

ReRAM Compute ASIC Fabrication

DESIGN DOCUMENT

sddec23-08

Client: Dr. Duwe

Adviser: Dr. Wang

Team Members: Joshua Thater, Aiden Petersen, Matthew
Ottersen, Regassa Dukele

sddec23-08@iastate.edu

<https://sddec23-08.sd.ece.iastate.edu/>

Revised: 04/23/2023

Executive Summary

Development Standards & Practices Used

IEEE 1481-2019 - IEEE Standard for Integrated Circuit (IC) Open Library Architecture (OLA): This standard fits our project since it goes over ways for our integrated circuit to be analyzed for timing and power consumption across a set of design automation (EDA) tools.

IEEE 1076.4-2000 - IEEE Standard VITAL ASIC Modeling Specification: This standard applies to our project since we will be designing an ASIC chip, and we will need it to test it with highly accurate and efficient simulation models.

IEEE 1149.4-2010 - IEEE Standard for a Mixed-Signal Test Bus: This standard will fit our project since it will house both analog and digital components, and we will have to thoroughly test all components both independently and together.

Summary of Requirements

Functional Requirements (Specification)

- Create an Efabless multi-project wafer (MPW) submission that contains a ReRAM crossbar that can be interacted with through the caravel harness.
 - This is only applicable if there is an MPW shuttle open
- MPW Submission must pass Efabless pre-check
- Create ADC, DAC, sample and hold, and TIAs to interact with ReRAM crossbar via the caravel harness.
- The ReRAM crossbar will be functional and able to perform multiply and accumulate operations.
- The ReRAM crossbar's inputs and weights will be assignable
- Individually design and verify the following components using SPICE
 - Up to a 1-bit ADC
 - 1-bit DAC
 - 1-bit Transimpedance amplifier
 - 1-bit sample and hold
 - ReRAM crossbar

Applicable Courses from Iowa State University Curriculum

- EE 330 - Integrated Electronics
- EE 435 - Analog VLSI Circuit Design
- EE 465 - Digital VLSI Circuit Design
- CprE 281 - Digital Logic
- CprE 288 - Embedded Systems
- CprE 381 - Computer Organization & Assembly Level Programming
- COM S 252 - Linux Operating System Essentials

New Skills/Knowledge acquired that was not taught in courses

- Open-source analog design using XSchem, Ngspice/Xyce, Magic, and Netgen
- Sky130 process
- RISC-V ISA
- ReRAM
- Design of ASIC chip
- Tape-out process of silicon chip

Table of Contents

1	Team	6
1.1	TEAM MEMBERS	6
1.2	REQUIRED SKILL SETS FOR YOUR PROJECT (if feasible – tie them to the requirements)	6
1.3	SKILL SETS COVERED BY THE TEAM (for each skill, state which team member(s) cover it)	6
1.4	PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM	6
1.5	INITIAL PROJECT MANAGEMENT ROLES	6
2	Introduction	7-8
2.1	PROBLEM STATEMENT	7
2.2	REQUIREMENTS & CONSTRAINTS	7-8
2.3	ENGINEERING STANDARDS	8
2.4	INTENDED USERS AND USES	8
3	Project Plan	8-13
3.1	Project Management/Tracking Procedures	8
3.2	Task Decomposition	9
3.3	Project Proposed Milestones, Metrics, and Evaluation Criteria	9
3.4	Project Timeline/Schedule	10
3.5	Risks And Risk Management/Mitigation	10-11
3.6	Personnel Effort Requirements	11-13
3.7	Other Resource Requirements	13
4	Design	13-22
4.1	Design Context	13-16
4.1.1	Broader Context	13-15
4.1.2	User Needs	15
4.1.3	Prior Work/Solutions	15
4.1.4	Technical Complexity	15-16
4.2	Design Exploration	16-17
4.2.1	Design Decisions	16
4.2.2	Ideation	16-17

4.2.3 Decision-Making and Trade-Off	17
4.3 Proposed Design	18-21
4.3.1 Design Visual and Description	18-20
4.3.2 Functionality	20
4.3.3 Areas of Concern and Development	21
4.4 Technology Considerations	21-22
4.5 Design Analysis	22
4.6 Design Plan	22
5 Testing	23-25
5.1 Unit Testing	23
5.2 Interface Testing	23
5.3 Integration Testing	23
5.4 System Testing	23-24
5.5 Regression Testing	24
5.6 Acceptance Testing	24
5.7 Security Testing (if applicable)	24
5.8 Results	24-25
6 Implementation	25
7 Professionalism	26-30
7.1 Areas of Responsibility	26-29
7.2 Project Specific Professional Responsibility Areas	29-30
7.3 Most Applicable Professional Responsibility Area	30
8 Closing Material	30-32
8.1 Discussion	30
8.2 Conclusion	30
8.3 References	31
8.4 Appendices	31-32
8.4.1 Team Contract	32-34

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

Definitions:

- ReRAM: Restive ram that is able to do computations in the analog domain instead of the digital domain
- Caravel Harness: Test harness that contains circuitry. Our design will be placed in this harness once fabricated
- XSchem: Open-source circuit schematic editor
- Ngspice: Open-source circuit simulator
- Xyce: Open-source circuit simulator (supports Verilog-A netlists)
- Magic: Open-source layout editor
- Netgen: Open-source LVS tester

Figures:

- Figure 1: Gantt Chart (page 10)
- Figure 2: ReRAM Crossbar High-Level Diagram (page 18)
- Figure 3: 1T1R Cell (page 19)
- Figure 4: User Area Top Level Diagram (page 20)
- Figure 5: Example Simulation Results (page 25)
- Figure 6: Caravel Harness Diagram (page 31)

Tables:

- Table 1: Possible Risks (page 11)
- Table 2: Personnel Effort Requirements (pages 11-13)
- Table 3: Broader Context (pages 13-15)
- Table 4: Decision Making for ADC Architecture (page 17)
- Table 5: Areas of Responsibility (pages 25-26)

1 Team

1.1 TEAM MEMBERS

- Joshua Thater
- Aiden Petersen
- Matthew Ottersen
- Regassa Dukele

1.2 REQUIRED SKILL SETS FOR YOUR PROJECT

- Willingness to learn and work together
- Strong persistence through challenging work
- Schematic editor
- Layout editor
- DevOps
- MCU programming
- PCB design

1.3 SKILL SETS COVERED BY THE TEAM

- Whole team
- Whole team
- Joshua Thater, Matt Ottersen
- Joshua Thater, Matt Ottersen
- Aiden Petersen
- Aiden Petersen
- Joshua Thater

1.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

- We are going to use The Agile method with short 1 week sprints.

1.5 INITIAL PROJECT MANAGEMENT ROLES

- Joshua Thater: Software tool researcher, mixed-signal designer
- Aiden Petersen: DevOps and digital designer
- Regassa Dukele: VLSI designer
- Matt Ottersen: VLSI designer

2 Introduction

2.1 PROBLEM STATEMENT

Ideal:

- Iowa State University would have access to fabricated ReRAM chips for research purposes
- Iowa State University would have institutional knowledge of how the analog design flow works for the Skywater 130nm process

Reality:

- It is difficult to get any fabricated chip, especially one with ReRAM, because of how new of a technology it is.
- Iowa State University has never produced a fabricated analog chip on the Skywater 130nm processed

Consequences:

- It's difficult to do novel research that involves ReRAM because of the lack of fabricated chips.
- It's very difficult for students and faculty to create analog chips using the Skywater 130nm process because of the poor public documentation and the lack of internal documentation.

Proposal:

- Use Efabless's MPW shuttle program to submit a ReRAM chip proposal.
 - If it gets approved, it would give us access to fabricated ReRAM chips
 - Along the way, we would document our workflow, contributing to ISU's internal knowledge of analog fabrication in the Skywater 130nm process.

2.2 REQUIREMENTS & CONSTRAINTS

Functional Requirements (Specification)

- Create an Efabless MPW submission that contains a ReRAM crossbar that can be interacted with through the caravel harness.
- MPW Submission must pass Efabless pre-check
- The crossbar will be interacted with through the wishbone bus, which is then turned into analog signals using a DAC and read back into the harness using an ADC.
- The ReRAM crossbar will be functional and able to perform multiple and accumulate operations.
- The ReRAM crossbar's inputs and weights will be assignable
- Individually design and verify the following components using SPICE
 - Up to a 3-bit ADC
 - 1-bit DAC
 - ReRAM crossbar
 - 2 finite state machines that communicate with the wishbone bus, one for receiving and one for sending.

Qualitative Aesthetics Requirements

- Quickly compute, multiply, and accumulate operations
- Document tools and tool flow necessary to create the MPW submission.
 - Documentation should be readable and easy to follow

UI Requirements

- Create drivers that provide an API for programmers to interact with the crossbar.

2.3 ENGINEERING STANDARDS

IEEE 1481-2019 - IEEE Standard for Integrated Circuit (IC) Open Library Architecture (OLA): This standard fits our project since it goes over ways for our integrated circuit to be analyzed for timing and power consumption across a set of design automation (EDA) tools.

IEEE 1076.4-2000 - IEEE Standard VITAL ASIC Modeling Specification: This standard applies to our project since we will be designing an ASIC chip, and we will need it to test it with highly accurate and efficient simulation models.

IEEE 1149.4-2010 - IEEE Standard for a Mixed-Signal Test Bus: This standard will fit our project since it will house both analog and digital components, and we will have to thoroughly test all components both independently and together.

2.4 INTENDED USERS AND USES

The main intended user for the results of our project will be Professor Duwe and Professor Wang. They first want to test the potential computing power of a ReRAM crossbar. They also want to use our experiences and knowledge of our project to help future students design ASIC chips through the Efabless ecosystem, whether it be for research or in pursuit of creating a club for chip fabrication.

Another user would be potential future students who are interested in analog/mixed-signal chip fabrication. It is our hope that they will be able to use our project as either an example or a jumping-off point for their own design. With the help of our documentation and written experiences, they should be able to hit the ground running when they want to work on fabricating their own chip through the Efabless process.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

We will use agile because of our frequent meetings with our advisor and client; it allows us to receive feedback more quickly and implement it into our project. We will use Git, Github, and MS Teams to track progress.

3.2 TASK DECOMPOSITION

We will divide analog design and implementation tasks among Joshua, Matthew, and Regassa, dividing each of the analog parts among each member. Aiden will be responsible for all digital design and software components of the project. All members will help with creating a bring-up plan and documentation

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

- Phase 1: Research tools and software

- Research all of the open-source software that will be required, along with tools provided by Efabless. Also will need to research other ReRAM crossbar designs.

-Phase 2: All components designed

- We implement all components in XSchem, simulate with Ngspice and create layouts using Magic. These layouts must pass LVS checks.

-Phase 3: All components in the top-level integrated into one part

- Implement all of the components into the top level. This top-level must include numerous spice simulations to test functionality and a top-level layout that passes LVS checks.

-Phase 4: The system is fully verified

- Must create detailed simulations for the top level to ensure that the crossbar works properly within the Caravel harness.

-Phase 5: Submission for MPW

- Must pass MPW precheck and submit it to potentially be fabricated.

- Phase 6: Bring-up plan & documentation

- Have to come up with a plan on how we will test our design once it comes back from fabrication. Also, create documentation on our experiences so that future teams will have an easier time.

3.4 PROJECT TIMELINE/SCHEDULE

ReRAM Crossbar ASIC Fabrication

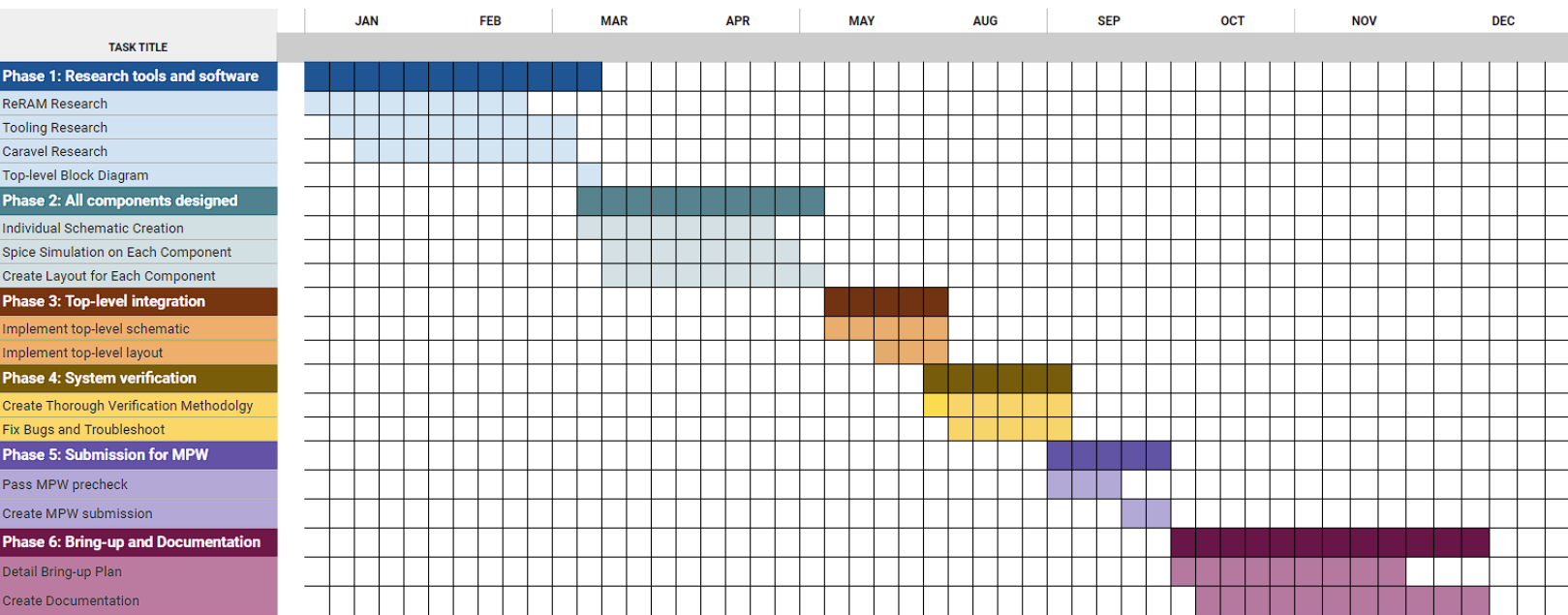


Figure 1: Gantt Chart

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

In phase 2, it is possible that we will fail to get all components implemented in a timely manner due to the lack of proper tooling. We may have to switch over to using a different VLSI design suite (Cadence) and then translate them over to open-source tools.

Phase 3 is relatively low risk compared to the others. It should be relatively straightforward as long as milestone 2 is done properly

Phase 4 has the potential to be difficult to fully verify due to the complexity of the incoming digital signals along with the lack of features of the analog verification tools. We currently do not have a solution to mitigate this risk

Phase 5 should be easy as long as precheck passes. If we can't get it to pass, it will be very difficult to solve. Currently no solution to this risk.

Phase 6 should have little risk involved, as we are only coming up with plans on how to test our design when it comes back from fabrication

Possible Risk	Probability
Efabless program gets suspended	3%
Unable to implement all components in time	10%
Top-level functionality is not desirable	25%
Design not able to pass precheck	10%

Table 1: Possible Risks

3.6 PERSONNEL EFFORT REQUIREMENTS

Phase 1:

Task	Time	Explanation
ReRAM Research	20hr	Must research how ReRAM MAC work in order to know how to use it in design
Tooling Research	20hr	Must figure out how to use open source tools in order to design schematics
Caravel Research	20hr	Understand how to caravel framework works and how to implement with user area (our ReRAM crossbar)
Top-level design	3hr	Actually design and create a broad block diagram schematic of our user area

Phase 2:

Task	Time	Explanation
Create a schematic for each component	2hr / component	Implementing each component with schematics using XScheme
Run spice simulations on schematic and troubleshoot	15hr / component	Verifying the functionality and achieving desired results using spice
Create a layout for each component	5hr / component	Create a functioning layout for each component that

		passes LVS
--	--	------------

Phase 3:

Task	Time	Explanation
Implement top-level schematic	20hr	Combine each component into a top-level schematic based on milestone 1 block diagram.
Implement top-level layout	20hr	Create a layout of top-level spice that matches passes LVS

Phase 4:

Task	Time	Explanation
Create thorough verification methodology	20 hr	Create a system to verify the implemented spice. Must be very comprehensive, so ensure that a fabricated design is functional.
Fix all bugs in the system	20 hr	Using the previously implemented verification methodology, fix all of the bugs present.

Phase 5:

Task	Time	Explanation
Pass MPW Precheck	~10hr (unsure on the amount of time needed)	Must pass MPW precheck for submission
Create MPW submission	1 hr	Submit the project. It should be fast.

Phase 6:

Task	Time	Explanation
Detail bring-up plan	30 hr	Must come up with a plan on

		how we will test our design. We may have to build PCBs to assist with this.
Create documentation	15 hr	Create documentation on how to use tools/software for future teams.

Table 2: Personnel Effort Requirements

3.7 OTHER RESOURCE REQUIREMENTS

Everything is done in free and open-source software that is poorly documented. Most of the resources that will be used come from previous senior design teams working on similar projects, along with the open-source hardware slack channel, which contains many experts in the field that can help us with our problems.

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

ReRAM is a non-volatile RAM that has the potential to perform computations in the analog domain. This can be beneficial as this requires less data movement and less power consumption as compared to typical computations done in the digital domain. Our project is to create a ReRAM crossbar on an ASIC chip using the SkyWater 130nm process. We are hoping to tape-out this design and get it fabricated so that Professor Duwe and Professor Wang can evaluate the crossbar for computational potential. Outside of the two Professors, we are also designing for future students interested in open-source ASIC chip fabrication, as our project will help pave the way as an example. With our design, we are hoping that Iowa State will be able to create a curriculum or a club that is focused on the fabrication of chips. This will help the societal need of young engineers looking to get trained in the process of designing and fabricating ASIC chips.

List relevant considerations related to your project in each of the following areas:

Area	Description	Examples
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	Increasing/reducing exposure to pollutants and other harmful substances, increasing/reducing safety risks, increasing/reducing job opportunities

	<p>This project does not have a huge impact on public health, safety, or welfare. However, ASIC chips will continue to have important uses in many applications around the world that do improve these areas.</p>	<p>Our design will be added to the open-source hardware in the Efabless program, which may lead to inspire or help other engineers learn how to design ASIC chips.</p>
Global, cultural, and social	<p>How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.</p> <p>As this is an open-source project, we are following the core ideals of open-source development. These include transparency, collaboration, and continual growth. As such, our project will follow these ideals and will help pave the way for other engineers to contribute to the open-source community.</p>	<p>Development or operation of the solution would violate a profession’s code of ethics, implementation of the solution would require an undesired change in community practices</p> <p>This project will add to the open-source hardware repository on the Efabless website. Our design may be used as a template for future engineers, whether inside this university or outside, as a potential template for future ReRAM crossbar designs.</p>
Environmental	<p>What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.</p> <p>ReRAM has the potential to consume much less power than its digital counterparts. There are many companies that are looking at ReRAM and trying to gauge whether or not it will be beneficial to their designs in the future. If this design is adopted, then there may be a substantial improvement in power consumption for RAM throughout the world.</p>	<p>Increasing/decreasing energy usage from nonrenewable sources, increasing/decreasing usage/production of non-recyclable materials</p> <p>Our project will be implementing a basic ReRAM crossbar. This ReRAM will have the hopes of utilizing much less power compared to other RAM types of similar size and computational power.</p>
Economic	<p>What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects</p>	<p>Product needs to remain affordable for target users, product creates or diminishes opportunities for economic advancement, high development cost creates risk for organization</p>

	<p>on communities, markets, nations, and other groups.</p> <p>As this is an open-source project, all of the tools are completely free to use. Along with this, there are similar designs that have been created through this open-source workflow as well. These designs may lead to faster innovation and may lead to companies feeling the need to innovate their designs as well.</p>	<p>As we are designing our project through this open-source workflow, all of our designs and process will be free to view and use for future use.</p>
--	---	--

Table 3: Broader Context

4.1.2 User Needs

Professor Duwe and Professor Wang need a way to create a ReRAM crossbar through the Efabless program because they want to test it for computational power and also better understand how designs are fabricated through this process.

Future students interested in ASIC chip design need a way to understand how to create chips through the open-source workflow because learning how to tape-out chips before entering the industry is an important skill to have.

4.1.3 Prior Work/Solutions

There are only a couple of similar designs on the market, but they are proprietary and may not allow for the types of testing that our client wants. As this is a developing technology, most ReRam modules are not commercially available.

Pros

- Open source
- Non-Proprietary
- While designing the ReRam, we will gain a lot of information for the client on how to do Analog VLSI design using open source tools

Cons

- Untested; we won't know if it works until we get the physical chip and test it
- May not have as good performance as other products

4.1.4 Technical Complexity

The design consists of both digital and analog circuit design. The components of the design include:

Resistive Ram Module

- This requires Research into how resistive ram works

ADC and DAC

- We will need to design Op-Amps with our preferred performance characteristics that will amplify signals and work as comparators
- Knowledge of how to design ADC and DAC for specific applications
- Design of Encoders and Decoders
- Design of sample and hold

Use of open source tools

- Ability to learn, setup, and troubleshoot software/process flow that has little to no documentation

Creating a layout for the circuit that passes LVS and DRC

- This will require design knowledge and skills for ASIC chip design

One challenge is the theoretical implementation of ReRam. This will require research of how ReRam works and then designing a circuit that allows a system to utilize the ReRam. Then the next challenge comes through having to bring that design to silicon by using an open-source design process. As the open-source PDK we are using has very little documentation on how to bring an analog design through its process flow.

4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

Our design has many circuit components in it, and all of these components have multiple ways to implement them. One of the key design decisions we made was what type of DAC would be used to pipe the digital voltages (high or low) to analog voltages so that the ReRAM crossbar would compute in the analog domain. For now, we plan on using a simple inverter as a 1-bit DAC, as it is simple to implement, and the DAC for our design does not need to be complex. Another key design decision we made was the architecture of the ADC. Just like with the DAC, we need a way to convert between analog and digital realms. For this project, we need a way to convert the analog computations back to a digital value to be read. One last key design decision we made was how we were going to load weights onto the ReRAM cells. We have not finalized this decision, but we believe that we may be able to do this using the logic analyzer that is housed in the Caravel harness.

4.2.2 Ideation.

For data converter architectures, there are many options to choose from. Specifically for ADC converters, there are almost too many options to choose from. Each of these options comes with

its own benefits but also its own drawbacks. One ADC may be faster but less accurate, while another may be able to do more complex conversions. So, picking the architecture for our ADC was important as we wanted it to be able to best match the needs of our project. From our research, we found that there were five main architectures that are widely used. These architectures were:

- Successive Approximation (SAR) ADC
- Dual-Slope ADC
- Pipelined ADC
- Flash ADC
- Delta-Sigma ADC

4.2.3 Decision-Making and Trade-Off

In order to choose which architecture to use, we weighed the pros and cons of each. After some research, we created this table:

<u>ADC Architecture</u>	<u>Pros</u>	<u>Cons</u>
SAR	Handles waveshapes very well, takes very high sample rate, high resolutions	Can be slow, larger/more complex design at low resolutions
Dual-Slope	Very accurate, and has high-resolution	Slow sample rate and speed, can be hard to implement
Pipelined	Very fast speeds, good resolution	Inherent latency due to architecture
Flash	The fastest speed, no latency, easy to implement at “1-bit” level	Circuit gets much bigger with resolution increases, limited to 8-bit resolution
Delta-Sigma	Very high resolution, reduces quantization noise	Limited sample rate, does not handle unnatural waves

Table 4: Decision Making for ADC Architecture

As can be seen from this table, each architecture comes with its own pros and cons. For our project, we are only using a 1-bit ADC, so we do not care about resolution. We also want something that is relatively small and easy to implement, as this circuit will be copied and pasted for however many columns our crossbar is. Because of all of this, we went with the flash ADC. While it can get larger with more bits, since we are only using it for 1-bit conversions, this will not be a concern. We also choose this one because of its fast speed, low latency in conversions, and because it is relatively simple to implement in a 1-bit form. We care mainly about speed in our project since we are testing the ReRAM crossbar for the best computational performance.

4.3 PROPOSED DESIGN

Discuss what you have done so far – what have you tried/implemented/tested?

4.3.1 Design Visual and Description

The figure below shows a basic ReRAM crossbar with analog circuitry around it that allows for a digital value to be input and a digital value to be output from the analog computation.

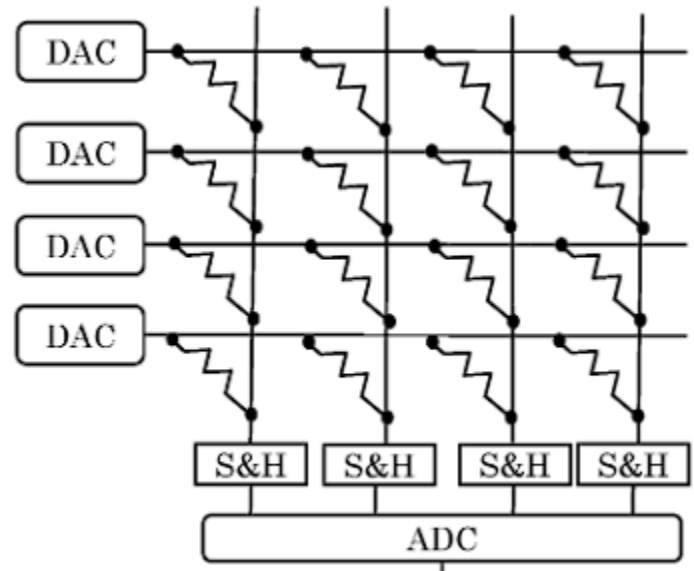


Figure 2: ReRAM Crossbar High-Level Diagram

The figure above shows the basic top-level schematic of how we will create our design to test the ReRAM crossbar for computational potential. First, we input digital values into the DAC. As they go through the DAC, the digital values are converted into analog voltages, which go through the ReRAM crossbar. At each column of the crossbar, current will accumulate, depending on the state of the crossbar and the digital values inputted. At each column, a transimpedance amplifier will be placed so that we can convert the current back to a voltage. Once that voltage is obtained, it will go through a sample and hold circuitry (S&H) and then be converted back to a digital value through a 1-bit ADC. This digital value is then inputted back into the Caravel harness to be read.

The figure below shows what one cell of the ReRAM crossbar will look like.

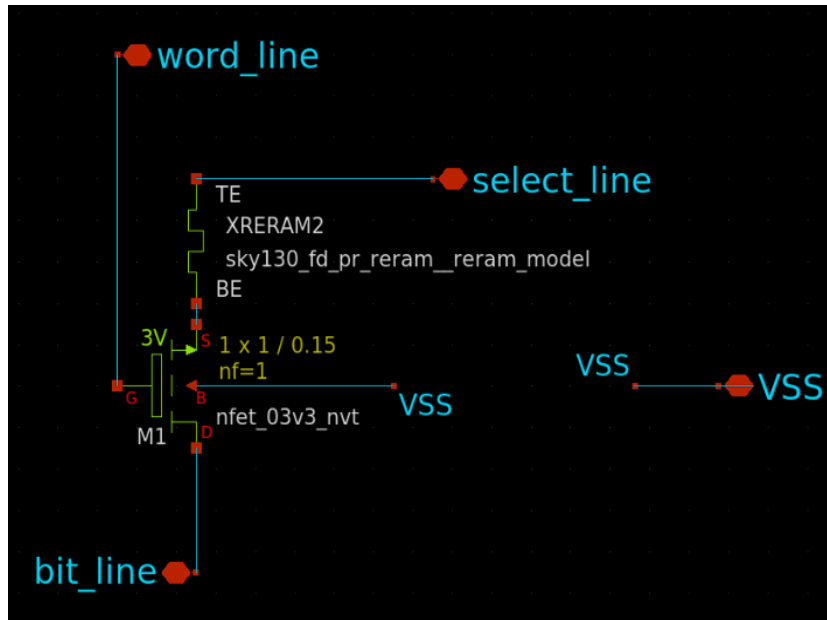


Figure 3: 1T1R Cell

As is seen in the figure, the cell consists of the ReRAM and a transistor. Thus, it is named a 1T1R cell. The gate of the transistor is going to be attached to the logic analyzer of the Caravel harness, which will write a weight to it. In the overall circuit, we will write the weights for an entire row of these cells at once. The select_line will be what is inputted into the cell. So, that means that the output of the 1-bit DAC will go into the select_line. Finally, the output of the cell will be the bit_line which will accumulate current. These currents will be added to the currents of other cells along that column that will then be converted to a voltage.

The figure below shows our whole system and how it will be integrated with the Caravel harness infrastructure:

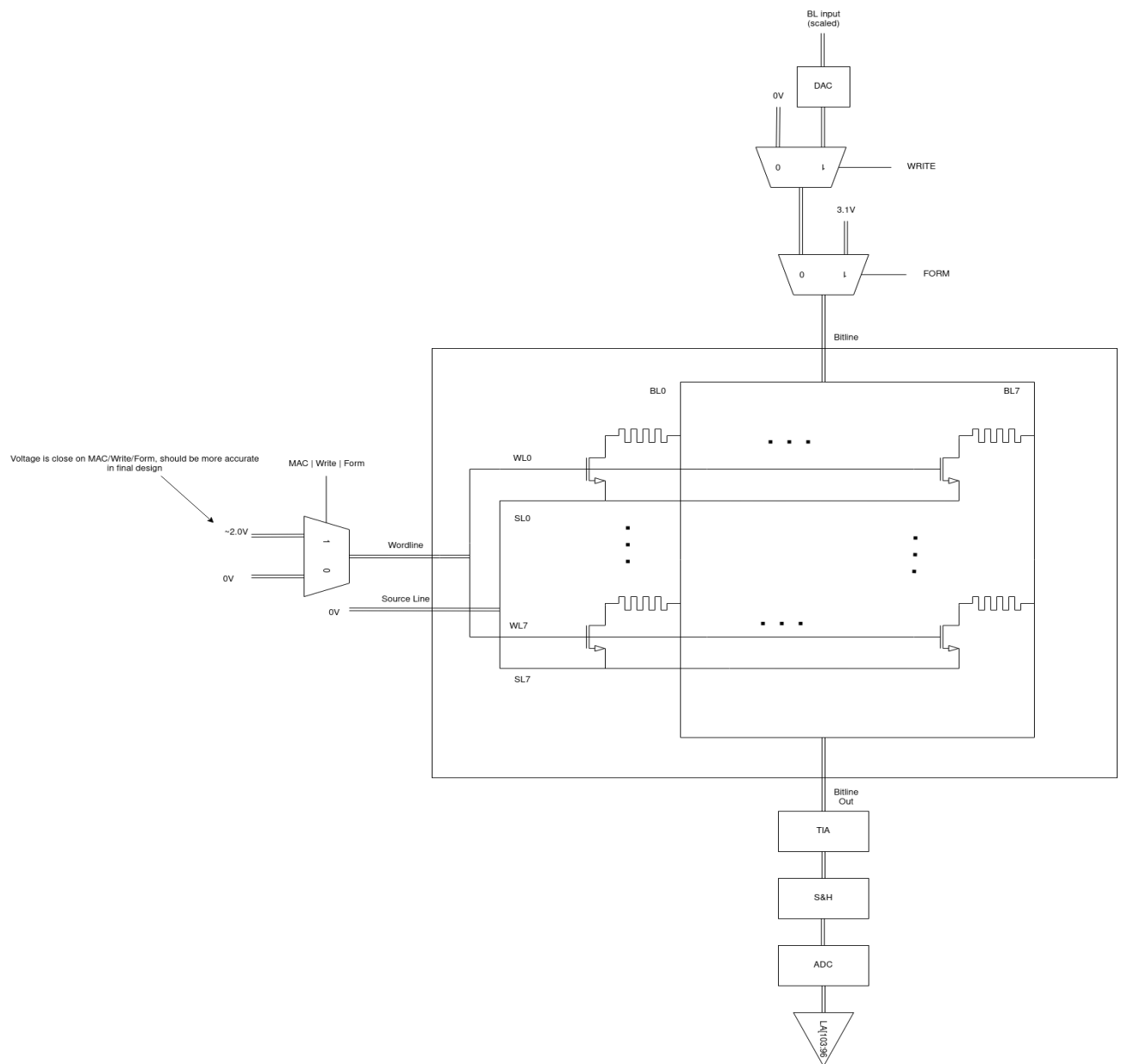


Figure 4: User Area Top Level Diagram

4.3.2 Functionality

In the real-world, a user will have our ASIC chip printed out on a board with numerous controls that are dictated by the Caravel harness. This board is provided in the fabrication process through the Efabless program. The user will be able to select digital inputs that they want to input into the chip to be computed by the ReRAM crossbar. They may also want to write weights to the ReRAM crossbar, which they will also be able to do with the circuitry around our chip. Once they input the values, they will then be able to read the digital output of these values on one of the output pins of the board. This design will meet all functional requirements of the ReRAM crossbar, as values will be able to be input and weights are able to be written to the cells.

4.3.3 Areas of Concern and Development

The primary concern of our project is fully instating all of the analog devices that will be used in this project and getting them past precheck. For each analog device, we will have to create a schematic, simulate that schematic, create a layout of that device, and pass DRC and LVS tests on that layout. Once we have that done for each component, we have to create a top schematic and run through everything again. Once that is done, we have to connect everything up in the analog wrapper that is provided by Efabless and try to pass precheck. As can be seen, by this description, there are a lot of steps, and that means a lot of places where things can go wrong or get bottlenecked for some time in our design. We will attempt to address this concern by being proactive and working ahead of schedule. If we run into any trouble, we will use a Slack channel that is used for open-source chip design for help.

4.4 TECHNOLOGY CONSIDERATIONS

Simulators:

- Ngspice
 - better integrates into Efabless framework
 - easier to setup
 - cannot simulate VerilogA
- Xyce
 - Harder to setup (needs to configure it properly)
 - Is able to simulate VerilogA
 - Requires other software (Trilinos)
 - Faster

For our simulator we chose Xyce because we are able to simulate VerilogA, which is how the ReRAM model is implemented. ReRAM is not able to be simulated in Ngspice because the mechanism is not implementable in SPICE.

User area communication:

- Logic analyzer
 - Easy to use
 - Easy to implement for hardware
 - has 128 bits
- GPIO
 - Can communicate with user area using configuration
 - Not very many available communication bits
- Wishbone
 - More flexible, follows a standard
 - Difficult to implement
 - Hard to do a handshake with analog circuitry

We decided to use the logic analyzer because it would be very difficult to use the wishbone, and the logic analyzer is strictly better than GPIO for our purposes.

4.5 DESIGN ANALYSIS

We are still working on implementing our design from 4.3, so we can not conclusively say whether our plan has worked out or not yet. However, the progress that we have made thus far is promising, and we believe we will be able to follow through on our plan. We are currently focusing our efforts on fully designing all of our analog components. This means that we are creating schematics of these components, verifying their functionality with testbenches, creating layouts, and running simulations on their extracted post-layout results. We have currently gone through this process of our 1-bit DAC and 1-bit ADC. We are still working on getting our transimpedance amplifier, ReRAM cell, and sample and hold circuitry.

As we go through this process, we may realize that certain components we design may need to be tweaked to work better with the overall design. So, we will have to iterate over our analog designs to ensure that everything functions as expected when put together. As we have not made it far enough to simulate multiple components together, we have not had to make any of these modifications or iterate over our planned design.

4.6 DESIGN PLAN

Our design plan is to design and test all analog components individually. Once we have verified that the individual components meet our functional requirements (this includes testing of the extracted post-layout results), we will move to testing the functionality of the individual components when put together. The goal of this is to ensure that the functionality of one device will not interfere with the functionality of another device. If we notice that there are undesired results when we simulate multiple components together, we will tweak the components until we do get the desired results. Once we have verified that all components work together, we plan on creating a top-level layout and hooking up the Caravel harness. Once our circuit is hooked up to the Caravel harness, we will run multiple test cases to ensure the functionality of our circuit still remains intact when hooked up the harness.

Going through this design plan should ensure that we meet all requirements of our project. We will be able to show the functionality of our circuit when not hooked up to the Caravel harness and when it is hooked up to the harness. With our layout, we will ensure that all DRC and LVS tests are passed so that our design will be able to pass precheck. With all of these parts of our design plan, we should be able to meet the goals of this project and successfully tape-out or design.

5 Testing

5.1 UNIT TESTING

Our design will consist of many units. The main units will be the analog circuitry, such as the DACs, ReRAM cells (1T1R), transimpedance amplifiers, sample and hold circuitry, and ADCs. These units will be thoroughly tested using the open-source tools we have available to us. For each device, we will have to make a schematic of it using XSchem. From that schematic, we will then create a testbench in Xschem to observe its behavior. These testbenches will be simulated using Ngspice or Xcye. Once we have verified if that device meets our design requirements and expectations, we will then have a layout created for it using Magic. We will then obtain the post-layout parasitics using Magic and simulate that parasitic netlist using Ngspice.

5.2 INTERFACE TESTING

The interface between each of our components will be wires. The number and name of the wires will vary based on what is needed from them. In our project, we have a user area which needs to be communicated with via a RISC-V CPU. This communication will be done over a configurable 128bit logic analyzer. Whether each bit is an input or output is configurable.

The composition of multiple units is being tested the same way as individual components, which is creating testbenches in XSchem, simulating through Ngspice, creating layout using Magic, verifying LVS, and extracting parasitics to rerun simulations with.

5.3 INTEGRATION TESTING

The main integration path in our design comes from the connection of all of our analog components to the outside Caravel harness. This path is fairly linear when we are computing data. Digital data will be entered into our circuit from the Caravel harness, where our chip will do computations in the analog domain before returning a digital value back to the harness. Before hooking up our circuit to the Caravel harness, we will ensure that the analog circuitry is behaving as expected. This will be done by creating a top-level schematic of all the components connected together in Xschem. We will then create a testbench that simulates the behavior of this circuit. This testbench will simulate the conditions that the circuit will be in when connected to the Caravel harness. We will run these simulations using Ngspice and Xcye. Once this is done, we will use Magic to connect together all of the components and extract post-layout parasitics. We will then simulate these parasitics, and if the results are still acceptable, we will move on to connect our circuit to the outside harness. We will run further testing on this to ensure that everything still behaves as expected.

5.4 SYSTEM TESTING

This part is awkward; unlike with fully digital designs, we are unable to fully verify our product. We must test in two separate parts. We currently have a fully functional digital implementation of

our project, which has the same behavior as the ReRAM crossbar, so it demonstrates the same functionality in digital simulation. The fully digital simulation allows us to determine if our port mappings are going to work, along with if our drivers will work.

For the second part, we need to determine if our analog ReRAM crossbar component's behavior matches our digital implementation. This must be done through analog simulations since it uses analog components. Our strategy here consists of writing many top-level tests to ensure all of the operations are working.

5.5 REGRESSION TESTING

The testing will be very iterative to ensure that everything works properly together. In our design, we need to make sure that each unit works not only independently but they all work together. For this regression testing, we will first build and simulate a schematic of each individual unit to ensure it works as intended using Xschem. Next, we will hook up two units together that will be connected in our final design. For example, we will hook up the 1-bit DAC with the 1T1R cell. We will then create a testbench and run simulations using Ngspice and Xcye to view how they behave together. If we observe that they are not behaving as expected, we will have to tweak the individual components and run simulations until they do behave as expected. Once we have verified that the two components work together, we would then connect one more component and run through the same process again. We would repeat this process until all components have been connected and simulated with acceptable results. Running through the testing this way, we can ensure that all components will work together, but it will also allow us to have an easier time troubleshooting if something deviates from the expected behavior, as we will be able to identify the component that is causing the problem.

5.6 ACCEPTANCE TESTING

Our design will be tested for non-functional acceptability through the precheck process the Efabless uses to verify that our design meets certain requirements. Some of these prechecks include DRC and LVS checks. This precheck essentially runs multiple DRC checks to ensure that our layout does not violate any rules and is able to be fabricated on silicon. For the functionality, we will have to run simulations and test cases on our design to verify that it behaves as expected. These simulations will be done using Verilog to simulate how our circuit will operate within the Caravel harness. From these simulations, we will obtain waveforms that will help verify the functionality of our chip.

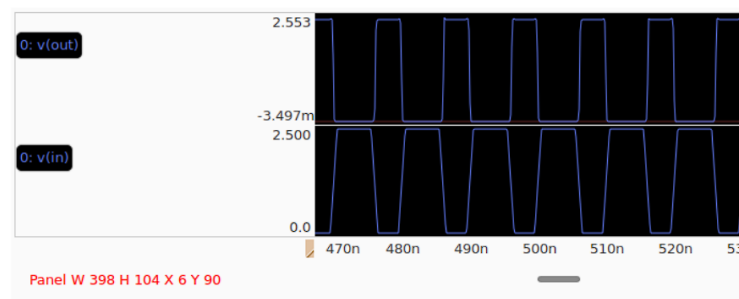
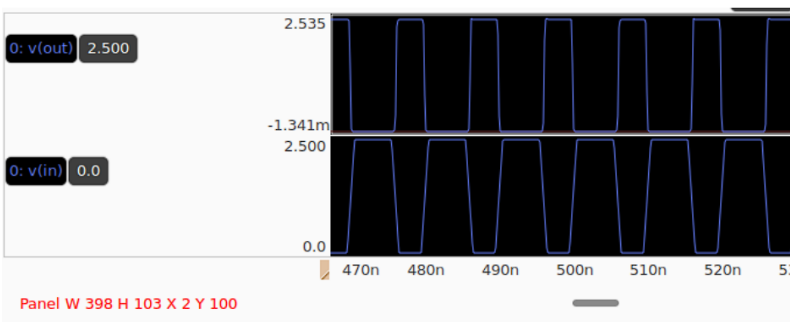
5.7 SECURITY TESTING (IF APPLICABLE)

5.8 RESULTS

As we are still working on our design, we do not have the full results of testing yet. However, we do know what form the results of tests will be. For each component, we will have the ideal test results and the post-layout parasitic test results. Once we have those results for each component,

we will begin to combine components and run ideal tests and post-layout parasitic tests. We will obtain these ideal and post-layout results for the combination of components until all of our devices have been connected. We will then connect our design to the harness and run different test cases on our circuit. With all of these results, we will be able to determine whether or not our design meets the requirements.

Below is an example of the ideal test results and the post-layout parasitic test results of a basic inverter.



Original simulation of inverter (no parasitics)

Simulation of inverter with parasitics

Figure 5: Example Simulation Results

6 Implementation

We have already begun some of the implementations of our design. We have begun work on designing all of our analog components. We have successfully gotten our 1-bit DAC through the entirety of our testing plan for an individual component. We are close to getting our 1-bit ADC through this process as well.

For next semester, we plan to continue the work that we have started this semester on our analog parts. We plan to get all of our components through individual testing and start testing their functionalities when put together. Once we have the desired functionality, we will hook up the top-level layout of our design to the Caravel harness. Once this is done we will then run through test cases for our circuit when hooked up to this harness to ensure that the functionality of our circuit remains. Finally, we will do any last touch-ups needed that will be required for passing precheck and getting a successful submission of our design through the Efabes program.

7 Professionalism

7.1 AREAS OF RESPONSIBILITY

Pick one of IEEE, ACM, or SE code of ethics. Add a column to Table 1 from the paper corresponding to the society-specific code of ethics selected above. State how it addresses each of the areas of seven professional responsibilities in the table. Briefly describe each entry added to the table in your own words. How does the IEEE, ACM, or SE code of ethics differ from the NSPE version for each area?

Area of responsibility	Definition	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
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
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	2. To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist 3. To be honest and realistic in stating claims or estimates based on available data
Health, Safety, Well-Being	Minimize risks to safety, health, and	Hold paramount the safety, health, and welfare of the public.	1. To accept responsibility in making decisions consistent with the safety,

	well-being of stakeholders.		health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment
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 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 so as to enhance the honor, reputation, and usefulness of the profession.	7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others 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 10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Table 5: Areas of Responsibility

- Work Competence
 - The IEEE Code of Ethics covers this responsibility with policies 5 and 6. Policy 5 talks about how to fully understand the technologies used, how they are used, and the consequences of using the technology Policy 6 talks about making sure that an individual is competent in their knowledge before taking on tasks.
 - These codes of ethics essentially cover what is in the NSPE canon, except they do not explicitly state to ensure avoiding deceptive acts.
- Financial Responsibility
 - The IEEE Code of Ethics covers this responsibility with policy 4. This policy just means not to take in bribes, whether it be monetary or for other benefits such as promotions.
 - This part of the Code of Ethics differs heavily from the NSPE canon, as it does not cover acts in the best interests of employers or clients as trustees. Nothing in the code of ethics went into detail about keeping up financial responsibility for others.
- Communication Honesty
 - The IEEE Code of Ethics covers this responsibility with policies 2 and 3. Policy 2 covers avoiding conflicts of interest to the best of one's ability, and if they do arise, make sure to notify the proper parties. Policy 3 talked about making sure to be honest in claims made according to available data.
 - The IEEE code of ethics covers this NSPE canon fully, as they both talk about ensuring to make truthful claims only based on objective facts.
- Health, Safety, Well-Being
 - The IEEE Code of Ethics covers this responsibility with policy 1. Policy 1 talks about taking responsibility for the health and safety of the public when making any decision. If a decision is made that might lead to the endangerment of the public, it must be fully disclosed.
 - There is a slight difference in the wording between the IEEE Code of Ethics and the NSPE canon. The NSPE canon states to hold paramount the safety and health of the public, while the IEEE code of ethics only says to accept full responsibility when making decisions that impact the health and safety of the public.
- Property Ownership
 - The IEEE Code of Ethics covers this responsibility with policy 9. Policy 9 mentions avoiding injury to others or their properties by ensuring not act in harmful ways.
 - The NSPE canon and the IEEE code of ethics differ slightly here. The NSPE canon has slightly looser wording around property ownership as it just mentions to act for each employer or client. Whereas the IEEE code of ethics makes a specific point to not endanger property through malicious acts.
- Sustainability
 - The IEEE code of ethics covers this responsibility with policy 1. Policy 1 was already talked about under the responsibility of health, safety, and well-being.

- There was no NSPE canon that covered sustainability, while there is an IEEE code of ethics policy that specifically mentions taking responsibility to not harm the environment.
- Social Responsibility
 - The IEEE Code of Ethics covers this responsibility with policies 7, 8, and 10. Policy 7 covers taking honest criticism of technical work, acknowledging errors in work, and properly crediting all contributions to work. Policy 8 talks about treating everyone fairly and not discriminating for any reason. Finally, policy 10 covers assisting others to follow the policies outlined in the IEEE code of ethics.
 - The NSPE canon makes a point to explicitly state for people to conduct themselves so as to enhance the honor, reputation, and usefulness of the profession. The IEEE Code of Ethics does not explicitly state the reason for acting with social responsibility, but it lists out ways to act responsibly.

7.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

- Work Competence:

It is definitely important. The work must be done in a highly constrained environment to ensure it is able to be submitted to Efabless for fabrication. It also has a strict deadline for when we can submit it. This combination of factors makes work competence important for our project.

We are doing a medium job of doing this. We are still in a discovery phase, so we aren't making tons of progress, but we are still learning with future prospects to work faster and more efficiently in the future.
- Financial Responsibility:

Not applicable. Everything we are doing is open source and is free.
- Communication Honesty:

This is applicable. We need to make sure we are honest when we are demonstrating our accomplishments to our project advisor and client.

We are doing a great job of this. We do not lie or try to bend the truth when communicating with them
- Health, Safety, Well-Being:

Not applicable. Our project does not have the potential to affect anybody's health or well-being.
- Property Ownership:

Not particularly relevant. We are working with open source, so it is expected that we treat other people's intellectual property with the license they provide. These licenses typically say that we can do whatever we want with the information. Nobody owns anything.

- Sustainability:
Not particularly relevant. We don't really have the ability to protect resources.
- Social Responsibility:
We are producing a product that has the potential to benefit humanity by improving computation technology. We are doing a good job of this because we are working on producing this particular technology.

7.3 MOST APPLICABLE PROFESSIONAL RESPONSIBILITY AREA

The most applicable responsibility is Social Responsibility. In the context of our project, it just means that we are contributing to society. We are doing this specifically by contributing to the open-source hardware community. So far, we have been able to discover and report a major bug in a framework, which has led to many users of the framework not be able to use it. This bug was later fixed, allowing other users to use the framework again. We are also contributing by expanding the number of people who have knowledge of these analog fabrication tools in Iowa State, expanding the knowledge base of future senior design projects who want to attempt similar analog ASIC problems as us.

8 Closing Material

8.1 DISCUSSION

We are expecting to have a ReRAM crossbar that we can communicate with via our Caravel harness. The ReRAM crossbar should be able to perform multiply and accumulate operations. We will be able to load the weight of the matrix into the ReRAM row by row, then load in the vector being multiplied. The caravel harness will then receive the result.

Currently, we do not have this implemented because the project is not done.

8.2 CONCLUSION

Currently, we have completed our research for our project, and now we are working on implementing individual components. We currently have DAC fully implemented, with the ADC being worked on. We have not started on the TIA or the sample and hold. We have completed a full schematic design and a digital behavioral implementation for testing purposes.

By the end of this project, we need to fully implement all of the components we listed along with a ReRAM crossbar, then connect these components together to form our user area. We then need to be able to interface with it through the Caravel harness using the logic analyzer.

8.3 REFERENCES

A. Shafiee et al., "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, Korea (South), 2016, pp. 14-26, doi: 10.1109/ISCA.2016.12.

F. Zahoor, T. Z. Azni Zulkifli, and F. A. Khanday, "Resistive random access memory (RRAM): An overview of materials, switching mechanism, performance, multilevel cell (MLC) storage, modeling, and applications - discover nano," SpringerLink, 22-Apr-2020. [Online]. Available: <https://link.springer.com/article/10.1186/s11671-020-03299-9>.

"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. [Online]. Available: <https://sky130-fd-pr-ram.readthedocs.io/en/latest/index.html>.

8.4 APPENDICES

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc., PCB testing issues etc., Software bugs etc.

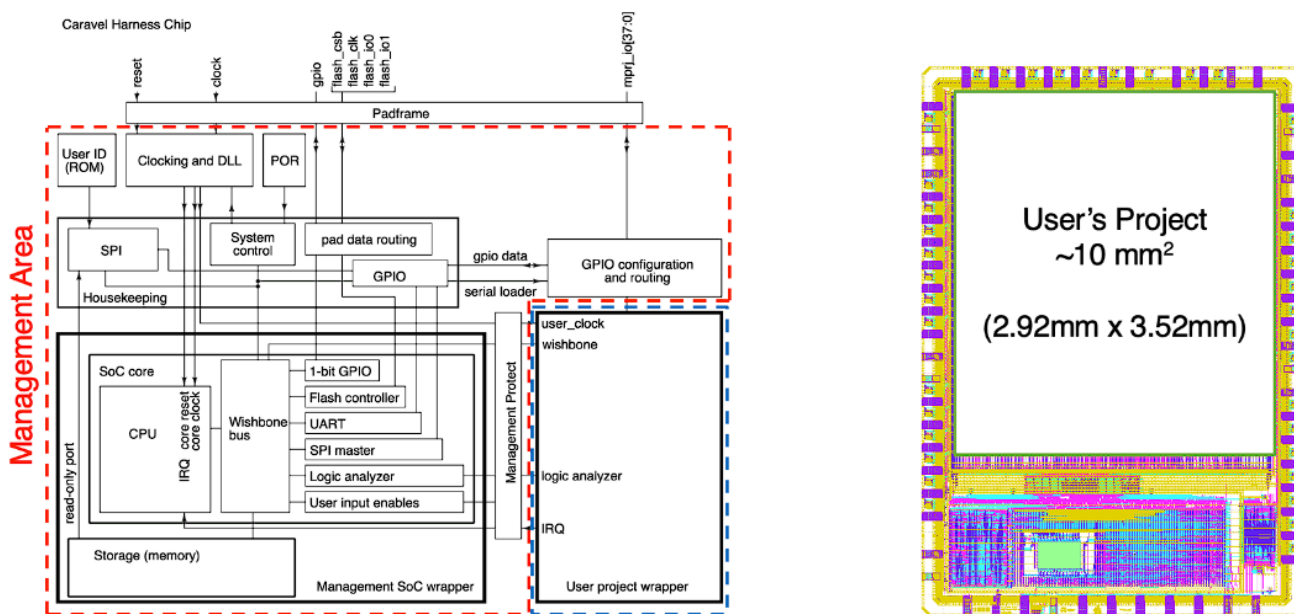


Figure 6: Caravel Harness Diagram

Documentation/Software/Hardware Manuals:

Magic documentation: <http://opencircuitdesign.com/magic/index.html>

Netgen documentation: <http://opencircuitdesign.com/netgen/index.html>

Ngspice documentation: <https://ngspice.sourceforge.io/docs/ngspice-40-manual.pdf>

Xschem documentation: https://xschem.sourceforge.io/stefan/xschem_man/xschem_man.pdf

Xyce documentation: https://xyce.sandia.gov/files/xyce/Xyce_Users_Guide_7.6.pdf

Sky130 nm integration with Magic and Netgen (created by us):

https://docs.google.com/document/d/1_9CwGsosCGFOkvt3-wE445qPI8kBoizo8M2xqubthE/edit

8.4.1 Team Contract

Team Members:

1) Joshua Thater _____ 2) Aiden Petersen _____

3) Matthew Ottersen _____ 4) Regassa Dukele _____

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
 - a. Monday 11-1 in TLA
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
 - a. Discord group chat
 - b. Microsoft Teams
3. Decision-making policy (e.g., consensus, majority vote):
 - a. Consensus
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
 - a. As a team, take notes as needed
 - i. Notes are stored in shared Google Doc

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
 - a. Expected to be on time for all meetings unless the team is notified beforehand.
 - b. During the meeting, each member should contribute ideas, promote questions, and make progress on the project.
 - c. It is expected that during meetings, the focus is solely on the project and no outside classwork or other distractions.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
 - a. It is expected that each member will contribute to team assignments equally.
 - b. All timelines and deadlines for deliverables should be on time unless otherwise communicated.
3. Expected level of communication with other team members:

- a. Check group chats at least once a day for any updates.
- b. Weekly updates on what each team member is working on for that week.
- 4. Expected level of commitment to team decisions and tasks:
 - a. Each team member should voice their opinion when it comes to decisions on the project.
 - b. If a team member says they will do a task, they are expected to fulfill that task.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
 - a. Joshua Thater - team organization & documentation
 - b. Aiden Petersen - Software documentation. Primary client communicator.
 - c. Matt Ottersen - Individual Component Design, Simulations
 - d. Regassa Dukele - Individual Component Design, Simulations
2. Strategies for supporting and guiding the work of all team members:
 - a. Make sure to always be in communication with the team and document what progress is being done.
 - b. If a member runs into trouble with something, they should share the issue in one of the group chats, so another team member may assist.
3. Strategies for recognizing the contributions of all team members:
 - a. Having a record of completed tasks and research.
 - b. Making sure that each member feels their contributions are helpful towards completing the project.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
 - a. Joshua Thater: Effective communication, Verilog, experience in analog and digital circuit design, some experience with mixed-signal circuit design, Linux terminal, PCB design, solid troubleshooting skills.
 - b. Aiden Petersen: Very experienced in Linux and troubleshooting it, verilog, C, digital design, general software engineering experience, computer architecture.
 - c. Regassa Dukele: I have some experience in schematic and layout design, including creating schematics, developing board layouts, and troubleshooting design issues.
 - d. Matt Ottersen: Experience in Analog circuit design, testing and layout. Some experience with digital, and mixed signal design. Also some experience with Verilog
2. Strategies for encouraging and support contributions and ideas from all team members:
 - a. Making sure that each person has shared their opinion.
 - b. Help members express their ideas in a positive environment.
 - c. Hold brainstorming sessions where everyone can contribute ideas and express their feeling
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?)
 - a. By informing the team that they feel their contributions are being obstructed.

- b. If issues still persist, contact the advisor.
- c. Identify any potential issues or concerns around collaboration in the team and tell them to fix the problem.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
 - a. Obtain all of the background information needed and begin working on the Efabless MPW submission
 - b. Fully design ReRAM ASIC to be fabricated, but not necessarily have it implemented.
2. Strategies for planning and assigning individual and teamwork:
 - a. Assignments will be based on the skills each individual has in a specific area.
 - b. Assignments and plans will meaningfully progress the project.
 - c. Individuals will be assigned equal amounts of work.
3. Strategies for keeping on task:
 - a. Weekly client meetings where we will discuss progress and define deliverables by the next week.
 - b. Weekly in-person group meetings to discuss and assign tasks, along with asking for help if needed.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
 - a. We will discuss these infractions with the member in a weekly meeting, try to figure out why they are happening and how to resolve them.
2. What will your team do if the infractions continue?
 - a. We will discuss these infractions with our advisor and see if we can resolve the issues. If we cannot, we may need to remove them from the project.

- a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
- b) *I understand that I am obligated to 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.*

1) Joshua Thater _____	DATE 02/19/2023 _____
2) Aiden Petersen _____	DATE 02/19/2023 _____
3) Regassa Dukele _____	DATE 02/19/2023 _____
4) Matt Ottersen _____	DATE 02/19/2023 _____