

ASIC Fabrication

DESIGN DOCUMENT

sddec22-17

Dr. Duwe & Dr. Huang

[Dawood Ghauri](#) - Researcher / Design Workflow

[Constantine Mantas](#) - Researcher / Team Organization Leader

[Soma Szabo](#) - Researcher / Component Design

[Courtney Violett](#) - Researcher / Testing

sddec22-17@iastate.edu

<https://sddec22-17.sd.ece.iastate.edu>

Revised: 04/22/2022

Executive Summary

Development Standards & Practices Used

- IEEE Standard VITAL ASIC (Application Specific Integrated Circuit) Modeling Specification - We are designing an ASIC using a modeling standard so this will allow us to more clearly communicate our design to other engineers.
- IEEE Standard Testability Method for Embedded Core-based Integrated Circuits - We are designing an ASIC using a testing standard so this will allow us to more accurately verify our testing process.
- IEEE Standard for Integrated Circuit (IC) Open Library Architecture (OLA)- This is useful for our ASIC design because it covers ways for integrated circuit designers to analyze chip timing and power consistently across a broad set of electric design automation (EDA) applications.

Summary of Requirements

- The design is selected and manufactured by eFabless in their Open MPW shuttle program.
- The digital design is able to take in a block header and calculate the next hash of a block by testing various nonce values to compute a valid hash.
- The fabricated ASIC is able to perform the expected tasks as compared to the simulated pre-silicon design.

Applicable Courses from Iowa State University Curriculum

- EE 330: Integrated Electronics
- CPRE 381: Computer Organization and Design
- CPRE 288: Embedded Systems I
- CPRE 480: Graphics Processing and Architecture

New Skills/Knowledge acquired that was not taught in courses

- Researching use of new design software using github and community communication channels such as slack.
- The ability to troubleshoot problems whose solutions are much more complex than a simple google search.
- Utilizing Openlane to harden a digital design which allows for place and route of the design to be done digitally and a schematic of the design to be output.
- Utilizing Open-Source hardware development software which often relies on an interplay of numerous libraries and lacks adequate documentation.
- Creating and using a linux docker to run a local dev environment.
- Creating a plan for pre and post-silicon bring-up that highlights how the chip will be tested and debugged.
- Learning hardware design skills and languages such as Verilog and SystemVerilog to create our design.

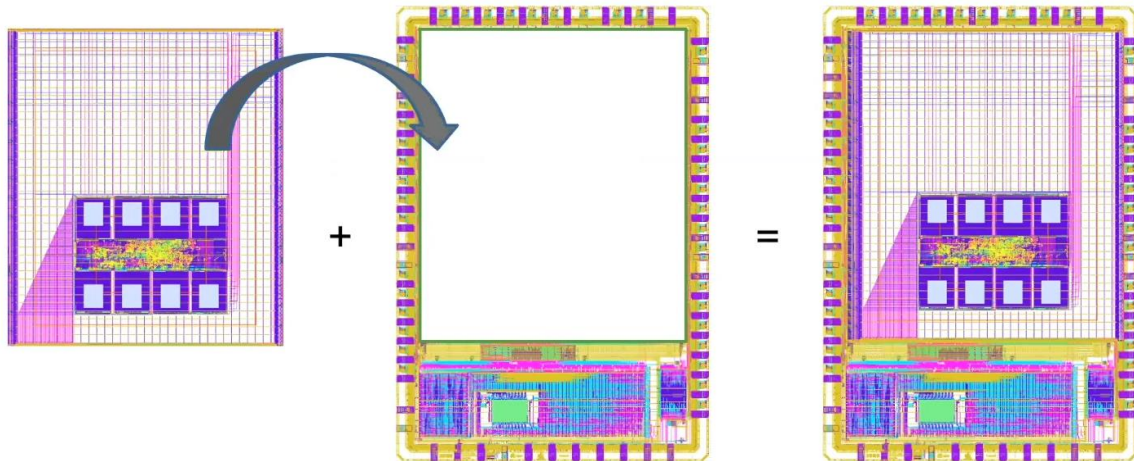
Table of Contents

1 Team	11
1.1 Team Members	11
1.2 Required Skill Sets for Your Project	11
1.3 Skill Sets covered by the Team	11
1.4 Project Management Style Adopted by the team	11
1.5 Initial Project Management Roles	11
2 Introduction	12
2.1 Problem Statement	12
2.2 Requirements & Constraints	12
2.3 Engineering Standards	13
2.4 Intended Users and Uses	13
3 Project Plan	14
3.1 Project Management/Tracking Procedures	14
3.2 Task Decomposition	14
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	15
3.4 Project Timeline/Schedule	16
3.5 Risks And Risk Management/Mitigation	17
3.6 Personnel Effort Requirements	18
3.7 Other Resource Requirements	19
4 Design	19
4.1 Design Context	19
4.1.1 Broader Context	19
4.1.2 User Needs	20
4.1.3 Prior Work/Solutions	20
4.1.4 Technical Complexity	21
4.2 Design Exploration	21
4.2.1 Design Decisions	21

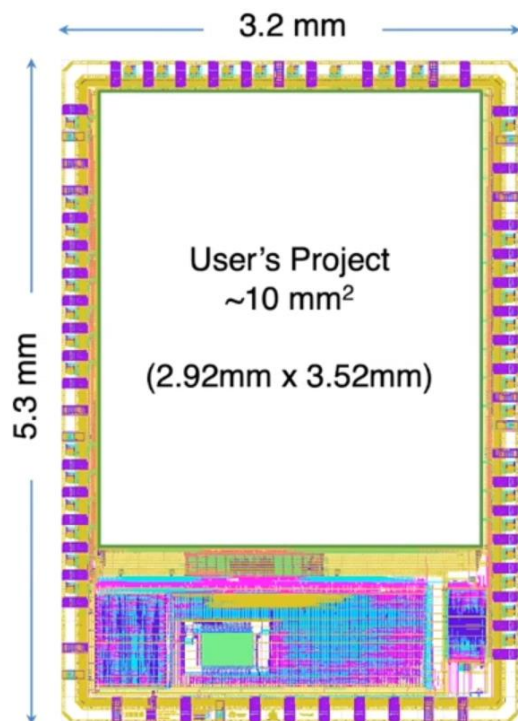
4.2.2 Ideation	22
4.2.3 Decision-Making and Trade-Off	22
Proposed Design	23
4.3.1 Design Visual and Description	23
4.3.2 Functionality	25
4.3.3 Areas of Concern and Development	25
4.4 Technology Considerations	25
4.5 Design Analysis	26
Design Plan	26
5 Testing	28
5.1 Unit Testing	29
5.2 Interface Testing	30
Integration Testing	30
System Testing	31
Regression Testing	31
Acceptance Testing	31
Security Testing	31
Results	32
6 Implementation	32
7 Professionalism	33
Areas of Responsibility	33
7.2 Project Specific Professional Responsibility Areas	37
7.3 Most Applicable Professional Responsibility Area	38
8 Closing Material	38
8.1 Discussion	38
8.2 Conclusion	39
8.3 References	39
8.4 Appendices	39

List of figures/tables/symbols/definitions

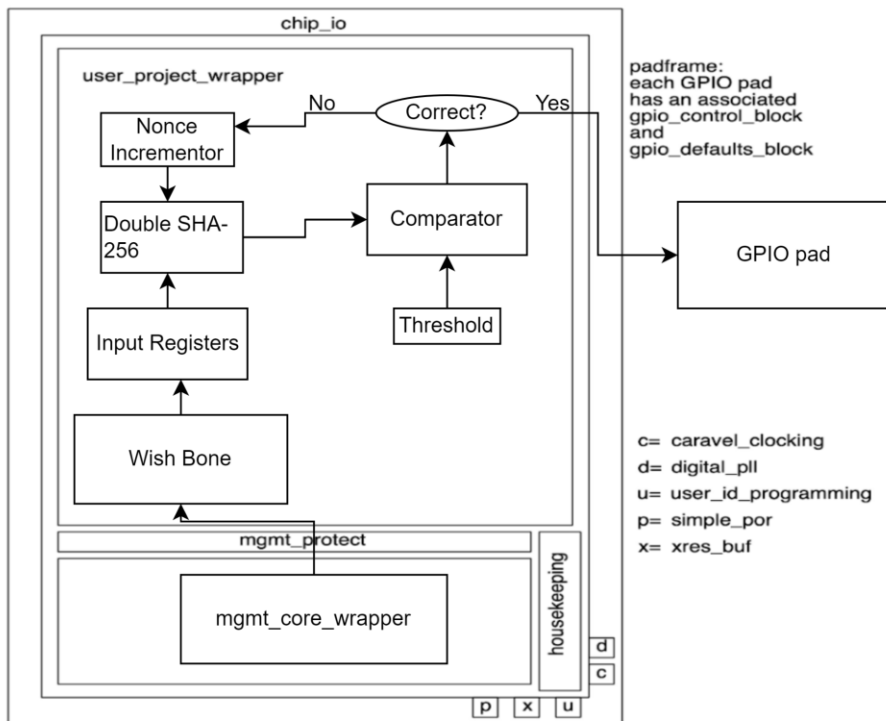
Caravel process image:



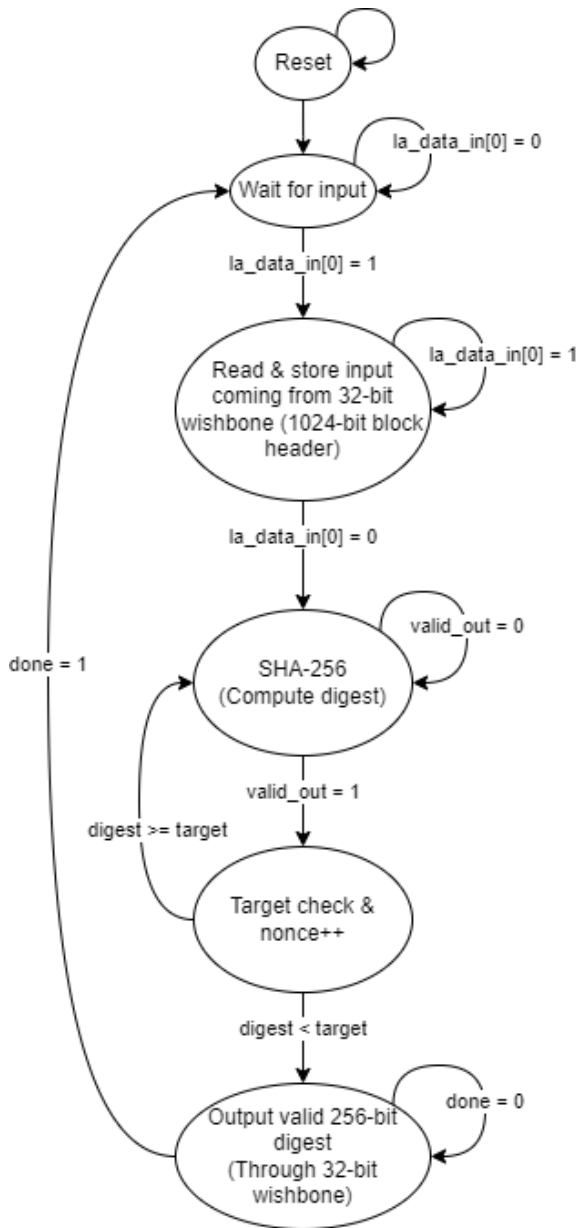
Caravel Harness:



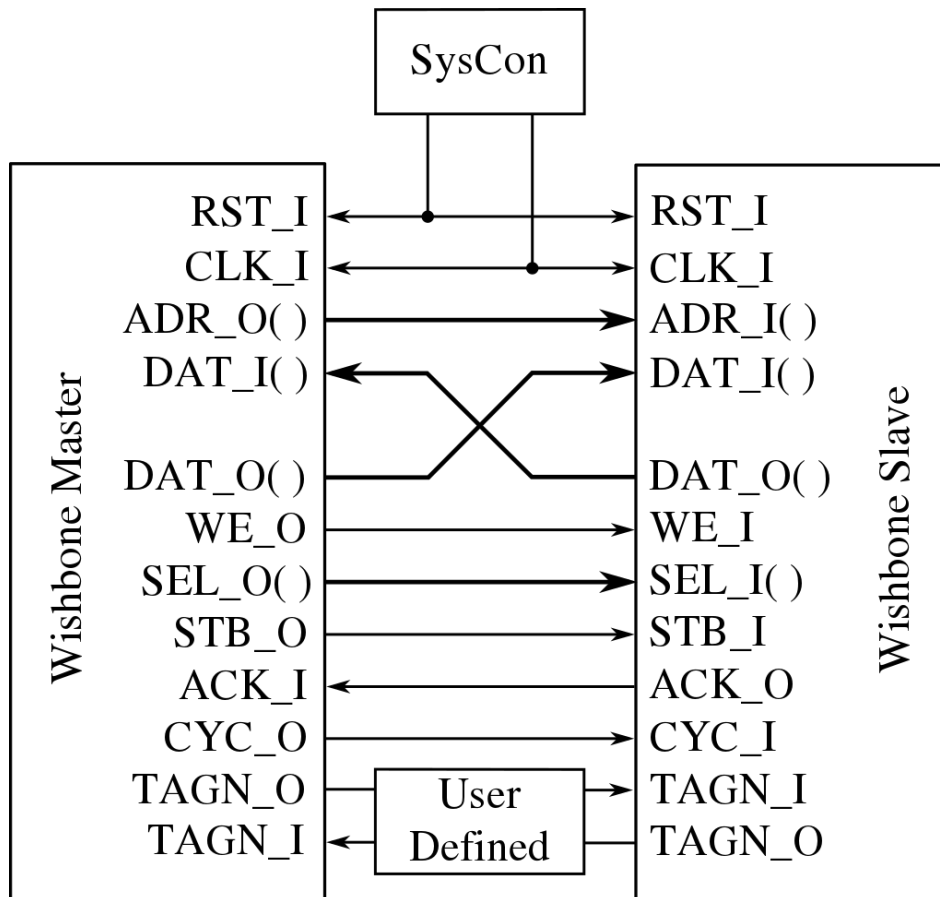
Initial design sketch:



State machine design plan:



Wishbone model:



Bring up plan:

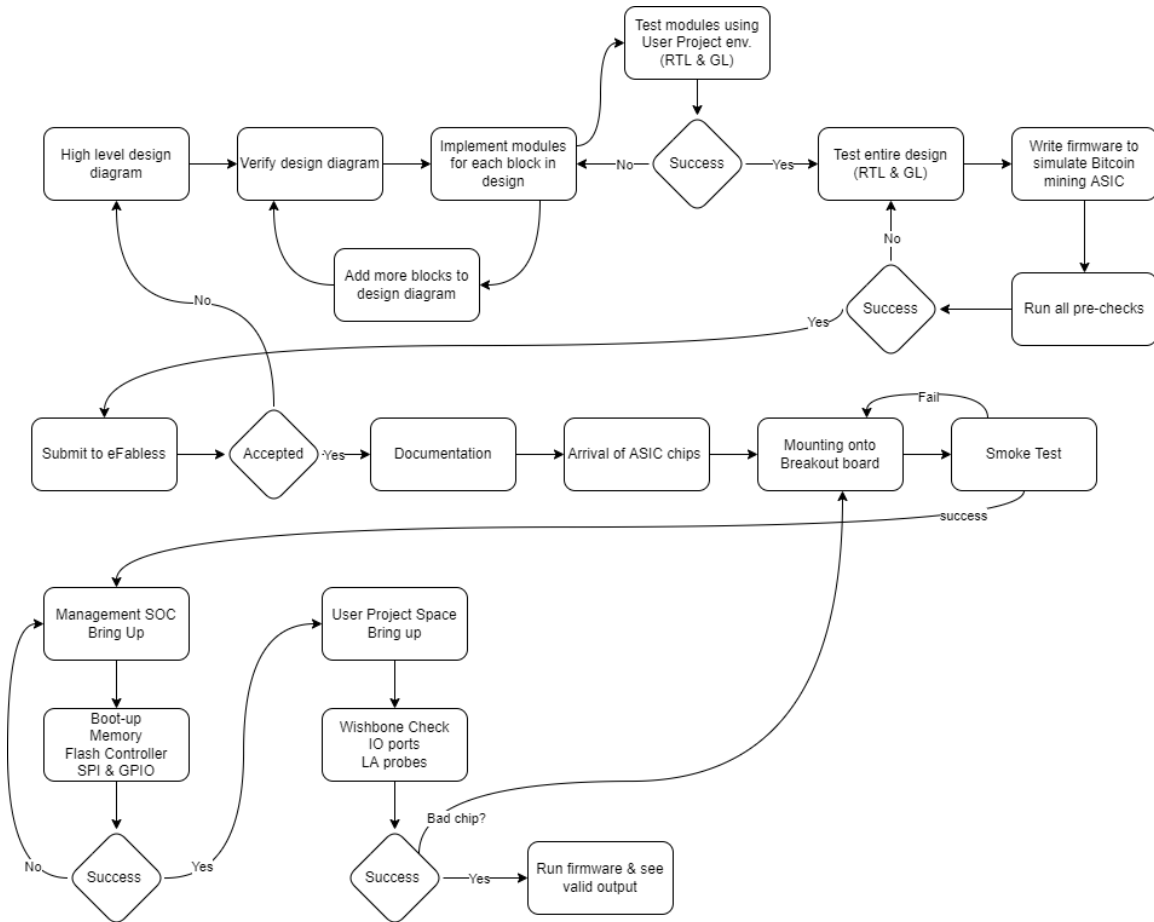
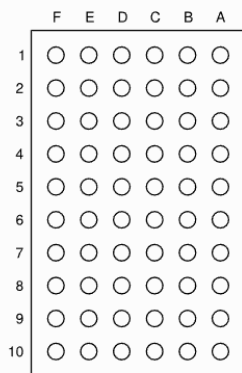


Image of pinout for ASIC:

Ball assignment (6x10 WLCSP)



Package as viewed from the bottom.

1 Team

1.1 TEAM MEMBERS

Constantine Mantas - a Computer Engineering major with an interest in hardware design. Through various classwork, internships, and projects, Constantine has gotten experience with various programming languages like Java, C, and Python, and computer hardware design with technologies like ModelSim, Verilog, and VHDL

[Soma Szabo](#) - a Computer Engineering major, pursuing the Master of Engineering concurrent degree program. His classes and work experiences have provided him with knowledge in programming techniques, software languages such as Java, TCL, C/C++, Python, and Dart, as well as hardware description languages such as VHDL and Verilog. He also has experience with computer architecture and related tools such as ModelSim, Vivado, Cadence, and Altium Designer (PCB design).

Courtney Violet - Is a Computer Engineering major. He has worked through a variety of class work and projects, both for class and personal projects, has gained experience in programming languages such as C, Java, HTML, CSS, computer hardware design, and circuit design. He also has experience with tools such as android studio and ModelSim.

[Dawood Ghauri](#) - a Computer Engineering major with a minor in Physics. His class experiences have developed his knowledge with respect to programming experiences in C, C++, Java, Python, VHDL, and Verilog

1.2 REQUIRED SKILL SETS FOR YOUR PROJECT

Some of the skill sets required for this project are:

- Ability to use and debug Verilog/SystemVerilog
- Ability to set up development environment in Linux
- Ability to use C to drive testing for Verilog/SystemVerilog components
- Understand the fabrication process for an ASIC

1.3 SKILL SETS COVERED BY THE TEAM

All of the required skills are covered by each team member.

1.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

The Project management style that we will use for this project is an agile style. We began the project with a waterfall approach when doing the research, but now that we have begun integration of components into the project framework we have begun using an agile methodology where each week we go over tasks that have been completed from the previous week, and then discuss tasks to complete for that week.

1.5 INITIAL PROJECT MANAGEMENT ROLES

Role	Team Member
Team Organization	Constantine Mantas
Design Workflow	Dawood Ghauri
Individual Component Design	Soma Szabo
Testing	Courtney Violett

2 Introduction

2.1 PROBLEM STATEMENT

Our group is designing a process for future students to fabricate a microchip through the Open MPW Shuttle program. To do this, we are creating a bring-up and testing plan for chip fabrication that will provide a streamline process for fabricating an application specific integrated circuit (ASIC). The focus of our design will be to create a test bed for evaluating approximate Bitcoin mining implementations and act as pioneers using Open MPW for future students wishing to create custom ASICs.

2.2 REQUIREMENTS & CONSTRAINTS

Functional Requirements (Specification):

- Full Chip Simulation passes for RTL and GL (gate-level)
- The hardened Macros are LVS and DRC clean
- The project contains a gate-level netlist for user_project_wrapper at verilog/gl/user_project_wrapper.v
- The hardened user_project_wrapper adheres to the same pin order specified at pin_order
- The hardened user_project_wrapper adheres to the fixed wrapper configuration specified at fixed_wrapper_cfgs
- XOR check passes with zero total difference
- The design passes the mpw-precheck

Resource Requirements:

- The project repo adheres to the same directory structure in this repo.
- The project repo contains info.yaml at the project root.
- Top level macro is named user_project_wrapper.
- Openlane summary reports are retained under ./signoff/

Qualitative Aesthetics Requirements:

- Not relevant for our project.

Economic/Market Requirements:

- Ensuring all technologies used are open-source. No monetary gain from design.

Environmental Requirements:

- As we are not in control of the fabrication process itself (managed by eFabless), there are not any environmental requirements relevant to us.

UI Requirements:

- Our design will not have a UI; therefore, UI requirements are not applicable.

2.3 ENGINEERING STANDARDS

IEEE Standard VITAL ASIC (Application Specific Integrated Circuit) Modeling Specification - We are designing an ASIC using a modeling standard so this will allow us to more clearly communicate our design to other engineers.

IEEE Standard Testability Method for Embedded Core-based Integrated Circuits - We are designing an ASIC using a testing standard so this will allow us to more accurately verify our testing process.

IEEE Standard for Integrated Circuit (IC) Open Library Architecture (OLA)- This is useful for our ASIC design because it covers ways for integrated circuit designers to analyze chip timing and power consistently across a broad set of electric design automation (EDA) applications.

2.4 INTENDED USERS AND USES

The primary benefactor of the results of our project will be our customers and specifically our advisor and customer Prof. Duwe. His hope is to use the knowledge and example of our project to be able to continuously have teams developing microchips through Efabless.

Another major benefactor of our project will be the preceding team to ours as they will not only use our project as an example for their own project. They will also benefit from our in depth documentation on how to develop within Efabless, but also on our documentation of the bring up process that they will be performing on our chip.

A potential large user group is bitcoin miners as our project focus is to create a chip that improves the hashing time. They could use this hardware to improve the efficiency of their mining operations. This is dependent on the improvement of the hashing speed and the compatibility of the chip with current mining rigs.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

As we were becoming familiar with the provided design environment, we initially worked in a waterfall workflow. However, once we became familiar with building the environment, we worked in an agile workflow. We started in a waterfall management style as it has allowed us to become more familiar with how to develop within Efabless. Once we all become familiar with it, an agile management style will be easier to work in. This will allow us to quickly move through and test various components as we implement and integrate them into the larger project.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

We are currently tracking tasks using a Microsoft Teams task board and working collaboratively on our codebase using Github.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

3.2 TASK DECOMPOSITION

Design an Open MPW ASIC submission

1. Setup of efabless dev environment
 - a. Setup Github repo
 - b. Clone caravel_user_project template into repo
 - c. Setup local linux environment to build project
 - d. Test baseline/template project to ensure everything works
2. Plan what we will design
 - a. High level system design
 - b. Functionality design for each component
3. Code components
 - a. Verilog development of each component used in the high level design
4. Test components
 - a. Testbenches for each individual component to verify functionality
 - b. Potential component specific feedback stage
5. Bring components together into a system
 - a. Put components together in the larger wrapper to create a final system
 - b. Test integrated components within the system to verify they are communicating correctly.
6. Test the system
 - a. Create testing that will cover all critical areas and ensure functionality of the system
7. Feedback from Dr. Duwe
 - a. Ensure the system and components meet expectations.

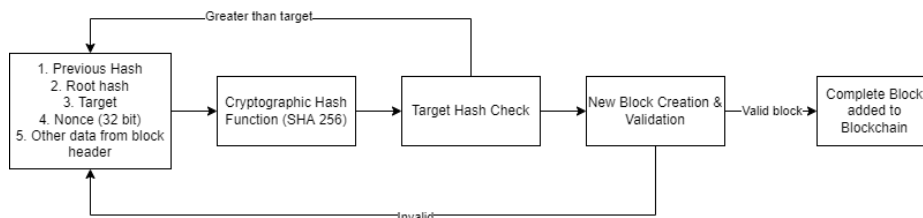
- b. Revise components, testing, or other aspects and repeat steps 2-6 (based on additions/revision)

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

The system to implement this design will consist of three SHA-256 hashing modules and a comparator module.

The hashing algorithm mentioned above should be implemented so that it copies the SHA-256 hash protocol used for bitcoin mining. No matter what size of input is given, the SHA-256 hash should output a 256 bit output in a one way hash.

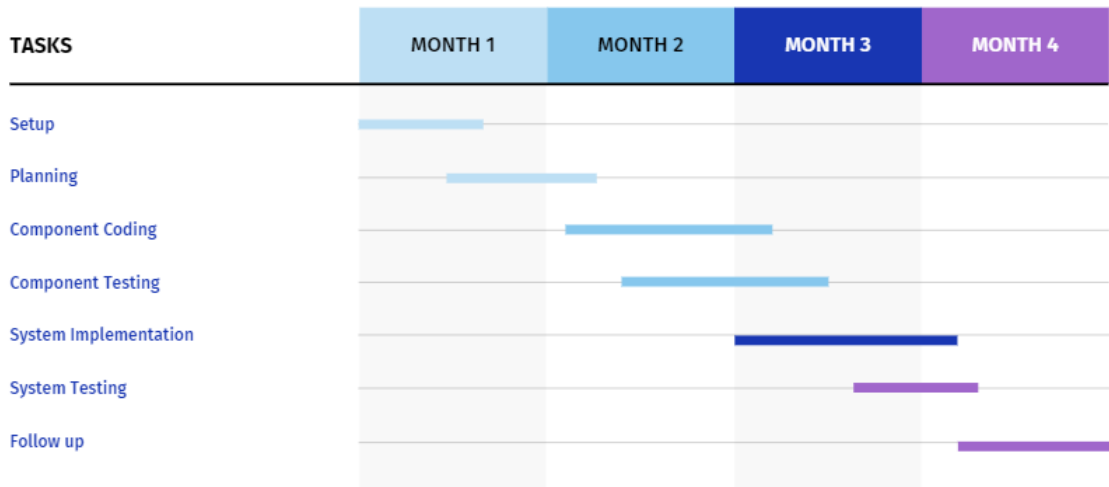
The comparator should take in a 256 bit threshold value and the final hash output and compare to see if the output is within the threshold. The output should be 1 if it is within threshold and 0 if it is not.



3.4 PROJECT TIMELINE/SCHEDULE



GANTT CHART



Week starting from 3/21/22	Task
1	High level design
2	Begin component design
3	Component design
4	Finish component design
5	Begin component coding
6	Component coding
7 (end of 491)	Component coding
8 (beginning of 492)	Component coding
9	Feedback from Dr. Duwe (additional component design)
10	Component coding (if needed) & testing
11	Component coding (if needed) & testing

12	Component coding (if needed) & testing
13	Component testing
14	Begin bringing components together into a system
15	Bringing components together into a system
16	Finish bringing components together into a system
17	Testing system
18	Testing system
19	Feedback from Dr. Duwe (provide any additional testing/implementation)
20	Testing system & additional implementation (if needed)
21	Testing system & additional implementation (if needed)
22	Testing system & additional implementation (if needed)
23	Finish entire system

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

Agile projects can associate risks and risk mitigation with each sprint.

1. Setup of efabless dev environment - difficulty setting up and installing workstations .6

We have already come into considerable roadblocks when setting up our dev environments as the information for setting it up is scattered and inconsistent. There is no way to eliminate this task as it is critical to do anything for our project. Although there are development tools available from efabless these tools aren't great for development. There aren't any other alternatives for this task.

2. Plan what we will design - .3
3. Code components - .75

There is no way to eliminate this as we need to come up with code for the project. We are able to find some off the shelf solutions as there are free IP's available with coded components. There aren't really any other alternatives as all the components need to be coded in verilog.

4. Test components - .5

There is no way to eliminate this task as we need to verify that our components are working correctly. There aren't any ways to get any off the shelf tests as our components will be of our own making. There isn't an alternative solution to this either. This is a critical part of hardware design that must be handled to ensure the fabricated hardware will work as expected since there is no way of modifying the chip afterwards.

5. Bring components together into a system - .4
6. Test the system - .2

3.6 PERSONNEL EFFORT REQUIREMENTS

Task	Total Estimated Person hours
Setup dev environments	10 per person
Plan our overall design and each component	20
Code Components	70
Test Components	70
Bring components together into system	70
Test System	70

Setting up a dev environment includes us familiarizing ourselves with the eFabless ASIC development process. Setting up a Github repository that makes use of an eFabless template project. Setting up a linux environment that has all the tools to run through all of the checks and tests required for the project.

Planning the design will include a high level schematic of what components we need to make. Schematics for how each component will be implemented with simpler logic modules.

Coding the components will entail designing verilog files for every subcomponent.

Testing each of these subcomponents to ensure proper individual functionality is the initial testing components phase.

Bringing the components together into a system entails bringing all subcomponents together to test actual component functionality then hooking all components together to verify system functionality.

Finally testing incremental functionality of the system will ensure that once everything is put together our final product works as intended.

3.7 OTHER RESOURCE REQUIREMENTS

None, as this is a purely digital project. However, proceeding teams will need tools such as wave form analyzers and generators to be able to test the proceeding teams designs.

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

The communities that are being designed for are primarily students that will be using our project as an example, and for our clients/professors that will be seeing the validity of having this as future senior designs. It could also affect the bitcoin mining community as if this design is proven to increase the efficiency of mining. It may address the societal concern of the environment as if it does prove to be a more efficient way of bitcoin mining it will reduce the amount of energy used and reduce the emissions that it contributes. It also will address the safety and welfare of the teams that will be coming after us as we will be making sure that their health and safety are being protected.

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

Area	Description	Examples	Considerations
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	During the bring up process there is risk of injury if one of them components serious malfunctions. Reducing those risks is a priority.
Global, cultural, and social	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.	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	Implementation of the solutions would potentially change the ethics of mining. It will also affect the globally as anything that makes bitcoin mining more efficient makes bitcoin more feasible as a currency. This intern would affect the way that society views currency.
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable	Increasing/decreasing energy usage from nonrenewable sources, increasing/decreasing	Decreasing the energy usage of bitcoin mining will improve the environmental impact that it has.

	practices related to materials manufacture or procurement.	usage/production of non-recyclable materials	
Economic	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 on communities, markets, nations, and other groups.	Product needs to remain affordable for target users, product creates or diminishes opportunities for economic advancement, high development cost creates risk for organization	It may lead to more efficient bitcoin mining creating more money for bitcoin miners. It also could affect the development of ASIC chips as if more universities use this it could lead to great economic development in chip fabrication and development.

4.1.2 User Needs

Students need a way to understand how to create an ASIC and get it fabricated by Efabless, because as it stands now there isn't much cohesive documentation on how to do so.

Prof/client needs a way to demonstrate how to have students fabricate their own ASICs, because they want to understand the feasibility of having a senior design group every year making their own ASIC's.

Bitcoin miners need a way to mine bitcoin faster, because they are doing so to make as much money as possible.

4.1.3 Prior Work/Solutions

There is no bitcoin specific hardware on the market at the moment, but many groups have researched the potential of hardware designed to maximize the efficiency at which bitcoin is mined.

Because most of our work for this project is focused around the complicated nature of fabricating an ASIC we do plan on finding publicly available vhdl IP that we can use in our design. We plan to make custom parts in the design where time allows but most of the uniqueness of our design will come from how the design interacts with the chip harness. Our highest priority is having a functional ASIC design submission that can be selected with as much uniqueness to design as we can fit in our time frame.

Unique components we plan on designing for example include a more efficient tree adder that can run through the sha 256 hash at much faster speeds than the base design.

Advantages

- Much more efficient bitcoin mining than most processing hardware
- Application specific design

Disadvantages

- Has potential for further improvements in hardware design
- Isn't an ASIC so it's very lame

4.1.4 Technical Complexity

The design for a bitcoin mining ASIC requires us to design two major components: a comparator and a sha 256 hash machine. The comparator is a relatively simple design that will check if the hash output is less than our ceiling and output the hash if it is otherwise it will ask for another nonce to perform the next hash.

The 256 hash design is much more complicated and can be broken down into two components: an expander and compressor. The expander will ensure that any input value will be padded to become 256 bits; it will also be running the input target through an algorithm that requires multiple different components to operate on different bit fields simultaneously. The application of the algorithm is explained in the cited research above.

Finally the compressor will be composed of 4 types of computation circuits that are all performing various levels of bit addition in order to complete the sha 256 hash.

The most complicated part of all of this however will be integrating the design within the efabless caravel harness in order to submit a design that can actually be fabricated into an asic. This requires multiple levels of testing and port alignment to make sure that the design is fully functional within the harness.

4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

One key design decision that has been made is to not include a random number generator on the design as we have a limited amount of chip space and would add an unnecessary layer of complexity to our design. Another design decision that will need to be made down the line is the type of adder. We have tentatively decided to go with a tree adder as those are the adders that have been used in the research that we have seen thus far. However, we still need to determine the feasibility and efficiency of using them in our design. Another area of the design that still needs to be determined is the layout of the components on the chip. The specific placement of the components will be handled by Efabless tools, however there are several options when it comes to those tools. We will need to determine the one that fits our design the best once we have implemented the components of the design.

4.2.2 Ideation

For choosing the adder that will be used in our final design there is a large list of potential options such as

- ripple carry adder
- carry skip adder
- carry increment adder
- carry lookahead adder
- carry-save adder

4.2.3 Decision-Making and Trade-Off

Each of these different adders have different pros and cons to them.

In the context of space used (most to least):

1. carry-save adder
2. carry skip adder
3. carry increment adder
4. ripple carry adder
5. carry lookahead adder

In the context of power needed(most to least):

1. carry-save adder
2. carry skip adder
3. carry lookahead adder
4. carry increment adder
5. ripple carry adder

In the context of computational delay (most to least):

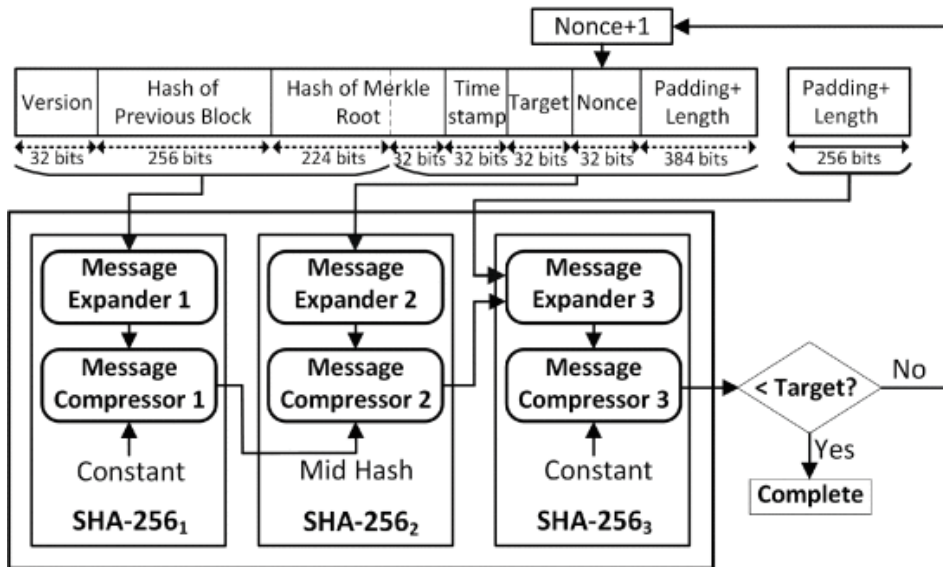
1. ripple carry adder
2. carry skip adder
3. carry lookahead adder
4. carry-save adder
5. carry increment adder

The adder we are going to choose is going to be the carry lookahead adder because it requires the least space, and has only moderate power consumption as well as delay compared to the other adders.

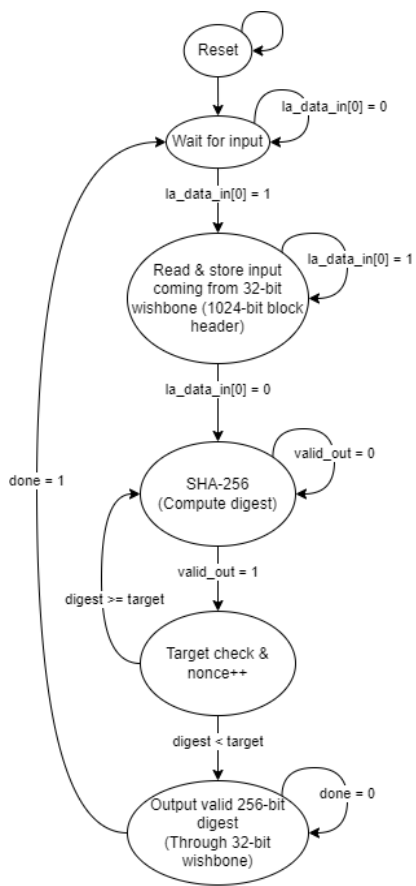
4.3 PROPOSED DESIGN

4.3.1 Design Visual and Description

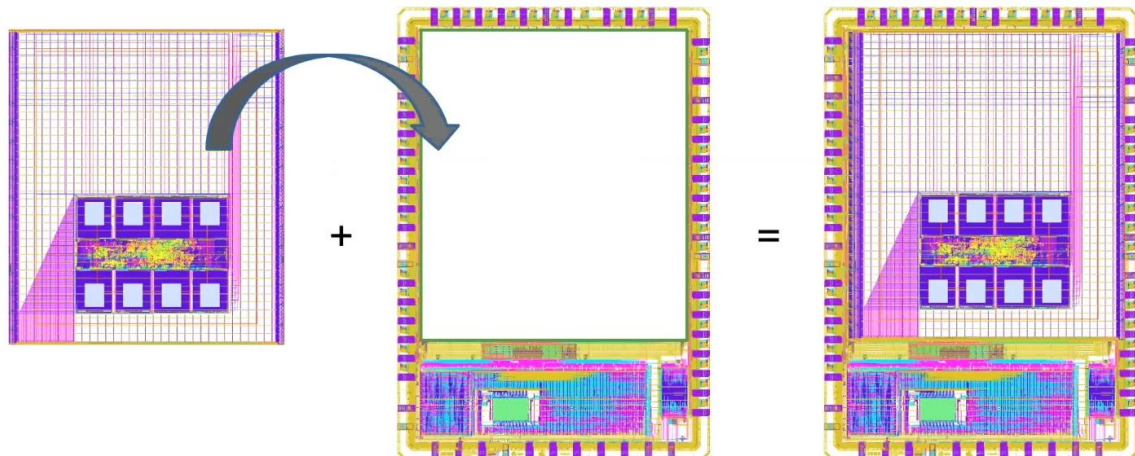
Below is our current high level design of the cryptographic hash function (SHA-256) design from the research paper we are basing the project on. It consists of 3 SHA-256 hash units used to compute the new hash of the block, send it to the comparator, and if the calculated hash is less than the target, the valid 256-bit hash will be outputted (to the management SoC). Otherwise, 1 will be added to the nonce and the hash will be recalculated.



To store the block header (1024-bit input to the 3 SHA-256 modules) we will have 2 types of registers. Ones that are 32-bit wide to store the version, timestamp, target, and nonce along with others that are 256-bit wide to store the hash of the previous block and merkle root. These will be populated with data coming from the management SoC through the wishbone bus. A high level state machine diagram for the mining process is shown below with the SHA-256 computation state and target check being broken down in the diagram above.

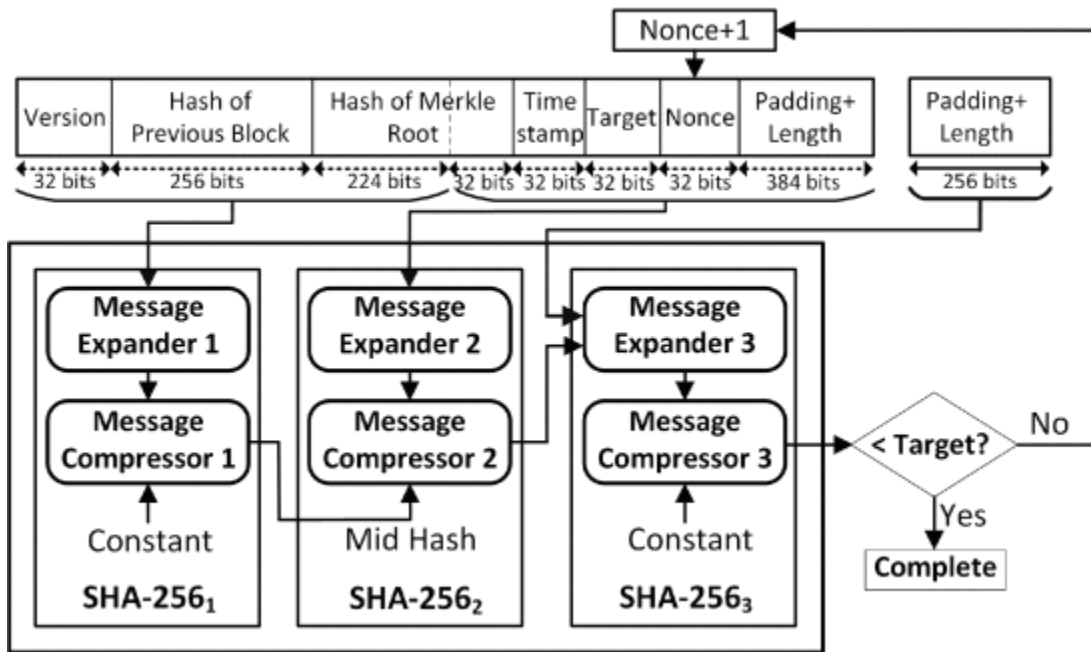


Below is an example of how an arbitrary design (left) would be integrated into the caravel harness (middle), resulting in a complete design that can be fabricated and function as an ASIC (right).



4.3.2 Functionality

The design is intended to take in a previous hash input and calculate the next hash of a blockchain by running through various nonce values. The current design seems to satisfy all the functional requirements of the hashing process as demonstrated in this picture again.



4.3.3 Areas of Concern and Development

Our major area of concern is how to properly integrate this bitcoin mining design into the eFabless caravel as it is a completely new process. We are currently doing a lot of research and experimentation with the caravel harness in our own eFabless work environments and hope to continually make progress on this issue. There are no current questions.

4.4 TECHNOLOGY CONSIDERATIONS

There are a wide variety of hardware design tools available for digital AISC developers such as ModelSim, Vivado, Cadence, and more. These allow hardware designs to be simulated, synthesized, and run through checks such as DRC and LVS. With all this, the logic design can be translated to a physical layout that can be used to fabricate a chip. The Open MPW shuttle projects from eFabless rely on their own tools for submitting a valid chip design; thus, our team was constrained to learning, understanding, and using the provided Caravel User Project as a basis for our design. The logic and layout synthesis tools were OpenLane and the SkyWater 130 Process Design Kit was used for the standard cells in our design.

The largest tradeoff in using these tools was the learning curve and the correct setup of the local environments so everyone on the team could contribute to and test the design. Another challenge

was communicating with the eFabless support teams in case of any errors. On the other hand, once the tools were correctly set up, the design process became much simpler and provided a streamline benchmark that ensured our project was acceptable for submission. Since the choice in hardware design tools were constrained, we did not have the option to consider alternatives and committed to understanding the provided software and hardware.

4.5 DESIGN ANALYSIS

The Bitcoin mining ASIC design is in the testing and simulation stages as it has not yet been manufactured due to the rigorous verification stages it must go through. These include logic and gate level simulations as well as connecting logic analyzer testing points in the design to debug and bring up the chip post fabrication. To ensure a round trip of the entire design process can be completed, we created an adder and accumulator for testing purposes. This confirmed that a custom design can be created and translated to a physical layout within the user project area of the chip. Therefore, we are confident that our final Bitcoin mining ASIC design will be successfully synthesized and laid out in the user project area.

There are many optimizations and design choices we considered while implementing the chip. These include multiple SHA-256 hashing cores capable of simultaneously working together to compute more hashes that may be valid for the specified block. This would speed up the mining process but would increase the verification complexity, add more signals to bring out to the logic analyzer, and may take up too much physical space in the user project area. The goal is to create a simpler version of a single SHA-256 hashing core with lots of debug options to ensure it works correctly. We plan to perform as many pre-silicon tests as possible to ensure the fabricated chip will behave as expected and will have the ability to determine what, if anything, goes wrong. Once this is successful, more complex versions of our design could be created and sent for fabrication if time permits.

4.6 DESIGN PLAN

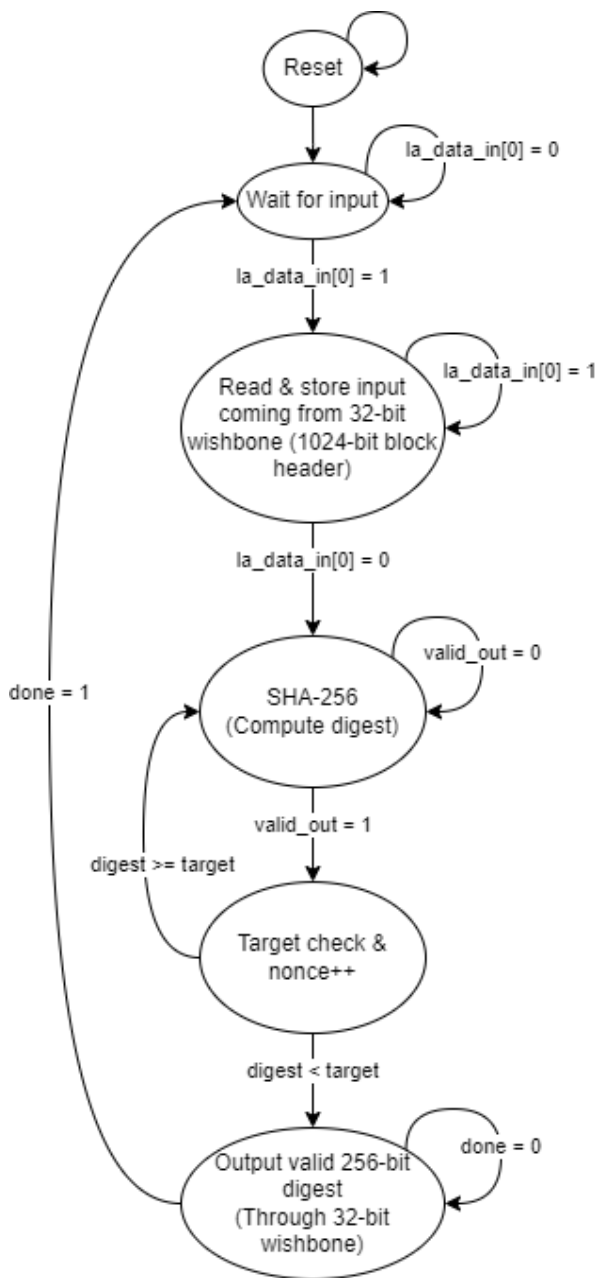
Our design plan begins with finding a suitable IP for the double SHA-256 that makes up the core of our design. Once we have fully tested and verified that the IP functions correctly we will integrate it into the user project area and begin the processes of changing adders within the hasher to be able to demonstrate different adder efficiencies. From there we will begin to build out the other module that is dependent on the hasher functioning correctly, the comparator. Once we have verified that the comparator is functioning correctly we will integrate it with the hasher and test that given a valid nonce to a target the hasher will correctly hash it and the comparator will be able to compare this hashed nonce to the target range correctly. From there we will build out the other components that the hasher is dependent on to work correctly within the user area. These components include the nonce incrementer, which is also dependent on the comparator, the input registers that hold relevant input information, and any control signals needed to correctly propagate information around the design. Each of these components will be tested separately from the overall design to ensure that they are functioning correctly before being integrated into the user area. Once a new component is integrated into the user area the entire system will be tested again to verify that the new component has been integrated correctly. Finally once every component has been tested and integrated into the user area the entire system will be tested to ensure that everything is working correctly within the Caravel Harness.

Once our design for the Bitcoin mining ASIC is integrated into the Caravel Harness, it will be reset and held in a waiting for input stage until a logic analyzer signal is asserted high. This will occur when the firmware sets bit 0 of the logic analyzer to 1 and the reading of the block header will follow. This state requires bit 0 of the logic analyzer to stay high and will read 32-bit data fragments from the wishbone bus, storing the input into the corresponding register for the block header. A handshake will be performed to ensure the data was stored successfully and the next block of data is ready to be stored. Once the entire block header is stored, bit 0 of the logic analyzer must be deasserted and the double SHA-256 IP core will begin computing the output hash or digest. An output valid bit will be set and the digest will be compared with the target. If the digest is less than the target, the valid output hash will be sent over the wishbone bus to the management SoC and stored in a variable. This will be compared to the expected output hash to ensure correctness. While this computation occurs in the user project area, various stages of the computation will be probed using the logic analyzer. These include the nonce to ensure it is incrementing correctly, the results of various adders within the SHA-256 module design, and the digest output. This will ensure our design is functioning correctly and can debug post fabrication.

Module Constraints:

- I. N-bit registers
 - A. Able to read stored values as output
 - B. Able to take as input n bit values and write new values to register to store
 - C. Able to toggle write ability on and off with single bit control
 - D. Able to reset registers to stored values being set to NULL
- II. Nonce Incrementer
 - A. Takes a single bit input each cycle to determine whether or not it will increase the nonce value. Acts similar to a write enable control input.
 - B. Initialized value of 0 and each cycle it receives a 1 input it will increase the stored value by exactly 1.
 - C. Outputs the stored nonce value
 - D. Able to be reset with a reset control
- III. Comparator
 - A. Takes in a 256 bit target value and 256 bit hash value and compares the target to the hash value.
 - B. One output will always be the initial hash value
 - C. The second output is determined by two cases
 1. If target>hash, then output a 1 to signal a valid hash value
 2. Otherwise output a 0 to signal that the nonce value needs to be incremented
- IV. Double SHA-256 IP
 - A. Outputs a 256 bit hash value of the input where it has run through the SHA-256 hash twice. The SHA-256 definition we use is based on the National Institute of Standards and Technology definition and citations here https://csrc.nist.gov/glossary/term/secure_hash_algorithm_256.
 1. Takes a 1024 bit input.

Design plan state machine model:



5 Testing

The pre silicon bring-up of the bitcoin mining ASIC will rely heavily on testing to ensure all components are functionally correct and compatible with one another. This is a portion of the requirements for our project, so we must ensure the test plan is thorough and that the design includes debug ports. Otherwise, the fabricated chip may behave unexpectedly without knowing the root cause. A successful bring-up process will involve lots of testing. Specifically, modularizing each component, performing unit tests focusing on edge-cases and analyzing C test benches and

waveforms at the RTL and gate level to ensure correct functionality. The system as a whole will also need extensive testing to ensure each component functions together as expected. A more detailed analysis of our test plan is discussed through this document.

5.1 UNIT TESTING

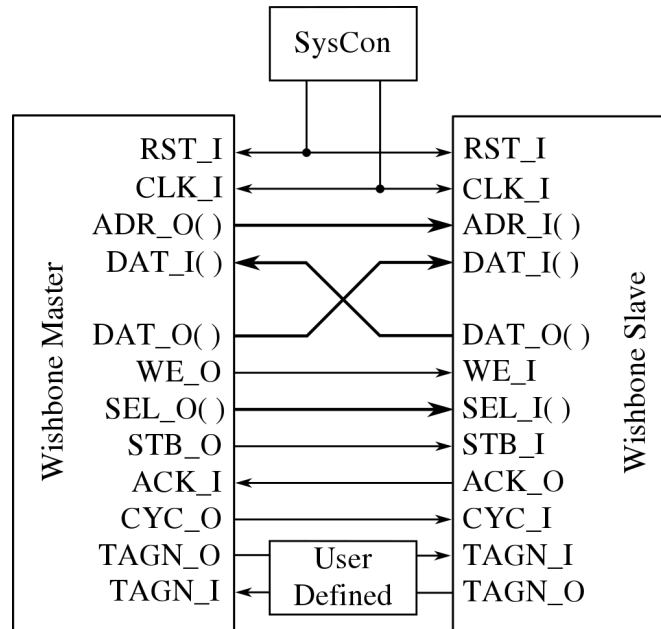
The following units will be tested:

- Comparator
- Double SHA-256 Hashing unit
- Adder
- Registers to store input
- Bitcoin Miner communication with the Caravel Harness
- SRAM or on chip memory (potential module depending on requirements)

Tools used for testing:

- Built in logic analyzer and wishbone communication of the Caravel harness allows us to develop test cases for each unit in C code and testbenches in Verilog. Using this we can attach virtual probes to each unit and run tests on them.
 - Give input and specify expected output
- Waveform simulation output files at the RTL and gate level. These will be used to confirm functionality of each module before integrating into the design.
- OpenLane is software that will be used to test that all components fit within the limited space of the design, check that our resource allocation is within specifications of the harness, as well as show how they expect to be mapped out
- Verilator/ModelSim is also being used to test the IP for a double SHA-256 hasher to determine its correctness and security.

5.2 INTERFACE TESTING



DUT (Designs Under Test) will be specified within the testbenches written in Verilog and C on the “Wishbone Master” side (i.e., the firmware). Along with the wishbone, another interface which assists in the testing process is the Logic Analyzer. This helps to drive inputs and receive outputs from the master and slave respectively. Some important ports to consider above are the CLK_I (the synchronous clock), DAT_I/O (data input/output), and ACK_I/O (acknowledge port). The clock allows the testbench (on the firmware side) to supply our project with a simulated clock, data input/output allows data communication between both the firmware and the user project area, and the acknowledge port serves as a handshake between both sides to ensure proper data communication.

5.3 INTEGRATION TESTING

The most critical integration path will be the Double SHA-256 hashing unit as it is made of many different subcomponents of adders. It is also the main functionality of the design and if it is not working then the design itself is somewhat meaningless. Another critical path will be the comparator as it will be comparing the hashed NONCE to the hashed target value. It is critical as even if we are hashing correctly if the comparator isn't working the design will also fail. These integration pathways will be tested using test cases in c that use the wishbone and logic analyzer on the management SOC of the Caravel harness to determine the values that are being generated by both integration paths. Before they are integrated together, each of the integration paths will be tested on their own to validate their individual functionality. The test benches themselves also generate waveforms that allow us to go through and verify/debug values of different components of our design.

5.4 SYSTEM TESTING

System level testing will run test cases structured to look similar to expected use cases. For example, we will specify a target and a previous hash as inputs to our miner and run them through a separate online hashing tool to get an expected output for our miner. In our test cases we will check the outputs of each cycle of hashing, the result of the comparator, and if the nonce value is increasing. If all of these are functioning as expected for a sufficient number of test cases then we feel that our design will be functional. This is a critical part of our design and ties into the requirements since it revolves around the successful bring-up of an SoC.

Tools:

- Built in logic analyzer and wishbone communication of the Caravel harness allows us to develop test cases for each unit in C code. Using this we can attach virtual probes to each unit and run tests on them.
 - Give input and specify expected output

5.5 REGRESSION TESTING

Modularity is the key design principle we are employing to ensure we do not break old functionality within our designs. In other words, we are aiming to develop new features/additions by creating modules for each new component, utilizing low coupling (not too many connections between components), and high cohesion (a module's responsibilities are focused on a small subset of things – in other words, it is specialized in doing one or a few tasks). One of the critical features we will need to make redundant (so they do not break) is the Wishbone communication that we implement to communicate with the firmware. Since this will be the primary source of communication between the user and the rest of the SoC, it is integral that our implementation is failsafe. The main requirements/tools in order to achieve this redundancy will be case-specific testbenches in verilog and C with an emphasis on edge-case testing.

5.6 ACCEPTANCE TESTING

We will be demonstrating the non-functionality of our design through OpenLane. OpenLane is software used by Efabless to determine if the design is within the specifications of the Caravel harness. This test is also required to submit the design to Efabless. We will be verifying the functionality of our design through test cases and waveforms. The test cases will be in C and be using the wishbone and logic analyzer on the harness to check correct values. We will also be using waveforms that are generated during the test benches to manually verify that the correct values are being achieved. We also will be running through the test benches and waveforms with our advisor/client during the rest of the design process. The waveforms and test benches must also be submitted to Efabless to verify the functionality of our design for them to fabricate it.

5.7 SECURITY TESTING

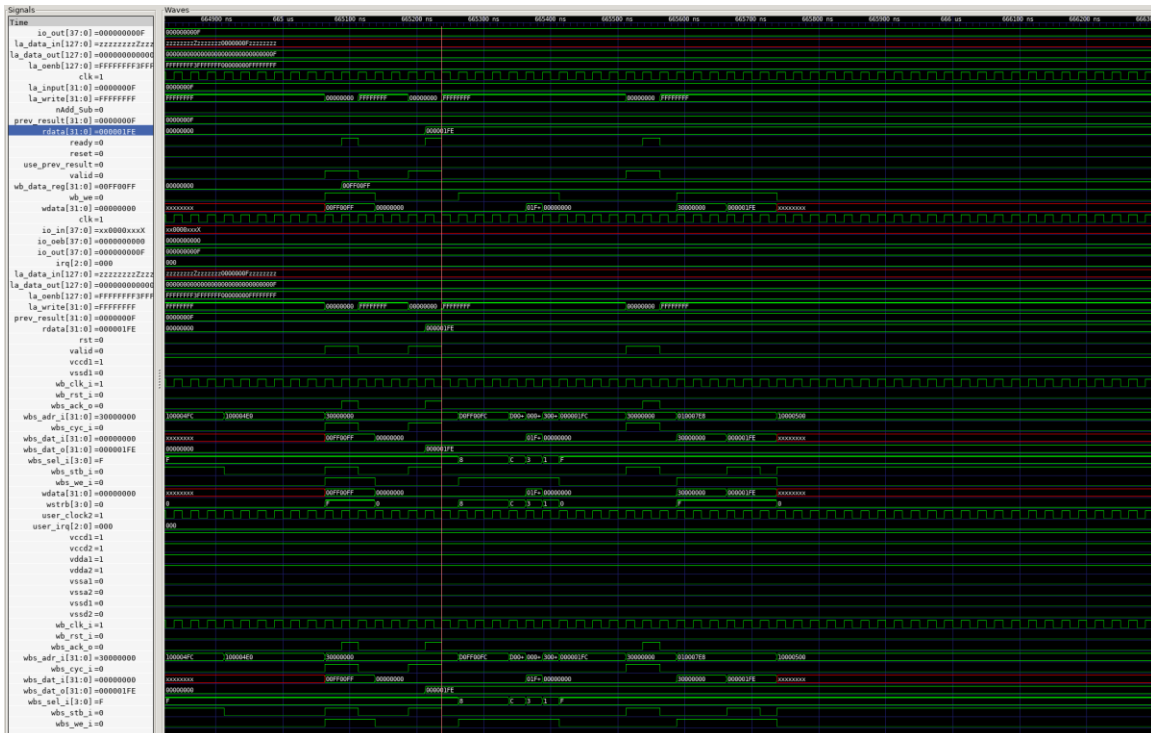
The only security concern that is present in our project is with the imported IP that we are using

for the double SHA-256. The security risk associated with using an imported IP in the design is if there are any additional components of it that would give a third party access to it somehow, or if it would cause damage to the design when powered up. We are addressing this by going through each of the design files to determine if they are strictly necessary to what we will be using it for as well as testing the IP thoroughly before integrating it into our system. The license on the IP allows us to make changes to it so if we do find some unnecessary or malicious addition we are able to remove them. We also will be doing a basic search of the author of the design to determine if they are a trustworthy source.

5.8 RESULTS

The results of our testing will come in two forms: one should be in a text reply that all of our C test cases have passed as expected, and the second will be an output of a waveform file for each component being tested that will show how every variable changes throughout the test cases. We can design our test cases based on the requirements we need to meet so that we can ensure compliance and demonstrate sufficient functionality. For example in an adder we can show test cases doing basic addition and subtraction as well as edge cases such as where we may expect overflow.

Example waveform of testing the custom adder:



6 Implementation

The plan from this point forward will be to test the functionality of this IP to ensure it functions as intended and presents no security issues. Integrate it into our project wrapper and harden it to

check that it will be within the space constraints of the project specifications. We still need to design and test a comparator which will take in a target and hash and output 1 and the hash if target>hash otherwise it will output a 0 and 0. A nonce incrementer which will increment the nonce value by 1 for each cycle it receives a 0 from the comparator. Registers that will store our input and final hash output. Then we will have to combine all of these components in the project wrapper and ensure their functionality as a system with more testing there. Finally we will submit this design to efabless and wait for our ASIC to be manufactured before testing it on an FPGA board.

7 Professionalism

This discussion is with respect to the paper titled “Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment”, *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416-424, 2012

7.1 AREAS OF RESPONSIBILITY

Area of responsibility	Definition	NSPE Canon	IEEE
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	<p>5. to improve the understanding of technology; its appropriate application, and potential consequences;</p> <p>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;</p> <p>7. to seek, accept, and offer honest criticism of technical work, to</p>

			<p>acknowledge and correct errors, and to credit properly the contributions of others;</p> <p>10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.</p>
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	
Communication Honesty	Report work truthfully, without deception, and are understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	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 well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	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;
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	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;
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 so as to enhance the honor, reputation, and usefulness of the profession.	2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist; 4. to reject bribery in all its forms;

Each of the 10 IEEE codes fit under one of the categories of the NSPE version except for the financial responsibility category where IEEE seems to have no code taking this into consideration. The IEEE code of ethics also seems to focus more on ensuring workers uphold high standards of integrity, responsibility, equality, respect, credit association, and work in a safe, non-discriminatory environment.

Each entry is a rule in the IEEE code of ethics placed where they would correspond to their NSPE counterparts.

Work Competence: Focuses on collaborating with others for gaining high quality experience and maintaining as well as continuously improving our technical knowledge. Respectfully reviewing, critiquing, and giving credit for the work of others is also essential for improving abilities as well as maintaining integrity and respect.

Financial Responsibility: None

Communication Honesty: Focuses on being truthful and practical when addressing individuals, team members, or clients. Additionally, when using data, it is important to ensure competent, concise, and effective communication.

Health, Safety, and Wellbeing: Making sure to minimize any potential risks to the end users or stakeholders, whether they be physical, emotional, or their general welfare. These risks can come during the development and later use of our product and we will need to take into consideration resources that they will be able to use to minimize those risks when we are not present for them.

Property Ownership: Treat all people fairly and equally. Avoid injuring others property or reputations.

Sustainability: Accept responsibility for how the product you are designing may affect the environment.

Social Responsibility: Avoid conflicts of interest and reject coercion in all of its forms.

Work Competence: IEEE code has four rules that correspond to work competence in a more focused way on the engineering integrity of products produced by an engineer at work.

Financial Responsibility: The IEEE code of ethics does not have anything that would go along with the NSPE area of Financial Responsibility, it instead focuses more on other areas such as Work Competence.

Communication Honesty: The wording for these two versions is different but the basic message is very much the same.

Health, Safety, and Wellbeing: Instead of focusing on minimizing health risk there is a greater focus on accepting responsibility for potentially harmful actions.

Property Ownership: IEEE has a lot more about respecting the differences amongst people than they do about property.

Sustainability: Sustainability is only seen at the very end of one of the codes of conduct, rather than as its own entire code. This code overall is actually more applicable to Health, Safety, and Wellbeing.

Social Responsibility: IEEE code has two rules that correspond to social responsibility in more focused forms like rejecting bribery and disclosing perceived conflicts of interest.

7.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

Work Competence: This is an extremely relevant part responsibility for our project as we have to ensure that our work is of great quality to pass all the checks eFables has put in place for their OpenMPW shuttle. The checklist given by eFables has parts like the design passing precheck, a concise project repo with a great directory structure, and proper naming conventions. Our team is performing at a level of “High” in this area as we set up our local workflow structure and get started on our application for the ASIC.

Financial Responsibility: This applies to our project professionability context in that we may be given tools that our client has paid for to be better able to complete the project for them. However, we don’t have a financial responsibility in delivering the end product to our client.. Due to this, there isn’t much that applies for financial responsibility to the context of our project. Our team is performing highly for our financial responsibilities as we haven’t incurred any financial costs for our project.

Communication Honesty: This greatly applies to our project because we must all communicate effectively while ensuring there is minimal ambiguity regarding the tasks we have completed or will work on. Our team values honesty and ensuring that everyone is included as well as supported as the tasks are performed. Our team is performing highly in this area because we have frequent meetings, assign tasks evenly, and have proof of the work all members have completed. Furthermore, each member of the team understands the importance and value of this professional responsibility.

Health, Safety, and Wellbeing: This only applies to our project in specific portions. That aspect of the project is the bring up process. During the bring up, there will be a mild risk to the safety and wellbeing of the individuals doing the bring up as they will be a separate group. The team thus far has been performing at a high level for this area as we have been keeping in mind the information that will be required for the bring up. We have also made it a top priority to make the instructions and information about the processes as clear and digestible as possible.

Property Ownership: This area may be somewhat applicable to our team because there has not been any property or proprietary information we had to deal with, but our chip design may include intellectual property (IP) of others which we will have to take into account. Apart from licensing our own work and ensuring the design IP is accounted for, there is no need for handling confidential or sensitive information. Therefore, our team is performing medium because there will not be many applications for this responsibility area and can handle it well if needed.

Sustainability: This is not applicable to our team as there is no core sustainability component that our project is addressing. As eFables is directly handling the fabrication component, and our chip is a very small part of the silicon that will be fabricated, there is a negligible impact on the environment that our project will have. Additionally, as the project is sponsored by Google, the SkyWater fabrication facility is focused heavily on making sure their practices are rooted in sustainable practices. To summarize, our project has no core sustainability component, and the sustainable components related to the fabrication are handled directly by eFables; therefore, this is not applicable (N/A) to our team.

Social Responsibility: This area is somewhat applicable to our project as depending on how well we are able to document the process of building and getting a chip fabricated through eFables it will be a great service to future groups doing this exact thing. In terms of our team's performance in this area we are performing at a high level. We have been able to compile an immense amount of resources for being able to create a user project within the context of the caravel harness and user project space. We have been condensing this information as we have examined it to give later groups an easier and more efficient way of digesting this information. We will be continuing this process throughout the project so as to maintain a high level of performance for this area.

7.3 MOST APPLICABLE PROFESSIONAL RESPONSIBILITY AREA

Communication Honesty is one area of professional responsibility that is extremely important for our project above all other areas. As a team of four, working closely with the numerous interconnected parts of this project, it is important that we communicate with one another about both research done and work implemented as it will likely directly impact another team member's area of specialization within the project. Our team has demonstrated this extensively over the past few weeks by having weekly client meetings where we discuss our direction as well as team meetings where we focus more in depth about our project's components. One of the biggest impacts recently that we found from this responsibility was the interconnection of understanding of the wishbone, caravel harness, workflow, and logic analyzer parts of the project. By continuing our commitment to this responsibility, we will be able to more efficiently and effectively implement each of our parts related to the project and meet the deadline given to us by eFables.

8 Closing Material

8.1 DISCUSSION

The result our team hopes for is to have a submitted ASIC design be selected by eFables for their Open MPW shuttle, manufactured, and then once we receive test its functionality to ensure that it works as intended.

In our specific design case we hope that our Bitcoin mining ASIC is able to be placed onto an FPGA board and have a previous hash and target sent as inputs to the ASIC and then receive as an output a hash that corresponds to a valid blockchain addition. The test cases we will use to test this are actual blockchain submissions which highlights all of these hashes in each case.

8.2 CONCLUSION

The work we have done so far is researching the Open MPW submission process, previous submissions, provided workspace, and constraints. We have selected a design, researched about bitcoin mining, figured out what components would be needed to make the design functional and what their constraints are. We have devised a testing plan for the digital design before submitting the ASIC as well as a plan for once the ASIC is manufactured and sent back to us. We have designed a simple adder in the new workspace to demonstrate that we do have the ability to design and test components successfully within this new workspace. We have selected the IP we will use for our double SHA-256 hashing, and begun integrating it. We have also begun designing the comparator we will be using in our final design.

The rest of our plan from this point forward will be to test the functionality of this IP to ensure it functions as intended and presents no security issues. Integrate it into our project wrapper and harden it to check that it will be within the space constraints of the project specifications. We still need to design and test a comparator which will take in a target and hash and output 1 and the hash if target>hash otherwise it will output a 0 and 0. A nonce incrementer which will increment the nonce value by 1 for each cycle it receives a 0 from the comparator. Registers that will store our input and final hash output. Then we will have to combine all of these components in the project wrapper and ensure their functionality as a system with more testing there. Finally we will submit this design to eFabless and wait for our ASIC to be manufactured before testing it on an FPGA board.

Our biggest constraint is dealing with the unknown eFabless workspace and different systems within it in order to bring this ASIC together and test it using their system to create an official system. This is a very necessary pain and the main reason for the project in the first place as we are venturing into completely uncharted territory with this eFabless ASIC design.

8.3 REFERENCES

H. L. Pham, T. H. Tran, T. D. Phan, V. T. Duong Le, D. K. Lam and Y. Nakashima, "Double SHA-256 Hardware Architecture With Compact Message Expander for Bitcoin Mining," in *IEEE Access*, vol. 8, pp. 139634-139646, 2020, doi: 10.1109/ACCESS.2020.3012581.

[Double SHA-256 Hardware Architecture With Compact Message Expander for Bitcoin Mining | IEEE Journals & Magazine | IEEE Xplore](#)

8.4 APPENDICES

8.4.1 Team Contract

Team Name: Digital ASIC Designers

Team Members:

1) Constantine Mantas

2) Soma Szabo

3) Dawood Ghauri

4) Courtney Violet

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
 - a. Client Meeting: Friday in Durham with Duwe (1 pm; 1 hour).
 - b. Team Meeting: Weekend meeting (variable; 1 hour).
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
 - a. Discord
 - b. Snapchat
 - c. MS Teams
 - d. MS Planner
3. Decision-making policy (e.g., consensus, majority vote):
 - a. Majority vote with professor Duwe making a decision if there is a tie.
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
 - a. Constantine will be the main record keeper with everyone else adding to it as they feel necessary.
 - b. Times will be maintained by each individual in a shared excel spreadsheet which will also detail what each person has accomplished in the week.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
 - a. Face to face meetings:
 - i. Be as punctual as possible, especially when meeting with professors or other people outside of our team.
 - ii. Try to arrive a few minutes early or as discussed.
 - b. Online meetings:
 - i. Similar to face-to-face but could have a few minutes (max 5) of slack.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
 - a. Ensure they are completed or worked on before the deadline.
 - i. If someone is having issues, they should let the team know as soon as possible.
 - b. Preferably complete work to a high standard and follow guidelines, specifically when completing issues on GitHub and merging branches.
3. Expected level of communication with other team members:
 - a. Ensure effective communication about all project work and meetings.
 - b. As stated previously, if any issue arises, let team members know as soon as possible.
4. Expected level of commitment to team decisions and tasks:
 - a. Varies with task urgency. High urgency tasks will be expected to be completed within a week while others may require more time to explore solutions. Communicating progress weekly is essential to demonstrate commitment.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Role	Team Member
Team Organization	Constantine Mantas
Design Workflow	Dawood Ghauri
Individual Component Design	Soma Szabo
Testing	Courtney Violett

2. Strategies for supporting and guiding the work of all team members:
 - a. Make use of human resources including asking team members and professor Duwe for assistance when stuck.
 - b. Communicate roadblocks as quickly as possible so that the team can work together to find a solution.
 - c. Use issues on GitHub and planned tasks so we know what each member is working on and no duplicate work is done.

3. Strategies for recognizing the contributions of all team members:
 - a. Assign weekly tasks that set expectations for each member with the usual expectation being that they be finished unless a roadblock occurs.
 - b. Can verbally praise people for their work.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

Skills/Experience/Perspectives	Team Member
VHDL coding experience, Github workflow, effective communicator, leadership experience in a hardware design environment (TA for 381), Embedded systems industry experience	Constantine Mantas
VHDL, Verilog, UNIX/Windows Server Management, Undergraduate Research Experience, Leadership Skills from TA (381).	Dawood Ghauri
VHDL, Verilog, scripting software, GitHub workflow, effective communicator, storage industry experience, PCB design	Soma Szabo
VHDL, verilog, GitHub workflow, Hardware design in 381, effective communicator. Experience in making WebDev applications, experience working with embedded side projects(Arduino)	Courtney Violett

2. Strategies for encouraging and support contributions and ideas from all team members:
 - a. Inquisition: Ask each other questions about what implementations and ideas they may have so all people get a good understanding of how things work.

- b. Inclusion: Encourage members to speak up and take on tasks or discuss what they think are good action paths.
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. Contact team members through any of the means of communication for most issues if it requires professor Duwe's attention make sure to @ him and team in Microsoft Teams.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
 - a. Design and begin coding of a submittable and likely to be selected Open MPW chip design.
 - b. Learn more about chip design, firmware, and the process of fabricating an integrated circuit.
 - c. Work effectively as a team and resolve conflicts peacefully.
2. Strategies for planning and assigning individual and team work:
 - a. Weekly progress evaluations and task assignments at meetings, and try to assign tasks relevant to people's strengths with potentially multiple people working together on more difficult tasks.
3. Strategies for keeping on task:
 - a. Task board with assignments and deadlines in Microsoft Teams group.
 - b. Remind team members of any tasks if they seem behind schedule or drifting off topic.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
 - a. Reach out through a means of communication to ensure expectations are understood and infractions are minimized in the future.
2. What will your team do if the infractions continue?
 - a. If they do not have a strong reason for the infraction, the team will reach out to professors like Duwe or Shannon to discuss consequences.

- 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) Constantine Mantas | DATE 2/12/2022 |
| 2) Soma Szabo | DATE 2/12/2022 |
| 3) Dawood | DATE 2/12/2022 |
| 4) Courtney Violet | DATE 2/12/2022 |