iFPGA - Intermittent Intelligent FPGA Platform

Design Document

sdmay20-38

Client: Henry Duwe

Advisers: Henry Duwe

Team Members/Roles

Justin Sung - Embedded Systems Analyst

Zixuan Guo - Systems Diagram Expert

Jake Meiss - Electrical Engineer

Andrew Vogler - Hardware Applications Scientist

Jake Tener - Software Technician

Team Email: jsung7@iastate.edu

Team Website: http://sdmay20-38.sd.ece.iastate.edu

Executive Summary

Development Standards & Practices Used

Hardware and software we will use in this project:

- Wifi harverst
- FPGA Circuit Board
- Microphone
- Voltage Booster
- Capacitor array
- Transmitter

Engineering Standard

- Honesty about the functionality and usefulness (#'s 3 & 6)
- Emphasis on Teamwork (#'s 7, 8, & 9)
- To make the highest quality product within our capability (#'s 5 & 6)

Summary of Requirements

- Low Power
- Ability to checkpoint progress in a program
    - Intermittent execution on an FPGA platform with frequent power cycling

Applicable Courses from Iowa State University Curriculum

- CPRE488
- CPRE381
- EE330
- CPRE281
- CPRE288

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

Table of Contents

1 Introduction 4

1.1Acknowledgement 4

1.2 Problem and Project Statement 4

1.3 Operational Environment 4

1.4 Requirements 4

1.5 Intended Users and Uses 4

1.6 Assumptions and Limitations 5

1.7 Expected End Product and Deliverables 5

2. Specifications and Analysis 5

2.1 Proposed Design 5

2.2 Design Analysis 6

2.3 Development Process 6

2.4 Design Plan 6

3. Statement of Work 6

3.1 Previous Work And Literature 6

3.2 Technology Considerations 7

3.3 Task Decomposition 7

3.4 Possible Risks And Risk Management 7

3.5 Project Proposed Milestones and Evaluation Criteria 7

3.6 Project Tracking Procedures 7

3.7 Expected Results and Validation 7

4. Project Timeline, Estimated Resources, and Challenges 8

4.1 Project Timeline 8

4.2 Feasibility Assessment 8

4.3 Personnel Effort Requirements 8

4.4 Other Resource Requirements 8

4.5 Financial Requirements 9

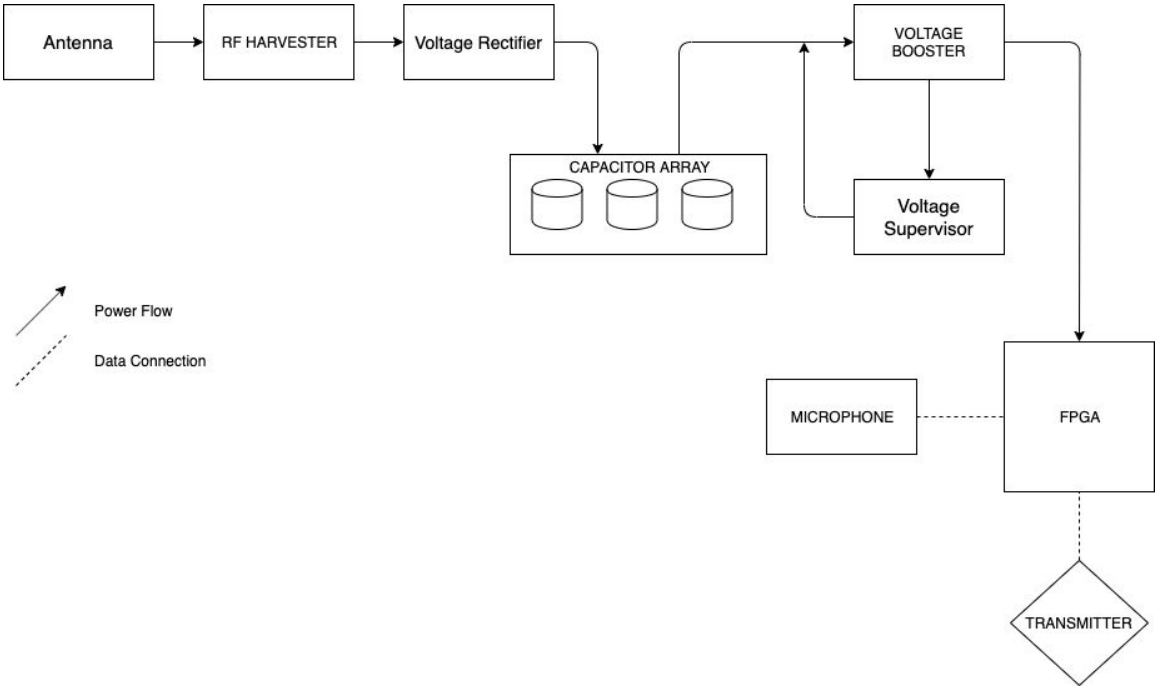
5. Testing and Implementation 9

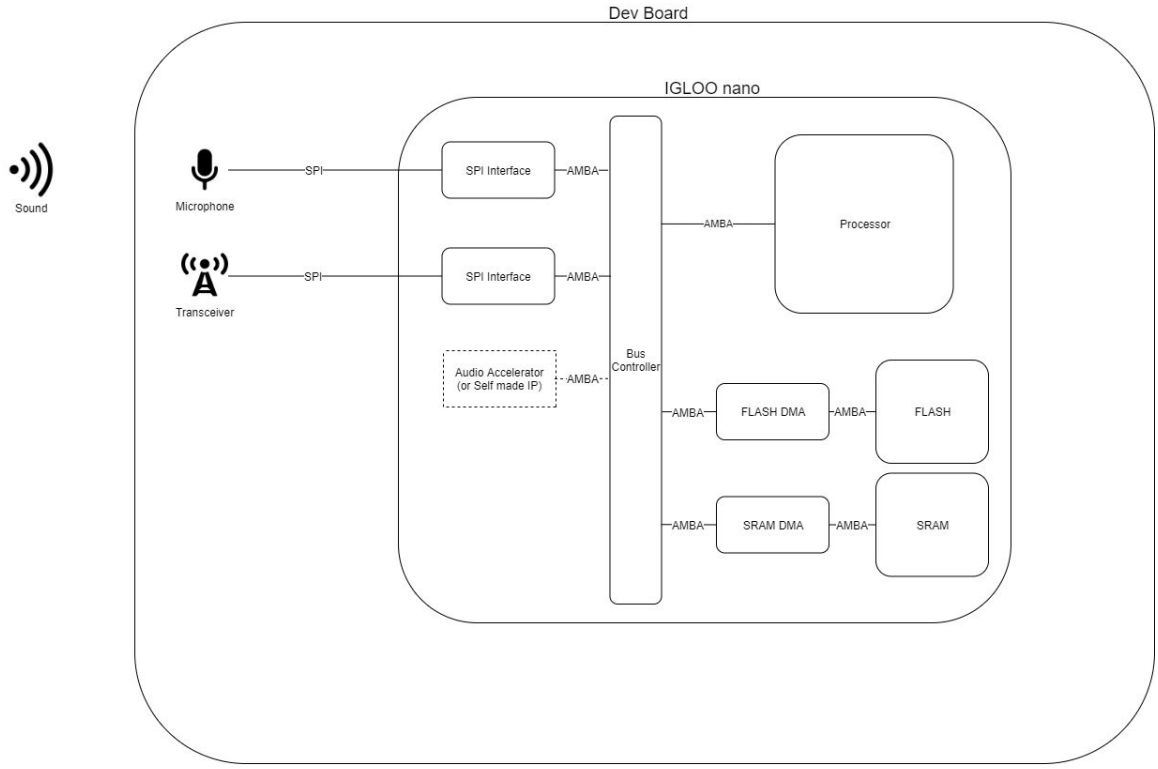
5.1 Interface Specifications 9

5.2 Hardware and software 9

5.3 Functional Testing 9
5.4 Non-Functional Testing 9
5.5 Process 10
5.6 Results 10
6. Closing Material 10
6.1 Conclusion 10
6.2 References 10
6.3 Appendices 10

List of figures/tables/symbols/definitions





1 Introduction

1.1 Acknowledgement

Narayanan Vishak - Assisting in the design development.

Henry Duwe - Advised and set concrete goals for the team to work towards.

1.2 Problem and Project Statement

Develop a hardware and software solution to intermittently execute a program targeted on a low power FPGA platform that can withstand multiple power cycling events.

As IoT applications become more prolific and integrated into society, the demand for more efficient and powerful devices grow. To address these demands, advancements in all aspects of embedded systems (e.g. memory space and speed, computational power, size, etc.) have become promising areas of study. Power consumption is one such area that has attracted a lot of attention recently. Reducing the power consumption of IoT devices results in longer device life, broaden IoT applicable areas, etc. . Thus, our project, iFPGA, attempts to study some pressing issues related to reducing power consumption.

The iFPGA is a low power designed FPGA platform powered completely by a radio frequency harvester device to intermittently execute an audio recognition program with resilience against frequent power cycling events. If successful, the iFPGA prototype will uncover novel design solutions to address unreliable power sources, and broaden the feasible areas where IoT can be applied. The prototype will lead to more advanced designs that will build and improve upon any shortcomings of the final prototype. purely for research, the iFPGA will be applied to audio recognition.

1.3 Operational Environment

Since the iFPGA is purely a research based project, our prototype has no real-world application. It is purely to design a potential solution to address unreliable power sources.

1.4 Requirements

Functional Requirement

- Self energy harvesting
    - Rectenna
- Data transmission
    - Bluetooth
    - RF
- Program execution that can pause and continue while power toggling
    - Non-volatile LUT
    - Non-volatile memory

    - Off-chip memory

Non-Functional Requirements

- Low Power
    - Must be able to detect, perform computations, and transmit data at low power
- Distance of Transmissions
    - Must be able to transmit data sufficient distances
- Sensor Range
    - Sensor must be capable of capturing nearby audio
- Little to no human intervention after deployment

1.5 Intended Users and Uses

This is the product designed for collecting and analyzing audio based data on an FPGA with an emphasis on low power design. It can be applicable to areas such as IoT and a prototype for developers to expand the study and project.

1.6 Assumptions and Limitations

Assumptions

- Purely a feasibility project, no real world problem being addressed.

Limitations

- cost of prototype shall not exceed 200USD
- as low power as possible

1.7 Expected End Product and Deliverables

A prototype that can accelerate a portion of the audio recognition pipeline that demonstrates the ability to complete the program successfully with frequent power cycling events throughout its execution. Powered by a radio frequency harvester feeding into a capacitor array device. A demo to show these capabilities suffice in a successful end product.

Comprehensive documentation describing and explaining how to interface and justifying our design choices will be included with the end product prototype.

2. Specifications and Analysis

2.1 Proposed Design

- Market survey comparisons of potential components such as: FPGA, microphone, transceiver, etc.
- FPGA power simulations and computational power analysis
- Audio recognition software pipeline analysis to determine which step is most feasible on the platform.
- RF harvester, voltage booster, and capacitor array component analysis

2.2 Design Analysis

Have not started designing.

2.3 Development Process

We are using a waterfall-agile mix for our workflow. Since many of the tasks in the early phase we are walking into with a lot of unknowns, we have to go back and forth with decisions as we learn more about FPGA design. We carry out a new set of tasks each week. Some of the smaller decisions don't have a specific due date (hence the waterfall method) but the bigger decisions, such as choosing an FPGA have a more concrete deadline (more of the agile mindset).

2.4 Design Plan

Deploy the platform in the field and be able to: (1) capture audio from its environment, (2) perform computation on the data, and (3) transmit the results from the computation to an external device.

3. Statement of Work

3.1 Previous Work And Literature

Paper that explains how to design an FPGA accelerated audio-related computational device.

https://ieeexplore.ieee.org/document/7763589

Market survey comparisons of components for the platform. Explained in previous sections in detail.

Self energy producing devices exists, but none that try to achieve something as costly, both in power and computational power, such as what iFPGA proposes. No other device in its class can sustain itself solely by radio frequency derived power while attempting costly cenergy and computations.

3.2 Technology Considerations

Low power and budget are the two main factors that guided the technology choices made. Since the end product is a prototype to assess the feasibility of such a platform, the budget was restricted and considerations were made in light of it. The low power constraint also affected the technology choices.

3.3 Task Decomposition

1. Decide on components that fit the project constraints and functional/non-functional requirements.
2. Design the hardware platform.
3. Design the software to handle the audio pipeline acceleration and intermittent programing.
4. Integrate (2) and (3).

3.4 Possible Risks And Risk Management

- Deeper knowledge of audio recognition pipeline
- Deeper knowledge of hardware/FPGA programming and development
- Graduation

3.5 Project Proposed Milestones and Evaluation Criteria

Key Milestones:

- Component finalization list
- FPGA component integration
- Power analysis and component integration
- Intermittent programing design
- Completed hardware and software integration

3.6 Project Tracking Procedures

Weekly group meetings, weekly advisor meetings, and weekly work period meetings.

3.7 Expected Results and Validation

Deploy the platform in the field and be able to: (1) capture audio from its environment, (2) perform computation on the data, and (3) transmit the results from the computation to an external device. Frequent demonstrations will guarantee that each milestone is functioning as expected and that progress is being made in the right direction and in the right way.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

	Sept 9-20	Sept 21-Oct 21	Oct 22-Nov 11	Nov 12-30	Nov 31-Dec 20	Jan 6 - May
Research Requirements						
Simulate Program & Research FPGA's						
Understand power levels & build electrical design						
Finalize diagram & Intended design blueprints						
Build individual pieces, test, and demo						
Buy parts and begin to mesh project together						

This schedule will keep our project on target while also allotting us enough time to sufficiently accomplish each important part. Our goal is to have each piece working well enough to be able to convince our client that we can build a working prototype during semester 2 by the end of semester 1.

4.2 Feasibility Assessment

The project goals in themselves should be achievable. We will have a working battery-less FPGA that can perform some sort of computation. However, our current application is having it perform sound classification on the device. We have many worries about the power intensitivity of this, and think we will have to downgrade what the FPGA will do. We will spend some time analyzing this and re-evaluate often.

4.3 Personnel Effort Requirements

Our personal effort will be very high throughout the year. This has been deemed a difficult senior project by our advisor, so it will be tough for us. We will be spending a very high volume of time during the fourth cycle (Build individual pieces, test, and demo) as we get ready to show off our project ideas in order to begin prototyping.

4.4 Other Resource Requirements

Our resources will be Libero, an FPGA design tool through a company called Microsemi. We will also be using a Microsemi low-power FPGA as well.

4.5 Financial Requirements

The financial requirements for our project is to keep everything we need to buy under $200.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case 5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested 8. Perform any necessary

retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 Interface Specifications

- Libero SoC V11.9
- ModelSim
    - Verifies that the HW works as intended.
- Synplify PRO
    - Verifies the power constraints are satisfied with the HW design

5.2 Hardware and software

- Igloo Nano ALGN250 Starter Kit
    - This was a reasonably priced FPGA and had all the specs we were looking for.
- Libero SoC V11.9
    - This is a tool that MicroSemi offers for free to use on all their FPGA products
- Powecast P2110B
    - A 915 MHz RF energy harvester that will be used to provide power to downstream devices
- Lab Instrumentation
    - Includes measurement devices such as multimeters and oscilloscopes used for testing and analysis

5.3 Functional Testing

Examples include unit, integration, system, acceptance testing

5.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

5.5 Process

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

5.6 Results

– List and explain any and all results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project

- – If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the **implementation Issues and Challenges**.

6. Closing Material

6.1 Conclusion

So far, we have gotten very far in our preliminary design. On the software side, we have got a working training set for classification of sound bytes as well as a few sound bytes to test against our training set. The next part we are figuring out how to attach the software to our FPGA. Duwe brought up using an external device to have a kernel since he is most concerned with whether the FPGA can do computations or not. Concerning the FPGA side, we have chosen to use a MicroSemi Igloo Nano product. After much consideration, Microsemi had the most support and largest range of products, that way, if we had to change the FPGA further down the road, we would still be using the same software tool. We think we will end up staying with the Igloo Nano anyway. Next steps on the FPGA side is to simulate some of the exact computations we will be doing on Libero. For the Electrical side, it has been mostly working on calculating the power of the boot sequence, steady state, and computation state. Since through this project we are trying to show that we can perform low power computations on a batteryless FPGA, the hardest thing we need to prove is that the FPGA can handle the power of running computations. Since we just recently chose an FPGA, there is testing to be done to get some actual numbers, which is the next task for the EE side.

6.2 References

N/A

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

6.3 Appendices

https://www.microsemi.com/product-directory/libero-soc/5507-libero-soc-v11-9-archive#documents

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

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