# FPGA-based Timing and Signal Generator for a Compact Electron Paramagnetic Resonance (EPR) Imaging System

Hamzah Siddique[1], Randall Pursley[1], Devasahayam Nallathamby[2], Murali Krishna-Cherukuri[2], Thomas Pohida[1]

[1]Signal Processing and Instrumentation Section, Center for Information Technology, National Institutes of Health

[2]Radiation Biology Branch, Center for Cancer Research, National Cancer Institute, National Institutes of Health

## Background

### Electron Paramagnetic Resonance Imaging

In the medical field, imaging the human body is a fundamental tool in allowing medical professionals to diagnose and improve the health of their patients. One such method in the works is Electron Paramagnetic Resonance Imaging, an imaging technique that utilizes free radicals, charged molecules, to create an image of the area, specifically looking at the oxygenation of the site being imaged. EPR imaging requires the use of an RF pulse signal to excite the electron spins of free radicals, these electrons go into a period of relaxation that then give off electromagnetic radiation that's processed into an image using signal processing techniques.

### Field Programmable Gate Array (FPGA)

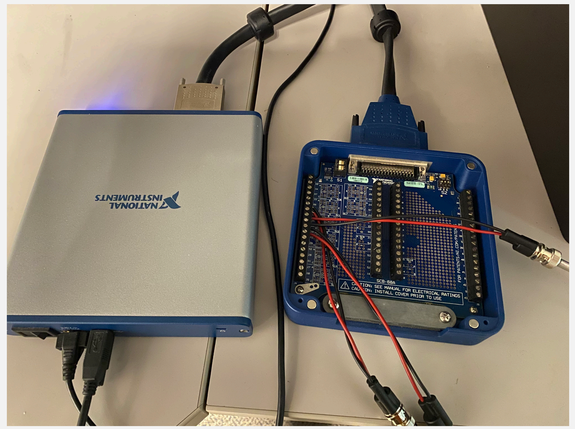
FPGAs are semiconductor devices that contain arrays of programmable logic blocks. Due to its reconfigurable nature, it has uses across many industries such as Aerospace, Communications, Prototyping, and others . To configure an FPGA, you must use a hardware description language.

### LabVIEW

This summer I've been able to work alongside my mentor to learn how to use a software called LabVIEW.
This graphical programming language allows a user to create modules that assist in testing, measuring and controlling electronic devices. Using this software, I was tasked to create a module that would be compiled onto an FPGA which generates timing and control signals for the RF signals that will be used in an EPR system.

## Results

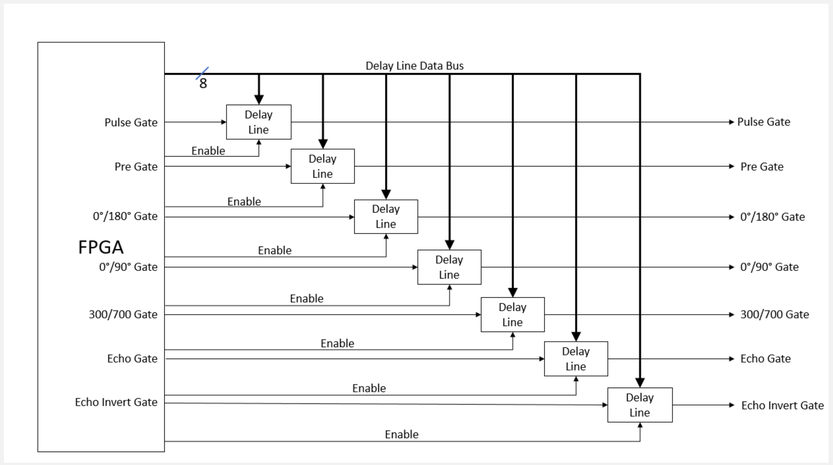
- Initial approach was more modular but lacked the tight synchronization this application requires.
- With the help of my mentor, I was able to redesign the FPGA architecture with a little more complexity but better synchronization.
- All outputs can be independently configured with customized pulse widths and delays while maintaining synchronization.




FPGA used to create pulse signals

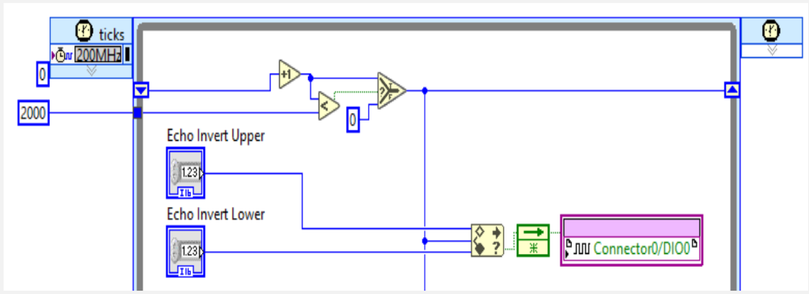

Oscilloscope displaying the signals

## Conclusion and Future Work




A block diagram of the delay line integrated with the FPGA

- Current FPGA implementation allows for 5ns resolution. This is near the limit for FPGA clock rates
- Addition of a delay line chip for each signal will increase the resolution to 150ps.




A portion of the LabVIEW code used to produce pulses

## Acknowledgements

I'd like to thank my mentors, Randy Pursley and Thomas Pohida and our collaborators, Dr. Devasahayam Nallathamby and Dr. Murali Krishna-Cherukuri for allowing me to contribute and be a part of their project.

National Institutes of Health