

RERAM COMPUTE CROSSBAR FABRICATION

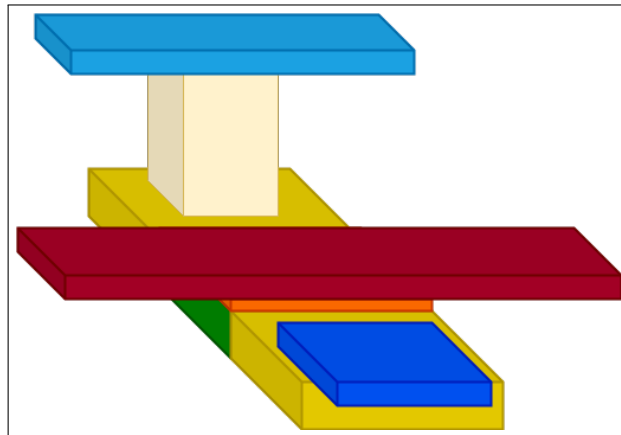
Authored by

Konnor Kivimagi, Jason Xie, Gage Moorman, and Nathan Cook

Advised by

Dr. Henry DUWE and Dr. Cheng WANG

January - Dec 2024



Team contact: sddec24-13@iastate.edu

Team website: <https://sddec24-13.sd.ece.iastate.edu/>

Executive Summary

Data is an invaluable resource for modern-day technology. Computers are constantly reading, writing, creating, and moving data around to perform complex operations. However, this can lead to something called the Von Neumann bottleneck, which occurs when the memory and CPU of a device are separate components. The CPU is often times much faster than the transfer speed of data between memory and the CPU causing a slowdown while the CPU is waiting to receive data. This bottleneck causes large drawbacks to performance as it limits the speed of operation.

Our design aims to help alleviate this bottleneck by storing data and performing computations all in the same spot. To do this, we are going to create a computational crossbar array using ReRAM (Resistive Random Access Memory) cells. This crossbar will allow us to perform matrix multiplication directly in the memory of the device by directly changing the voltages flowing across the cells. Along with alleviating the bottleneck this will also create a computation that is energy efficient.

Our design includes the use of an ADC (Analog-to-digital converter), a trans-impedance amplifier, the ReRAM crossbar, as well as a RISC-V (Reduced Instruction Set Computer) processor used to control the device. These components are designed using the open-source tools used by the Efabless sky130nm PDK (Process Design Kit).

There is a lot of potential for this design, one of the major use cases being the potential for use in the development of AI models. This is because these models often need to perform many computations a second, and if there is potential for them to be able to do so without being bottlenecked, they will be able to learn and provide answers much more quickly.

Learning Summary

Development Standards and Practices Used

- IEEE 1076.4-2000 - IEEE Standard VITAL ASIC Modeling Specification
 - Promotes the development of highly accurate, efficient simulation models for ASIC components in VHDL.
- IEEE 1481-2019 - IEEE Standard for Integrated Circuit (IC) Open Library Architecture(OLA)
 - Methods to analyze chip timing and power consistently across a broad set of electric design automation (EDA) applications
- IEEE 1149.4-2010 - IEEE Standard for a Mixed-Signal Test Bus
 - Defines a mixed-signal test bus architecture that provides the means of control and access to both analog and digital test signals
- IEEE 1364-2005 - IEEE Standard for Verilog Hardware Description Language
 - Defines verilog hardware description language
- IEEE 1666-2023 - IEEE standard for C++ language
 - Defines provide a C++ based standard for designers and architects who need to address complex systems that are a hybrid between hardware and software

Summary of Requirements

- Functional requirements
 - ReRAM array that can perform read, write and computation operations accurately
 - Design has low overall power consumption
 - DAC (Digital-to-Analog Converter) outputs correct voltage thresholds.
 - ADC with a sufficient resolution to distinguish between all possible MAC outputs
 - Characterization of peripheral circuits
- UI requirements
 - The ability to see what is happening on the inside for conducting research
 - Require IO ports for probe testing

Applicable Coursework from Iowa State University

- EE230 - Electronic Circuits and Systems
- EE330 - Integrated Electronics
- EE435 - Analog VLSI Circuit Design
- EE465 - Digital VLSI Circuit Design
- CprE 281 - Digital Logic

New Skills/Knowledge Acquired

- ReRAM technology
- Open-source design using Xschem, Magic VLSI, and Ngspice
- Skywater 130nm process
- ASIC (Application Specific Integrated Circuit) design

Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Users and User Needs	1
2	Requirements, Constraints, and Standards	2
2.1	Requirements and Constraints	2
2.2	Engineering Standards	2
3	Project Plan	3
3.1	Project Management/Tracking Procedure	3
3.2	Task Decomposition	3
3.3	Project Proposed Milestones, Metrics, and Evaluation Criteria	3
3.4	Project Timeline/Schedule	4
3.5	Risks and Risk Management/Mitigation	4
3.6	Personnel Effort Requirements	5
3.7	Other Resource Requirements	6
4	Design	6
4.1	Design Context	6
4.1.1	Broader Context	6
4.1.2	Prior Work/Solutions	8
4.1.3	Technical Complexity	8
4.2	Design Exploration	9
4.2.1	Design Decisions	9
4.2.2	Ideation	9
4.2.3	Decision-Making and Trade-Off	11
4.3	Proposed Design	11
4.3.1	Overview	11
4.3.2	Detailed Design and Visuals	12
4.3.3	Functionality	18
4.3.4	Areas of Challenge	19
4.4	Technology Considerations	20
5	Testing	21
5.1	Unit Testing	21
5.2	Interface Testing	23
5.3	Integration Testing	23
5.4	System Testing	23

5.5	Regression Testing	24
5.6	Acceptance Testing	24
5.7	Results	24
5.7.1	Crossbar Testing	24
5.7.2	Comparator Testing	25
5.7.3	TIA Testing	26
5.7.4	ADC Testing	28
5.7.5	ADC coupled with TIA Results	29
6	Implementation	30
6.1	Design Analysis	30
7	Professional Responsibility	31
7.1	Areas of Responsibility	31
7.2	Project Specific Professional Responsibility Areas	32
7.3	Most Applicable Professional Responsibility Area	33
8	Conclusions	33
8.1	Summary Of Progress	33
8.2	Value Provided	34
8.3	Next Steps	34
9	References	34
10	Appendices	35
10.1	Operation Manuals	35
10.2	Previous Versions	35
10.2.1	TIA versions	35
10.2.2	ReRAM versions	35
10.3	Other Considerations	35
10.4	Code	36
10.5	Team	36
10.5.1	Team Members	36
10.5.2	Required Skill Sets	37
10.5.3	Skill Sets	37
10.5.4	Project Management Style	37
10.5.5	Initial Project Management Roles	37
10.5.6	Team Contract	37

List of Tables

1	Phase 1: Research and Setup	5
2	Phase 2 and 3: Component Design and Verification	5
3	Phase 4: Integration	5
4	System Verification	6
5	Phase 5: Finalise	6
6	Broader Context	7
7	ADC architecture comparisons	11
8	ADC Requirements	17
9	Areas of responsibility	32
10	Required Skills	37

List of Figures

1	Task decomposition	3
2	Project Timeline	4
3	Example of Flash ADC circuit from [1]	10
4	Example of SAR ADC circuit from [1]	10
5	General Diagram for the overall design	12
6	Diagram of crossbar configuration	13
7	Diagram of 1T1R memristor	14
8	Example of vector-matrix computation	14
9	ReRAM set operation	14
10	ReRAM reset operation	15
11	ReRAM MAC operation	15
12	ADC Layout	16
13	Comparator Schematic	17
14	TIA Schematic	18
15	Illustration of memristor forming process	19
16	Testbench for 2x2 ReRAM crossbar	25
17	Comparator Slew rate	25
18	Testbench for TIA	26
19	Open loop gain of TIA	26
20	Open Loop Phase margin of TIA	27
21	4-bit ADC Results	28
22	ADC Results cont.	28

23	Combined TIA and ADC results	29
24	Testbench for 4bit ADC	29
25	Testbench for TIA and ADC	30

1 Introduction

1.1 Problem Statement

Resistive random access memory (ReRAM) is an emerging, non-volatile, low-power memory technology. ReRAM has a significant appeal due to its scalability at smaller processes and a much higher capability for memory density.

This project aims to create a research vehicle for silicon, proving computational ReRAM for research and education purposes. ReRAM allows for computation within the memory, eliminating the need to send data to and from a digital ALU (Arithmetic logic unit).

The current issues this project faces are a need for produced ReRAM chips, limited opportunities to get chips fabricated, and a test vehicle to gain an understanding of and better test the physical characteristics of ReRAM primitives in Skywater technology.

ReRAM technology is primarily used as a memory device. Therefore, understanding how it interacts with peripheral circuits while being implemented for computation is essential for our project and will be one of the goals of this project. Due to ReRAM being an emerging technology, there are multiple questions that we would like to answer. One major question is whether or not using ReRAM for computation is better than existing memory technologies. A few ways to measure this could be to compare computation accuracy, energy efficiency, and power consumption between ReRAM and a similar technology such as SRAM (Static Random Access Memory) or DRAM (Dynamic Random Access Memory).

1.2 Users and User Needs

Professor Duwe and Professor Wang:

Professor Duwe and Prof. Wang are the primary users of this project. They are also the clients of this particular project. The results of our project would provide them with a practical example of a ReRAM compute crossbar and knowledge of how the Efabless process can be used to design and fabricate chips in the Skywater 130nm process.

Graduate students:

These users would be the graduate students that are under Professor Duwe and Professor Wang. These students would need documentation on the Efabless process and tool flow to apply that knowledge to their projects. This documentation would provide them with the benefit of easily being able to learn the process and allowing them to spend time on their research.

Undergrad students:

These students will be able to bring up our chip and others fabricated under Duwe as a part of their undergraduate curriculum. These students will need easy-to-follow lab documentation similar to EE 201 and 230. The benefit these students receive is the ability to bring up and test a physical chip in parallel with their regular course work, providing them valuable experience and real-world experience of the concepts they are learning.

Future senior design teams:

An important user group are the senior design teams that follow our project. They will need to have access to our designs as well as an understanding of how we designed our circuit. This will help them to get a grasp on the design much quicker than we were able to, allowing them to start thinking about improvements to the design sooner.

2 Requirements, Constraints, and Standards

2.1 Requirements and Constraints

- Functional requirements:
 1. ReRAM array that can perform read, write and computation operations accurately
 2. Design has a low overall power consumption
 3. DAC outputs correct voltage thresholds.
 4. ADC with a higher than 1-bit resolution
 5. Low-power trans-impedance amplifier
- Deliverables:
 1. Git repo with all project files
 2. Project documentation
 - Bring up documentation
 - Lab walkthrough documents
 - Probe testing documentation
 - Top-level circuit diagram
 - Supporting documentation for any common problems with tools
 3. Simulations showing correct device operations
- User requirements:
 1. Bring up documentation for success/failure
- Precheck approved GS2:
 - All subcircuits need to pass DRC (Design Rule Check) and LVS (Layout Versus Schematic)
 - The final design must be integrated with Caravel Harness and Pass LVS
- UI requirements:
 - The ability to see what is happening on the inside for conducting research
 - Require IO ports for probe testing

2.2 Engineering Standards

- IEEE 1076.4-2000 - IEEE Standard VITAL ASIC Modeling Specification
 - Promotes the development of highly accurate, efficient simulation models for ASIC (Application-Specific Integrated Circuit) components in VHDL.
- IEEE 1481-2019 - IEEE Standard for Integrated Circuit (IC) Open Library Architecture(OLA)
 - Methods to analyze chip timing and power consistently across a broad set of electric design automation (EDA) applications
- IEEE 1149.4-2010 - IEEE Standard for a Mixed-Signal Test Bus
 - Defines a mixed-signal test bus architecture that provides the means of control and access to both analog and digital test signals
- IEEE 1364-2005 - IEEE Standard for Verilog Hardware Description Language
 - Defines the Verilog hardware description language

3 Project Plan

3.1 Project Management/Tracking Procedure

The project management style that we will use is AGILE. This methodology was chosen due to its iterative nature, allowing us to constantly check and verify that we have completed our goals before we continue to the next step. The tracking will be done through

3.2 Task Decomposition

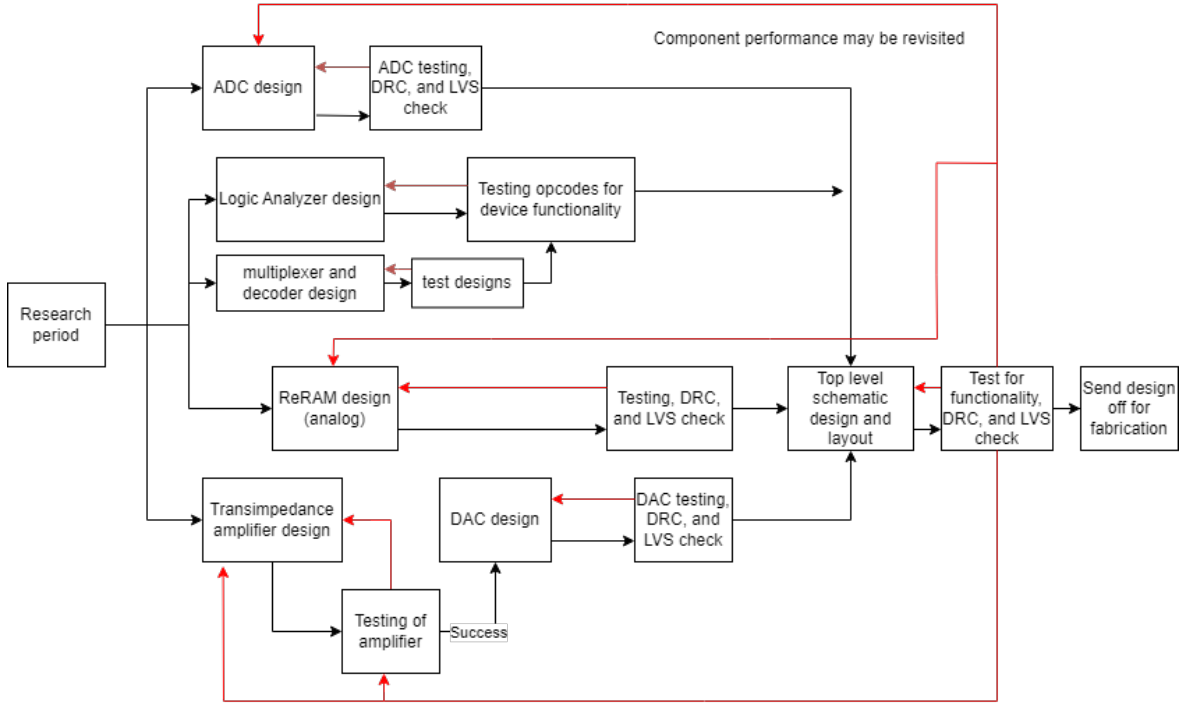


Figure 1: Task decomposition

Analog design will be split between Gage, Jason, and Konnor. Jason and Nathan will work on digital design. Everyone will work on the documentation.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Phase 1: Analog Tool Setup and ReRAM research
 - Verify that we can run all the tools needed for this project
- Phase 2: Project and component research
 - Research applicable ADC, DAC, and TIA (Trans-Impedance Amplifier) architectures
 - Develop a better understanding of ReRAM architectures
 - Determine the best interface methodologies and devices
- Phase 3: Component design and verification
 - All components must pass both LVS and DRC checks to be verified

- Schematics created in Xschem and OpenLane
- Layouts created and compiled in Magic VLSI
- Phase 4: Integration of all components
 - Integrate all components into one top-level design and create a schematic and layout for this design
 - The schematic and layout must pass LVS
 - Simulate the system rigorously to verify that the integrated design is operating as expected
- Phase 5: Finalise documentation
 - Create design documents
 - Create a bring-up plan for the ReRAM

3.4 Project Timeline/Schedule

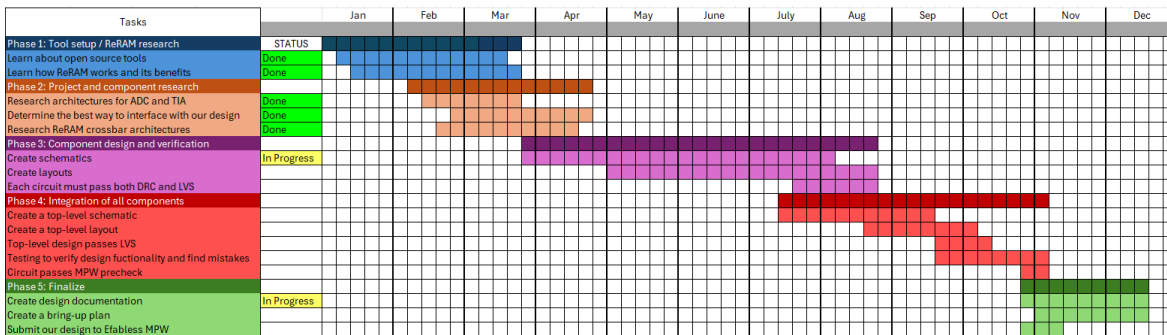


Figure 2: Project Timeline

3.5 Risks and Risk Management/Mitigation

In concerns of design and verification, peripheral circuits may not get done on time. We can mitigate this risk by planning well and designing circuits in other programs, such as cadence. We could also reduce the complexity of designs and reduce time to completion.

There is a risk concerning the digital functionality and performance of the memristor as the model provided by the PDK is currently inaccurate. It has been shown to simulate behavioral errors. Similarly, in the analog models of the CMOS transistors provided by the PDK, there is an inconsistency in simulation results during weak and moderate inversion conditions. A solution for both cases, albeit laborious and implausible considering the time constraints, we can attempt to resolve the model issues by either contributing to the open-source spice model or developing an entirely new model ourselves. A more likely solution would be to create sound documentation of the current issues of the model and derive plausible solutions for future teams.

When considering the fabrication process, there arises a risk in which the design constraints are not met due to fabrication errors. While preventing fabrication errors is beyond our sphere of influence, we can mitigate this risk by providing a sound bring-up plan with intricate specifications concerning testing methods and expected results.

3.6 Personnel Effort Requirements

Phase 1 Research and Setup		
Task	Time	Component
ReRAM and crossbar research	30 Hrs	Research the essential operation of ReRAM as well as how it is implemented as a crossbar for computation
Analog tool flow setup	20 Hrs	Setup of the essential tools we will need for design. These include Xschem, magic, and Ngspice
Consideration of how to design peripheral analog circuits	15 Hrs	Discuss the different structures and routes to take in our design process

Table 1: Phase 1: Research and Setup

Phase 2 and Phase 3 Component Design and Verification		
Task	Time	Component
Schematic creation	5 Hrs / Component	Create a schematic for each component in Xschem
Simulation and debugging of schematic	10 Hrs / Component	Run simulations with Ngspice on the schematics and make changes as necessary to achieve the desired functionality
Layout creation	10 Hrs / Component	Create a layout for the schematic in magic
Verification of component	10 Hrs / Component	Verify that the layout passes DRC and that the element passes LVS

Table 2: Phase 2 and 3: Component Design and Verification

Phase 4 Integration		
Task	Time	Component
Top-level schematic	25 Hrs	Implement a top-level schematic with all of the components
Top-level layout	30 Hrs	Implement the layout of the top-level schematic and verify that it passes LVS.

Table 3: Phase 4: Integration

System Verification		
Task	Time	Component
Top-level schematic	25 Hrs	Implement a top-level schematic with all of the components
Top-level layout	30 Hrs	Implement the layout of the top-level schematic and verify that it passes LVS.

Table 4: System Verification

Phase 5 Finalise		
Task	Time	Component
Create documentation	20 Hrs	Create documentation for our project detailing our process. As well as improving upon existing tools/software documentation
Create a detailed bring-up plan	35 Hrs	Document how our device should be tested to prove functionality and discover any discrepancies between simulations and testing of the fabricated chip

Table 5: Phase 5: Finalise

3.7 Other Resource Requirements

This project was completed using open-source tools available to the general public. Due to the nature of these tools being open-source, there is limited documentation and resources available online. Our most valuable resources were the open-source hardware slack channel and the documentation on the design flow from previous teams.

4 Design

4.1 Design Context

4.1.1 Broader Context

ReRAM is a memory technology that has the potential to be able to perform computations in the analog domain. This can be beneficial as it minimizes data movement and, in turn, requires less power consumption than traditional digital computations. We aim to create a ReRAM crossbar to be fabricated through Efabless so that future students can bring up and test our design’s functionality. Relevant considerations for multiple areas are listed in Table 6.

Area	Description	Examples
Public health, safety, and welfare	Our product does not have many direct impacts on public health, safety, and welfare. However, it is a possibility that a working implementation of our design could be helpful in many applications in the future	This technology brings with it the potential to create a new job market as breakthroughs in this could lead to a drive for companies each to have the best version of it
Global, cultural, and social	This is an open-source project so that we will adhere to the values of transparency, collaboration, and continual growth	This project will be fabricated through Efabless and must be submitted as a project to the open-source repository. This means that it could provide inspiration to others going forward
Environmental	The chip fabrication process is generally toxic and can produce harmful waste chemicals. However, our design will only be fabricated once. ReRAM also has the potential to draw less power than its digital counterparts	Our project will implement a ReRAM crossbar that will hopefully allow for computations to be done in a more power-efficient way
Economic	Our project is being designed through open-source tools which are free to use	Our design will be added to the open-source documentation and will be free to view and use by anyone in the future

Table 6: Broader Context

4.1.2 Prior Work/Solutions

Existing references and designs of ReRAM are very sparse and often impossible to find. ReRAM is a developing technology, and many details are not readily available to the general public. However, we are using the open source forums through the company Efabless, where there has been some attempt at designing a ReRAM module. While it is not much, we have small stepping stones to go off of between this and the previous group's findings.

4.1.3 Technical Complexity

Our design is a complex mix of both digital and analog components. These components include:

- Design of a ReRAM module:
 - Requires research into how this memory technology works
 - Requires testing and design of multiple array sizes for a better understanding
- Design of a digital logic analyzer
 - Requires knowledge of Verilog and how to implement logic algorithms
- Design of conversion circuits
 - Must be designed for a specific resolution goal
 - Requires design of encoders and decoders
 - Requires knowledge of ADC architectures and which best fits our project
- Use of open-source tools
 - Requires finding and reading documentation to troubleshoot setup issues
 - These tools are much less polished than the tools used by large companies and, as such, are unintuitive at times
- Design of a TIA
 - We'll need to determine the ReRAM current ranges for input and understand how to design around them.
- Mixed-signal component integration
 - Because the design flows of digital and analog circuits are bit different in our environment, we need to determine a method of developing an environment for mixed-signal tests
- Interdigitization and common centroid approach to analog circuit layouts
 - This approach requires complex interconnections between a rather significant number of devices
 - The inclusion of dummy devices and parallel connections is unintuitive in terms of LVS and precheck in the design flow

4.2 Design Exploration

4.2.1 Design Decisions

In determining our project’s aspects, we determined the peripheral circuitry based on the ReRAM operative parameters. The peripheral circuitry serves as the interface between external accessibility and the operation of the ReRAM. At the highest description level, we first have the DAC, which allows us to control the system through external digital excitation. Conversely, we also require an ADC, which will enable us to realize the resulting analog output of the ReRAM. These well-studied circuits have plenty of references to allow us to design a system that satisfies our defined constraints.

However, once the ReRAM has processed the input, it returns a current that cannot be easily translated into a digital output, as existing ADC architectures are primarily designed to take voltage inputs. To remedy this, we will need a TIA that converts a current into a voltage at a standard reference. Lastly, we also require miscellaneous circuitry that allows us to assume as much control over the ReRAM Crossbar as possible. We will use encoders and multiplexers to enable the ability to decipher and administer the previously mentioned external digital commands to the voltage controls on the ReRAM crossbar.

Overall, design decisions for each of these peripheral circuits will be based on the design constraints established by the project and parameter expectations from the desired ReRAM Crossbar architecture.

4.2.2 Ideation

As an example of our ideation process for the circuits, as mentioned above in Section 4.1.1 [Broader Context](#), we will cover the ADC. There are many pre-existing architectures we considered when designing our ADC. We started by referencing designs from a course textbook we were familiar with, leading us to discover the Flash, Pipelined, and SAR (Successive Approximation Converter) ADC. The simplest of these, the Flash ADC, is a converter based on a parallel resistor and comparator string. Of the three architectures, the Flash ADC has the fastest conversion speed and is generally the easiest to implement. However, Flash ADC calls for an exponentially increasing amount of comparators to the number of bits. Not only does this aspect cause a more significant demand amount of area, but it also means the circuit becomes exponentially more power-hungry. Furthermore, due to the number of comparators on the input, there is a large number of parasitic capacitances, which could deter the speed of the ADC and require another power-hungry buffer to drive the input. An illustration of a general 3-bit flash ADC is shown in Figure 3. [1]

On the other hand, instead of a single conversion stage, the Pipelined ADC implements several circuits that algorithmically determine outputs via the simultaneous conversion of successive inputs. This is usually achieved through a series connection between an $N/2$ -bit ADC and $N/2$ -bit DAC, which results in an N -bit output resolution. If we use this system with a Flash ADC, we have effectively halved the area and power consumption in proportion to increasing the resolution while maintaining a relatively high throughput. Furthermore, because of the circuit’s modularity and redundancy, multiple stages can be resolved in each stage, where any offset in the comparators can be rectified through digital correction. However, this digital correction cannot determine the nonlinearities in the DAC and inaccuracies in the interstage gain, which limits the overall performance of the ADC. Moreover, due to its algorithmic nature, Pipelined architectures typically require an input clock at the minimum, which also introduces latency issues. [1]

Finally, the SAR ADC is another algorithmic-based converter that relies on some digital control logic to perform a binary search on a series input. The architecture, in general, can be relatively simplistic, containing a DAC, a digital control logic circuit, and a comparator. Furthermore, because the architecture is iterative and algorithmic, the SAR ADC scales linearly with resolution per the DAC specifications. This means much higher resolution and accuracy can be achieved with less power and space. Conversely, because of the iterative nature, this architecture has a relatively low throughput compared to the Flash and Pipelined ADC architectures. A general SAR ADC circuit is shown in Figure 4. [1]

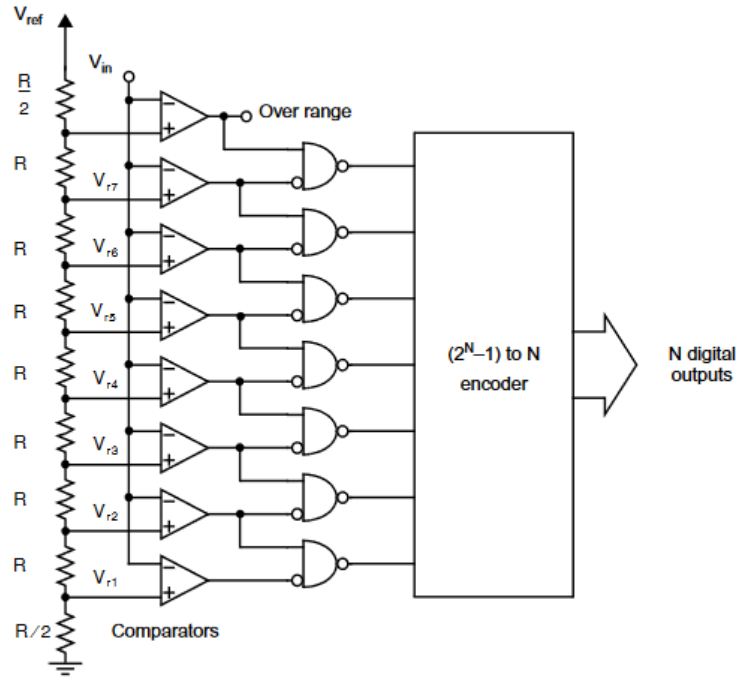


Figure 3: Example of Flash ADC circuit from [1]

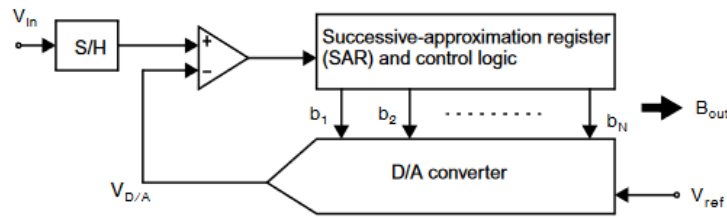


Figure 4: Example of SAR ADC circuit from [1]

Aside from the three architectures taken from our textbook, we also considered some ADCs used in research papers on ReRAM Crossbar. One example that stood out to us was the TDC (Time to Digital Converts). This architecture was proposed as time-domain data conversions are known to be more energy efficient compared to the traditional voltage to the digital domain. The conversion was done by establishing signal codes through the delay from an established point. This means the system can be implemented as a fully digital flow and remove the need for some peripheral circuits like the TIA entirely. Unfortunately, the typical TDC circuit is relatively complex and would require us to understand concepts beyond our curriculum and current experience.

Furthermore, implementing a TDC would require operating everything in a different domain. This would lead us to stray from the conventional control logic for the ReRAM crossbar and design and incorporate other peripheral circuits like a DTC (Digital to Time Converter). In other words, while the TDC is a compelling option to satisfy our constraints, it was undoubtedly our least considered option due to the difficulty of design and implementation. [2]

4.2.3 Decision-Making and Trade-Off

Referring to the design space exploration done in the previous section, we can derive a simple comparison in the following table.

As we have mentioned in our constraints, the goal of our overall system is to be low-powered, area-efficient, at a medium resolution (about 3-4 bits), and have a relatively fast throughput (about 10 MHz). When we evaluated our analog design experience as a team, we found that we may only be capable of relatively simple circuits with absolute success. For these reasons, we ruled out the SAR and TDC ADCs due to their complexity and inexperience with the design process. We then decided to focus on the Flash ADC despite its lackluster characteristics compared to the Pipelined ADC. This was mainly due to our unfamiliarity with the open-source tools needed to develop our circuit. Another reason was that we would have needed to create a Flash ADC to design a Pipelined ADC in the first place. Therefore, we concluded that first designing a Flash ADC would satisfy our design requirements at this stage of the development process.

ADC Architecture	Positives	Negatives
Flash	+ Highest throughput + Simplest design conceptually	- High input capacitance - Exponentially increasing power consumption and area requirement with system resolution
Pipelined	+ High throughput + Digital correction	- Require a clocked input - Latency between stages
SAR	+ Very accurate + Low power consumption and area requirement	- Slower throughput - Relatively challenging to implement
TDC	+ Incredibly energy and area-efficient + Immune from input signal noise + Can be implemented in a fully digital application	- Relatively slow throughput - Very difficult to implement - Implementation would require changing several other circuits

Table 7: ADC architecture comparisons

4.3 Proposed Design

4.3.1 Overview

In general, and at the highest level of abstraction, our final circuit should be able to take a digital instruction, perform an analog computational or storage operation, and provide a digital output. Specifically, this is done by feeding a digital value that will be converted to an analog voltage by the DAC. These voltages run through the crossbar and the individual ReRAM cells creating a current. The crossbar, controlled by the row and column select modules, may then manipulate these currents such that vector computation or a storage operation could be performed. The specifics as to how these operations are realized are described in Section 4.3.2 [Detailed Design and Visuals](#), detailed design and visuals. Unfortunately, there isn't a trivial way of reading these output currents. Therefore, they'll have to be converted back into a relative voltage through a TIA. These voltages can then be passed into an ADC which converts it back into a digital voltage, allowing it to be read by a logic analyzer. Figure 5 shows a top-level block diagram of our design.

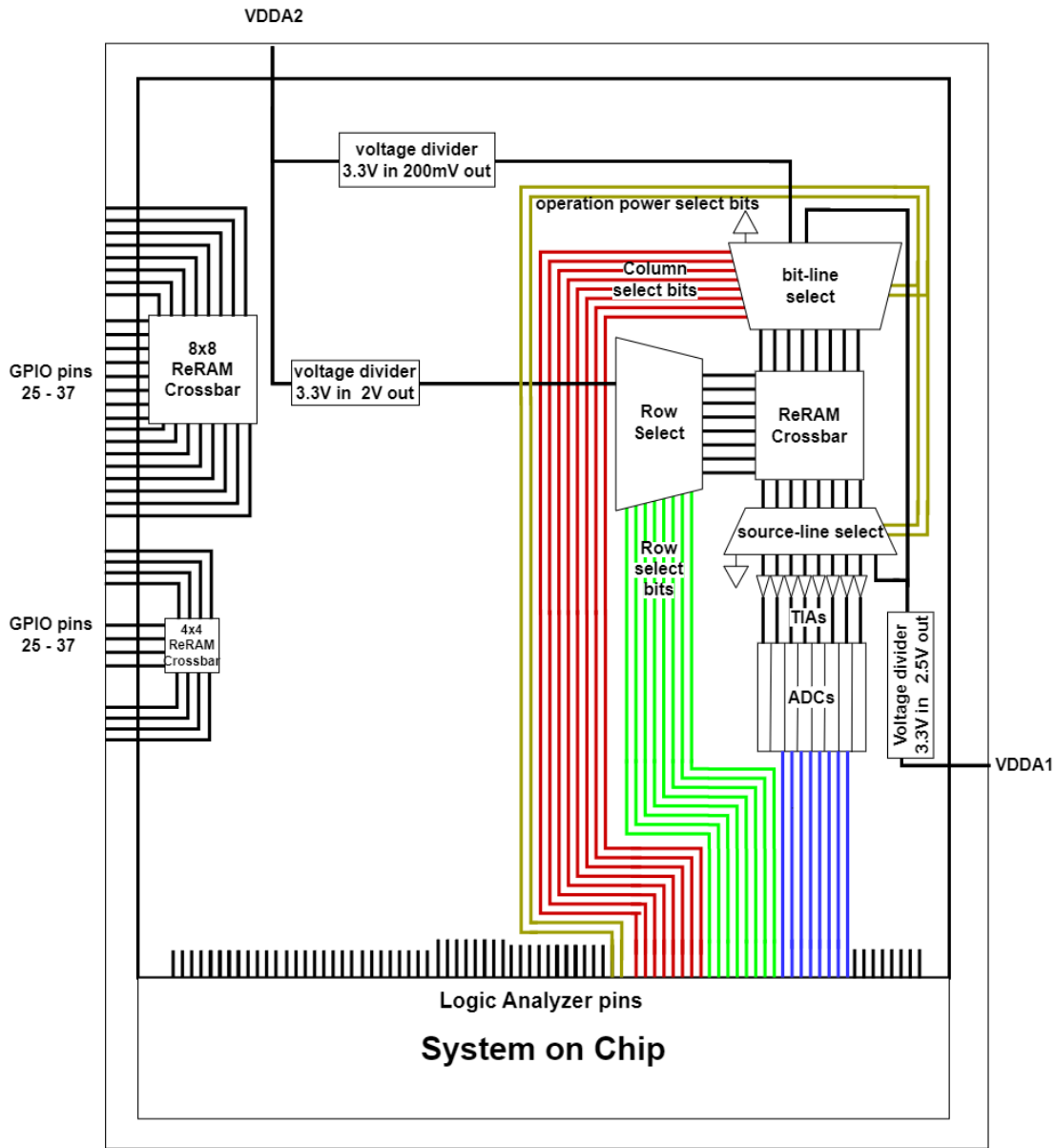


Figure 5: General Diagram for the overall design

4.3.2 Detailed Design and Visuals

ReRAM

An individual ReRAM cell is shown in Figure 7. Our crossbar will consist of an eight-by-eight array of these cells, as described previously in the overview. It can be seen that each cell contains both a resistor and a transistor, hence why this architecture is referred to as a 1T1R (one transistor one resistor) cell. The cell's Word Line (WL) will be connected to a logic analyzer that determines whether the cell should be on or off and powers it accordingly. The Bit line (BL) is connected to the DAC and is the voltage passed into the cell. The Source Line (SL) is where the current will accumulate and be output to the TIA.

We will use these cells to perform a MAC operation. This operation is illustrated in Figure 8,

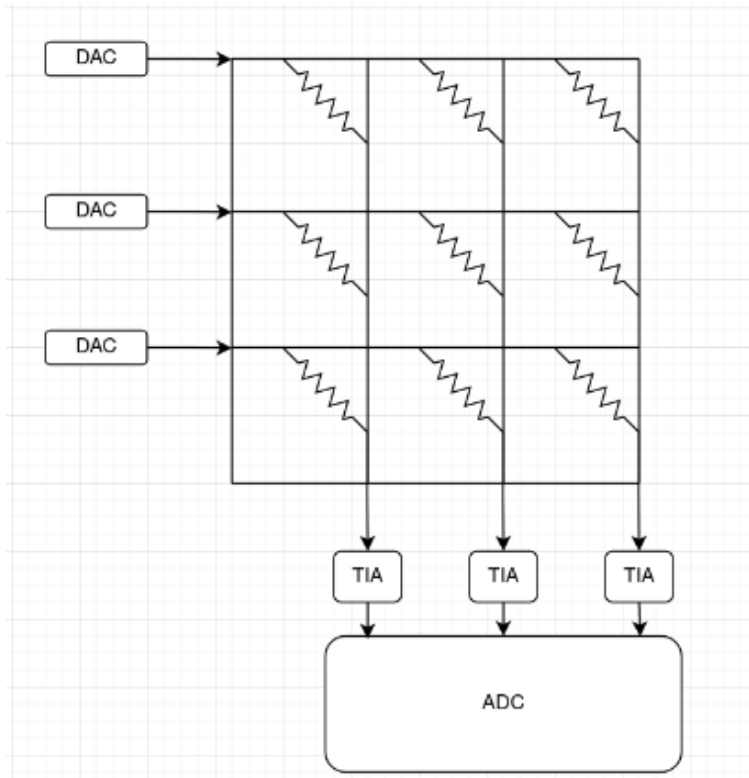


Figure 6: Diagram of crossbar configuration

where the input voltages, V_1 , V_2 , and V_3 , are multiplied by the conductance's, G_1 , G_2 , and G_3 of the ReRAM, and the resulting currents are then summed down along the column to create one output.

The conductance of a ReRAM cell depends on whether the cell is in an LRS (low resistance state) or an HRS (high resistance state). A LRS is viewed as a "digital 1" and a HRS as a "digital 0". All ReRAM cells start in a pristine state and must initially form into a LRS. After the initial form, the cell can be set and reset to switch between a digital 1 or 0. This process is illustrated in Figure 15.

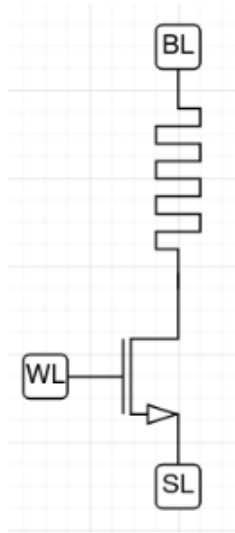


Figure 7: Diagram of 1T1R memristor

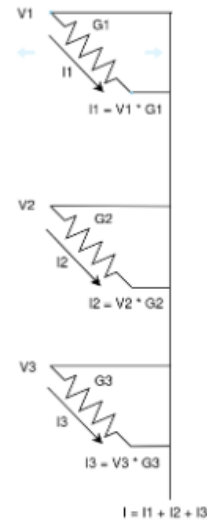


Figure 8: Example of vector-matrix computation

There are a couple of important operations that we need to be able to perform on the ReRAM crossbar. These are the read, write, and MAC. Figures 9 and 10 depict the two different write options, set and reset, which are being applied to the top left cell of the 2x2 matrix. Figure 9 shows the set operation where a positive voltage is applied across the memristor by applying 2.5V at the bitline and grounding the source line. Figure 10 shows the reset action where a negative voltage is applied by sending 2.5V from the source line while the bitline is grounded. The red arrows depict the voltage direction and the blue numbers are the equivalent digital logic that the cell is being set to.

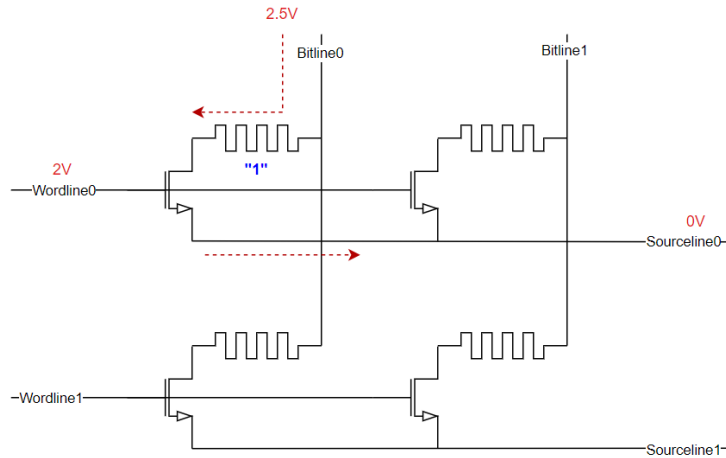


Figure 9: ReRAM set operation

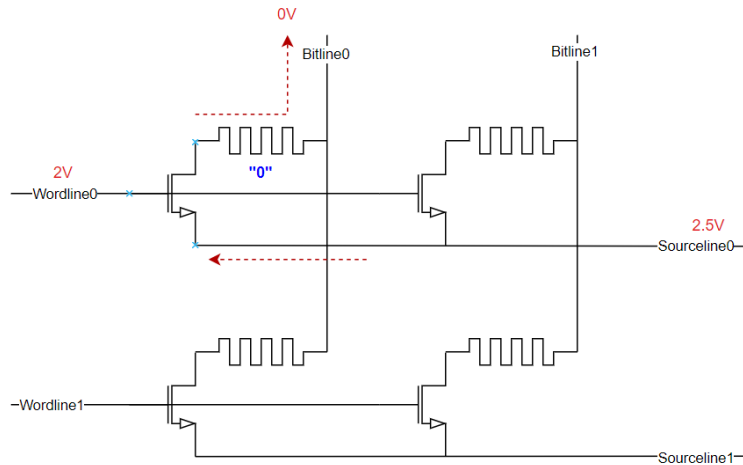


Figure 10: ReRAM reset operation

The two other operations are the read and MAC. The read is the same as the MAC operation but only on a single cell. Figure 11 depicts the MAC operation on the same crossbar as the write operations. Unlike the read and write operations that are performed on a singular cell, the MAC operation is performed on all cells simultaneously. This means that every wordline is turned on. The bitlines are where we apply our vector. In Figure 11 this vector is a 1 by 2 vector where both values are 0.2V. These voltages multiply with the respective currents of cells in their columns and these currents add horizontally along the sourceline resulting in a vector of currents. As seen in Figure 11 a digital one produces around 18uA of current and a zero produces 0uA.

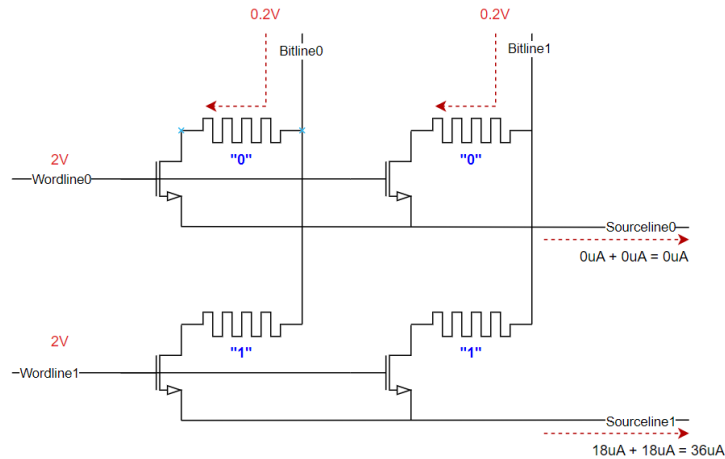


Figure 11: ReRAM MAC operation

System on Chip

All of the operations that the circuit will be performing will be decided by the System on Chip (SoC). The chip will be flashed with code; the Caravel project provides a framework for writing and simulating the code. From this code, a test can be created, verified to be working, and flashed onto the chip, which will run the tests and relay the results back to the onboard memory or back to the device connected to the chip. To do this we are using the logic analyzer to send control signals to decide which operation will be performed and on what cell or column. In the code, the logic analyzer pins being used must be enabled, and then you can begin writing to them, which is relatively simple,

all that must happen is a bit field can be set, which represents the data on certain pins of the logic analyzer. There are things that can be done to better secure the data being sent like disabling the logic analyzer pins after the data is sent to avoid sending more data and overwriting the data being processed. We chose to do this and use a wait function that was provided by a small logic analyzer specific library. The full dataflow of the circuit is data is sent out to the circuit, the logic analyzer pins used are then turned off and the pins that will be receiving the output will be activated so they can collect the result and send it back to the users device.

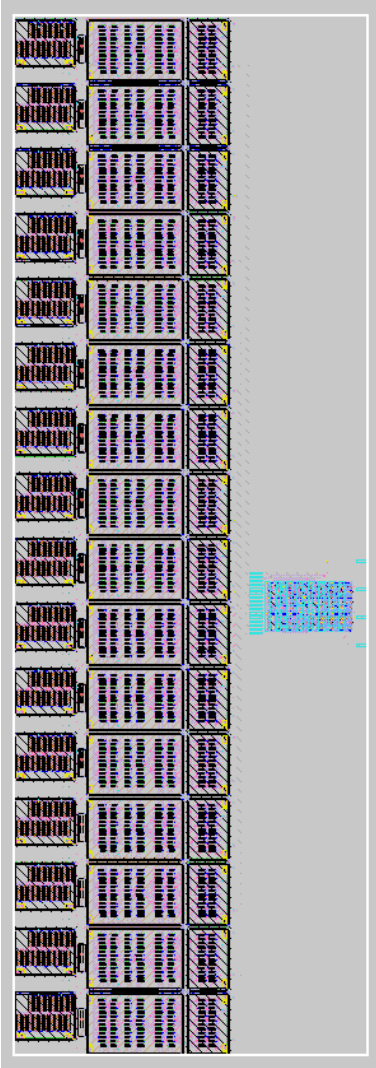


Figure 12: ADC Layout

ADC

In order to read the MAC results, the analog current needs to be converted to a digital voltage. To do this, an ADC must be used.

For our design, a flash ADC was selected, the flash ADC was chosen because of its high speed, simplicity, and low-resolution specification we were given. Before beginning the design, the requirements were analysed carefully and taken into consideration when deciding on the subcircuits that make up the flash ADC. The general flash ADC diagram can be found in Figure 3. A layout of the ADC can

Resolution	4 bits
Speed	40 MHz
Power	No power constraints were given, but we still took it into consideration

Table 8: ADC Requirements

be found in Figure 12. In particular, from left to right, the image illustrates the strongARM latch comparator preamp, the comparator itself, and the priority encoder.

To meet the resolution requirement we need 2^n Comparators and $2^n + 1$ resistors. The goal for choosing the resistance value of the resistors is to have a high resistance value that allows the gates of the comparators to charge in time. Following the equation $I = C * (DV/Dt)$ where C is the gate capacitance of the comparators and parasitics. If the resistors are too large, then there is not enough current to charge the inputs of the comparators, and this will considerably reduce the speed of your ADC. If the R-value is too small then there will be unnecessary power consumption and the resistors will take up a lot of area.

Comparator

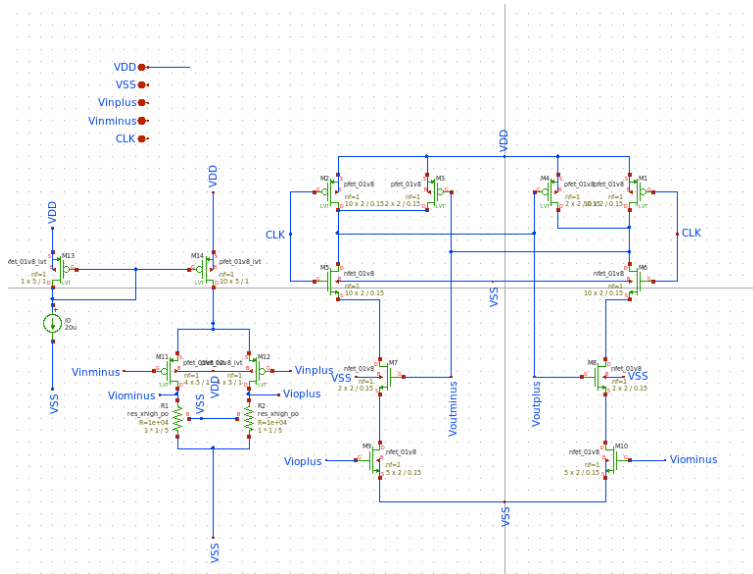


Figure 13: Comparator Schematic

The strongARM latch in Figure 13 was chosen over an open loop Op-Amp (Operational Amplifier) due to its reduced static power consumption and rail-to-rail output swing. These traits are desirable because of their reduction in power consumption and increased slew rate. To size the transistors a heuristic approach was taken because of the architectures inherent non-linearity from the positive feedback configuration. The process was streamlined as an analysis of the circuit was performed, resulting in the need for fewer iterations. We were required to have an operating bandwidth of 40 MHz. To solve for this, an initial guess of $251\mu A$ was chosen. The amount of current increases the speed $I = C(dV/dt)$ where C is the capacitance seen at the output, and the slew rate dV/dt is the requirement that must be met. Transistors M2:M1 and M5:M8 in Figure 13 are used as switches to precharge the output, removing any hysteresis and allowing the differential pair (M9:M10) to remain in saturation as long as possible, allowing the inputs to be amplified. A preamplifier was needed in order to reduce loading on the inputs due to switching and to allow ample headroom when making the comparisons between

Vin- and Vin+.

TIA

In order to read the MAC results the current from the crossbar needs to be converted into a voltage so the ADC can convert it. In order for the TIA to be effective it must meet our bandwidth requirement as well as have a high gain to reduce noise and increase accuracy when negative feedback is applied. Initially, a common source (CS) amplifier with a current mirror load was selected to be the TIA but was found to not have sufficient gain. The next iteration of the TIA is a Folded Cascode Amplifier depicted in Figure 14 with PMOS inputs and a CS amplifier (NMOS inputs) with a current mirror load as the second stage. The folded cascode was chosen for the first stage due to its high DC gain (due to its high transconductance (g_m) and high output impedance (r_o)) and its improved headroom over the telescopic cascode architecture. The high g_m also contributes to its high Gain Bandwidth Product (GBW), which can be solved with $g_m/(2\pi Cc)$. The CS amplifier was chosen as the 2nd stage because of its high bandwidth and high output signal swing. The addition of a 2nd stage caused complexity due to adding a right half plane zero, making the amplifier unstable, which caused oscillations. To combat this, miller compensation with a nulling resistor was used which allows some of the signal to feedback at higher frequencies. The nulling resistor is added to cancel a pole-zero pair increasing the bandwidth and phase margin of the amplifier.

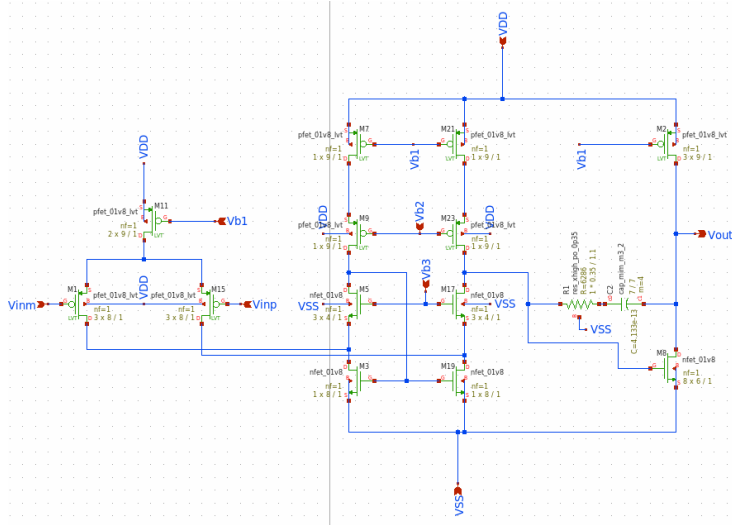


Figure 14: TIA Schematic

4.3.3 Functionality

The primary function this device serves is to provide users with an intuitive and lab-compatible interface for users interested in experimenting with or characterizing the ReRAM crossbar. This is to say, users are expected to be able to provide conventional instructions in the form of a highly abstract language (Like Python or C) that is supported by the included RISC-V ISA and receive a binary value sufficient in the precision representative of our crossbar depth. However, it does not need to have an expansive user interface. Aside from controls and logic analyzer output supplied by the on-chip RISC-V, the device also needs test points clearly defined in the documentation and easily accessible on the bring-up board. The test points should provide insight into the circuit infrastructure crucial to the operation of the crossbar and its interface. For example, signals going in and out of the ADC, DAC, TIA, and possible output values of the individual memory cells or columns if the pin space is limited. This will assist users in troubleshooting issues in hardware utility and thoroughly understand how the

device is functioning to verify correct operation.

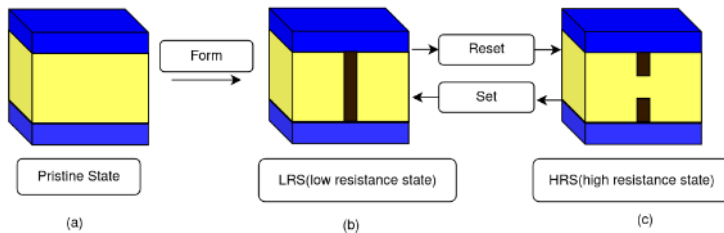


Figure 15: Illustration of memristor forming process

4.3.4 Areas of Challenge

The main challenge facing this project was the time constraint. This project is a very in-depth and technically challenging process. There were numerous setbacks where a design would be found to be unusable, and a new approach would be needed. This challenge was overcome by pushing forward and spreading the workload evenly.

One area of challenge was getting the ReRAM cell to pass precheck. Precheck is a series of tests given by eFabless that need to pass successfully in order to have your design approved for precheck. Upon the first attempt at getting the ReRAM to pass precheck, multiple of these checks were failing, the most notable one being the LVS. After numerous hours of testing and asking questions on the open-source forum, the issue was resolved. The issues that caused the problem were numerous files that were outdated and needed to be updated to a specific version in order to work properly. A guide to solving these issues was created to be given to future teams to help them avoid spending time debugging this challenge.

The main challenge of the comparator design was the amount of comparators at the r-string. This added an undesirable amount of parasitics to the resistor-string which severely hindered our operating speed. To combat this, a pre-amplifier was added to the comparator to increase its input impedance with the goal of reducing the loading and increasing speed. This issue was still present after the pre-amp integration. The next step to try and alleviate this issue was to add 2 inverters at the output to act as buffers to reduce the output loading of the priority encoder. This issue was still not solved. Our last and final iteration was to add a 5T op amp at the output of the r-string to again try and alleviate the capacitive loading.

Another minor issue we faced was determining a general flow such that digital and analog components could work in the same workspace. However, the default caravel harness project has distinctly separate flows for design implementation between digital and analog components. In particular, while both flows can eventually output a GDSII description of a component, a file type that could be compiled in the analog top-level hierarchical caravel harness, ensuring the correct behavioural operation of a mixed-signal design was less obvious when consulting with the available documentation. In relevance to our project, our ADC requires a digitally implemented priority encoder. Therefore, in order to ensure the ADC correctly converts to binary encoding, we had to determine a testbench in the analog environment, Xschem. To remedy this, we found a relatively straightforward workaround through an AWK script the author of Xschem had made. This script takes the Verilog netlist generated by OpenLane and converts it into a schematic describing the logic through combinational gates. This, in turn, could be plugged into the schematic for behavioural testing. For further information on this script, please refer to the appendix.

Lastly, we wanted to implement the layout of our analog components with an interdigitated and common-centroid-inspired approach. However, this meant devices had to be split up and wired in parallel, which introduced a much more complex placement and routing process. Furthermore, the environment used in this implementation process, Magic VLSI, had very rudimentary placement and

wiring tools compared to the Cadence suite our team was used to. For this reason, there would be great difficulty creating a layout accurately and efficiently. One solution we experimented with was the use of TCL scripts to assist in tiling and repetitive designs. For reference, a previous senior design group had already developed a couple of Python scripts that generated TCL scripts that could be executed in Magic VLSI to create PMOS and NMOS cells from the ground up. However, our design relied more on the generation of primitive instances instead of custom cells. For that, we referenced a publicly available project to develop a rudimentary TCL script that utilized the generic cell generation (`gencell`), `paint`, and `draw` functions built into Magic VLSI. Using this script, we were able to successfully create the layout for the ADC's comparators in this interdigitated and common-centroid-inspired approach. However, the script could not trivialize the routing process enough to justify the additional time commitment needed to implement the common centroid layout on the rest of the analog components. For specifics on why we decided to use such an approach to the layout, please refer to [Section 4.3.2 Detailed Design and Visuals](#). The TCL script and some commentary on the script can be found in the appendix.

4.4 Technology Considerations

This section describes the tradeoffs of the tools that we used in our design process.

- Simulation tools:
 - Ngspice [An open source spice simulator]
 - + Used for schematic level analysis/simulation
 - + Easily integrates with schematic software
 - Documentation on analysis setups can be a bit difficult to find
 - Error logs can be unintuitive or non-descriptive, making troubleshooting a bit difficult
 - Netgen [A program used for LVS]
 - + Relatively lightweight and easy to run
 - Documentation is scarce
 - Some functions have been implemented as a quick solution for a particular issue but have yet to be fully incorporated
 - Icarus Verilog [A compiler implementation for Verilog]
 - + Used for RTL level and gate level simulation of Verilog components
 - + Well embedded in OpenLane toolchain and hence, digital design flow
 - No native UI, so a wave viewer is needed to read outputs
 - GTKwave [A waveform viewer for standard Verilog VCD files]
 - + Used to view waveforms from digitally defined and hardened components
 - + Intuitive UI and integrated into the digital design flow
 - Klayout [A GDS and OASIS file viewer]
 - + Used to view GDSII layout files
 - + Intuitive UI and integrated into the digital design flow
- Design and Layout tools:
 - Xschem [A schematic capture program capable of hierarchical design]
 - + Used for analog component design
 - + Integrates well into the Efabless framework
 - UI was relatively unintuitive

- Documentation can be outdated or difficult to find
- o Magic VLSI [A TCL-based VLSI layout tool]
 - + Used to develop analog layouts
 - + Can read/write GDSII streams for mixed-signal environments
 - + DRC checks are made in real-time, so design adjustments can be made quickly
 - UI is very unintuitive
 - The tool doesn't have much utility to aid complex designs or automate repetitive actions
- o Cadence [Industry standard design software]
 - + It can be used for schematics and layout
 - + It could be used for creating proof-of-concept designs
 - It does not work with the technology/ process we are using

5 Testing

5.1 Unit Testing

The following analog components are tested individually via Xschem, Ngspice, and GAW to ensure correct behaviors:

- 2:1 analog multiplexers
 - o Tests should be done by introducing excitation signals equivalent to relative operational codes from the control module and power signals
 - o Results are expected to follow the general 2:1 multiplexer output per the relative operational codes and power signals
- Memristor
 - o Tests should be done by varying a voltage input to simulate the differing operating modes we have established (See Section 4.2.2 [Ideation](#))
 - o At this stage of the project, expected outputs should clearly define specific currents that intuitively derive the differing operating modes
- Transistor switch
 - o Tests should be done by simply performing a DC analysis to determine the switch impedance and drain current
 - o Resulting switch impedance is expected to be as low as possible, and the drain current should be as high as possible
- 4-bit Flash ADC
 - o Test should be done to identify the key ADC parameters
 - ENOB (Effective number of bits)
 - SNR (Signal to noise ratio)
 - INL (Integral nonlinearity)
 - DNL (Differential nonlinearity)
 - THD (Total Harmonic Distortion)
 - SFDR (Spurious Free Dynamic Range)
 - o A triangle wave will be used at the input of the ADC

- Spectral analysis will then be used to determine parameters
- TIA
 - Tests will be done by inputting an expected current based on previous DC analysis on the ReRAM crossbar and measuring the output voltage
 - Expected output voltages should indicate the differing input currents and provide a generous and intuitively quantized range of voltages for the ADC
- StrongARM latch comparator
 - Tests will be done by introducing a sine wave 5 to -5 Vpp and an ideal clock at 10 MHz to the circuit
 - Results should show that the comparator successfully switches between low and high output voltages when the input wave passes the threshold voltage of 3.3 V
- 1-bit DAC (Basic CMOS inverter)
 - Critical parameters to be established from this device are the behavioral performance in which a simple DC analysis should suffice.
 - Results should show that the 1-bit DAC correctly follows general inverter behavior.

The following digital components are to be tested individually via GTKwave to ensure correct behaviors:

- External FPGA control module
 - Tests will be done via a Verilog testbench that demonstrates all possible output operational codes
 - Expected results should follow the determined operational codes shown in –
- Logic analyser
 - Fully digital device, so testing for this device will consist of thorough test benches for each input.
 - Most testing will be applicable once system integration occurs
 - testing before system integration is to test the functionality of the opcodes that we produce
- 16:4 Priority encoder
 - Tests will be done via a Verilog testbench that demonstrates an output sequence
 - Expected results should follow the general 16:4 priority encoder table
- 3:8 decoders
 - Tests should be done via a Verilog testbench that demonstrates an output sequence
 - Expected results should follow the general 3:8 decoder table

Once the behaviors of each component have been determined, a layout will be realized in Magic VLSI. Layouts will be completed once DRC and LVS tests have been passed.

5.2 Interface Testing

As a whole, this project will consist of the following primary components, as stated in section 4.2 [Design Exploration](#): Logic analyzers, analog multiplexers, 3:8 decoders, 1-bit DACs, a ReRAM crossbar, TIAs, and 4-bit ADCs. Many of these components, specifically the 4-bit ADC and ReRAM crossbar, are built of several smaller components and thus require several tests conducted to ensure the systems are operating as expected. There are few interfacing components as the product we are delivering will be an integrated circuit with GPIO pins and the ability to flash code. We can test the GPIO pins through physical testing, but through our simulations, we can only assume that they will work as the technology simulates they will.

- **System on Chip:**

The user will be interfacing with the SoC by flashing code created by the User to perform operations on the ReRAM array and test it's functionality. The SoC will take code created by the user, execute it and make the appropriate changes to the pins interfacing with the user area, finally outputting the results attained by the tests into the SoC memory or outputted to a device connected to the chip package.

- **4-bit ADC:**

As this project utilizes a 4-bit flash ADC, the following components are used: 16 comparators, a resistor ladder, and a 16:4 priority encoder. Once the ADC has been assembled, it is paramount that the encoding, INL, and adequate ENOB are determined. A DC analysis will be conducted with a sweeping input voltage that simulates expected voltages based on the TIA output to test this.

5.3 Integration Testing

Integration testing in our project, aside from general component test benches, concerns the following critical junctions:

- **TIA to ADC:**

This connection is vital to ensure correct ADC quantization regarding the TIA output range. In particular, this test will be done primarily in Xschem and Ngspice as both components are analog. The test will introduce the existing excitation signal referenced from the ReRAM crossbar during an individual test cited in Section 5.1 [Unit Testing](#). The expected results should follow the relative output voltages established by the determined ADC encoding sequence.

- **Wrapper to overall system:**

To pass the MPW precheck such that our project is valid for fabrication, we'll need to ensure that the input and output pins are correctly integrated. Specifically, the connections of interest are between the input power pins and the voltage level shifters and between the GPIO and logic analyzers. These connections can be tested through Xschem and Ngspice. The test plan should introduce an expected voltage through input pins based on listed pin behaviors and measure the corresponding outputs to ensure the correct voltages are delivered.

5.4 System Testing

The overall system tests will be performed primarily in Xschem and Magic VLSI. Firstly, the complete circuit behavior will be established using a similar test, determined in section 5.3 [Integration Testing](#), for the wrapper to the overall system. Secondly, the final layout will be constructed using layouts of each primary component. Once the final layout passes DRC and LVS checks, it will be flattened, and a parasitic extraction (PEX) via Ngspice will be performed.

5.5 Regression Testing

We can cite the previous test results of our primary components to prevent new additions from breaking original functionality. This is plausible as the circuit was developed modularly, and input dependencies are clearly defined in section 4 [Design](#). Due to these defined expectations and influences, it should be possible to troubleshoot future functionality errors introduced by additional components quickly. However, it should still be noted that the following critical functions remain error-free or unchanged:

- ReRAM operations: The memristor operations determined by the established control voltages must be preserved, as doing otherwise will lead to cascading issues in control and interfacing circuits.
- Clock speed: Since the ADC comparators rely on the clock and have been sized accordingly, the clock speed should remain unchanged lest the pervasive effects of changing ADC characteristics can be accounted for.

5.6 Acceptance Testing

The functional results of this project can then be verified once all previous tests have been completed. These results can be illustrated and condensed via measurements from the Xschem, as mentioned earlier, analysis to confirm behavioral design constraints have been satisfied. Next, DRC and LVS report logs can easily demonstrate the overall success of component and project integration. Finally, fabrication constraint satisfaction can be illustrated through post-layout simulations like PEX and the MPW precheck. It should also be noted that systemic noise is another concern that must be addressed. This parameter can be tested by measuring and calculating the SNR of the system in its entirety or through the individual test benches indicated in the [Results](#) section.

5.7 Results

5.7.1 Crossbar Testing

The 1T1R cell was tested with Ngspice using the testbench shown in Figure 16. Using this two by two matrix, the cells were manipulated to perform a simple MAC in order to verify that the architecture was correct. Upon initial testing, it was discovered that due to the transistor models being hardcoded to only work drain to source, it was not possible to reset a cell. Applying a negative voltage from the bit line instead of a positive voltage from the source line was able to get around this error. However, this led to all cells in a column being reset partially even with their word lines disconnected due to the architecture of the 1T1R, where all the cells are directly connected to the bit line with no method of breaking that connection. The crossbar was not able to be fully simulated properly due to these model issues. It can be concluded that because the 1T1R works properly for both set and reset and the 2x2 matrix works, albeit the slight cell degradation from the reset operation, the fabricated circuit will operate properly when tested. A more in-depth discussion can be seen [here](#).

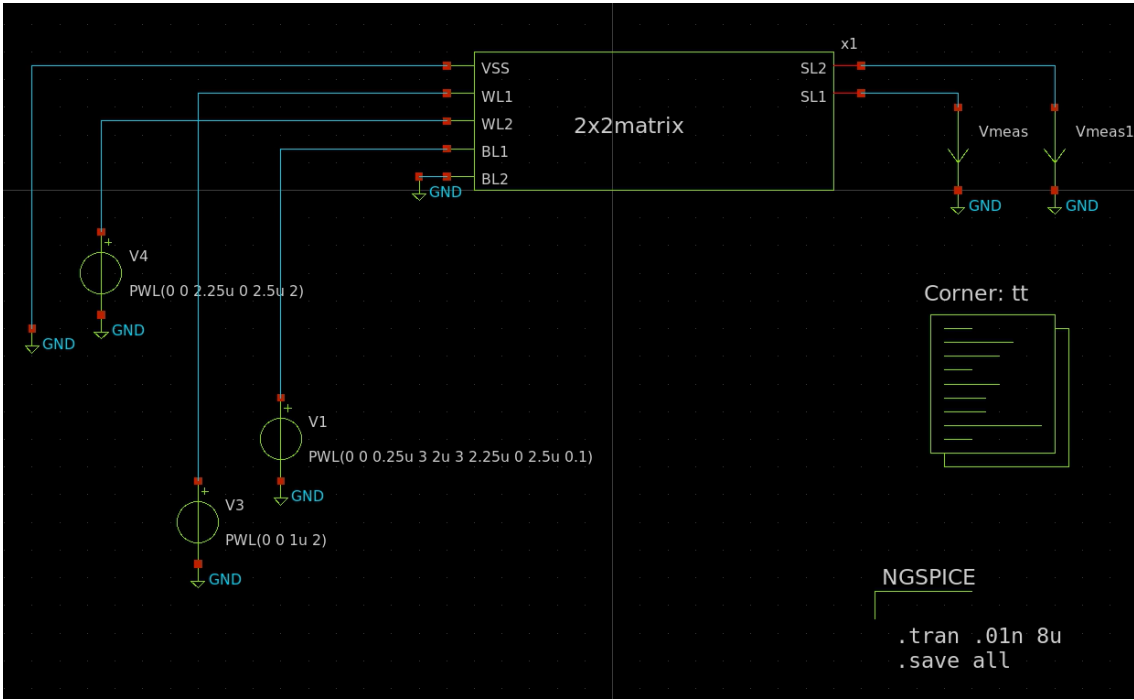


Figure 16: Testbench for 2x2 ReRAM crossbar

5.7.2 Comparator Testing

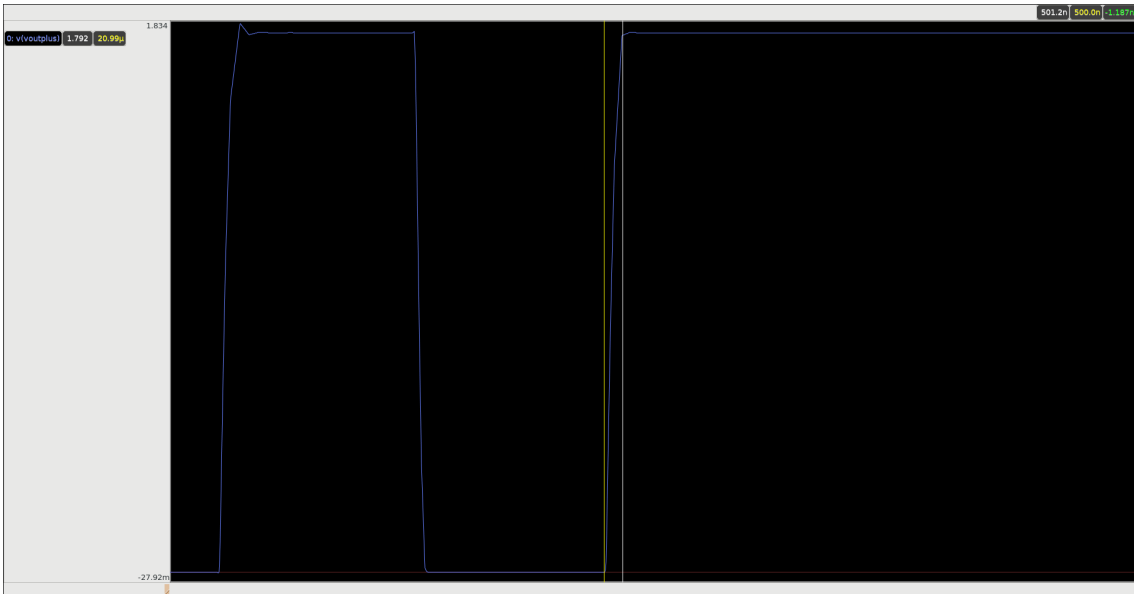


Figure 17: Comparator Slew rate

Figure 17 shows the slew rate of the strongARM latch. The input are a line from 0 to 1.2(Which is max value of the pre-amplifier input common mode range) and the refrence voltage of $V_{cm}/2$. The calculated slew rate which is calculated by the equation $(V_2 - V_1)/(t_2 - t_1)$ gives us a slew rate of $1493.32V/\mu s$

5.7.3 TIA Testing

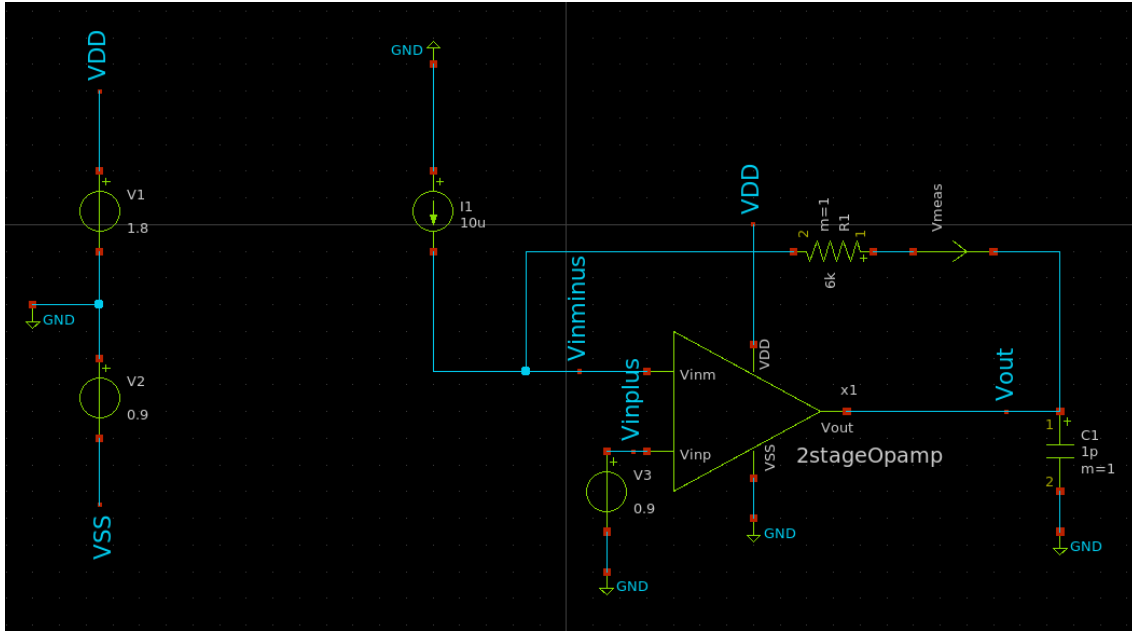


Figure 18: Testbench for TIA

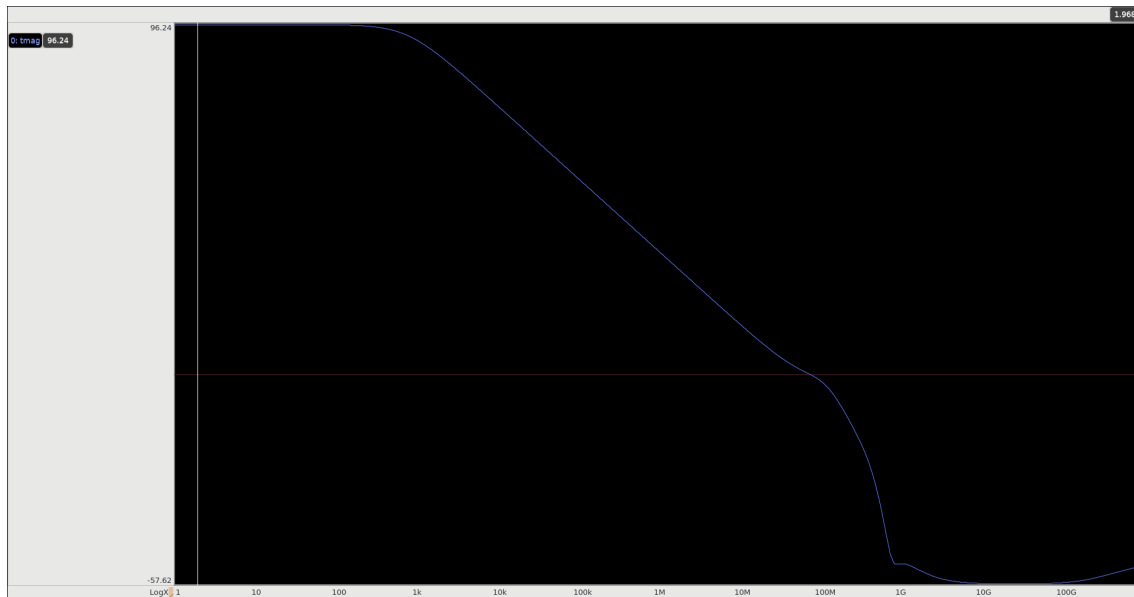


Figure 19: Open loop gain of TIA

The image above is the Frequency response of the TIA in an open loop configuration.

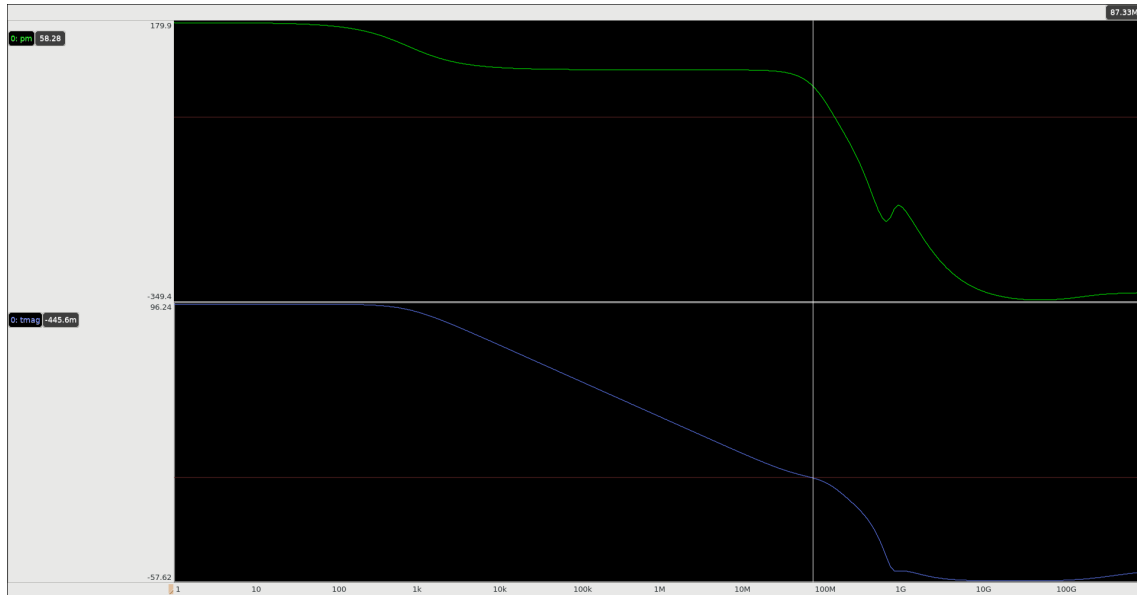


Figure 20: Open Loop Phase margin of TIA

5.7.4 ADC Testing

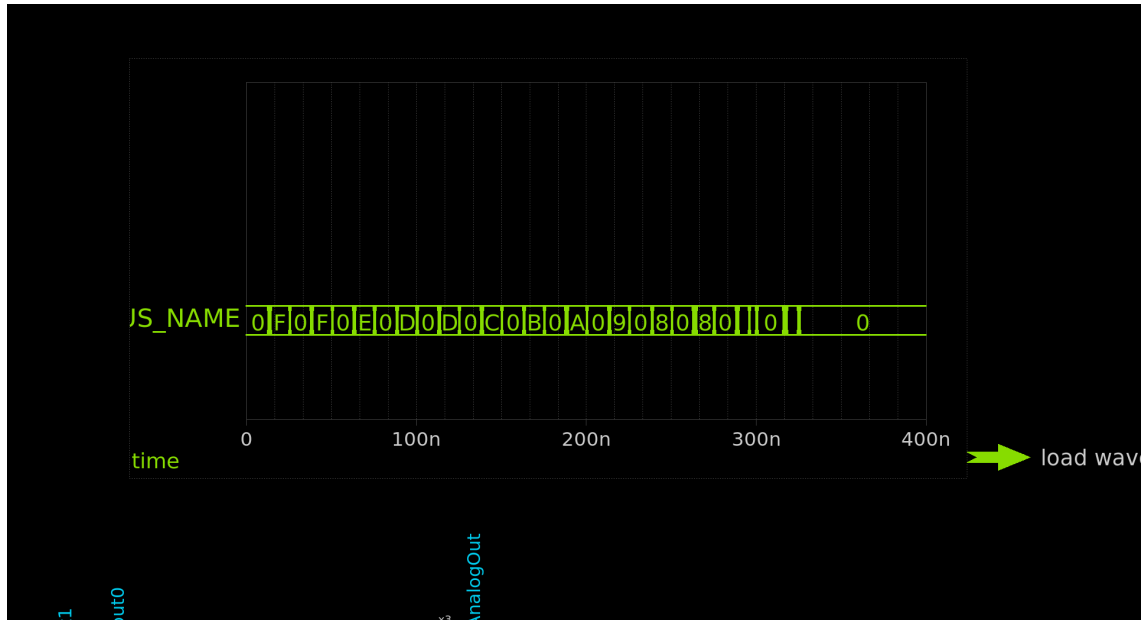


Figure 21: 4-bit ADC Results

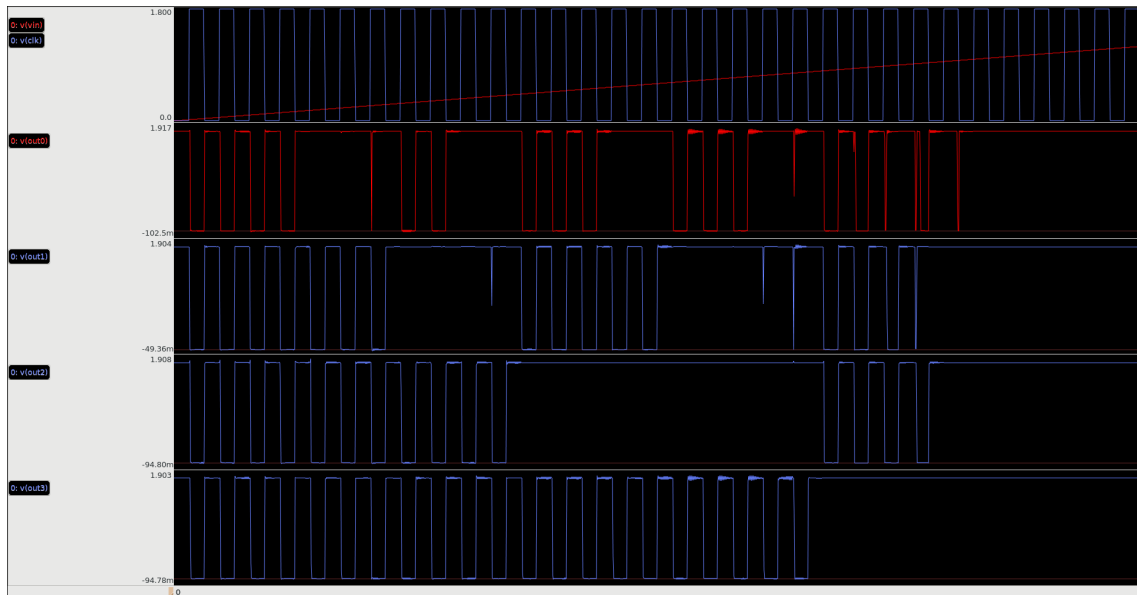


Figure 22: ADC Results cont.

5.7.5 ADC coupled with TIA Results

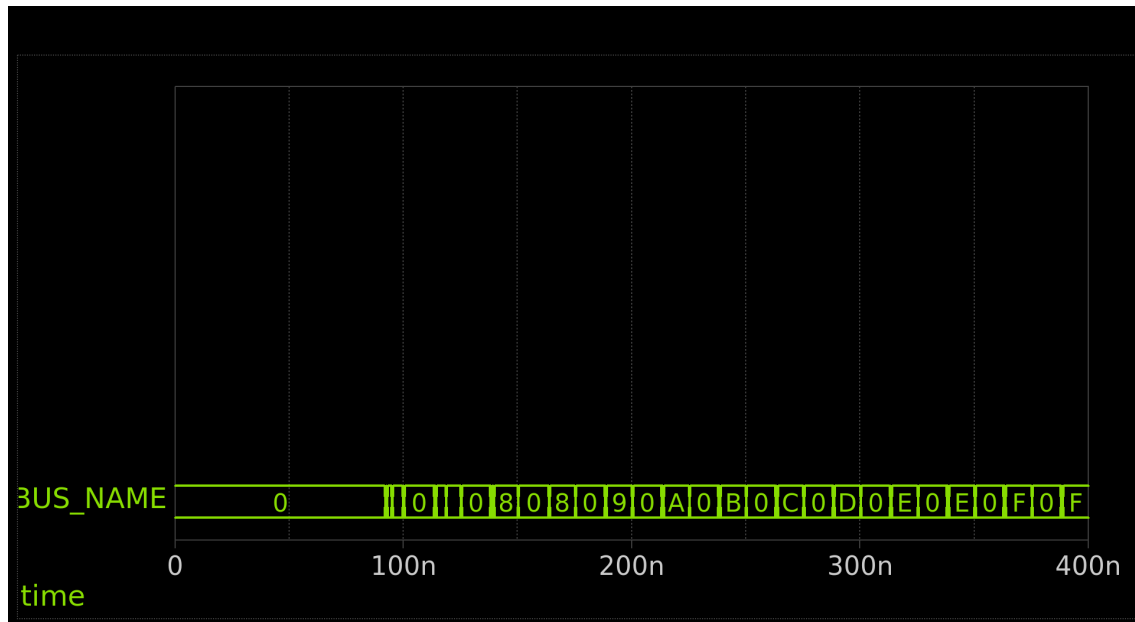


Figure 23: Combined TIA and ADC results

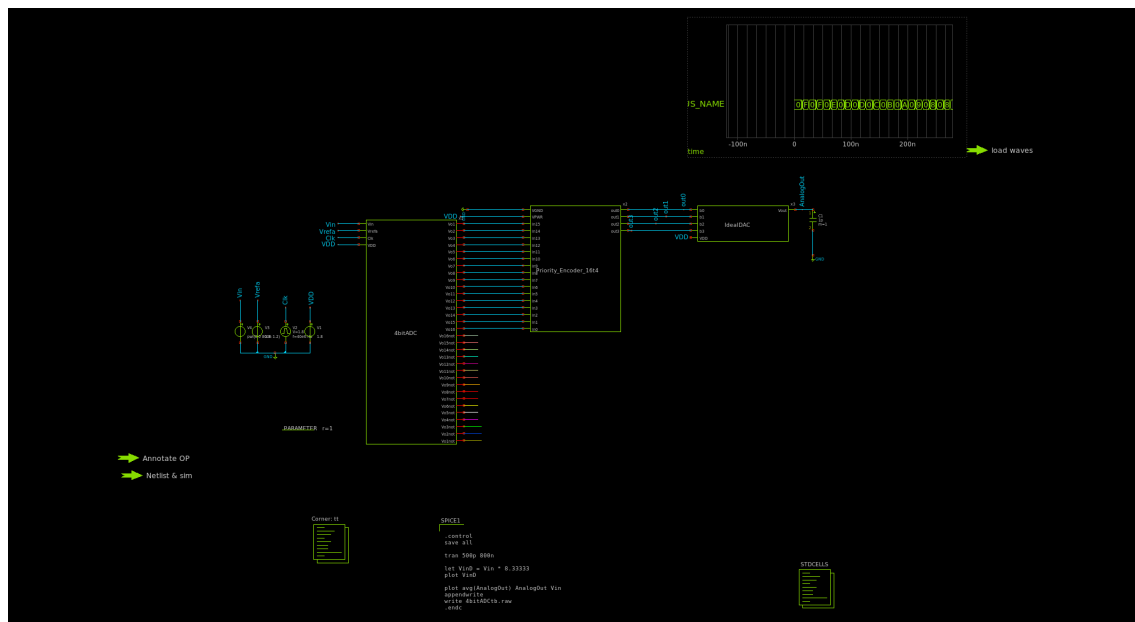


Figure 24: Testbench for 4bit ADC

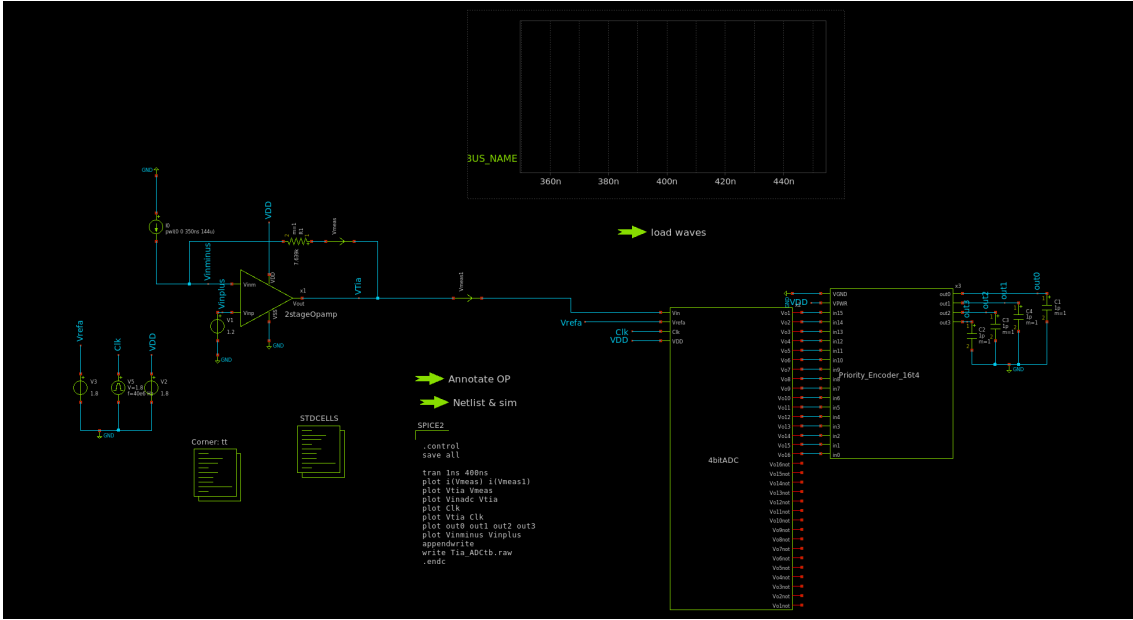


Figure 25: Testbench for TIA and ADC

6 Implementation

We have created and implemented the 8x8 crossbar as well as the peripheral circuits to control it and decode its output. We were able to complete and test all of the individual circuits and components up to the point of verifying that they passed precheck. This means that we have finalized the schematics and layouts of these components and that they pass the required checks in order for them to be fabricated on a chip.

However, we have not been able to produce a final fully integrated design where we tie all of these individual components together. This goal was unable to be accomplished due to a shortage of available time left to debug the numerous errors and issues that came with attempting a full-scale integration. Our components and designs will be left in a repository with the hope that the team that follows us will be able to finalize putting them together and sending them off to be fabricated.

6.1 Design Analysis

As mentioned in the previous section we were unable to fully implement our design. Looking at the performance of individual components, however, our design functioned quite well. The ReRAM cell was tested and verified to work properly as an individual cell; however, due to the modeling issues covered in Section 5.7 [Results](#), successful analysis of a crossbar was not plausible. We were able to get consistent current measurements from the single cell however, allowing for accurate design of the peripheral circuits. Regarding the ADC system, we were able to confirm that the individual components worked soundly, as seen in Section 5.7 [Results](#). Unfortunately, we were unable to achieve satisfying results during the integration tests. We found that our ADC has an ENOB of 31/2 bits. It's possible just using an open loop comparator or increasing the current in the StrongARM may improve results. Though the TIA behaves up to spec and then some, it could be optimized further to reduce power, increase bandwidth, and reduce area.

7 Professional Responsibility

7.1 Areas of Responsibility

Area	Description	NSPE Canon	IEEE Code of Ethics
Work competence	Perform work of high quality, integrity, timeliness, and professional competence	Perform services only in areas of their competence; Avoid deceptive acts.	5. to improve the understanding of technology, its appropriate application, and potential consequences 6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience or after full disclosure of pertinent limitations
Communication Honesty	Report work truthfully, without deception, and is understandable to stakeholders	issue public statements objectively and truthfully. Avoid deceptive acts.	3. to be honest and realistic in stating claims or estimates based on available Data; 2. to avoid real or perceived conflicts of interest whenever possible and to disclose them to affected parties when they do exist
Health, safety, and well-being	Minimize risks to safety, health, and well-being of stakeholders	Hold paramount the public's safety, health, and welfare.	1. to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment 9. to avoid injuring others, their property, reputation, or employment by false or malicious action

Financial responsibility	Deliver products and services of realizable value and at reasonable costs	Act for each employer or client as faithful agents or trustees	4. to reject bribery in all its forms
Property ownership	Respect property, ideas, and information of clients and others	Act for each employer or client as faithful agents or trustees.	9. to avoid injuring others, their property, reputation, or employment by false or malicious action
Sustainability	Protect the environment and natural resources locally and globally		1. to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;
Social Responsibility	Produce products and services that benefit society and communities	Conduct themselves honorably, responsibly, ethically, and lawfully to enhance the profession's honor, reputation, and usefulness.	8. to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression 9. to avoid injuring others, their property, reputation, or employment by false or malicious action 10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Table 9: Areas of responsibility

7.2 Project Specific Professional Responsibility Areas

- **Work Competence**

This responsibility is essential for our project. Our project is particular, so the design of components needs to be well thought out and of high quality. We must also stay on task for our design as we have a strict deadline of when the project must be submitted for fabrication. We are doing an excellent job at ensuring we are progressing, and barring any unforeseen circumstances, we

should be able to meet the deadline.

- **Financial Responsibility**

This is not an applicable responsibility as everything we do is through open-source tools and, therefore, free. The only cost that we will have is time.

- **Communication Honesty**

This applies to our design as our project is open source, and to have the chance to have it fabricated, we will need to submit our design to the open-source documentation. This means we will need good documentation and clarity of our design so that when other people look at it, they might understand what it is for. We must also be transparent when discussing our design with our advisor and avoid lying about what progress we have made. We have been doing an excellent job with this and communicating effectively and clearly with our clients.

- **Health, Safety, and Well-being**

This product will not be able to cause any potential health risks to the users.

- **Property Ownership**

All property we will be using is open source and free for anyone to use. However, we will still respect the work of those who have gone before us.

- **Sustainability**

Not a relevant responsibility, chip fabrication can be harmful to the environment. However, we are not in charge of the manufacturing process and, therefore, do not affect the conservation of resources.

- **Social Responsibility**

Our product can potentially improve computational technology by creating a device that can do computation with less chip area and lower power consumption.

7.3 Most Applicable Professional Responsibility Area

The most applicable responsibility to our project is Work Competence. This is because ASIC design is a very intensive process that requires a lot of intricate details and thought processes. The scale of the project is also quite large, and to meet our goals, we must ensure that we are staying on task and making high-quality progress. Our design will be added to the open-source documentation when we are finished, so we must create high quality work so that others may use it as inspiration.

8 Conclusions

8.1 Summary Of Progress

After many hours of work, a ReRAM crossbar and its surrounding peripheral circuits were designed. This ReRAM crossbar will be able to have values written to individual cells, and when a small voltage is applied, it will perform a compute. We have verified that the individual components will work as intended and can interface with the crossbar. Integrating all of these components together in a single design has proven to be difficult and may become unrealizable with the time crunch. However, there will be a design team that follows in our footsteps, allowing for instructions to be left in order for them to take our components and use them to fabricate a design.

In terms of peripheral circuitry, our group has made strides in behaviorally realizing much more complex circuits using the analog tools. In particular, a strong-arm latch comparator, a two-stage operational TIA, and a cascoded Wildar current mirror.

8.2 Value Provided

Our design provides valuable information for our clients. We were able to provide the components that were asked for such as an ADC with a higher than one bit resolution, a TIA to go along with the ADC and the 8x8 crossbar. We have provided a stepping stone in the overall lifetime of this project for future teams to use to continue the overall growth of the program. In addition to the components that we designed we also solved many common issues and wrote supporting document for these issues to further stream line the design process for future teams. Our efforts on this project will also provide useful to the co-curricular chip program that our clients plan to start by providing more concrete knowledge and support on these open source tools.

8.3 Next Steps

The next step for this project is to fabricate our designs and test them thoroughly to verify that the functionality of the real world product matches the models. This is important because there is uncertainty of the accuracy of the models and having this knowledge would highly benefit and influence the designs of future teams as it would allow them to be able to characterize and expect how the technology works in physical use.

There are many areas of this project that could be improved upon by future teams. These could be an even higher resolution ADC or a ReRAM cell that can store up to four different values. Something else that would prove useful for the future would be to continue on improving the tool infrastructure and documentation.

9 References

- [1] T. C. Carusone, K. W. Martin, and D. Johns, Analog Integrated Circuit Design, 2nd Edition, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2011.
- [2] W. Li, P. Xu, Y. Zhao, H. Li, Y. Xie and Y. Lin, "Timely: Pushing Data Movements And Interfaces In Pim Accelerators Towards Local And In Time Domain," 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2020, pp. 832-845, doi: 10.1109/ISCA45697.2020.00073.
- [3] "Sky130_fd_pr_reram - sky130 reram (Skywater provided)," sky130_fd_pr_reram - SKY130 ReRAM (SkyWater Provided) - SkyWater SKY130 PDK 0.0.0-22-g72df095 documentation, <https://sky130-fd-pr-ram.readthedocs.io/en/latest/index.html> (accessed Apr. 14, 2024).
- [4] C. Xu et al., "Overcoming the challenges of crossbar resistive memory architectures," 2015 IEEE 21st International Symposium on High-Performance Computer Architecture (HPCA), Burlingame, CA, USA, 2015, pp. 476-488, doi: 10.1109/HPCA.2015.7056056. keywords: Arrays;Random access memory;Phase change materials;Switches;Resistance;Transistors.
- [5] IEEE Standards Association, <https://standards.ieee.org/> (accessed Apr. 23, 2024).

Manuals:

- **Magic Docs:** <http://opencircuitdesign.com/magic/index.html>
- **Netgen Docs:** <http://opencircuitdesign.com/netgen/index.html>
- **Xschem Docs:** https://xschem.sourceforge.io/stefan/xschem_man/xschem_man.pdf
- **Ngspice Docs:** <https://ngspice.sourceforge.io/docs/ngspice-40-manual.pdf>

10 Appendices

10.1 Operation Manuals

A short manual describing how to use our design can be found here: [Operation Manual](#)
A guide on navigating the issues with getting the ReRAM models to pass precheck: [Precheck Guide](#)

10.2 Previous Versions

10.2.1 TIA versions

Initially our TIA was chosen to be a common source amplifier with a current mirror load but after initial testing it was determined that a higher gain was needed to increase accuracy and decrease noise. Given the transfer function of a 2 stage amplifier with negative feedback ($g_{minput} * (g_{moutput} - C_c) / (s^2 * C_L * C_c + sC_c(g_{mout} - \beta * g_{min}) + \beta g_{min} * g_{mout})$) it can be observed that the higher the gm of both stages decreases the sensitivity as well as noise sensitivity. In order to achieve this we switched to a two stage design which allowed us to achieve a higher gain. However, this was still not sufficient enough and we had to change our design yet again. We finally settled on a folded cascode design as it allowed up to 90 decibels of gain with the same bandwidth as the two stage design.

10.2.2 ReRAM versions

The architecture of the ReRAM has changed over the course of this project. Initially, the source lines of the crossbar were connected down the columns, which caused the current vectors to be the sum of all the currents in the columns. However, it was discovered that this was not correct as performing matrix multiplication correctly would mean that the source lines would have to be connected horizontally along the rows instead. Figure 9 shows an example of a 2x2 matrix with the proper configuration.

10.3 Other Considerations

Abbreviations Glossary

- 1T1R - One Transistor One Resistor
- ADC - Analog to Digital Converter
- ALU - Arithmetic Logic Unit
- ASIC - Application Specific Integrated Circuit
- ASIC - Application Specific Integrated Circuit
- BL - Bit Line
- CS - Common-source
- DAC - Digital to Analog Converter
- DNL - Differential Non-Linearity
- DRAM - Dynamic Random Access Memory
- DRC - Design Rule Check
- DTC - Digital Time Converter
- ENOB - Effective Number of Bits

GBW - Gain Bandwidth
gm - MOSFET transconductance
HRS - High Resistive State
INL - Integral Non-Linearity
LRS - Low Resistive State
LVS - Layout Vs Schematic
MAC - Multiply accumulate computation
Op-Amp - Operational Amplifier
PDK - Process Design Kit
PEX - Post-Extraction
ReRAM - Resistive Random Access Memory
RISC-V - Reduced Instruction Set Computer
ro - MOSFET output impedance
SAR - Successive Approximation Register
SFDR - Spurious Free Dynamic Range
SL - Source Line
SNR - Signal to Noise Ratio
SoC - System on Chip
SRAM - Static Random Access Memory
TDC - Time to Digital Converter
THD - Total Harmonic Distortion
TIA - Trans impedance Amplifier (Current to Voltage Amplifier)
WL - Word Line

10.4 Code

A collection of our code can be viewed in our [repository](#).

10.5 Team

10.5.1 Team Members

- Electrical Engineers
 - Konnor Kivimagi
 - Gage Moorman
 - Jason Xie
- Computer Engineers
 - Nathan Cook

10.5.2 Required Skill Sets

- Analog design
 - Layout design
 - Schematic design
 - Basic circuit knowledge
 - Device sizing
- Digital design
 - Verilog
 - Knowledge of computer architecture

10.5.3 Skill Sets

Skill	Members
Layout Design	All members
Schematic design	All Members
Basic circuit knowledge	All Members
Device sizing	Gage Moorman Jason Xie
Verilog	All Members
Knowledge of computer architecture	Nathan Cook, Jason Xie

Table 10: Required Skills

10.5.4 Project Management Style

Our project management style was agile. We divided our time into short one-week sprints where, after each week, we discuss and document our progress and address any outstanding issues and concerns.

10.5.5 Initial Project Management Roles

- Gage Moorman - Team Organizer, main analog designer
- Konnor Kivimagi - Main documentation editor, mixed analog, digital designer
- Nathan Cook – Main client liaison, mixed analog, digital designer
- Jason Xie – Assistant documentation editor, main digital designer

10.5.6 Team Contract

Team Members:

1. Gage Moorman
2. Nathan Cook
3. Konnor Kivimagi
4. Jason Xie

Team Procedures:

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
Ideally, on Friday evenings, in person, Coover TLA and Durham 310 lab.
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., email, phone, app, face-to-face):
Email, Microsoft Teams, Face-to-face, and or Discord will be used to coordinate with other group members, ideally Microsoft Teams, so that we can keep all group information in one place.
3. Decision-making policy (e.g., consensus, majority vote):
A consensus will make decisions.
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared or archived):
Start a Microsoft Teams meeting, keep notes in Microsoft Teams as the meeting progresses, and assign Gitlab tasks. Use GitLab to chronicle work done.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
Ideally, attend all meetings on time, but if absences occur, the members should notify the group of the absence; the rest will document what was discussed and send absentee the meeting notes.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
Each member should try to complete their assignments and reach their deadlines within the ideal time commitment (8 hours) each week; if not met, it is not automatically an issue unless it becomes a recurring problem.
3. Expected level of communication with other team members:
Responses should be timely; a few hours are okay, but at least a response within the day would be ideal.
4. Expected level of commitment to team decisions and tasks:
Should be available and willing to meet on weekends to finish tasks or work later, ideal time commitment is 8 hours a week.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
may be changed as deliverables are updated throughout the project run. The client and advisor gave baseline deliverables; if those are met, other deliverables will be given.
 - Nathan Cook: Main client liaison, mixed analog, digital designer
 - Gage Moorman: Team Organizer, main analog designer
 - Jason Xie: assistant documentation editor, main digital designer
 - Konnor Kivimagi: Main documentation editor, mixed analog, digital designer
2. Strategies for supporting and guiding the work of all team members:
Agile project management, using GitLab to assign tasks, filling meeting notes with relevant information on tasks that must be completed.
3. Strategies for recognizing the contributions of all team members:
Weekly reporting during agile meetings and Gitlab push and issue resolution.

Collaboration and Inclusion

1. Describe each team member's skills, expertise, and unique perspectives.
 - Nathan Cook: Computer Architecture, semiconductor physics, Computer science
 - Gage Moorman: Signals processing, Digital Synthesis
 - Jason Xie: Computer Architecture, Microwave, and RF, Digital Synthesis
 - Konnor Kivimagi: Signals processing, Semiconductor physics
2. Strategies for encouraging and supporting contributions and ideas from all team members: Agile project management to issue ideas as well as troubles with deliverables as well as GitLab issue reviews when deliverables or portions of work are finished.
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
 Inform the person one on 1 to try and resolve the issue. If no resolution is found, move to a group setting, which might involve the client and advisor if it is a project-related issue that further questions can resolve; the last resort is going to 491 TAs or Professors to see what can be done to resolve it.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
 Finishing all deliverables expected for this semester between the group, client, and advisor and being ready for or already working on deliverables for the fall.
 - Current deliverables
 - Familiarize ourselves with project tools (Caravel, GTKwave, Klayout, magic, XScheme)
 - Begin basic work on designs, laying circuits out on paper and then converting them over to the tools
 - Future deliverables
 - Improve upon previous groups' work
 - Better resolution ADC
 - Fix ReRAM LVS
 - Bit serial computation DAC
2. Strategies for planning and assigning individual and teamwork:
 Meeting on Fridays to set group and individual tasks for the next week and go over work done during the week to see if there is any place lacking and if that place needs extra support from other group members
3. Strategies for keeping on task:
 Regularly checking in with the group and their progress, giving gentle reminders, using gitlab to keep track of work on the project and who is doing what issues and which ones are getting resolved.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
 Three strike system, 1st will be a gentle reminder, 2nd will be a group conversation about what needs to change 3rd refer to continued infractions
2. What will your team do if the infractions continue?
 Consult 491 professors about what we should do and continue to dialogue with the group and the professors.

- a) *I participated in formulating this contract's standards, roles, and procedures.*
- b) *I understand that I must abide by these terms and conditions.*
- c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract*

Nathan Cook
Gage Moorman
Konnor Kivimagi
Jason Xie

DATE 1/26/2024
DATE 1/26/2024
DATE 1/26/2024
DATE 1/26/2024