# Implementing the Atanasoff-Berry Computer With Modern Technology

**Design Document** 

sdmay25-29

Dr. Alexander Stoytchev

Connor Hand / Client Interaction and Team Organization

Zach Scurlock / Testing and Individual Component Design

Noah Butler / Testing and Individual Component Design

Peter Hurd / Testing and Individual Component Design / Budget Handling

William Mayer / Meeting Tracking and Note-Taking

sdmay25-29@iastate.edu

https://sdmay25-29.sd.ece.iastate.edu/

# **Executive Summary**

The Atanasoff-Berry Computer (ABC) was the world's first electronic digital computer, a milestone in computational history. Our project seeks to recreate the ABC using modern technology to preserve its educational and historical significance. The goal is to create a hands-on, interactive tool that aids educators and students in understanding foundational computing principles while honoring the historical importance of the original ABC. By providing a tangible example of early computational logic, this project enhances educational outcomes and inspires a deeper appreciation for the evolution of technology.

This project addresses the need for accessible tools that help bridge the gap between theoretical concepts and practical understanding in computer engineering. Our recreated ABC must accurately replicate the computational logic of the original while remaining user-friendly and interactive. To achieve this, we are using integrated circuits (ICs), and printed circuit boards (PCBs), while ensuring our design stays true to the spirit of the original machine. Input will be managed through an Android-based interface, emulating the ABC's punch-card system while improving accessibility. Core subsystems, such as the add-subtract mechanism, base converter, memory drums, and control system, are designed to function with modern components, yet follow the logic of the original machine.

Significant progress has been made in the project. We have researched the original ABC's design extensively, created a breadboard prototype and PCB for the add-subtract mechanism, and finalized a high-level design strategy. We have also developed a PCB specifically for testing the add-subtract mechanism, we will continue to do this for components that require a great amount of copies.

Our design successfully balances historical accuracy with modern practicality, ensuring it meets the needs of students, educators, and enthusiasts. Our design also utilizes LEDs to display calculations at each step of the computing process, furthering the ease of understanding for students. The next steps include prototyping more components, designing PCBs for those components, developing other non-PCB components, integrating subsystems, and conducting comprehensive system testing. Once completed, the project will be released under an open-source license, providing detailed documentation and plans for global educational and recreational use. This recreation not only revives the ABC's legacy but also contributes to advancing the education of future generations of computer engineers.

# Learning Summary

# Development Standards & Practices Used

- Circuit Practices:
  - Breadboarding for initial prototyping of circuits.
  - Use of ICs for implementing digital logic.
  - PCB design for final module assembly.
  - Adherence to proper grounding and signal integrity techniques during circuit design.
- Hardware Practices:
  - Sourcing reliable and cost-effective components that are compliant with industry standards.
  - Modular design to simplify testing and integration.
  - Compact design to fit within a typical educational workspace.
- Software Practices:
  - Use of version control for code and documentation.
  - Agile and waterfall hybrid project management for iterative development.
- Engineering Standards:
  - **IEEE 1149.1:** Standard for test logic in integrated circuits, ensuring valid interconnections between ICs on PCBs. [2]
  - **IEEE 1481:** Standard for analyzing chip metrics like timing and power consumption during IC design. [3]
  - **IEEE 370:** Standard for high-quality data measurement and signal integrity in circuits. [4]
  - **IEEE 1621:** Standard for user interface elements in power control of electronic devices. [5]
  - **IEEE 1680:** Standard for environmental assessment, focusing on sustainable material use and energy efficiency. [6]

# Summary of Requirements

- Our system must accurately replicate the functionality of the ABC using modern technology.
- Our system must allow users to input data, view each step of the computation, and receive output.
- The system should use modern components.
- The computer must be a physical, tangible object that users can interact with.
- The computer design should fit within a workspace suitable for educational demonstrations.
- The user interface must be simple and intuitive.
- It must look cool.

# Applicable Courses from Iowa State University Curriculum

- CPRE 281: Digital Logic
- CPRE 381: Computer Organization and Assembly Level Programming
- COMS 309: Software Development Practices
- COMS 227: Object-Oriented Programming
- COMS 228: Introduction to Data Structures
- EE 201: Electrical Circuits
- EE 230: Circuits and Systems in Electronics

### New Skills/Knowledge acquired that was not taught in courses

- KiCad Schematic and PCB Design
- Vacuum Tube Logic

# Table of Contents

1. Introduction	7
1.1. Problem Statement	7
1.2. Intended Users	7
2. Requirements, Constraints, And Standards	8
2.1. Requirements & Constraints	8
2.2. Engineering Standards	9
3 Project Plan	10
3.1 Project Management/Tracking Procedures	10
3.2 Task Decomposition	10
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	11
3.4 Project Timeline/Schedule	13
3.5 Risks and Risk Management/Mitigation	14
3.6 Personnel Effort Requirements	15
3.7 Other Resource Requirements	15
4 Design	16
4.1 Design Context	16
4.1.1 Broader Context	16
4.1.2 Prior Work/Solutions	16
4.1.3 Technical Complexity	17
4.2 Design Exploration	17
4.2.1 Design Decisions	17
4.2.2 Ideation	17
4.2.3 Decision-Making and Trade-Off	18
4.3 Proposed Design	18
4.3.1 Overview	18
4.3.2 Detailed Design and Visual(s)	19
4.3.3 Functionality	20
4.3.4 Areas of Concern and Development	20
4.4 Technology Considerations	21
4.5 Design Analysis	22
5 Testing	23
5.1 Unit Testing	23
5.2 Interface Testing	23
5.3 Integration Testing	23
5.4 System Testing	23
5.5 Regression Testing	23
5.6 Acceptance Testing	23
5.7 Security Testing (if applicable)	24
5.8 Results	24
6 Implementation	24

7 Ethics and Professional Responsibility	24
7.1 Areas of Professional Responsibility/Codes of Ethics	24
7.2 Four Principles	25
7.3 Virtues	25
8 Closing Material	25
8.1 Conclusion	25
8.2 References	25
8.3 Appendices	26
9 Team	27
9.1 Team Members	27
9.2 Required Skill Sets for Your Project	27
9.3 Skill Sets Covered by the Team	27
9.4 Project Management Style Adopted by the Team	27
9.5 Initial Project Management Roles	27
9.6 Team Contract	28

# List of Figures

Figure Title	Figure Description
Figure 1	Gantt Chart
Figure 2	Risk Management Chart
Figure 3	Personnel Effort Requirements Chart
Figure 4	Prior Works Compared to Our Project Pros/Cons Table
Figure 5	Weighted Decision Matrix
Figure 6	Block Diagram of the Atanasoff-Berry Computer
Figure 7	Adder-Subtractor Digital Logic Design
Figure 8	Functional Flow of Our ABC Recreation
Figure 9	Image of Completed Adder-Subtractor PCB
Figure 10	4-Equation Elimination Java Switch Case
Figure 11	Areas of Professional Responsibility and Their Relevance to Our Project
Figure 12	Broader Context Principle Pairs
Figure 13	Adder-Subtractor Module KiCad Schematic
Figure 14	Adder-Subtractor Module PCB Implementation Render
Figure 15	KiCad Schematic for the Adder-Subtractor Tester PCB
Figure 16	Adder-Subtractor Tester PCB Implementation Render
Figure 17	Bugged Code - Zero Coefficient Problem

### 1. Introduction

#### 1.1. PROBLEM STATEMENT

Our project is to re-create the ABC computer from scratch using modern technology. Our goal is to use the same logic that John Atanasoff used when developing the original ABC. We are designing our computer purely based on the resources that exist on the ABC. Most resources are written, there is a video from the 90s ABC reconstruction project, and we are also using primary sources from people who worked on the ABC reconstruction project in the 90s. The purpose of our ABC re-creation is educational, but also recreational. Most users will be students and professors in the pursuit of education, but also, computer enthusiasts may enjoy using our computer or learning about it. Our project may benefit our society by creating more capable computer engineers. Our project is open-source, so globally, our project can benefit other schools or enthusiasts by providing detailed documentation on how our ABC re-creation works. Further educating computer engineering students through our project is important because visually seeing a computer's inner workings could make a difference in a student's success or even interest. Our computer is capable of education because it will have physical displays or LEDs that show what our computer is doing at that moment in time. Overall, our project could have a global impact on the education of students in a computer engineering-related field.

#### 1.2. INTENDED USERS

**Computer Engineering Professor** 

- 1. The user is a professor in computer engineering, likely with in-depth expertise in computational systems and logic. They are responsible for teaching complex subjects, such as the architecture and function of computers, to students who may be new to these concepts. They need to convey these topics in a simplified and engaging way to help students understand.
- 2. Computer engineering professors need a way to demonstrate the function of a simple, easy-to-understand computer to their students because it helps them explain core computational concepts effectively.
- 3. The recreated ABC will serve as a tangible teaching tool, enabling professors to visually demonstrate how computers perform calculations and execute operations. This hands-on experience will make abstract concepts more concrete for students, enhancing understanding. The value of this product is in its ability to simplify complex ideas.

#### **Computer Engineering Student**

- 1. These users are students enrolled in computer engineering courses, often motivated to understand how computers work at a foundational level. They may have varying degrees of familiarity with computational logic and systems but are driven by their interest in technology and their desire to succeed academically.
- 2. Computer engineering students need to learn about computational logic to build foundational knowledge in their courses and to succeed in the academic environment.

3. Students will benefit from interacting with the recreated ABC because it provides a real-world application of the theoretical knowledge they are learning in their courses. By witnessing how a computer performs calculations, students will deepen their understanding of computational logic, helping their academic progress.

#### **Computer Enthusiast**

- 1. This group consists of individuals with a personal interest in the history and development of computer technology. They may not be formally involved in academia or the computer industry, but they enjoy learning about how computers evolved and the first in the field.
- 2. Computer enthusiasts need to learn about the ABC because they are interested in the history of computing technology.
- 3. Enthusiasts will find value in the project as it provides them with a tangible piece of computing history that they can explore and understand. By seeing a modern recreation of the first electronic digital computer, enthusiasts can gain a deeper appreciation for the technological advancements that have shaped modern computing. This ties back to the project's mission to educate and inspire through historical recreation.

## 2. Requirements, Constraints, And Standards

#### 2.1. REQUIREMENTS & CONSTRAINTS

#### **Functional Requirements**

- 1. The system must accurately replicate the functionality of the original Atanasoff-Berry Computer (ABC) using modern technology.
- 2. The system should be able to solve different systems of linear equations as the ABC did, using a similar computational logic.
- 3. The recreated computer must allow users to input data, view each step, and receive the correct solution as output.
- 4. The implementation must adhere to the IEEE 370 standard to ensure high-quality data measurement and signal integrity (constraint). [4]

#### **Resource Requirements**

- 1. The system should use modern components such as integrated circuits, PCBs, and semiconductor memory devices to emulate the original ABC.
- 2. Components must be sourced to align with IEEE 1149.1 and IEEE 1481 standards for testing and efficiency (constraint). [2][3]

#### **Physical Requirements**

- 1. The computer must be a physical, tangible object that users can interact with directly.
- 2. The design should fit within a workspace suitable for educational demonstrations, not exceeding typical desktop dimensions.

- 3. The system should visually display each step of the calculation process to enhance user understanding.
- 4. It must look cool.

#### **UI Requirements**

- 1. The user interface must be simple and intuitive, with clear instructions for students and enthusiasts who may not be familiar with this type of background.
- 2. The interface should include visuals on how computations are completed.

#### 2.2. Engineering Standards

Engineering standards are important because they ensure safety, reliability, and quality in everyday products and processes. They provide a common framework for all parts of the manufacturing process and guarantee collaboration across industries. Engineering standards help reduce errors and costs and ultimately enhance consumer trust. It is important to stay in line with engineering standards because they improve efficiency and help engineers maintain consistency in their work.

#### IEEE 754:

The IEEE 754 standard specifies methods for performing binary and floating-point arithmetic. It also specifies exception conditions and how they should be handled by default. This standard is intended to provide a common method for computation with binary and floating-point numbers that will yield the same result. This standard applies to computations being done directly through hardware, or through software. [7]

#### IEEE 1149.1:

The IEEE 1149.1 standard defines test logic for integrated circuits that standardizes approaches to multiple forms of testing. It provides a standard for testing interconnections between integrated circuits (ICs) that have been assembled onto a printed circuit board (PCB). It also provides a standard for testing the IC itself. This standard is intended to standardize the methods of testing ICs on or off of a PCB. [2]

#### **IEEE 1481:**

The IEEE 1481 standard defines methods for IC designers to analyze chip metrics. These metrics include timing and power consumption. Methods for which IC designers can express these metrics are also defined. This standard is intended to allow IC designers to more accurately and completely analyze semiconductor designs while expressing them in a way that is understandable to other IC designers. [3]

We believe IEEE 1149.1 and IEEE 1481 to have relevance to our project, but not IEEE 754. We believe IEEE 1149.1 and 1481 will have relevance to our project due to our use of ICs. We need a way to ensure that interconnections between ICs on our PCBs are valid, and we need to use timing and power information given to us by the IC designers when considering our design. IEEE 754 is not so relevant to our project because it defines the modern way of performing binary arithmetic. Our project intends to use the same method of binary computation that was used on the original ABC, which does not follow IEEE 754. We may use the ideas in IEEE 754 as inspiration when considering how Atanasoff did it, but we will not directly use this in our design.

#### IEEE 1621:

Standard for User Interface Element in Power Control of Electronic Devices. This standard guides the user interface designs' power control. Since we want our modified ABC to turn on/off or manage energy use, following this standard could help usability and consistency with other modern electronic devices. [5]

#### IEEE 1680:

Standard for Environmental Assessment of Electronic Products. This standard is friendly as it focuses on the environmental aspects of electronic products. Like recycling, energy efficiency, and environmentally friendly materials. This standard could help our project be more sustainable.

We have only designed a couple of select parts of our project because that is how our project is structured. We intend to use these standards while working on sections of our project's design, but for now, we do not need to make any changes to the designs we have already made. These standards will influence our design process once we start using PCBs. [6]

# 3 Project Plan

#### 3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

Our project management style that we have adopted is a hybrid of the waterfall and agile approach. Our project benefits from this approach because it allows us to work on different components at the same time (agile), but within those components, the process is fairly linear (waterfall).

Our team will track progress through the use of Discord and Github. Firstly, we will use Discord as our primary method of communication where we will let each other know what we have completed and what needs to be done next. Secondly, Github will be used as our primary repository of files pertaining to the project, such as documentation and KiCad files.

#### 3.2 TASK DECOMPOSITION

- 1. Research Modules
  - 1.1. Add-Subtract Mechanism
    - 1.1.1. Look into modern approaches to replicate the ABC's adder-subtractor design.
  - 1.2. I/O Methods
    - 1.2.1. Study user interaction and techniques that keep the spirit of the ABC.
  - 1.3. Base Converters
    - 1.3.1. Analyze methods for data conversion to align with the ABC's requirements.
  - 1.4. Bit Shifter
    - 1.4.1. Find out how the bit-shifting mechanisms for implementing the arithmetic.
  - 1.5. Control Unit
    - 1.5.1. Design a modern equivalent of the control unit to manage data flow and operational sequences.
- 2. Design Modules
  - 2.1. Add-Subtract Mechanism
    - 2.1.1. Build adder-subtractor circuit in modern terms.
  - 2.2. I/O Methods

- 2.2.1. Create a design that supports input and output for user interactions.
- 2.3. Base Converters
  - 2.3.1. Finalize the base converter design to accurately handle data transformations.
- 2.4. Bit Shifter
  - 2.4.1. Specify the bit-shifting module to integrate with other components.
- 2.5. Control Unit
  - 2.5.1. Define the unit's layout and operational logic to work with all modules.
- 3. Build Prototypes / Test Designs (Breadboarding)
  - 3.1. Add-Subtract Mechanism
    - 3.1.1. Assemble and test the add-subtract functionality on a breadboard.
  - 3.2. I/O Methods
    - 3.2.1. Prototype the I/O system and evaluate user interaction and functionality.
  - 3.3. Base Converters
    - 3.3.1. Test the base converter's accuracy and performance.
  - 3.4. Bit Shifter
    - 3.4.1. Verify the functionality of the bit-shifting mechanism.
  - 3.5. Control Unit
    - 3.5.1. Prototype the control unit to ensure it successfully manages operations and data flow.
- 4. Build PCB
  - 4.1. Add-Subtract Mechanism
  - 4.2. I/O Methods
  - 4.3. Base Converters
  - 4.4. Bit Shifter
  - 4.5. Control Unit
  - 4.6. Top-Level PCB

4.6.1. Integrate all modules, creating a complete system.

- 5. Release Under Open-Source License
  - 5.1. Document Progress
    - 5.1.1. Comprehensive project documentation.
  - 5.2. Release to Public
    - 5.2.1. Allows educational and recreational use.
- 6. Final Deliverables
  - 6.1. ABC Finished
    - 6.1.1. All components are complete and meet functional and quality standards.
  - 6.2. Released to the Public Under an Open-Source License

#### 3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

- 1. Research Modules
  - Milestone: Completion of research for all core modules.
  - Metric: Documentation of research findings for each module.
  - Evaluation: Achieved if the documentation provides clear guidance for all phases.
- 2. Design Modules
  - Milestone 1: Finalize design choices for each module.
    - Metric: Completion of design documents for all modules.
    - Evaluation: Consensus by the team for all design choices.
  - Milestone 2: Review and revise designs based on feedback.
    - Metric: Documented revisions and final approach for each design.
    - Evaluation: Each Consensus by the team for all design choices.
- 3. Build Prototypes / Test Designs

- Milestone 1: Breadboard prototypes assembled for each module.
  - Metric: Prototypes assembled and tested for functionality.
  - Evaluation: Prototypes meet 95% success rate in functional tests with adjustments.
- Milestone 2: Complete prototype testing for all modules.
  - Metric: Pass rate of at least 99% in functionality tests for each module.
  - Evaluation: Prototype achieves expected results.
- 4. Build PCB
  - Milestone 1: Manufacture and assemble PCBs for each module.
    - Metric: Each PCB meets design specifications and functional tests.
    - Evaluation: PCBs pass functionality, compliance, and safety tests, with adjustments documented as necessary.
  - Milestone 2: Integrate modules into the top-level PCB.
    - Metric: Fully integrated PCB meets performance and interconnection standards.
    - Evaluation: Final integration passes compliance tests (IEEE 1149.1 for testing interconnections), meeting all technical and functional requirements. [2]
- 5. Release Under Open-Source License
  - Milestone 1: Complete project documentation for public release.
    - Metric: Documentation is peer-reviewed and approved for clarity and completeness.
    - Evaluation: Documentation meets open-source standards, making the project accessible and understandable for external users.
- 6. Final Deliverables
  - Milestone: Final quality check and project sign-off.
    - Metric: Final system meets all project specifications and passes final user and performance tests.
    - Evaluation: Project receives final approval for completion and is available under open-source, fulfilling educational and functional objectives.

#### 3.4 PROJECT TIMELINE/SCHEDULE

	September	October	November	December	January	February	March	April	May
Research Modules									
Research Add-Subtract Mechanism									
Research Base Converter									
Research I/O Methods									
Research Bit Shifter									
Research Control System / Algorithm									
Design Modules									
Design Add-Subtract Mechanism									
Design Base Converter									
Design I/O Methods									
Design Bit Shifter									
Design Control System									
Build Prototypes / Test Designs									
Breadboard Add-Subtract Mechanism									
Breadboard Base Converter									
Prototype I/O Methods									
Breadboard Bit Shifter									
Breadboard Control System / Top-Level PCB									
Build PCB									
Build PCB For Add-Subtract Mechanism									
Build PCB For Base Converter									
Implement I/O Methods									
Build PCB For Bit Shifter									
Build PCB For Control System / Top-Level PCB									
Release Under Open-Source License									
Document Progress									
Release to Public									
Final Deliverables									
ABC Done									
Released To Public Under Open-Source License									

#### Figure 1: Gantt Chart

Our project's tasks will be performed in the following order: Research Modules, Design Modules, Build and Test Modules, Build PCBs, and Release Under Open-Source License. We will be researching the different modules from September-January. We will be designing the modules one by one after gathering enough information from October-February. We will be building prototypes and testing our designs directly after finishing said designs, this phase spans from October to March. We will build all of our modules and our high-level module on PCBs after testing our designs, this will happen from November to April. We will also be documenting our progress throughout the entire project, and we will be preparing to release it to the public in March-May. The final deliverables of our project are the working ABC reconstruction, and that it is released under an open-source license. The ABC reconstruction will be done in April, and we will release it to the public under an open-source license between April and May.

Task	Risks	Mitigation Plan
Research Modules	There is not enough information for us to 100% faithfully recreate a module. Risk factor: 0.9	If there is a similar component from that time period, we will use it for inspiration and make our own version of the module, otherwise, we will use modern techniques to achieve the same goal as the original.
Design Modules	Making a mistake while designing. Risk factor: 0.95	We have tools that are testing our designs. If we design something incorrectly, we can test it, and then fix our error.
	We run out of time in the school year to design all modules of the ABC. Risk factor: 0.05	
Build Prototypes / Test Designs	Making a mistake while building a prototype. Risk factor: 0.8	We will have debugging outputs in our prototypes when constructing them. If we make a mistake while building a prototype, we will be able to tell what is wrong, and we will fix the error.
	We run out of time in the school year to build prototypes of each module. Risk factor: 0.1	
Build PCB	There is a malfunction in the PCB that was ordered. Risk factor: 0.2	
	We designed the PCB incorrectly and ended up getting something wrong. Risk factor: 0.3	
	We run out of time in the school year to build all PCBs. Risk factor: 0.15	
Release Under Open-Source License	We do not follow the guidelines for releasing our project under an open-source license.	

#### 3.5 RISKS AND RISK MANAGEMENT/MITIGATION

Nisk factor. 0.1
------------------

#### Figure 2: Risk Management Chart

#### 3.6 Personnel Effort Requirements

Task	Details	Estimated Hours
Research Modules	Research the functionality and implementation of the modules	160 Hours
Design Modules	Design modules of the ABC based on our research and understanding of how each module should function.	200 Hours
Build Prototypes / Test Designs	Build prototypes of components on breadboards, and test components individually and all together.	400 Hours
Build PCB	Design schematic, order PCB, and run tests.	150 Hours
Release Under Open-Source License	Document the Process and Release our Designs	50 Hours

#### Figure 3: Personnel Effort Requirements Chart

About 16.6% of our time is spent on researching the modules, 20.8% of our time will be spent on designing the modules, building prototypes and testing the designs will take about 41.6% of our time, 15.6% of our time will be spent on building our PCBs, and about 5.4% of our time will go towards documenting our project and releasing it under an open-source license. These estimates are based on our past experience in completing similar processes.

#### 3.7 OTHER RESOURCE REQUIREMENTS

This project involves a myriad of different parts, materials, and other pieces that will be needed to eventually realize all of the requirements and expectations. Since this project is very heavy on the circuit design component, we will be using and recording an extensive parts library that will include all integrated circuit chips, light-emitting diodes, resistors, capacitors, headers, and connectors we will use throughout this project. Along with all those components, we will also make extensive use of wiring and printed circuit board materials to provide both prototype and finalized versions of our project. Finally, our project contains a significant amount of lab time that will require our attention to manage along with all the other facets to bring about our successful completion of this project and this course.

# 4 Design

#### 4.1 DESIGN CONTEXT

#### 4.1.1 Broader Context

Our design problem exists in the context of education and historical preservation. Implementing the Atanasoff-Berry Computer with modern technology will help connect students with the roots of modern computing by allowing professors to use our design when teaching students about the history of computers and digital logic. It will also benefit those who are simply interested in the history of computing, as our design will be open-source and allow them to take a deep dive into the innermost workings of our design.

Our project will affect the general well-being of students, professors, and those interested in the history of computing by indirectly improving their intellectual welfare by fostering curiosity about technology and engaging them in learning. Professors will be directly affected because they will gain an educational tool. Students will be able to gain knowledge using a physical example designed to help them learn. Those who are interested in the history of computers will also be affected because they will have the ability to learn about the first electronic digital computer like never before.

Our project reflects the values, practices, and aims of the cultural groups it affects positively. It furthers the education abilities of professors and allows students a chance to physically learn about computing. It also enhances the computer enthusiast community's understanding of the first electronic digital computer.

Our ABC implementation project may have a negative impact on the environment. We are unsure of the practices of the manufacturers of our components, and we may be indirectly harming the environment by using their services. Our project will also increase the energy usage from nonrenewable sources because it has to consume power to run.

Our project will not have any funding issues due to the somewhat low cost of components and PCB manufacturing. We plan to use two Android tablets in our design, which will consume the bulk of our budget. However, we will target Android devices that are just powerful enough for our simple apps in order to save on cost for us, and others who would like to build our ABC implementation on their own.

#### 4.1.2 Prior Work/Solutions

There have been two projects in the past here at Iowa State University relating to the ABC. The two projects are the original construction of the ABC by John Atanasoff and Clifford Berry, and the reconstruction performed by a team of engineers led by John Gustafson in the 90s. The goal of the reconstruction project was to create a 100% faithful reconstruction of the original ABC to prove that it was a functional machine. The reconstruction team completed their goal in 1997.

	Our ABC Reconstruction Project	90s Reconstruction Project
Pros:	<ul> <li>Smaller scale</li> <li>We are doing it all electronically</li> <li>Easier to debug with modern tools</li> <li>Lower cost</li> <li>Faster development cycle</li> <li>Ability to use modern tools like computer simulations</li> </ul>	<ul> <li>Closer to the time period of the original ABC construction</li> <li>Captures the spirit of the era in physical form</li> <li>Greater historical accuracy in design and components</li> </ul>
Cons:	<ul> <li>Fewer people</li> <li>Lack of historical authenticity in hardware</li> <li>May not fully represent the constraints of the original project</li> </ul>	<ul> <li>Requires sourcing obsolete components</li> <li>Increased risk of mechanical and electrical failures</li> <li>Longer project timeline due to complexity</li> </ul>

Figure 4: Prior Works Compared to Our Project Pros/Cons Table

The following references outline the previous works that have been completed pertaining to the ABC:

- [1] A. Burks and A. W. Burks, "The First Electronic Computer: The Atanasoff Story." Ann Arbor: University of Michigan Press, 1988.
- [2] J. Gustafson and C. Shorb, "The Atanasoff-Berry Computer In Operation," YouTube, https://www.youtube.com/watch?v=YyxGlbtMS9E&ab\_channel=ComputerHistoryMuseum (accessed Dec. 7, 2024).
- [3] J. Gustafson, Reconstruction of the Atanasoff-Berry Computer, http://www.johngustafson.net/pubs/pub57/ABCPaper.htm (accessed 2024).

#### 4.1.3 Technical Complexity

Our design consists of multiple subsystems, such as the add-subtract mechanism, base converter, memory drums, and our I/O solution. Each subsystem leverages distinct scientific, mathematical, and engineering principles:

- Add-Subtract Mechanism: Implements binary arithmetic through modern logic gates replicating the logic of the original ABC.
- **Base Converter:** Utilizes modern digital logic circuits and ICs to replicate the function of the original ABC's base converter circuit.

- Memory Drums: Simulates temporary data storage using modern memory technologies.
- I/O: Uses Android app interfaces for data input and output.

Our project includes challenging requirements that match industry standards. Our project must replicate the ABC's functionality using modern components while maintaining the spirit of the original design. This requires not only technical ingenuity but also historical research and adaptation. The use of modern ICs and PCBs introduces challenges in achieving functional fidelity while adhering to industry standards. Utilizing the modified Gaussian elimination process that the original ABC used further increases the complexity of math and algorithms used in our project.

#### 4.2 DESIGN EXPLORATION

#### 4.2.1 Design Decisions

- 1. Input
  - a. Input corresponds to how we will receive the equations to compute from the user (the original used physical punch cards).
- 2. User Feedback / Output
  - a. User Feedback / Output corresponds to how we will show the computer's calculations to the user and how we will give the final answer to the equations to the user.
- 3. Physical Layout
  - a. The physical layout corresponds to how we will be assembling our final design (most likely using PCBs).

#### 4.2.2 Ideation

Our project allows for open-ended conversations about design choices because we are implementing modern technology with the Atanasoff-Berry Computer. Our most talked about component in the project is how we will handle input and keep the user involved. The original input for the computer was punch card-based, which was the first option we discussed. We want to stay loyal to the original machine, so we sought out IBM punch card machines but struggled to find any. Our second consideration was printing dots on a piece of a standard sheet of paper and scanning the paper as the input. Relating to the previous consideration, our third consideration, and a little bit far-fetched, was 3d printing a punch card based on the user's input. Our fourth idea for the component was to generate a QR code or a PDF that our machine could read. Our fifth and final option is using tablets as input. We arrived at this idea because our team could program a punch card generator based on the original machine. That way, we could also stick with our main idea of visualizing the machine's operation.

#### 4.2.3 Decision-Making and Trade-Off

	Compentency	Cost	Viability	Desirability	Alignment	Total
Criteria Rating	3	4	5	4	2	
Punch-Card	5	5	2	1	3	
Weighted Rating	15	20	10	4	6	55
Printed Dots	3	2	2	3	3	
Weighted Rating	9	8	10	12	6	45
3D Printed	3	5	5	5	5	
Weighted Rating	9	20	25	20	10	84
QR/PDF Generate	3	2	2	4	3	
Weighted Rating	9	8	10	16	6	49
Tablet	2	3	1	2	2	
Weighted Rating	6	12	5	8	4	35

#### Figure 5: Weighted Decision Matrix

Competency refers to how it relates to our team's skill set. Cost refers to the cost required to adopt the new idea. Viability determines if the idea is applicable in real life. Desirability refers to how the user accepts and interacts with the new idea. Alignment refers to how the idea aligns with our project strategy. A lower total score is in our better interest and tablets align with that score.

#### 4.3 PROPOSED DESIGN

#### 4.3.1 Overview



Figure 6: Block Diagram of the Atanasoff-Berry Computer

Our current high-level design is based on the high-level design of the original ABC (see Figure 1). The modules that we need to recreate include the add-subtract mechanism, base-converter, input/output, memory drums, and the control unit. The add-subtract mechanism is responsible for adding base-2 numbers used in the Gaussian elimination algorithm. The base-converter is used for taking the base-10 input, and converting it into the format our computer will use, which is binary. The input/output modules consist of our input/output methods. We have not fully designed these methods yet, but we are leaning towards using an app to give the machine input and having a display that shows the output when the computer is finished. The memory drums are used to store numbers after a computation or step in the elimination algorithm. And finally, the control unit will be used to control the flow of the algorithm.

#### 4.3.2 Detailed Design and Visual



Figure 7: Adder-Subtractor Digital Logic Design

Sub-Systems Overview

- 1. Add-Subtract Mechanism
  - Function: The component performs binary addition and subtraction as part of the Gaussian elimination algorithm. Helps in computing the results of systems of linear equations.
  - Design Details: Modeled as a full adder, which relies on a series of logic gates to process inputs and generate sum and carry outputs. The image above is the diagram of the full adder-subtractor logic of the ABC. Each gate in the circuit plays a role in calculating the intermediate and final results, providing the correct binary output.
- 2. Base Converter
  - Function: Converts the input from decimal (base-10) to binary (base-2), aligning with the original ABC's computational format.
  - Design Details: This sub-system takes decimal inputs from the user and uses software logic in our program to transform these inputs into binary. The binary data is then routed to the add-subtract mechanism for further processing.
- 3. Input/Output Module:
  - Function: Allows users to input data and displays the results after calculations are complete.

- Design Details: Utilizing Android tablets for user input and output, this module emulates the punch card system of the original ABC, showing users how their inputs are converted into binary and processed through the various subsystems.
- 4. Memory Drums:
  - Function: Emulates the ABC's original memory function by storing intermediary data during calculations.
  - Design Details: Implemented using IC chips and modern memory storage solutions, this module temporarily holds calculation results and binary inputs, which are then fed into the add-subtract mechanism for successive operations.
- 5. Control Unit:
  - Function: Manages the data flow and operation sequences within the system.
  - Design Details: The control unit coordinates the various subsystems, ensuring data moves between components in the correct order. The control logic is implemented through a combination of IC-based circuits and programmed sequences to handle input processing, base conversion, and final output.

The control unit oversees the integration of all modules. Data from the user (input) flows into the base converter, where it is transformed into binary. This binary data is stored in the memory drums temporarily and then processed by the add-subtract mechanism. Once calculations are completed, results are displayed to the user through the output module.

#### 4.3.3 Functionality

Our design is intended to solve systems of linear algebra equations that are taken from the user's input. The equations that the user inputs will translated onto a virtual punch card, and then be converted to binary. Once the equation is in its binary form, the ABC will solve for the unknown variables present in the equation by utilizing adders and memory drums to perform the Gaussian elimination algorithm. Once the equation has been solved, the results will outputted onto a screen.



Figure 8: Functional Flow of Our ABC Recreation

#### 4.3.4 Areas of Concern and Development

Looking at the requirements for our project, we are very confident that our current design will meet or exceed them. Specifically, we are required to produce a faithful recreation of the ABC, and our design strategy is constantly balancing that requirement with the technology we have available to us. Additionally, our design for this project, which will be used as an educational aid for our fellow students, will sufficiently meet their needs with a focus on interactivity and continued documentation.

Currently, our biggest concern for this project stems from the inherent time constraint of the entire Senior Design Project cycle. Our project, while seemingly narrow in scope, contains a myriad of technical details that force us to continue our research far beyond what we initially envisioned. While it would be fairly simple had we been given a complete understanding of how the original ABC worked, we have had to find out that information ourselves, along with the complete hardware and software design of our version of the machine. To address this issue, we have adopted a rigid design strategy with some hard-set deadlines so that we can work as efficiently as possible. While we hope that this will be enough to guarantee our success, we are slowly coming to terms with the fact that a full-scale re-envisioning of the ABC may simply not be possible given not only the time constraint but also the monetary constraint. Thus, we have made the decision that, rather than make a subpar full-scale version, we would instead opt for a faithful, detail-rich, scaled-down version of the ABC to prove that it worked and help educate future students on digital logic design.

#### 4.4 TECHNOLOGY CONSIDERATIONS

#### 1. Integrated Circuits (ICs)

Strengths: Integrated circuits offer a compact, reliable, and energy-efficient solution compared to the vacuum tubes used in the original ABC. ICs allow us to implement complex logic functions within a smaller physical footprint, making the design more feasible and scalable for modern educational use.

Weaknesses: Using ICs, while efficient, moves away from the original design's reliance on discrete vacuum tube circuits. This departure sacrifices some historical accuracy in exchange for practicality, as vacuum tubes are difficult to source and maintain.

Trade-Offs: We opted for ICs due to the reduced power requirements and simplified design process. Additionally, ICs enable us to achieve the desired computational logic without the extensive space and cooling needs of vacuum tubes, making the ABC replication more manageable and user-friendly.

#### 2. Android Tablets

Strengths: Android tablets provide a versatile, interactive interface for both input and output, which simplifies user engagement and feedback. The use of tablets allows us to simulate the punch card system for input and visualize binary data transformations, which is crucial for educational clarity.

Weaknesses: While convenient, tablets are not a true reflection of the original punch card-based input system, which may detract from the historical authenticity of the replication. Furthermore, tablets require software development to emulate punch card functionality, adding complexity to the project.

Trade-Offs: Tablets were selected due to their accessibility, ease of programming, and ability to simulate various forms of input and output. This approach balances educational value with practical usability, allowing users to interact with the system while maintaining an approximation of the original input/output experience.

#### 3. Breadboard and PCB Design

Strengths: Breadboards allow for easy prototyping and testing, enabling us to validate circuit designs before creating the final PCB. PCBs provide a permanent, stable design for long-term use, which is essential for a reliable educational tool.

Weaknesses: Breadboards are limited in terms of space and connection stability, which can complicate complex circuits. PCB production, while necessary for durability, incurs additional costs and requires careful design to avoid errors.

Trade-offs: We are using breadboards for initial prototyping, allowing us to refine the circuit design without the permanence of a PCB. Once designs are finalized, we transition to PCBs for the

completed modules, balancing flexibility in development with the durability required for educational use.

#### 4.5 DESIGN ANALYSIS

Our team was given a task with a lot of unknowns. We are still finding new information, discussing new ideas, and constantly changing every day. We've proved that an old vacuum tube logic was a full adder-subtractor, which was a big step for our team. Since then, while researching, we breadboarded the ABC's adder-subtractor module. We still have many questions about digging up sources, uncovering the machine, watching videos, and even finding people still alive who were a part of building the original poorly documented machine. Our future plans include building more breadboards for parts we are fully discovering, decrypting the original input to the ABC, and building physical components to show at the end of the semester. There might be a little struggle with feasibility, but it lies in how fast we can digest this project.

# 5 Testing

#### 5.1 UNIT TESTING

The individual components of our design are each being tested separately. The design of our components are tested early in the design stage using programs to simulate digital logic. After finishing the design, we build the components on a breadboard and test the design physically by using LEDs and a multimeter to measure important voltages not represented by LEDs. After ensuring our design works, we get it printed on a PCB and test the PCB design in the same way as the breadboard. Our overall design requires that we use multiple of the same PCBs for some components, such as the add-subtract module, so for these, we will develop another PCB specifically for testing the functional PCB. The test PCBs will be incredibly simple and only read important voltages to be measured from the functional PCBs, and the test PCBs will represent those voltages as LEDs.

#### **5.2** INTERFACE TESTING

Our design features many different interfaces that include many of the individual modules that make up the machine. While our unit testing will ensure that each module works by itself, we will combine certain modules together to cover key interfaces like machine input/output, data conversion and storage, and finally, computation. Once we pass our unit testing, we will then move on to implementing each of these key interfaces by stringing together our modules and ensuring they function as expected. Unlike the unit testing, this interface testing will occur mostly on breadboards as it is not completely economical to build PCBs for this testing.

#### 5.3 INTEGRATION TESTING

From our requirements, we need to ensure that the machine can perform all the same functions as the original ABC while taking input and output in an analogous manner. Thus, the primary integration path for us will be to take in base ten input, convert it into base two, store it in the machine's memory, perform one computational step, and then output the base two data for later use. The secondary integration path consists of many of the same steps, however, the input needs to come in base two, so no base conversion will be necessary. This testing will, unfortunately have to wait until all of our modules have been implemented on PCBs to conserve our team's breadboarding resources.

#### 5.4 SYSTEM TESTING

All of our different stages of testing will be necessary to verify the overall system performance and ensure that we meet all of our requirements. Specifically, we will need to run unit tests for every module we create, interface tests on the three different key operations for our machine, and finally, some higher-level integration tests that make sure the machine can fully complete a computation step, as then we can chain those steps to fully solve a system of linear equations.

#### 5.5 REGRESSION TESTING

We can implement some regression testing by making sure that any new hardware design works properly according to our unit tests on breadboards first. Then, by making sure all designs are breadboarded first, we can ensure that they will not break any old functionality. For any software we might write, we will need to implement some common functional tests to be sure that the overall functionality of the new software only improves previous functions, not hinder them.

#### 5.6 ACCEPTANCE TESTING

We will demonstrate our meeting of design requirements by making sure we are testing as often and as thoroughly as we can, and ideally, within the presence of our advisor so that he too can verify our findings. By actively involving our advisor/client in the design and verification process, we hope to further deliver a useful educational aid to him and future students.

#### 5.7 SECURITY TESTING

There are virtually no security concerns with this project. As a primarily hardware-based, educational tool with a very specific use, this device is not at risk for any malicious activity.

#### 5.8 RESULTS

So far, despite the limited scope of our testing, it has been very successful. With our adder-subtractor module, we were able to run our unit testing on the component both on breadboards and then on the PCB version with the help of a specially designed PCB. Through testing, we were actually able to find numerous inefficiencies and flaws just in this one component itself, which gives us confidence that our testing scheme is thorough enough to continue using. Additionally, by actively including our advisor/client in our testing process, we have ensured that he is satisfied with the design even before we sent it to be fabricated on PCB, which helped save us some costs.

With the relative success we have seen in designing this one component, we now have the confidence to begin branching out separately to fully tackle the design of the rest of the necessary components we need to implement to achieve success in this project.

# 6 Implementation

Our team has developed our first PCB prototype of the adder-subtractor circuit. This prototype plays a huge role in our project because it is a core feature of the Atanasoff-Berry Computer. The ABC originally had 30 adder-subtractor circuits using vacuum tubes, we deciphered the original vacuum tube logic and translated it into modern circuit logic. Our modern implementation of the adder-subtractor circuit outputs correct values. From there, we used it to create a KiCAD file of the PCB that we ordered.



Figure 9: Image of Completed Adder-Subtractor PCB

The ABC solved linear systems of equations using a modified Gaussian elimination technique. Alice and Arthur Burks [1] described Atanasoff's elimination algorithm on pages 11 and 12 in their book. The linear system example provided in the book described how Atanasoff used forward and backward parts to complete the elimination process. However, the linear system was a set of 3 equations and worked too perfectly. Our team used the example in the book to infer how a 4 equation system would be solved using the same techniques in the book. We proved that a 4-equation system can be solved by implementing Java code that uses case-specific operations based on the number of equations. We are still working on proving if it works for 5 or more variable systems.

<pre>//For a four equation linear system</pre>				
for ( <u>i</u> = 1; <u>i</u> < 13; ++ <u>i</u> ) {				
switch ( <u>i</u> ) {				
<pre>case 1 -&gt; rowreduction(matrix,</pre>		row2: 1,		col: 0);
<pre>case 2 -&gt; rowreduction(matrix,</pre>	row1: 1,			col: 0);
<pre>case 3 -&gt; rowreduction(matrix,</pre>	row1: 2,			col: 0);
<pre>case 4 -&gt; rowreduction(matrix,</pre>				col: 1);
<pre>case 5 -&gt; rowreduction(matrix,</pre>			newrow: 8,	col: 1);
<pre>case 6 -&gt; rowreduction(matrix,</pre>		row2: 8,		col: 2);
<pre>case 7 -&gt; rowreduction(matrix,</pre>			newrow: 10,	col: 3);
<pre>case 8 -&gt; rowreduction(matrix,</pre>			newrow: 11,	col: 3);
<pre>case 9 -&gt; rowreduction(matrix,</pre>	row1: 10,	row2: 11,	newrow: 12,	col: 2);
<pre>case 10 -&gt; rowreduction(matrix,</pre>	row1: 12,		newrow: 13,	col: 1);
<pre>case 11 -&gt; rowreduction(matrix,</pre>	row1: 10,	row2: 13,	newrow: 14,	col: 2);
<pre>case 12 -&gt; rowreduction(matrix,</pre>		row2: 14,	newrow: 15,	col: 3);
}				
}				

Figure 10: 4-Equation Elimination Java Switch Case

# 7 Ethics and Professional Responsibility

Area	Relevance to Project
Work Competence	Our project is heavily Electrical, Computer, and Software Engineering. By having our team's skill set and knowledge, we ensure a high-quality outcome.
Financial Responsibility	The modern implementation of the ABC will require PCB orders from JLCPCB. It will require multiple revisions to ensure our project is complete. Before we place orders, we have our team separate into groups to revise KiCAD designs. We also send our designs to ETG and revise them with our client. This makes our spending smarter and more faithful.
Communication Honesty	We meet twice a week where we update our weekly report, fill out design documents, and take meeting notes. All of our updates are truthful in what we have done each week and the progress we've made. We communicate fully with our advisor and constantly update each other.
Health, Safety, and Well-being	By adhering to this area, we build confidence in our project. We reduce the risk of legal or ethical repercussions. We also reflect quality and user needs.
Property Ownership	Our final product will be released under a permissive open-source license. We acknowledge our own team's contributions by printing the name of the student(s) on each PCB that they worked on. We also have a team logo that will be printed on each PCB.
Sustainability	Our team orders parts from China. That way, we stay under budget and get fast, efficient PCBs.
Social Responsibility	To us, this means serving a societal purpose that can inspire creativity in people interested in computers.

#### 7.1 Areas of Professional Responsibility/Codes of Ethics

Figure 11: Areas of Professional Responsibility and Their Relevance to Our Project

The area where our team is performing well is Social Responsibility. One of our team's virtues is staying faithful to the ABC's original logic, just interpreted in a modern form. By staying true to this, we create a meaningful product. We want people interested in computers and students in computer-related degree programs to be able to easily understand how the first electronic computer operated.

One area our team needs to improve on is Sustainability. We don't think heavily on optimization of an algorithm because we're using an already existing one. Also, we order parts from China because it's fast and cheap. This is done because we're under a time constraint and do not have the luxury of spending a large amount of money.

#### 7.2 FOUR PRINCIPLES

Area	Description
Beneficence	Our project respects the values of people in computer related fields and people interested in computers. We reflect their values by attempting to stay as faithful as we can to the original design.
Nonmaleficence	Relating this area to the environment, we find some problems. We order parts from China because they are cheap, but we're not sure how friendly their practices are to the environment. We don't think twice about being supplied PCBs from China.
Respect for Autonomy	We are releasing our project under a permissive open-source license. This allows people invested in the project to modify our project themselves on their own account. This promotes creativity and allows for future teams to learn and base on our project.
Justice	With an Economic relation, we promote fairness by standardizing the parts we order. We conversed with the other team that Stoytchev, our advisor/client, is also supervising, and we standardized LED's and other wiring's for our prototypes.

Figure 12: Broader Context Principle Pairs

An important pair is Beneficence-Global, Cultural, and Social. This is important because our team wants to deliver a product that can be benefitted by students and anyone interested in computers. We will ensure it by considering multiple perspectives by breaking down the basic principles of the original ABC. A pair that is lacking is Nonmaleficence-Environmental. The main negative of this pair is that we have little to zero control over this. We work under Iowa State and use the basics. This is overcome in other areas because it's so insignificant to our entire project.

#### 7.3 VIRTUES

Team Virtues

- Collaboration
  - Ensures that team members work together, this helps each other's strengths to achieve a common goal. This promotes creativity by combining different perspectives.
- Integrity
  - Integrity builds trust in our team and with our client. It ensures that decisions are made ethically and that work is performed honestly and transparently.
- Adaptability
  - Adaptability allows our team to respond effectively to unexpected challenges, such as changing project requirements, technical issues, or new insights gained during prototyping.

# 8 Closing Material

#### 8.1 CONCLUSION

Our team has made significant progress in recreating the Atanasoff-Berry Computer (ABC) using modern technology. We divided into research on the original ABC's design and computational logic, built a breadboard prototype for the adder-subtractor circuit mechanism, and translated vacuum tube logic into modern circuit designs. Additionally, we've developed and tested a PCB for the adder-subtractor circuit and implemented Java code to simulate the Gaussian elimination algorithm for 4-variable systems. To achieve our goal, the best plan of action involves completing prototyping and integrating subsystems to create a cohesive design. Our current constraints are time frame and limited access to original documentation. In future iterations, improved time management will allow us to properly complete our project. With all these improvements, our team will deliver a faithful and impactful recreation of the ABC.

#### 8.2 References

- [1] A. Burks and A. W. Burks, "The First Electronic Computer: The Atanasoff Story." Ann Arbor: University of Michigan Press, 1988.
- [2] IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2001.
- [3] IEEE Standard for Integrated Circuit (IC) Designer Metrics, IEEE Std 1481-2009.
- [4] IEEE Standard for Electrical Characterization of Printed Circuit Board and Related Interconnects at Frequencies up to 50 GHz, IEEE Std 370-2020.
- [5] IEEE Standard for User Interface Elements in Power Control of Electronic Devices, IEEE Std 1621-2004.
- [6] IEEE Standard for Environmental Assessment of Personal Computer Products, Including Laptop Personal Computers, Desktop Personal Computers, and Monitors, IEEE Std 1680-2006.
- [7] IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2019.
- [8] J. Gustafson and C. Shorb, "The Atanasoff-Berry Computer In Operation," YouTube, https://www.youtube.com/watch?v=YyxGlbtMS9E&ab\_channel=ComputerHistoryMuseum (accessed 2024).
- [9] J. Gustafson, Reconstruction of the Atanasoff-Berry Computer, http://www.johngustafson.net/pubs/pub57/ABCPaper.htm (accessed 2024).

#### 8.3 APPENDICES



Figure 13: Adder-Subtractor Module KiCad schematic



Figure 14: Adder Subtractor Module PCB Implementation Render



Figure 15: KiCad schematic for the Adder-Subtractor Tester PCB



Figure 16: Adder Subtractor Tester PCB Implementation Render



Figure 17: Bugged Code - Zero Coefficient Problem

### 9 Team

#### 9.1 TEAM MEMBERS

- Peter Hurd
- Connor Hand
- Zach Scurlock
- Noah Butler
- William Mayer

#### 9.2 REQUIRED SKILL SETS FOR YOUR PROJECT

The required engineering students for the implementation of ABC with modern technology are: Electrical, Computer, and Software. Some other special skills needed include experience with breadboards, some PCB design experience, familiarity with computer design principles, and assembly programming skills.

#### 9.3 Skill Sets Covered by the Team

- Electrical Engineering
  - Peter Hurd
- Computer Engineering
  - Connor Hand
  - Zach Scurlock
  - Noah Butler
  - Software Engineering
    - William Mayer

Peter Hurd, Connor Hand, Zach Scurlock, and Noah Butler all have prior experience with breadboards. Peter Hurd has PCB design experience. William Mayer has Assembly programming skills. All team members are familiar with computer design principles.

#### 9.4 Project Management Style Adopted by the Team

Our team uses a mix of both Waterfall and Agile styles for our project management. As a team, we defined our project's scope, objectives, and requirements. We set milestones with a project timeline. From there, we broke down our work into short sprints, focusing on specific tasks. We often meet and gather feedback from our advisor to make flexible adjustments.

#### 9.5 INITIAL PROJECT MANAGEMENT ROLES

1. Project Sponsor

3.

- a. Iowa State University
- 2. Project Manager
  - a. Alexander Stoytchev
  - **Project Team Members** 
    - a. Connor Hand
    - b. Zach Scurlock
    - c. Peter Hurd
    - d. William Mayer
    - e. Noah Butler

#### 9.6 TEAM CONTRACT

#### **Team Members:**

- 1) Connor Hand 2) Zach Scurlock
- 3) Peter Hurd 4) Noah Butler
- 5) William Mayer

#### **Team Procedures**

#### 1. Day, time, and location for regular team meetings:

With advisor - Thursday at 1:30 PM D Senior Design Rm in Coover

Without advisor - Wednesday at 3:30 PM Senior Design Rm in Coover

#### 2. Preferred method of communication updates, reminders, issues, and scheduling:

Discord - Carrier Pigeons - Outlook

#### 3. Decision-making policy:

Majority Vote

#### 4. Procedures for record keeping:

William records notes - Shared in Google Drive and Notes channel

#### **Participation Expectations**

#### 1. Expected individual attendance, punctuality, and participation at all team meetings:

All individuals should be attending at least 90% of meetings. We understand some situations are unavoidable. Team members should participate in meetings by sharing what they have done since the last meeting and by helping with the project.

#### 2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

Assignments: Get done before the due date

Timelines: Try your best for timelines

Deadlines: Finish up before deadlines

#### 3. Expected level of communication with other team members:

Team members should respond to one another within the same day. Team members should be sharing important information with each other or asking each other questions over Discord. Expect responses at a reasonable time of day.

#### 4. Expected level of commitment to team decisions and tasks:

Full commitment when possible

#### Leadership

#### 1. Leadership roles for each team member:

Client Interaction / Team Organization: Connor Hand

Meeting Time Tracking / Note Taking: William Mayer

Testing / Individual Component Design: Noah Butler, Zachary Scurlock, Peter Hurd

Budget Handling: Peter Hurd

#### 2. Strategies for supporting and guiding the work of all team members:

Constant communication

Ask for help whenever

Rubber Ducky theory

#### 3. Strategies for recognizing the contributions of all team members:

Time management

Have responsibility for team actions

Talk about what we did individually at each meeting

#### **Collaboration and Inclusion**

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

team.

Connor Hand: Hardware design, some electronic circuits, computer theory

Zachary Scurlock: Hardware design, computer theory, software design

Peter Hurd: Soldering, PCB design, electronic circuits, computer theory

William Mayer: Software concepts, multiple coding languages, Assembly

Noah Butler: Soldering, hardware design, software design

# 2. Strategies for encouraging and supporting contributions and ideas from all team members:

We are using Discord for our communication so we can put our ideas in our Discord server, where they will be forever and everyone can see. We can also talk about our contributions and ideas during our meetings.

#### 3. Procedures for identifying and resolving collaboration or inclusion issues:

By having an open-minded conversation. We all agree that our meetings are a safe space to talk about any problems we have within the team.

#### Goal-Setting, Planning, and Execution

#### 1. Team goals for this semester:

Develop a functional prototype. Have a plan that we can execute next semester.

#### 2. Strategies for planning and assigning individual and teamwork:

Determine work to be done at meetings and create a list of items that need to be completed. We can take the items that we want to do or the team has agreed on and assign our names to them.

#### 3. Strategies for keeping on task:

Meet consistently and communicate constantly. Create personal goals to spend a certain amount of time on our project individually.

#### **Consequences for Not Adhering to Team Contract**

Open-minded conversations and heavy communication if infractions repeat. Communication with Stoytchev on specifics of infractions. If infractions continue after talking about it with the team, the rest of the team and Stoytchev will determine the next steps for the person committing the infractions.

- 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) Connor Hand	DATE 9/18/2024
2) Zachary Scurlock	DATE 9/18/2024
3) William Mayer	DATE 9/18/2024
4) Noah Butler	DATE 9/18/2024
5) Peter Hurd	DATE 9/18/2024