DRC/LVS	Simulations
Inverter	Done	Done	Done	Done	Done
Buffer (DAC)	Done	Done	Done	Done	Done
2-1 MUX	Done	Done	Done	Done	Done
Level Shifters	Done	Done	Done	Done	Done
OP-Amp	Done	Done	Done	Done	Done
TIA	Done	Done	Done	Done	Done
Inverting Amp	Done	Done	Done	Done	Done
1-Bit ADC	Done	Done	Done	Done	Done
1T1R Cell	Done	In-Progress	In-Progress	In-Progress	In-Progress
8x8 Crossbar	Done	In-Progress	In-Progress	In-Progress	In-Progress
8 Line WL	Done	Done	Done	Done	Done
8 Line BL	Done	Done	Done	Done	Done
8 Line SL In	Done	Done	Done	Done	Done
8 Line SL out	Done	Done	Done	Done	Done
Top Level	Done	Done	Done	In-Progress	In-Progress

Deliverable - Tutorial Documentation: Analog Design Flow

- Tutorial documentation is completed
  - Very in-depth document that goes over all steps of the analog open-source design flow
  - Has not been used by anyone outside of the team
  - Beneficial to future senior design teams, researchers, and others interested in analog chip fabrication

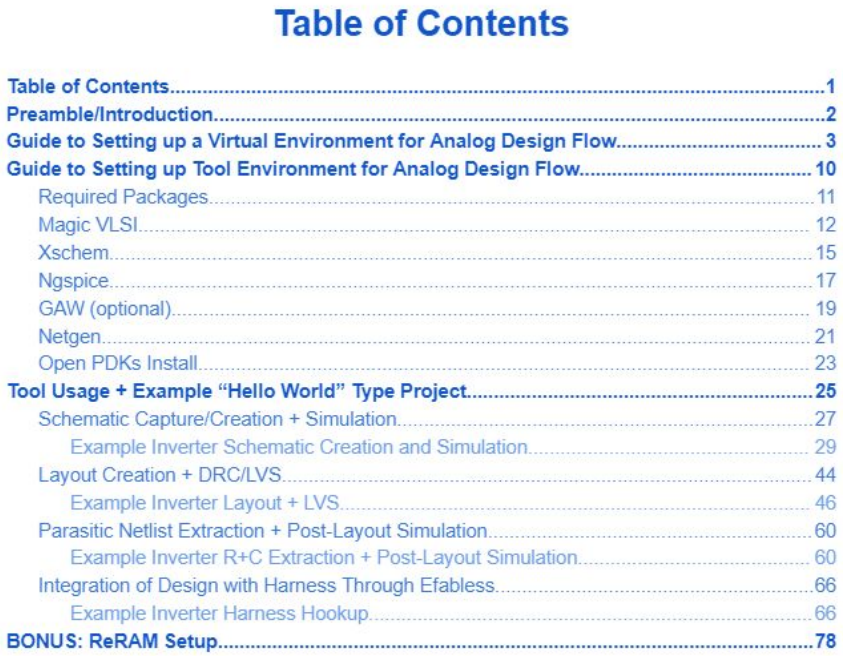
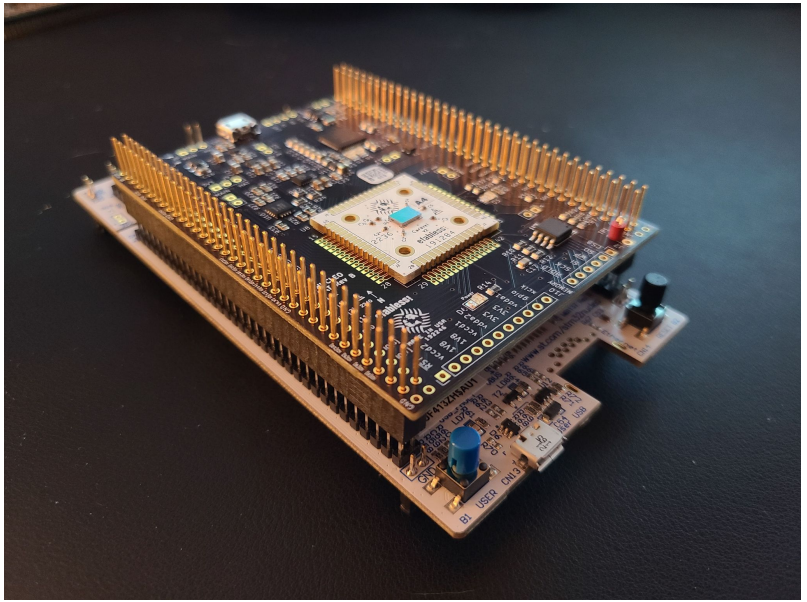


Table of Contents

Deliverable - Bring-up Plan

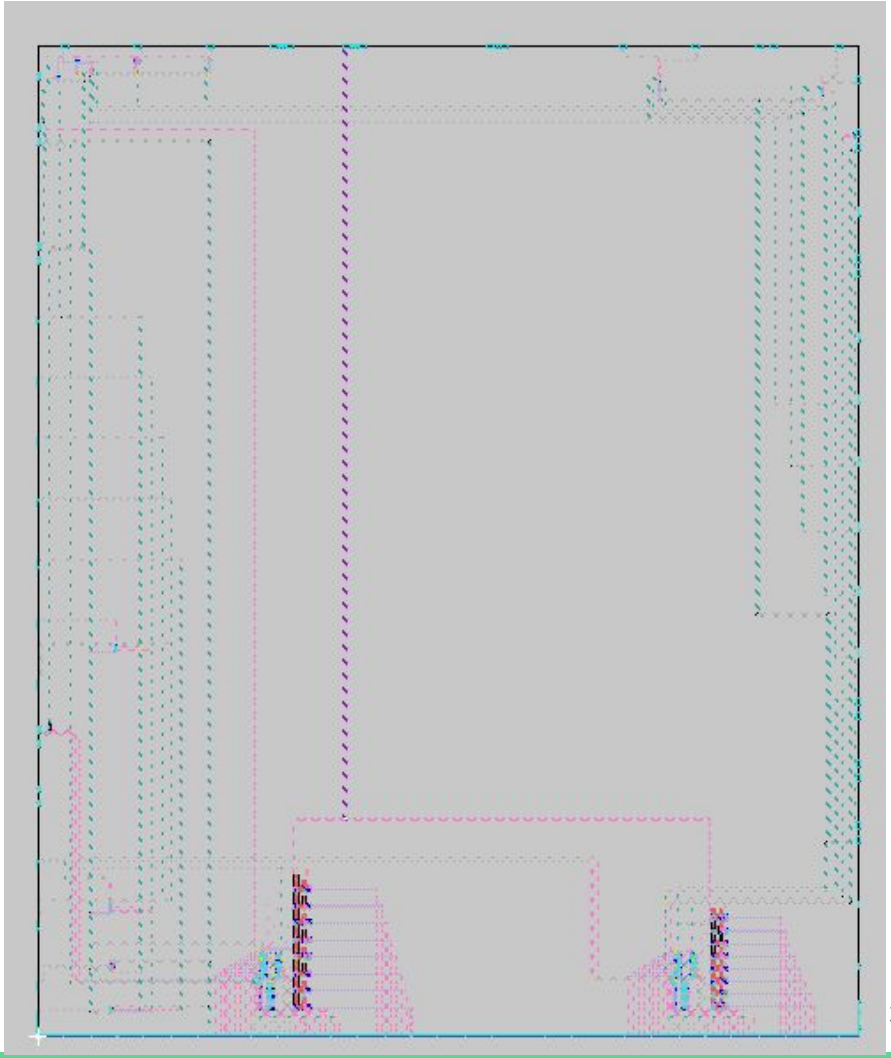
- Created a bring-up plan in order to document how to use a board received from eFabless with our design on it
- Contains:
  - How to setup repository with libraries to use board
  - How to flash firmware necessary to run crossbar
  - How to test standalone components to test ReRAM functionality
    - We have components not connected to LA so that we can manually test them.



Source: [6]

Conclusion

- 8x8 compute crossbar is not fully implemented, but excellent progress has been made towards its completion
- Tutorial documentation on open-source analog design flow is completed
- Bring-up plan along with other supplemental documentation has been completed
- Excellent groundwork has been laid out for future senior design team(s) to hit the ground running and make tweak/make improvements upon our design
- Both client & advisor are satisfied with the work we have been able to achieve



Questions?

Bibliography

[1] [PDF] a reconfigurable 4T2R RERAM computing in-memory macro for ..., https://www.semanticscholar.org/paper/A-Reconfigurable-4T2R-ReRAM-Computing-In-Memory-for-Chen-Lu/c640e83c98b0c55d54a755e69b57e6281e8b5eec.

[2] T. Boucard, “Spotlight on Resistive Ram (reram) – an interview with Weebit Nano,” Yole Group, https://www.yolegroup.com/player-interviews/spotlight-on-resistive-ram-reram-an-interview-with-weebit-nano/.

[3] T. Chou, W. Tang, J. Botimer, and Z. Zhang, Electrical Engineering and computer science, https://web.eecs.umich.edu/~zhengya/papers/chou_micro19.pdf.

[4] “Efabless.com,” Make Your Own Chips for Free, https://efabless.com/open_shuttle_program.

[5] “Technology specifications¶,” Technology Specifications - SkyWater SKY130 PDK 0.0.0-22-g72df095 documentation, https://sky130-fd-pr-reram.readthedocs.io/en/latest/technology_specifications.html#seting-reseting.

[6] “Caravel Nucleo Hat,” GitHub, https://github.com/efabless/caravel_board/tree/main/firmware/mpw2-5/nucleo.

Supplementary Slides

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/process 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

Contributions - Joshua Thater

- Figured out and documented the entire open-source analog design flow and integration with Caravel Harness
- Pushed designs through open-source analog design:
  - Inverter
  - Buffer
  - Transmission Gate
  - 2-1 MUX
  - ReRAM crossbars
- Created overall layout of design and integrated it with Caravel Harness

Contributions - Aiden Petersen

- Documented and ran design through digital design flow
- Created and tested drivers
- Created and tested digital Behavioral Model
- Contributed towards top level design and understanding of the crossbar
- Exploration into “quantized multiplication” using ReRAM crossbar

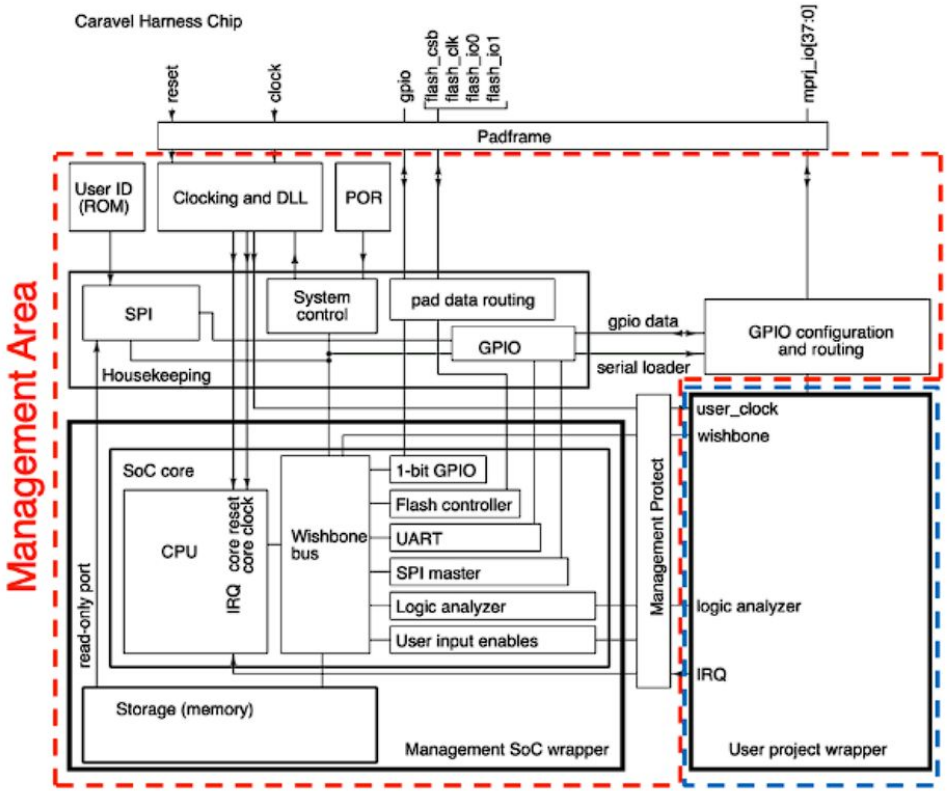
Contributions - Matthew Ottersen

- Worked on understanding Analog Process Flow
- Pushed components through process flow and created layouts for:
  - Voltage dividers
  - 5T Operational Amplifier
  - One Bit ADC
  - Transimpedance Amplifier
- Worked on Bring Up Plan
- Did Final Testing on Top Level Model

Contributions - Regassa Dukele

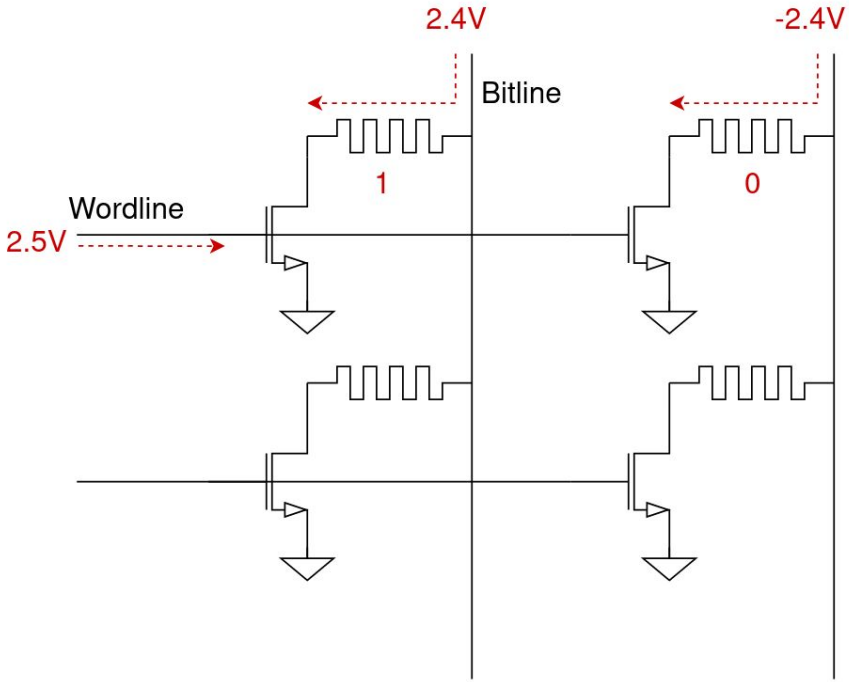
- Worked on the understanding analog design flow
- Pushed designs through open-source analog design:
  - 2-stage amplifier
  - 3-bit ADC
    - Designs were ultimately not used
- Worked on the conclusive and final phase of top-level testing

Caravel Harness Diagram



Simplified ReRAM Write

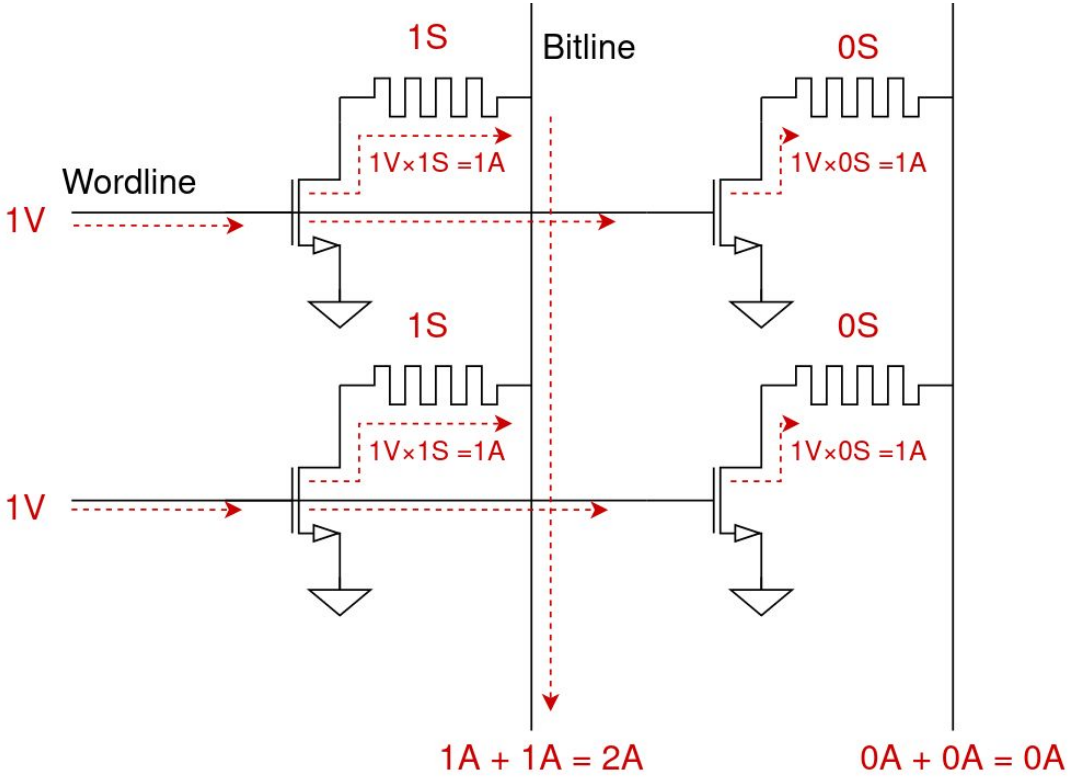
- Bitline contains value written to line
- Wordline is one-hot-encoded line number



(sourceline voltage is complementary to sourceline and is not included in this diagram)

Simplified ReRAM MAC

- 1-bit MAC
- Input comes through wordline
- Matrix weights are stored in ReRAM
- Outputs through bitline



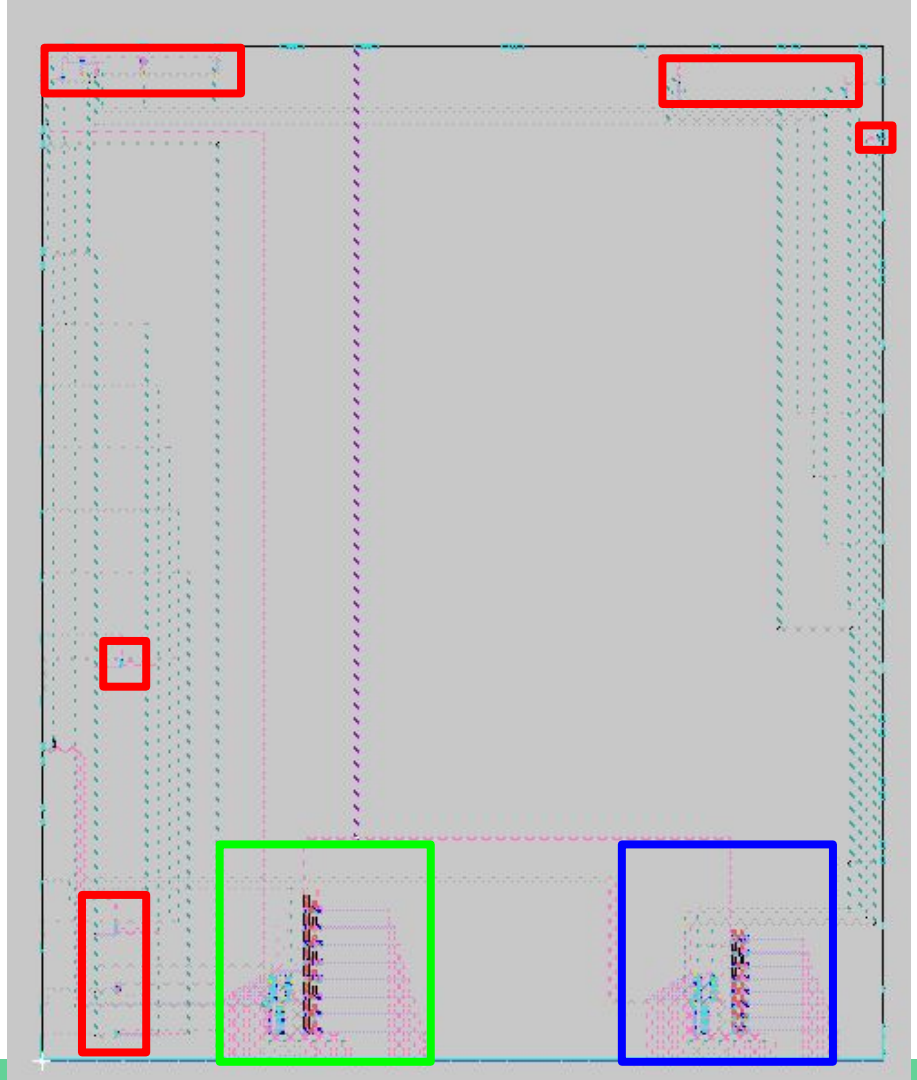
(sourceline voltage is complementary to sourceline and is not included in this diagram)

Design in Caravel Harness

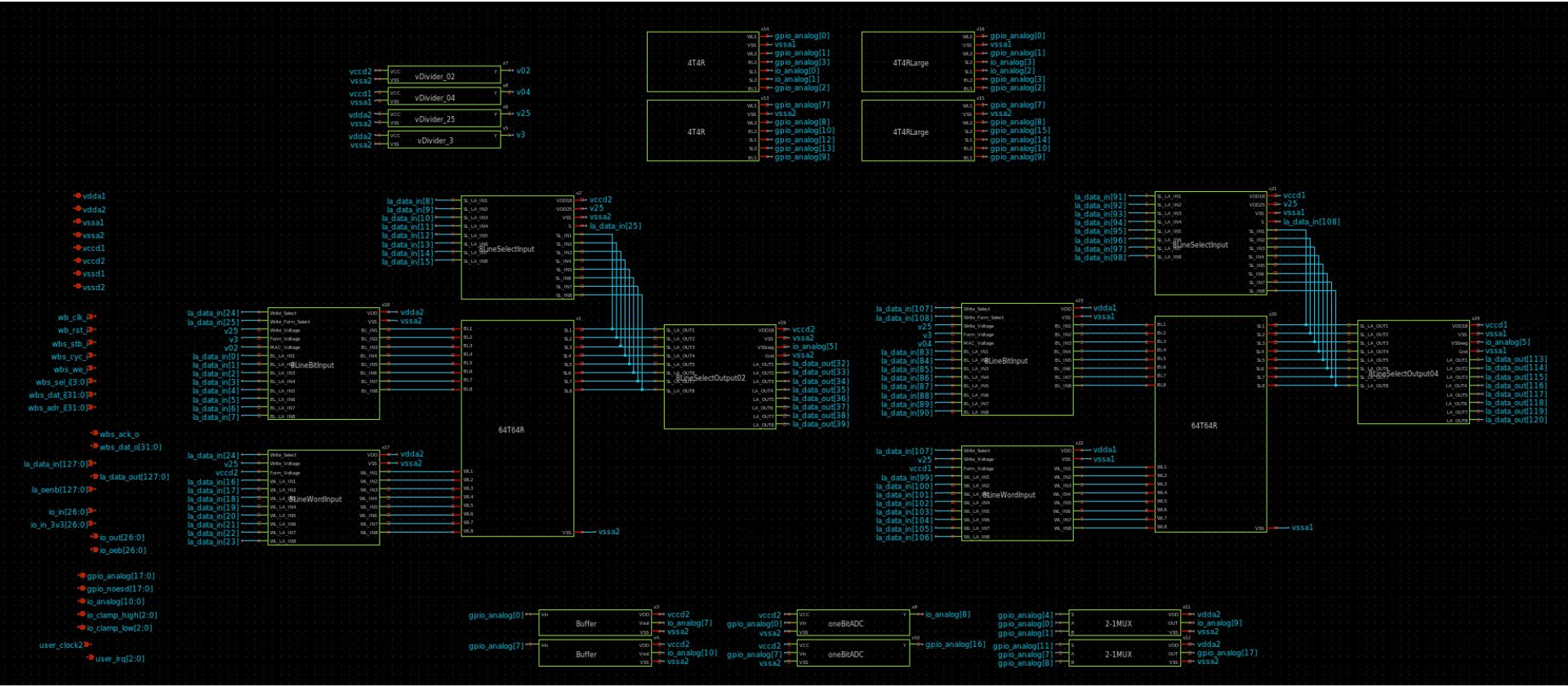
External circuitry hooked up to GPIOS

0.2 V Compute Crossbar

0.4 V Compute Crossbar

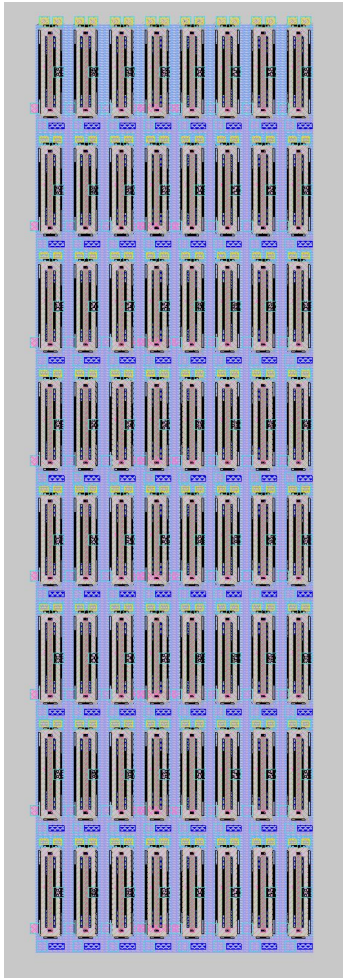


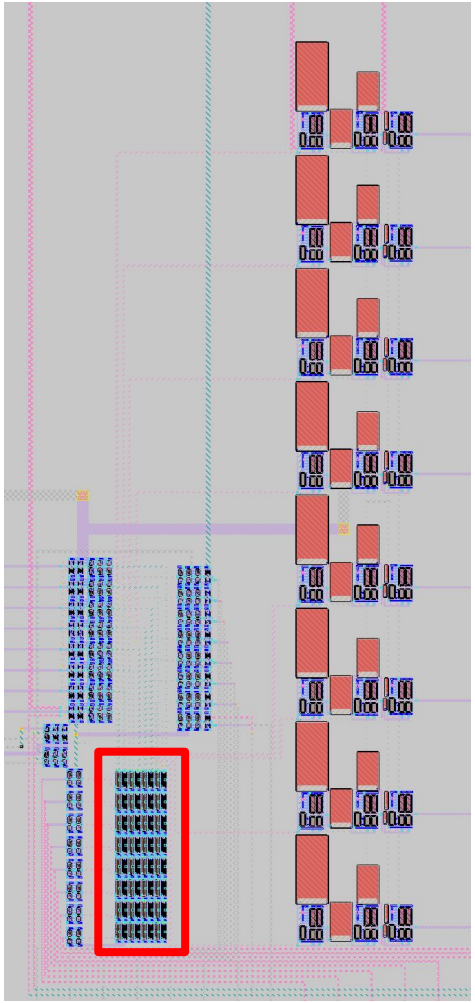
Schematic Level View of Wrapper



8x8 Crossbar

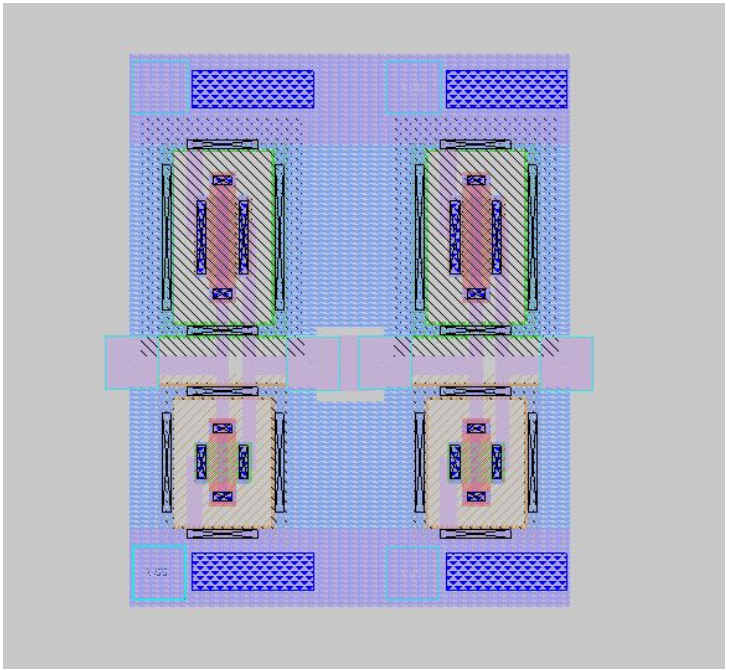
- ReRAM Cell
- NMOS Transistor

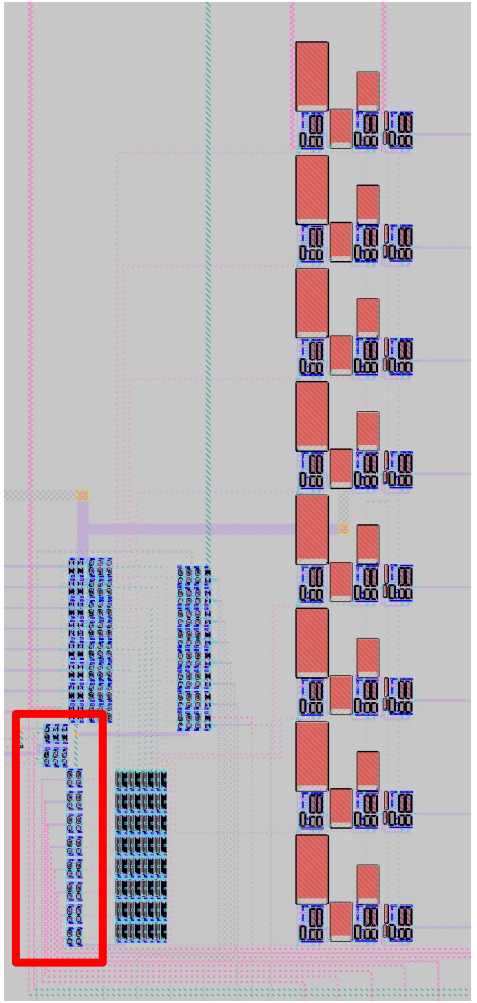




Word Line

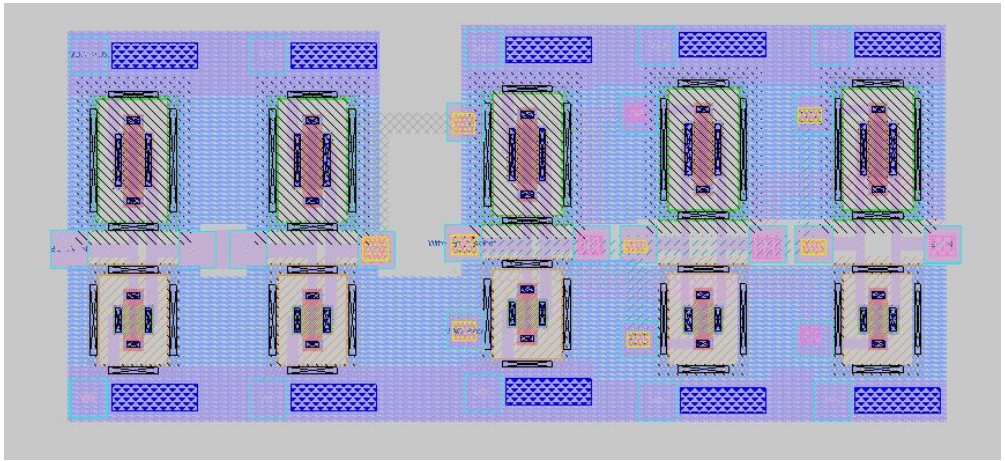
- DAC

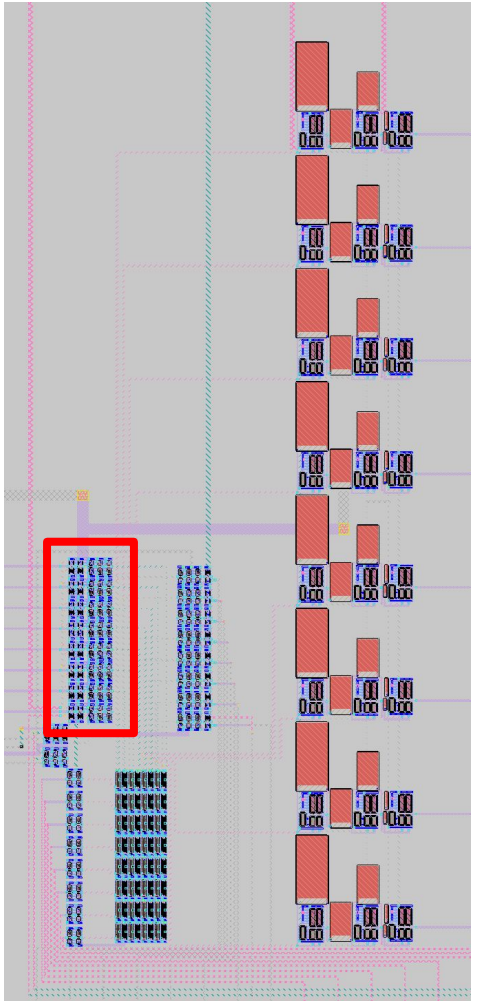




Bit Line

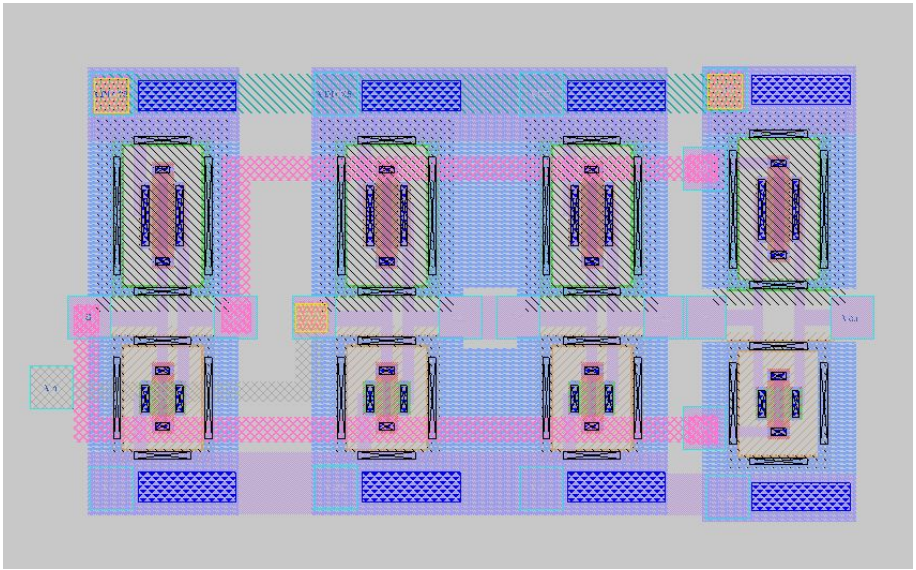
- DAC
- 2-1 MUX

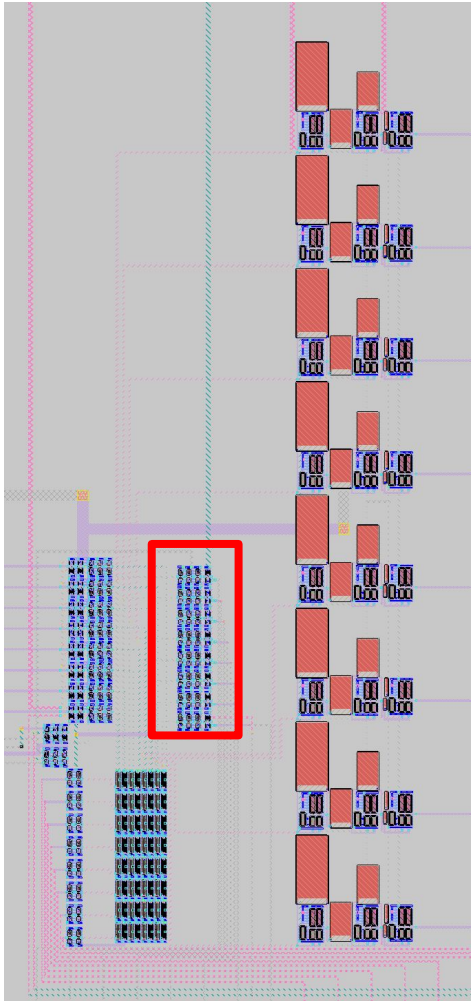




Select Line Input

- DAC
- Transmission Gate
  - With Inverter





Select Line Output

- Transimpedance Amplifier
- Inverting Amplifier
- ADC