Spring 5-11-2024

Design and Analysis of Passive Correlator Radio-Frequency Identification Tagging

Albert Kilgore
Washington University – McKelvey School of Engineering

WASHINGTON UNIVERSITY IN ST. LOUIS

McKelvey School of Engineering
Department of Electrical & Systems Engineering

Thesis Examination Committee:
Shantanu Chakrabartty, Chair
Roger Chamberlain
Darko Ivanovich

Design and Analysis of Passive Correlator Radio-Frequency Identification Tagging

by
Albert Kilgore

A thesis presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Science

May 2024
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to first thank my family and friends for supporting me through all of my masters degree. I have greatly appreciated everything they have done for me and I would not have been able to get to this point without them.

I would also like to thank my advisor, Dr. Shantanu Chakrabartty for giving me the opportunity to explore all of the different topics I have researched during my masters degree. I believe I have had an experience that normal masters students do not have and it is all thanks to him.

I also am very thankful for the Department of Electrical and Systems Engineering at WashU for supporting me with the Department Chair Fellowship. Having this scholarship has really helped me focus on performing my research and not needing to worry about funding to attend the university.

A special thanks goes out to the members of the Adaptive Integrated Microsystems Lab and Nagulu Lab for helping me understand helping me through all of the processes for my thesis. I would like to specially thank Zhili Xiao, Subhankar Bose, and Aswin Undavalli Chowdary for all of their contributions, assistance, and patience.

I would also like to thank Dr. Aravind Nagulu for his feedback and access to testing equipment used within this work. Additionally I would like to thank Professor Ed Richter for access to additional equipment.

This work was supported by DARPA/AFRL MAX Program Contract No: FA86502327309 and Bill & Melinda Gates Foundation Grant No: INV035476.

Albert Kilgore

Washington University in St. Louis
May 2024

ABSTRACT OF THE THESIS

Design and Analysis of Passive Correlator Radio-Frequency Identification Tagging

by

Albert Kilgore

Master of Science in Electrical Engineering

Washington University in St. Louis, 2024

Professor Shantanu Chakrabartty, Chair

Radio Frequency Identification (RFID) is a technology used in many industries to locate and track assets. Passive RFID tags are popular because they are inexpensive and flexible, but they have limited accuracy. My thesis aims to improve the accuracy of passive RFID tags by investigating two solutions: (a) Using orthogonal RFID configuration and exploiting phase information available to an RFID reader, in addition to the received signal strength indicator (RSSI) metric; and (b) Designing a novel voltage multiplier-based correlator circuit that can be integrated directly onto the tag. To further reduce power consumption and silicon area, we propose an architecture that combines the energy-harvesting capability of RFID power circuits with the calculation of correlation. This results in a tag peripheral circuit that consumes minimal power and generates a unique output voltage based on the correlation of input signals. These correlator architectures enable bundling of multiple data types into a singular input, such as clock, signal, and power. The designs are fully passive, and the only power consumption occurs when the measured correlation exceeds a pre-specified threshold. We analyzed these correlator architectures from design proposals to physical testing with designed IC chips to provide an initial hardware characterization.

Chapter 1

Background

Radio Frequency IDentification (RFID) is a type of system which is very present in the modern world. Many different industries use these tags for purposes such as logistics, wireless user authentication, and as bar codes for stores. One important application which is widely explored is their use for item tracking due to their small and non-intrusive size along with their ability to be fully passive, or operate without a directly connected power supply. Tracking of these tags has also been used in multiple applications from gesture recognition [1] to tracking patients for health monitoring [2] due to low skin penetration, allowing for the design of safe wearable.

Correlation is also a type of measurement typically measured in many RF applications though is not normally utilized with RFID systems regarding input signals. Correlators are often used in imaging and radar systems. Typical modern correlation systems are high power consumption units which utilize many hardware resources to compute the similarity between input signals. An interesting alternative method could be taken to implement these correlators to reduce the power, and that is to create systems which do not use digital processors to calculate the correlation and instead operate within the analog domain.

To address both of these topics we have proposed an Application Specific Integrated Circuit (ASIC) which can implement both the energy harvesting and correlation details that could be used on an RFID tag. First before going into the details of the designed topologies, we should first go into more depth about specifics regarding the topics of correlation and RFID to get a better understanding of the targets for our design.

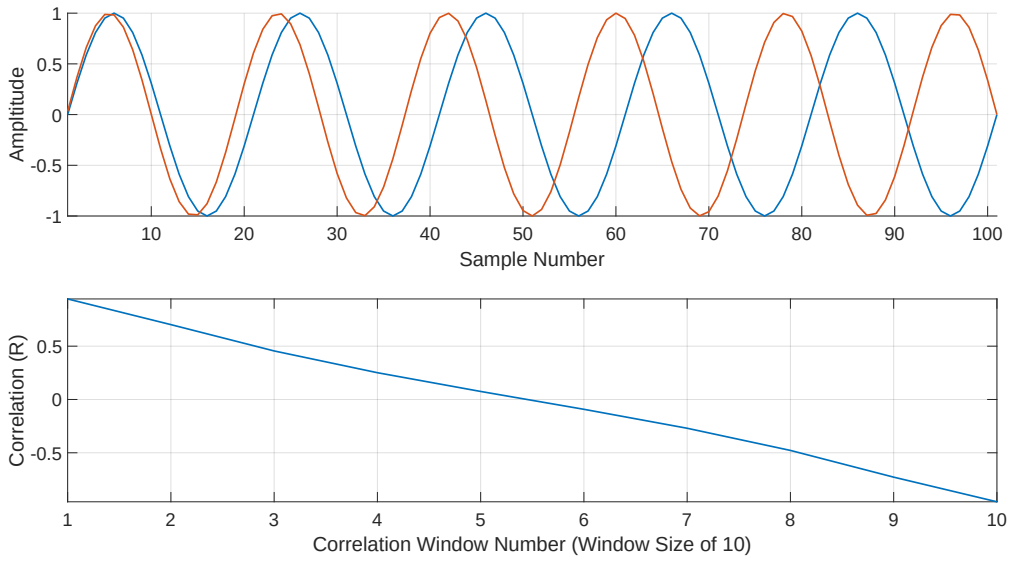

Figure 1.1: Example of a Correlation Calculation between two waveforms

1.1 Correlators

Correlation is the process of taking two input signals and finding the similarity between the two signals to generate an output which describes the similarity between them. An example of this can be seen in Fig. 1.1 which displays two sinusoidal inputs, one with a constant phase, the other with a swept phase. We can see here that as the phase difference increases between the sinusoids, the correlation decreases and reaches a minimum which occurs when the signals are anti-correlated (or out of phase by 180 degrees).

One important application of correlation is using Log-Sum-Exponential (LSE) estimators [3]. Within the field of machine learning LSE estimators can be used for transforms which are easy to optimize over traditional feed-forward neural networks in the training phase. Part of this work will focus on developing a hardware implementation of an LSE correlator which can be performed using diodes. The equation below displays the basic equation for LSE estimators.

$$LSE(\underline{x}) = log(exp(x_1) + exp(x_2) + ... + exp(x_3)), \quad where\ \underline{x}\ \epsilon\ \mathbf{R}^n \tag{1.1}$$

Exponential diodes can provide a simple way of calculating the LSE estimator due to the I-V characteristic of the device. LSE devices are able to provide correlation of the different

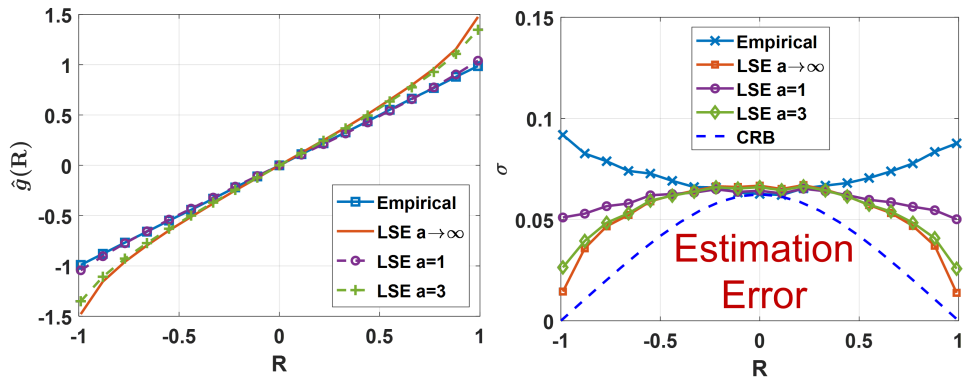

Figure 1.2: MP Calculations with Trans-linear Diodes

components of $\underline{x}$ which provide some information about the correlation of the inputs to the LSE function. The same could be applied with a diode structure based on the potential at the anode of the diode. Simple calculations of correlation have been attempted with this structure and are displayed in Fig. 1.2.

Another type of correlation to consider is Margin Propagation (MP) Correlation. This form of correlation involves taking the summation of multiple input "margins" to determine the confidence of a calculation of correlation between inputs [4]. Where

$$P_n = [x_n - k] \tag{1.2}$$

Where k is a threshold value constant across all iterates. This value will then be rectified utilizing the ReLU function which can be expressed as the following.

$$[z]_+ = max(z, 0) \tag{1.3}$$

Is a rectified margin with a calculation offset removed, the following equation can then be held for multiple inputs.

$$\sum_{n}^{N} [x_n - k]_+ = \gamma \tag{1.4}$$

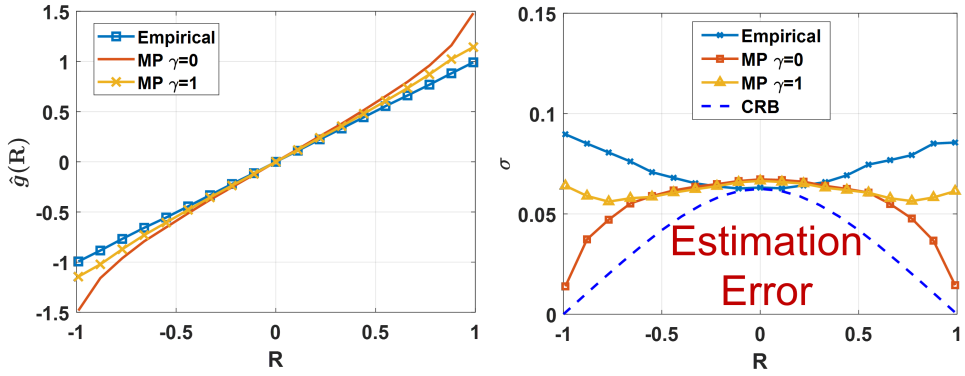

Figure 1.3: MP Calculations with Rectifying Diodes

Where γ is our MP result for the correlation. Similarly to LSE estimators, rectifying diodes can also be explored to calculate the correlation type. Fig. 1.3 shows an example of a design which calculates the MP algorithm.

Another more practical example of an application for correlation is in radar systems. Many different radar systems use the correlation of a signal to compare a received signal to a target signal. This correlation between the two signals can help the radar system determine if any received signal is the desired input for a certain result. This sort of application can also be robust to noise since essentially a correlation can be understood as a convolution and if the Signal to Noise Ratio (SNR) is very high, then the convolution will not be too greatly affected by the noise of the received signal. This allows for higher-quality communication systems in critical sectors where receiving certain signals using a radar is important.

Auto-correlation is also another important application of correlation. A single signal is correlated with itself in this application and requires time-shifting to occur. This could be useful especially for determining if an input signal is periodic or to find hidden features within a transmission. Auto-correlation will not be the focus of the designs within this work, but could be a potential application of the designs explored.

Most correlator circuits implemented using standard CMOS technology are typically developed using digital components. Recent research into these designs have begun to focus leveraging the use of stochastic computing to calculate the correlation of these signals due to the potential to increase the accuracy of computations while improving the density and power usage within an array [5]. Other research has also been performed to design these correlators using purely analog designs. The reasoning behind this change in circuit type

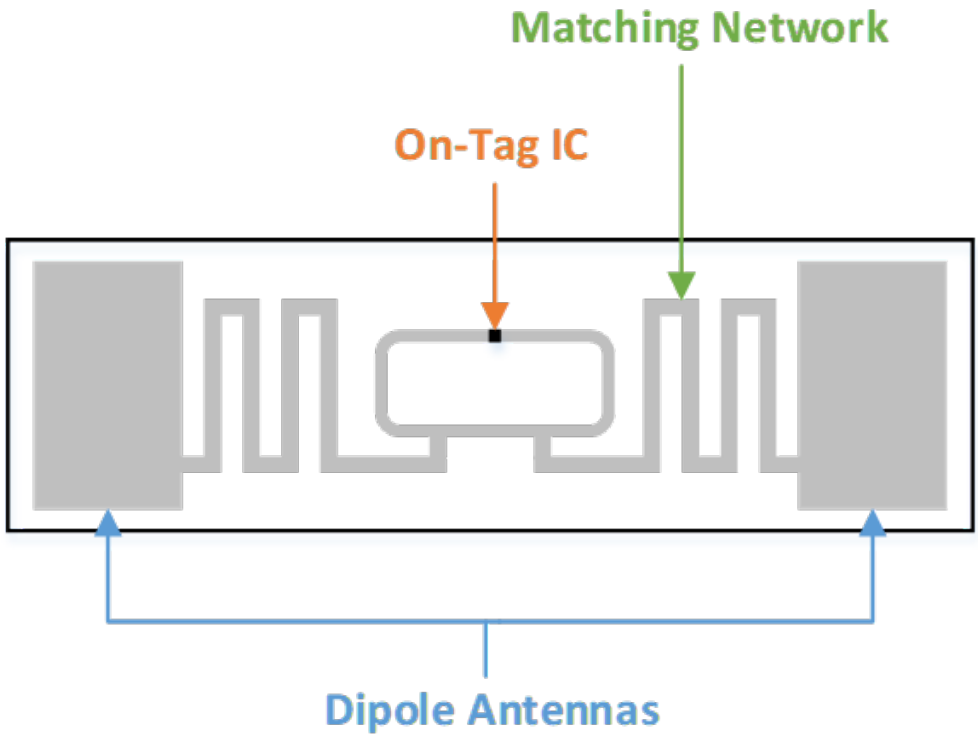

Figure 1.4: Three Main Components on an RFID Tag

comes from the potential of removing the clock signal from the designs which could reduce the power usage of the circuit since less switching occurs [6]. The main trade off of this design change is the increased area usage on ICs.

1.2 Radio Frequency IDentification (RFID)

RFID Tags are specialized ICs which allow for wireless transmission of data between an RFID reader and antenna on-board the IC. All RFID contain three main parts: on on-tag IC, a dipole antenna for transmitting or receieving signals, and a matching network to ensure maximum power transfer between the dipole antennas and the on-tag IC. Fig. 1.4 labels these three main components on a standard passive RFID tag.

These designs come in two forms which are passive and active. Active RFID tags have an on-tag power source and can provide a farther read-range. Passive tags do not have an

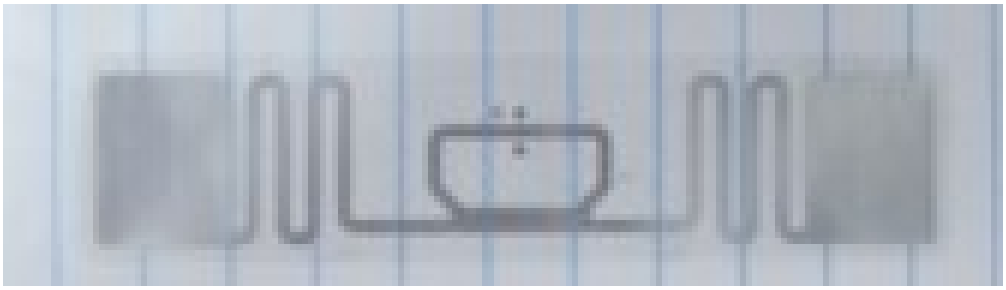

Figure 1.5: An example of a passive RFID tag on a piece of paper

on-board power source and are instead powered by the signal transmitted from the RFID reader and will be the focus of RFID tags within this work. Passive tags also can come in a flexible packaging, are non-intrusive, and be quite small. This can be very useful for logistic applications such as wireless identification on multiple mediums including shipping or even for clothing [7]. Fig. 1.5 shows an example of a commercial off-the-shelf passive RFID tag on a sheet of paper.

RFID tags also generally have a low amount of data overhead when reading their transmitted information on a processor. This allows for readers to be able to process up to thousands of tags per second. RFID tags are generally operated using a system of a transmitter which is used to broadcast an RF signal at a specific frequency value or range, a tag which acts as a receiver and transmitter, and a separate receiver which receives the transmitted signal from the tag. RFID readers and transmitters are typically connected to an RFID reader which allows for control over the transmitted or received data. These can be anything from a portable device to a stationary device attached to a computer for control and processing. The process of transmitting data from an RFID tag to a receiver is known as backscattering [8] and occurs from modulating the received signal by quickly switching impedance on chip. Fig. 1.6 displays the basic functionality of RFID data transmission.

RFID readers typically allow for high-throughput of tags, meaning that around a thousand tags can be read per second. Having such a high throughput allows for applications using dense RFID configurations which can be exploited to provide many different types of data. Though this concept will not be explicitly explored within this work, high tag count is a key motivator for the topics explored.

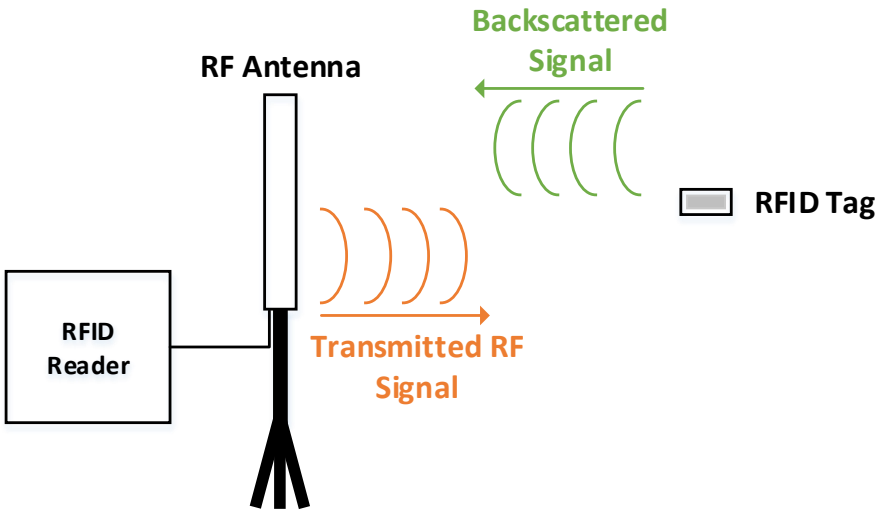

Figure 1.6: RFID Backscattering Example

1.3 Energy-Harvesting Circuits

Energy-Harvesting Circuits take an input power source and boost its voltage to a higher level to be used within a circuit. This is very useful for wireless communication since it allows for a low-power transmission to be made to provide power to the design. Energy-Harvesting Circuits are often used within passive RFID tags as its power source when boosting the energy received from the transmitting antenna. Typically RFID tags implement a Dickson Voltage Multiplier structure which convert an RF signal into a DC signal by creating diode-capacitor chains to rectify and store charge between multiple stages [9]. Fig. 1.7 displays an example of a two-stage Dickson Multiplier Structure.

Current research has been motivated in decreasing the losses per-stage by converting these chains into MOSFET-capacitor chains due to newer CMOS technologies supporting a lower threshold voltage value [10] but could come at the cost of lowering the highest attainable voltage through these devices due to CMOS operating limits.

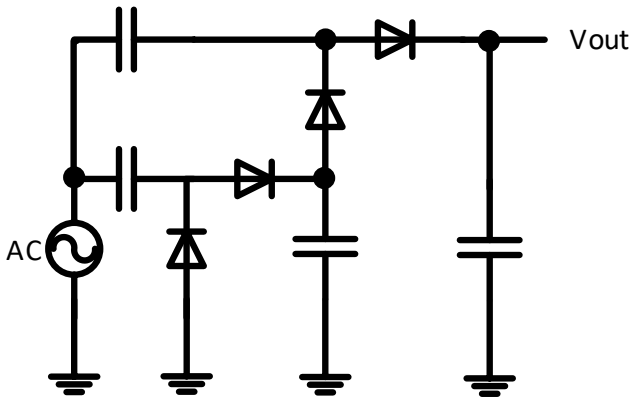

Figure 1.7: Two-Stage Dickson Multiplier

1.4 Motivation, Challenges, and Contribution

Many existing correlator topologies require the use of digital components and gates to calculate the correlation of two input signals. A significant issue that is being observed for digital components is while the process sizes of ICs continue to decrease, the power density of digital components continues to increase leading to a need to reduce the power of components. Our design intends to tackle this issue by creating a fully analog design for correlator circuits which eliminates the need for digital switching components which should greatly reduce the power usage of the overall system.

Previous research on the topic of RFID topics such as localization require the need for reference tags in known positions [11], additional hardware connected to the tags, or using a large amount of antennas arrayed to observe tags in order to accurately estimate the position of the tag [12]. The need for these additional parameters mean that these RFID setups are not easily re-configurable or require more area or power usage on an integrated circuit which could instead be used for additional functionality or to reduce the footprint of the design. Existing works have also found that using the correlation of antennas can be a simple means of identifying the position of an RFID tag but would require additional hardware or post-processing to calculate the correlation to ensure an accurate position estimate has been

made. Our design intends to remove the need for post-processing of the tag data and instead perform this correlation calculation on-chip and reduce the power usage of the design by only powering other peripherals if a correlated signal is detected.

This thesis focuses on a simple structure which is analyzed, simulated, fabricated, and tested for both uses as a basic correlator and for potential application in an RFID design. The design has been created in multiple design processes including 130nm, 65nm, and 22nm while including multiple different topologies to determine the effectiveness of the different design variations.

1.5 Thesis Overview

Within this work multiple topologies of our proposed correlator circuit will be explored and compared both in simulation and actual hardware. This process involves creating schematics and simulating the designs within Cadence Virtuoso, creating a layout of the design in differing processes, designing a PCB or writing firmware (if needed) for each of the hardware designs, then analyzing their performance with actual inputs. After determining their performance and functionality, a Software Defined Radio (SDR) will be used to determine the functionality of the developed ICs using external antennas to act as an RFID tag. Future improvements and works for these designs will be explored and recommended.

Chapter 2

Exploration of RFID Applications

Before the exploration into the circuits designed within this work the motivation for creating these designs must be established and explained due to the complex nature of why such a design is needed. This section will discuss the background that solidified the need to create the designs which are explored within the work.

2.1 RFID Tag Localization

As mentioned earlier, a hot-topic for RFID research is tag localization and tracking. This is due to the low risk factor of these tags due to their low skin-penetration, low cost, and allows for tracking to be achieved indoors. Many different applications have been explored to implement tag localization such as warehouse item tracking [13], navigation for robotics applications [14], and tracking gestures of a human. A large limitation of these applications is that they often require pre-placed reference tags within a space which adds the need of a setup or constant environment for the application to fully function [15]. This sort of limitation of needing references could be a large constraint in dense areas of interest since only a finite number of tags can be sampled per second by most RFID readers. This also means that systems are not fully portable. One issue we wanted to address was the removal of these reference tags and to be able to track an item with a high precision. As an initial exercise for this work we attempted to localize and track RFID tags locally which created the motivation for the designs which will be explored throughout the rest of this work.

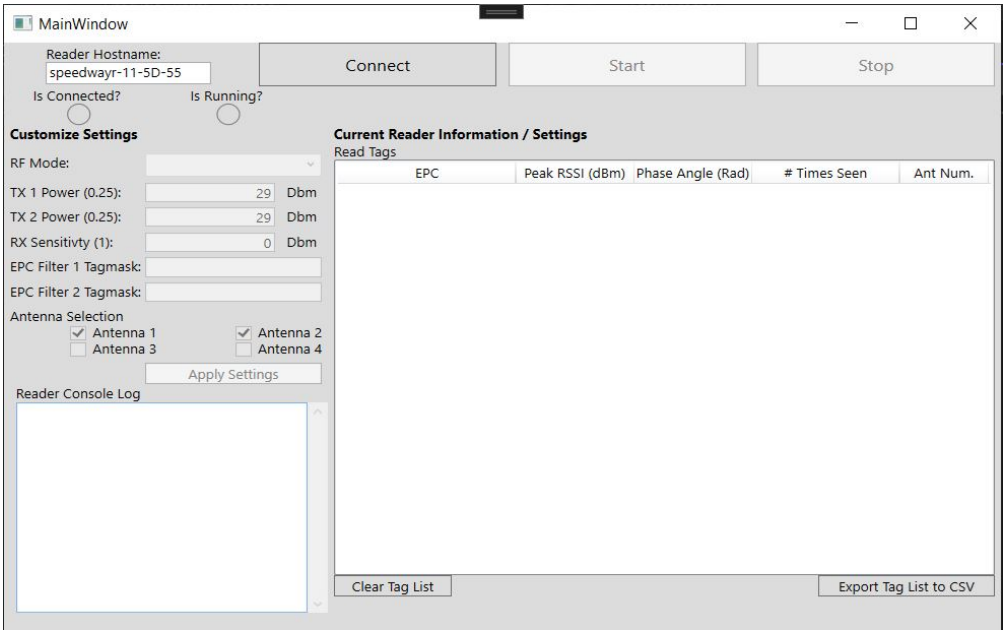

Figure 2.1: Developed Software for RFID Control

2.2 Hardware Used and Software Developed

To attempt to address these issues we used an Impinj Speedway R420 RFID reader along with two circular polarized antennas to both transmit and receive RF signals from the target tags. The reader used supports up to 4 antennas which could be useful for the application discussed within this work. Software was also developed using the C# language using the Impinj Octane Software Development Kit (SDK) to create software for controlling the reader and collecting tag data. Multithreading of data collection and processing is very important for this application since the receive rate of the reader itself is up to 1000 tags per second. Due to the high speed needed to receive and perform basic processing for output, the actual processing of tag data has been left to be completed in post-processing. Fig. 2.1 displays the developed software GUI for controlling the RFID reader and collecting tag data.

Since post-processing is needed to utilize the data collected, data is post-processed and visualized within MATLAB.

2.3 Processing of Tag Data

After collecting the data within the developed application, the tag data is then processed within MATLAB to extract features which can be useful for future applications. Three different data-types are readily available from Impinj RFID Readers on a tag read: Received Signal Strength Indicator (RSSI), Phase Angle, and Doppler Frequency. RSSI data provides a low precision (0.5 dB) result of reflected power from the RFID tag thus it isn't helpful for the desired application since a high precision is desired. It is however, a decent estimator of the position of a tag relative to the reader since it can estimate a rough radial distance from the reader since it is essentially a power measurement. Doppler Frequency provides an estimate of the change in position from an RFID tag and reports it in terms of a frequency value and provides more accurate results at a lower tag read rate. This data has a relatively high precision and could be useful for determining speed, but does not have a direction parameter and the resolution of the data is tied to the packet size which can slow down the sampling rate. Lastly, phase angle also provides a relative change in position similar to doppler frequency but has a higher resolution (0.0015 Radians or $39\mu m$) and can provide a metric for change in direction. Phase data is generally preferred in our case due to the ability to track objects with a high performance in tag read rate.

One main issue with phase data, however, is the high amount of processing required to utilize the tag-data in post processing. Phase angles are a periodic value ranging from 0 to 2π and can be represented by equation 2.1 [16].

$$\theta = \frac{4\pi R}{\lambda} + \theta_o = 4\pi R \frac{f}{c} + \theta_o \tag{2.1}$$

Where θ_o is a phase offset which is equivalent to

$$\theta_o = \theta_T + \theta_R + \theta_{tag} \tag{2.2}$$

Where θ_T, θ_R, and θ_{tag} are additional phase offsets caused by the RFID reader's transmitter, reader's receiver, and tag's reflection properties respectively.

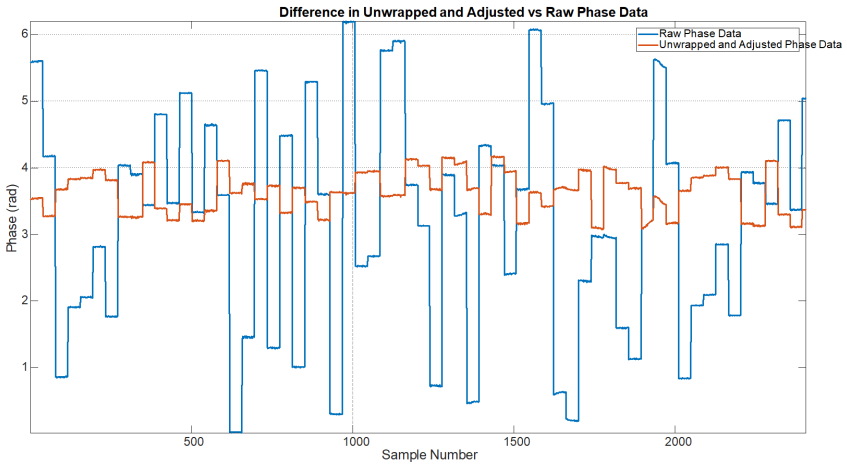

Figure 2.2: Example of Wrapped and Unwrapped RFID Phase Data

In the phase equation R represents the distance the RF signal travels from the transmitter to the antenna and λ is equal to the wavelength of the transmitted signal which is expanded and shows a dependence on the ratio of signal frequency to the speed of light, θ_o is an offset phase rotation which is a sum of offsets caused by the tag, reader, and distance the wave travels. This means that phase angle can come with the issue of 'wrapping' when the phase value approaches 0 or 2π and will jump in value when reaching one of the bounds. Fig. 2.2 shows an example of both wrapped and unwrapped phase data.

Another issue comes with the fact that the calculated phase value is dependent on the RFID reader frequency. Due to FCC regulations RFID readers must 'hop' around different frequencies which span between 902-928 MHz within random intervals to prevent any interference with other RF bands. This means that the phase will be constantly changing due to RF channel hopping. All of these additional requirements for the reader makes doing real-time processing of RFID tags difficult as many steps are needed to process the phase data since all phase data must be adjusted relative to the parameters of the collected data point and information from previous points for calibration.

2.4 Correlation within RFID Data

While performing applications which will be discussed later in this section, an interesting discovery had been found. RFID phase data generally can provide a direction, but sometimes it is wrongly estimated from the small movements of a tag. This could cause issues when performing item tracking, especially when the desired region of interest to track is small (such as milimeter scale). Since phase data has such a high resolution and sampling rate, we had decided to use this data type to track objects. To tackle the issue of a false prediction of direction being recorded by the reader, a second RFID tag is proposed to be used which is cross-coupled with another tag to verify the direction of movement from the body being tracked. BY measuring the data from a different angle, we can correlate the movement data of the body to determine if the recorded movement direction by the RFID tags are correct and to adjust the recorded direction. This comes at a cost of increasing the number of tags being used but can greatly increase the accuracy of movement measurements. Fig. 2.3 displays the cross-coupled RFID tag which will be used for the different applications shown in the future sections.

It is important to remember that these are basic fully commercial-off-the-shelf tags and do not contain any custom hardware. This sort of configuration means that these tags are essentially bar codes and all important data measured within this application is from the RF properties recorded by the reader.

2.5 RFID Item Tracking using Correlation

The first application using this cross-coupled tag explored is item tracking due to the ability to detect small movements with higher accuracy. Two RFID antennas are used for the tracking so reconstruction can be performed within a 2D plane to determine the new position of the object. One issue with this method is that phase data is not a good estimator of initial position due to the periodic phase wrapping, so the initial position will be assumed to be at the mid-point of the two antennas. The item of interest for these items will be moved in the x and y directions relative to the readers, where a positive x and y values are moving towards antenna 1 and antenna 2 respectively. This test was repeated with both a single-tag and the cross-coupled tag configurations to demonstrate the data that can be picked up through

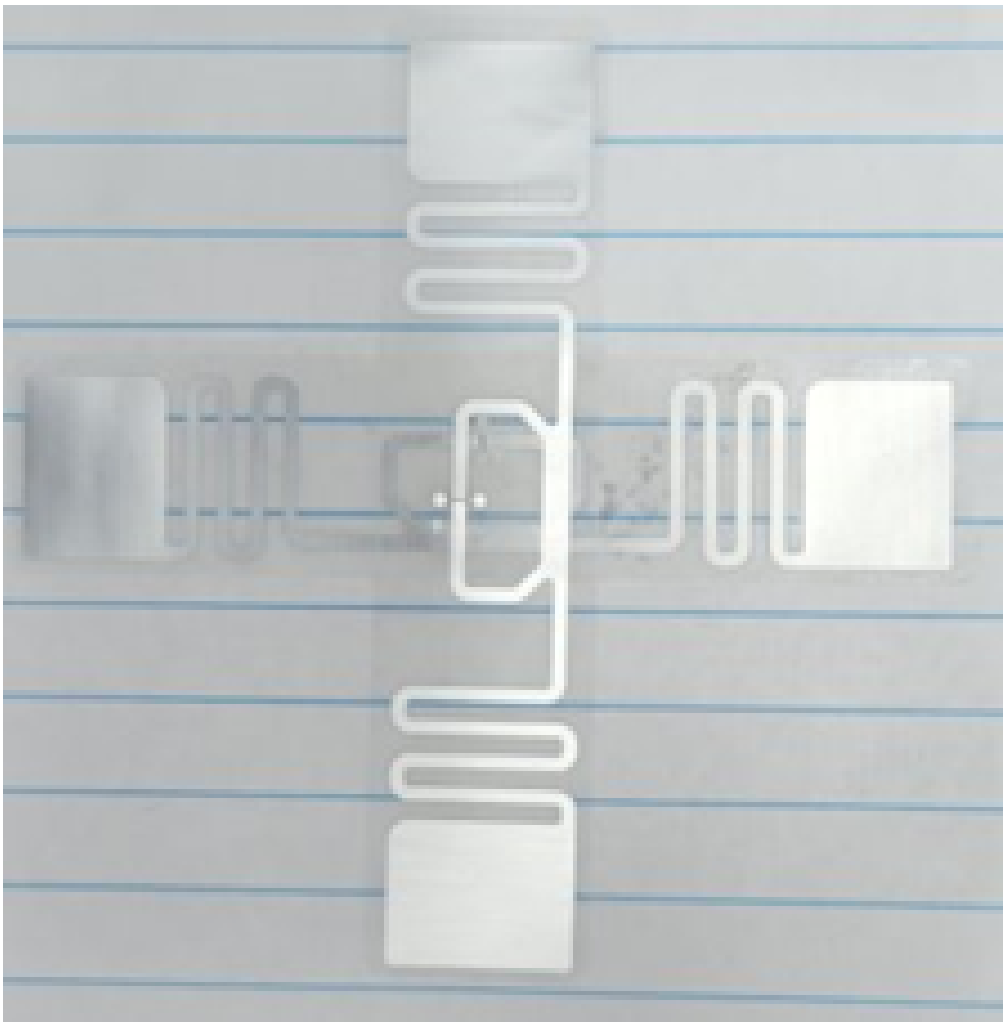

Figure 2.3: Cross-Coupled RFID Tag

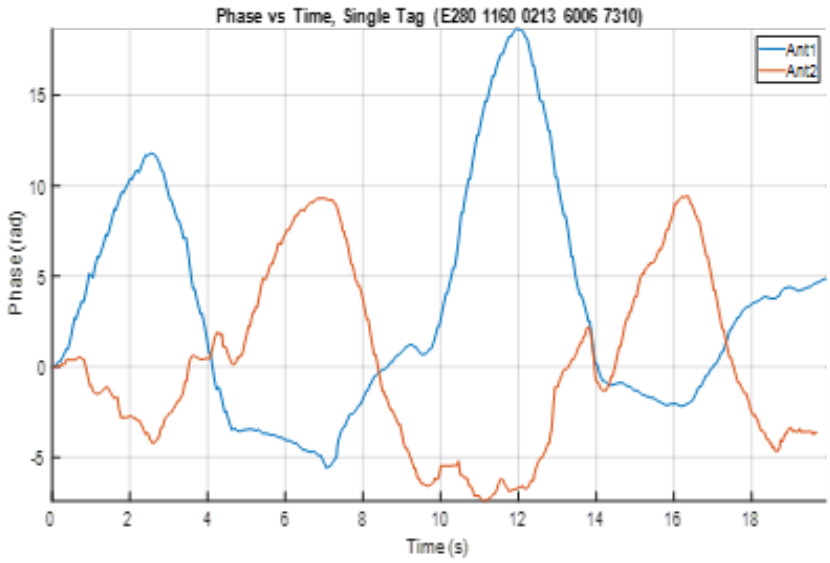

Figure 2.4: Movement Recorded for a single RFID tag moving in the x and y planes

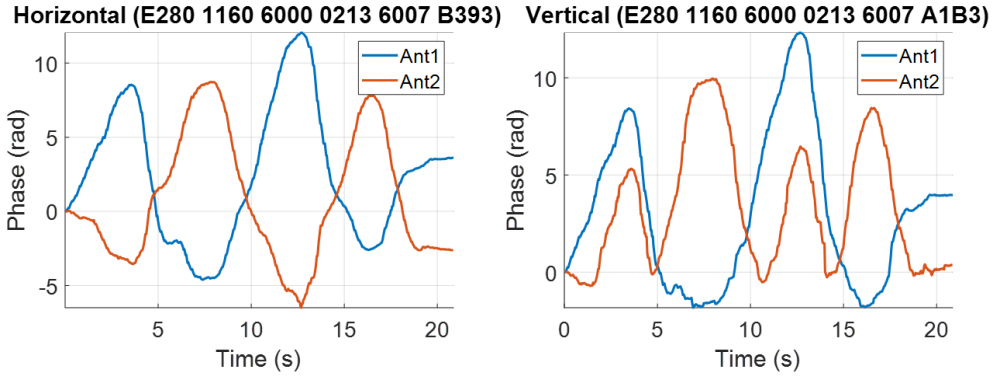

Figure 2.5: Movement Recorded for the cross-coupled RFID tag moving in the x and y planes

tag data correlation. Fig. 2.4 shows the generated phase data for the single tag and Fig. 2.5 shows the generated phase data for the cross coupled tag with each respective tag's data side-by-side.

We can see that for the single tag configuration does see a large peak when moving towards the antenna and decreases when moving away but the opposite antenna also experiences a change in phase data. This is not ideal since the measured tag is moving strictly in the x or y-planes and not diagonally towards the antennas. With the cross-coupled configuration, the tag is seen to have similar peaks across both antennas when moving in the direction of the antennas but also experiences opposite peaks from the opposite antenna. This can

be determined as a "false flag" movement detection with respect to the antenna and could be potentially cancelled out to improve the movement detection from the phase data. This allows us to track the objects within a 2D plane with a higher accuracy. Ideally this configuration could also be extended into 3D tracking with a third antenna being used but is not explored within this work. Corrleation can be seen to be a key aspect with improving the accuracy of movement detection using RFID with the only cost on the tagged object being an additional tag.

This sort of correlation can easily be applied within the medical field since the penetration of the RFID signals are small and it has a high level of precision. An example could be for tracking electrodes on a patient in case the positioning of the electrodes and their change over time could provide detailed information needed to diagnoise the patient.

2.6 Improving the Processing Rate of Correlated RFID Data

Due to high collection rate of RFID tag data, processing of RFID data while attempting to run real-time software becomes quite difficult due to the multiple steps of correcting phase data to be used within any application. Due to the RFID tag also needing to transmit the phase data back to the reader from the tag some additional offset which is variable based on tag position is also added to the recorded phase value and could also cause issues within data processing which can cause additional noise within the processed data. This can be seen from the phase equation expressed above, where there is an additional phase offset added both from the tag's reflection properties and the RFID receiver. These sorts of phase offsets may be negligible for large phase movements, but could be critical for small movement detection especially when this offset is non-constant due to constant tag position changes. One way to reduce this cost is to perform the calculation on the RFID chip embedded within the tag. This allows for one transmission to occur for the phase data, perform the processing on the tag itself, and the correlation to be transmitted back to the reader to interpret the calculated signal correlation which would remove any additional phase offset from the transmission of phase data from tag to reader. Another interesting application which could be targeted with such a tag is 'conditional powering'. This means that the tag

will only power on if a correlated signal is detected and will remain off if receiving non-correlated signals. This could easily be used in applications such as security since a known frequency band and phase utilizing multiple transmitters must be used to turn on the RFID tag and receive the encoded data. Since in this work we target passive tags this design needs to be functional without a constant voltage source and must be powered by the antenna signal. This means that we need to encode the powering data into the RF signal which is being correlated, having no constant power source also means that if a clock signal is being used within the attached circuit then it must also be encoded within the powering signal. The one exception for this concept is if there is some form of charge storage on tag with a Voltage Controlled Oscillator (VCO). These concepts will be utilized in the design of a custom Application Specific Integrated Circuit (ASIC) which implements both correlation and energy harvesting on a single architecture.

Chapter 3

Design of a Fully-Analog Correlator

Our proposed design includes multiple different variations to achieve different output characteristic goals. These include both common-mode and differential outputs and using different diode implementations, whether that be by using traditional, schottky, or MOSFETs as the diode implementation which mimic LSE and MP functionality. Different types of diodes could be used to implement these algorithm types, but it will be assumed we will use LSE functionality since exponential diodes will be utilized on silicon. These designs are also created to be cascaded, where the bias pins of each stage will be connected to the output pins of a previous stage to allow for amplification across stages to further improve energy harvesting characteristics. This design methodology of using multiple stages also includes a center capacitor between calculation paths to store charge from each stage. One target characteristic of all designs is to both create a correlation circuit but also have the potential of acting simultaneously as a voltage multiplication circuit. Another target characteristic in our design is to bundle clock, data, and power into a single analog input. This sort of design principle can both reduce the pin-count for designs which utilize this correlator circuit and contain a unified data-path to reduce the potential of error from these inputted signals such as synchronization from input and clock signals. Another target design principle that was considered is the removal of amplifiers being required for a voltage readout. Many existing correlator circuits require amplifiers to boost the voltage to a distinguishable value [17]. This can reduce the potential area and increase power on a chip and will also be a target of our design to address. All of these designs will be analyzed, simulated, and a layout will be created to be put on-chip.

Within this work we will perform simulations and analysis for a frequencies in a range of 915 MHz and 2 GHz. Though 2 GHz is not a typical RFID frequency, we would also like to explore the potential of the designed ASIC to also function within a communications application such as for satellite applications. For smaller process nodes for actual hardware

this can likely be further explored due to small switching thresholds being more ideal for high frequency applications.

3.1 Common-Mode Design

Our proposed correlator is designed to function as both a correlator and a voltage multiplier circuit and is shown in Fig. 3.1. This design utilizes a common-mode structure, meaning that the measured output will be the sum of the two output terminals. This is due to the left-side of the circuit measuring the positive correlation and the right-side measuring the anti-correlation of the input signals. This means that we essentially will have either a positive output voltage sum if the signals are positively correlated, zero voltage sum if the input signals are not correlated, and a negative voltage sum if the input signals are anti-correlated. The design may also contain a delay due to the capacitor-diode structure requiring a finite amount of time for the signal to fully propagate from input to output due to the capacitors needing to charge. Cascading of stages then means that the amount of delay present within the structure is dependent on the number of stages. This means that our design is not exactly a zero-lag correlator, but with a low amount of stages this delay can be neglected since each individual stage contains a low RC time constant from low diode resistance, but having this lag could lead to the development of a cross correlator architecture since lagged signals are needed but would require the decoupling of input signals.

This design takes heavy inspiration from the earlier discussed Dickson Multipliers to perform energy-harvesting functionality from the correlated signals. This design comes at a cost of requiring higher peak-to-peak AC inputs due to the use of diodes, which have a high turn-on forward voltage, which could be a major issue in smaller process nodes since voltage ranges are much more restricted. Two simple methods without altering much of the design can be made to alleviate this issue, one is to use a process node which supports schottky diodes and the other to use MOSFET transistors since both of these devices can also implement LSE and MP functionality though the designs using diodes within this work will focus more on LSE-like implementations. Some other implementations that could be used for rectification of inputs to calculate MP correlation could be utilizing custom devices such as active diodes which could further drive threshold voltages lower but would require a rectified energy-harvested power source to operate [18] which would remove the power for the circuit

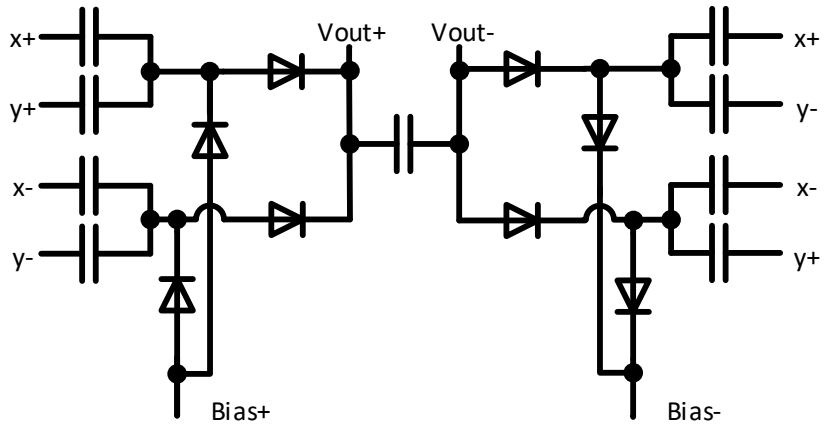

Figure 3.1: Single-Stage Proposed Correlator

being only transmitted from inputs or could add additional delay from the inclusion of startup circuits which is not ideal if a zero-lag correlator is required [19]. Some future designs will explore using transistors due to their smaller size along with more options to configure to change the operating characteristic of the architecture. An important note is that this configuration is known as a zero-lag correlator. This implies that there is little to no delay between calculating the correlation for a single stage. The addition of stage cascading however could introduce more lag which is proportional to the number of stages due to the diode-capacitor chain needed to traverse to reach the load of the final stage.

3.2 Common-Mode CMOS Design

One means of improving our initial design was to implement a CMOS variant which replaces the diodes of the original common-mode design with MOSFETs to both reduce the potential area of each stage and provide a lower threshold voltage for switching the "diode" on and off. Another advantage of using CMOS transistors is that the bulk pin can also be manipulated to further change the voltage threshold of the transistor. This bulk pin can be connected to a constant DC bias which allows for a "knob" which can manipulate the threshold voltage. Fig.

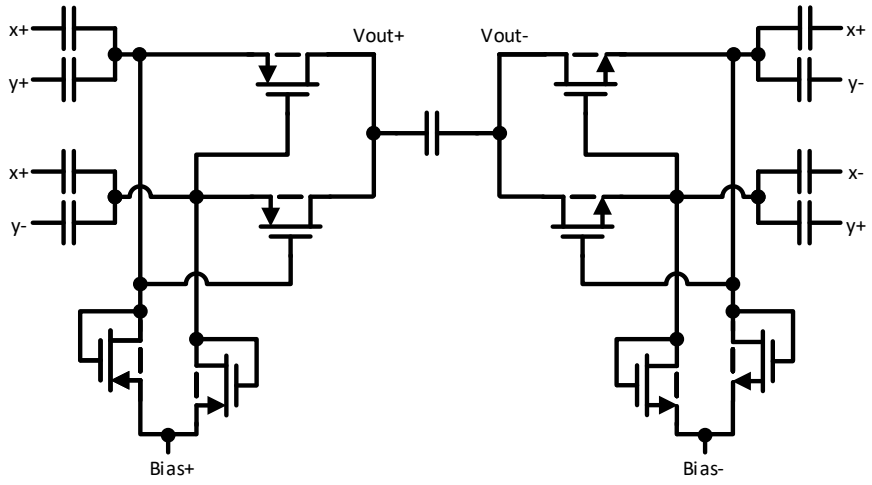

Figure 3.2: Single-Stage Common-Mode CMOS Correlator

3.2 displays the proposed common-mode CMOS design. The bulks of all PMOS transistors within this design is connected to a single DC-bias while the NMOS transistors are connected to a separate DC bulk-bias.

In this design the gate connection of the MOSFETs are connected to the node right after the input capacitors on the opposite horizontal-side of the circuit. This is important to note since this design can allow for additional verification of a correct correlation calculation since if a correlation is present, then an opposite calculation should be occurring on the other side of the circuit. This ensures that a valid correlation is being detected across the whole design. Due to the use of non-exponential devices that act more as a rectification, the CMOS designs explored within this work more closely resemble the MP function.

It is important to note that while ideally the threshold voltage could decrease when using MOSFETs it could come with some tradeoffs. One such important tradeoff is a possible reduction in operating range of the design. This is to be expected, especially when decreasing in process node size since the overall supply voltage must also be scaled down for all components from its size reduction. This means that while we may have a lower overall operating range, it will scale accordingly with the design node since any peripheral components which

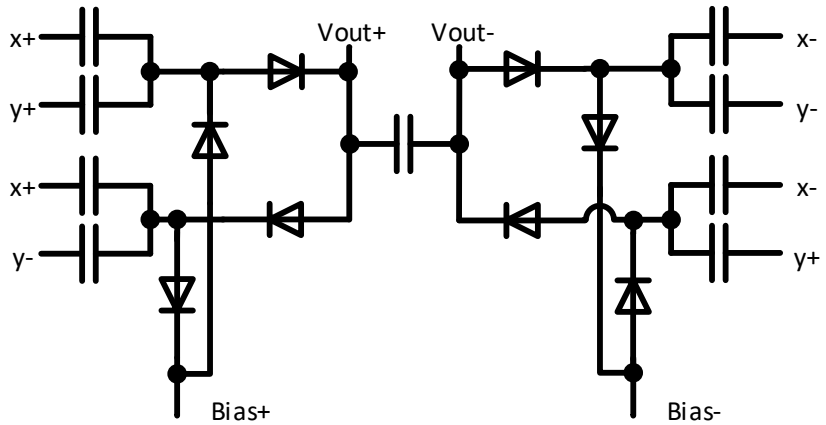

Figure 3.3: Single-Stage Differential Diode Correlator

read this output from our design must also have reduced input voltages. The main downside of this tradeoff is for extreme amplification ranges to power external peripherals.

3.3 Differential Design

In order to increase the Common Mode Rejection Ratio of the correlator to assist with noise reduction, a differential design was then explored as an alternative design. This design simply involves switching a few of the inputs to where each side has a matching input polarity along with a mixed-polarity path. This ensures that both correlation and anti-correlation can be calculated on both the left and right-ends of the design. Now instead of taking the sum of the outputs, we will take the difference. Fig. 3.3 displays a single-stage of the designed differential correlator.

Another important change in this design is that both sides of the circuit will be active as long as some correlation is present within the inputs. This is different from the common-mode design which is dependent on if the input signals have a higher correlation or anti-correlation present to determine the operating side of the design. This means that some current will flow between stages regardless and could potentially benefit the design when

considering the resonance of the circuit if, for example, and oscillator is connected to the bias pins so that a constant signal is correlating through the designs (as long as a correlation is present). Another feature of this design is that the pins on each side of the design could be decoupled from the opposite side, this means that we could essentially have 4 differential input pairs instead of 2 to calculate 2 sets of 2 differential inputs which could then be mixed by propagating through the structure.

3.4 Differential CMOS Design

Similar to the CMOS Common-Mode Design, a CMOS alternative design was created to address issues with high input voltages needed for the correlator to function. This is especially important for the differential correlator since the common-mode configuration performs calculation of correlation on one side for both paths if a correlation is present, creating a higher overall voltage before reaching the output. Since the correlation paths are split for the differential design, the voltages before reaching the output of each stage will be smaller and could run into issues of propagating through the MOSFETs which connect the bias of each stage. The bulks of the transistors are similarly connected to an external DC bias similar to the common-mode CMOS design. Fig. 3.4 displays a single-stage of the CMOS differential correlator.

Again we expect to see a lower operating range and each side of the design will have a lower magnitude than the common-mode design due to the correlated results being split between the left and right sides of the design. So overall we may see a lower differential output voltage with this design.

3.5 Cascading of Correlator Stages

Each of these designs can easily be correlated into an array structure. Using this type of structure can allow for two different over-arching modes of operation: one where the structure acts as a voltage multiplier and correlator, actively boosting the output voltage between stages or can be used with different inputs per-stage to calculate multiple different correlation functions and can be propagated to find a net correlation between multiple inputs.

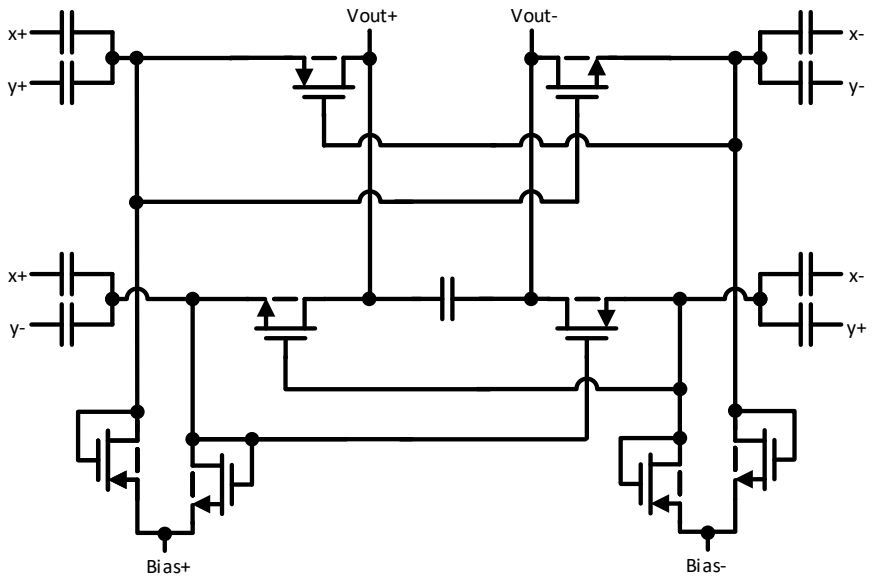

Figure 3.4: Single-Stage Differential CMOS Correlator

As previously mentioned, the output of each stage can be connected to the bias of the next stage to create an array. Fig. 3.5 shows an example of how multiple stages of the proposed correlator circuit can be connected.

In general within this work, this type of cascoding can be used for all multi-stage designs. Fig. 3.6 displays the general form of stage cascading that will be used.

An issue which could arise from connecting a large amount of stages is delay from beginning to end of correlation result propagation or potentially exceeding safe operating voltages of components used within the design. These sort of considerations are important to make when creating arrays of the structure.

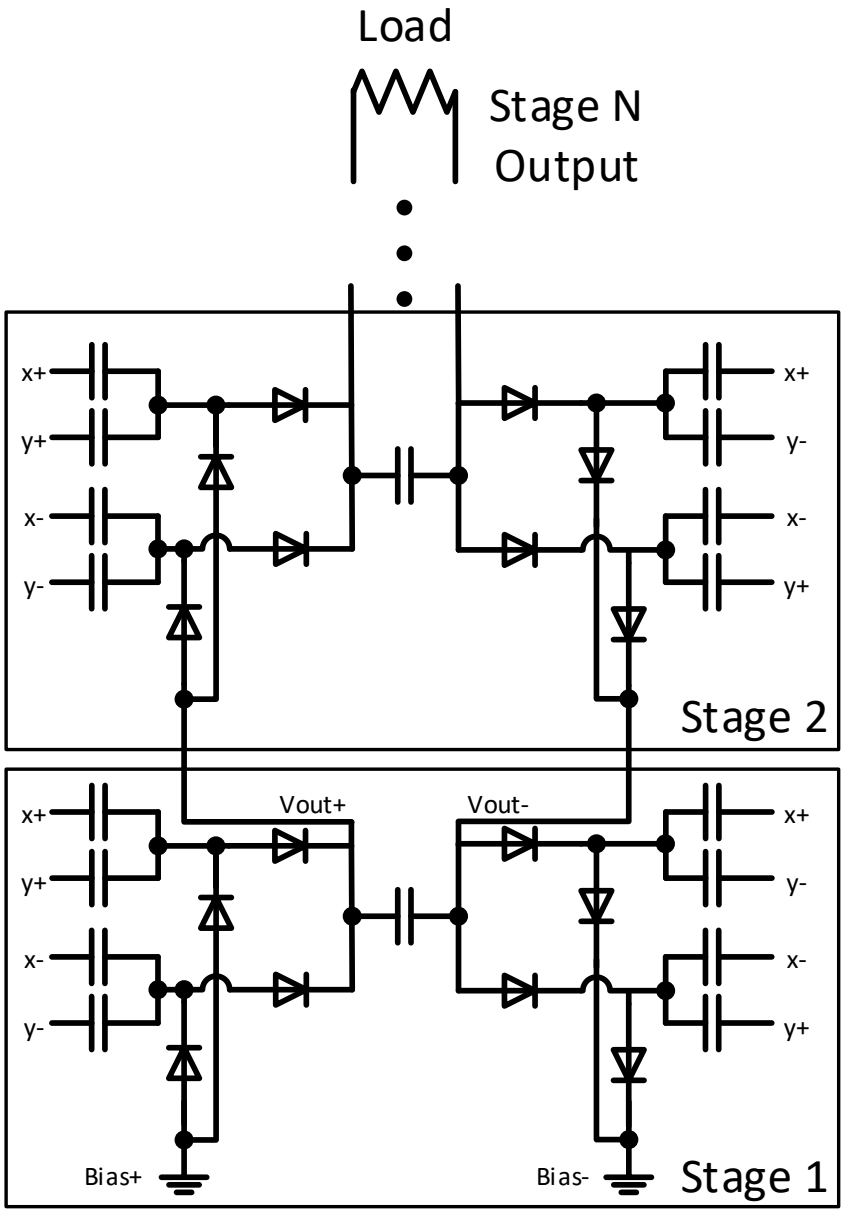

Figure 3.5: Example of a Two-Stage Correlator

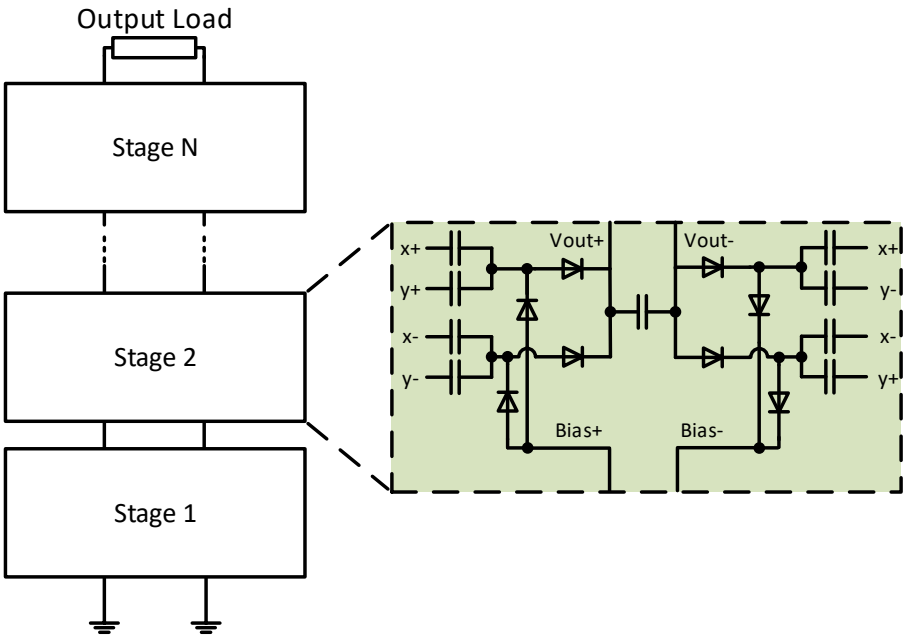

Figure 3.6: Example of a Correlator Array

Chapter 4

Analysis of Proposed Designs

The proposed designs were then analyzed to estimate their output waveforms along with their potential performance. It is difficult to reach closed-form solutions of the designs due to the non-linear nature of the design of the circuit. Any estimation of results will be performed computationally to provide a model of design performance and then simulated to verify the accuracy of the derived equations. As stated previously, we should expect the diode correlator variation to be more similar to the LSE function due to the use of exponential diodes and the CMOS design to be closer to the MP function since it is essentially a rectification.

4.1 Common-Mode Diode Design

The common-mode diode design involves a simple connection of diodes and capacitors which are cascaded between stages with the outputs of the previous stage connected to the bias pins of the subsequent stage. For the analysis of this design we will consider a single stage for simplicity. It is important to note that the cascade between stages will act as a "bucket" where the rectified output of the V_{out+} pin will move to the next stage and the inverse of the rectified V_{out-} will be moved to the next stage (current will flow into the last stage or the bias connection on the first stage). Fig. 4.1 displays the schematic of the circuit which will be analyzed, where the bias pins are connected to ground and the V_{out} pins will be an open-load.

To begin our analysis we will start by performing Kirchoff's Current Law (KCL) for the left-side of the circuit to measure the current flowing into the positive output terminal. For this derivation we will assume that this operation is occurring where the left-ended diodes are forward biased and current is flowing towards the positive output terminal (positive correlation) which is an open load. Since the node voltage of capacitor junction (V_a) is positive

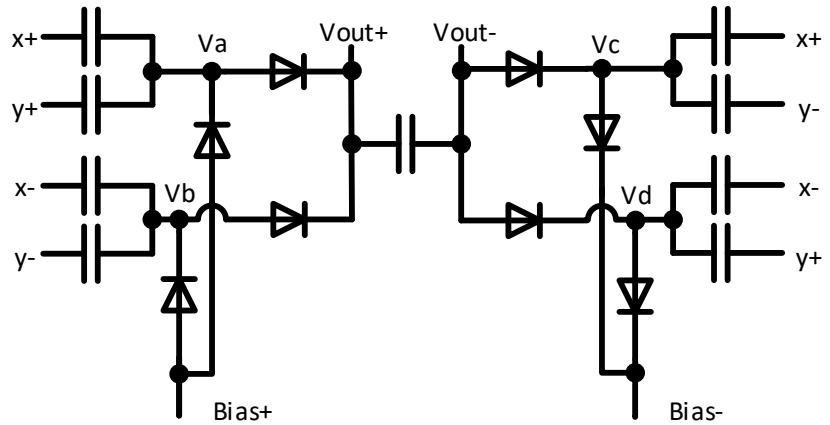

Figure 4.1: Single-Stage Diode Common Mode Correlator

to forward bias the diodes to the positive terminal, we can ignore the diodes connecting $Bias_+$ to node voltages V_a and V_b since these currents will only occur when the diodes are forward biased. The resulting equation becomes:

$$I_s(exp(a(V_a - V_{out+})) - 1) + I_s(exp(a(V_b - V_{out+})) - 1) = I_z \tag{4.1}$$

Where I_s is the leakage current of the diode, a is equivalent to the inverse of the diode thermal voltage, and I_z is the current through the center capacitor. By performing algebraic rearrangement, the following equation can result as a general formula for current through the center capacitor for the left end of the circuit.

$$I_s(exp(a(V_a - V_{out+})) + exp(a(V_b - V_{out+})) - 2) = I_z \tag{4.2}$$

A similar equation arises for the right-end of the circuit, however for this end we will assume that the diodes connecting to the negative output terminal will only conduct whenever the node voltages V_c and V_d are at a lower potential than the negative output terminal. This

means that this end of the circuit will only conduct when an anti-correlation is present. A very similar current equation for the capacitor current arises from these branches.

$$I_s(exp(a(V_{out-} - V_c)) + exp(a(V_{out-} - V_d)) - 2) = I_z \tag{4.3}$$

We can equate these terms of capacitor current and solve for the output node voltages to find a general equation of a single stage of the design. The following steps are taken to solve for these node voltages.

$$I_s(exp(a(V_a - V_{out+})) + exp(a(V_b - V_{out+})) - 2) = I_s(exp(a(V_{out-} - V_c)) + exp(a(V_{out-} - V_d)) - 2) \tag{4.4}$$

$$exp(a(V_a - V_{out+})) + exp(a(V_b - V_{out+})) - 2 = exp(a(V_{out-} - V_c)) + exp(a(V_{out-} - V_d)) - 2 \tag{4.5}$$

$$exp(a(V_a - V_{out+})) + exp(a(V_b - V_{out+})) = exp(a(V_{out-} - V_c)) + exp(a(V_{out-} - V_d)) \tag{4.6}$$

$$a(V_a - V_{out+}) + a(V_b - V_{out+}) = a(V_{out-} - V_c) + a(V_{out-} - V_d) \tag{4.7}$$

$$V_a + V_b - 2V_{out+} = 2V_{out-} - V_c - V_d \tag{4.8}$$

$$\frac{1}{2}(V_a + V_b + V_c + V_d) = V_{out+} + V_{out-} \tag{4.9}$$

While this is solved using explicit node voltages, this can provide a general form solution of the design. This formula also shows that we need to find the common mode of the design to

fully calculate the correlation. This general form can apply for all stages since the cascading of stages through bias connections causes for the values of the branch node voltages to be boosted. Another important aspect to note is since all of these nodes are conditionally activated for the diodes to turn-on, the node voltages are very important to maintain as roughly equal. This means that the best balanced will result exactly where $V_{x+} = -V_{x-}$ and $V_{y+} = -V_{y-}$ or the inputs are equally balanced.

We can further derive aspects of this non-linear system with respect to the node voltages of the system. For a stage without boosting (such as if the first stage is connected to a grounded bias) KCL at the input nodes can be expressed as.

$$C\frac{d}{dt}(V_{x+} - V_a) + C\frac{d}{dt}(V_{y+} - V_a) = I_s(exp(a(V_a - V_{out+}) - 1) \tag{4.10}$$

$$C\frac{d}{dt}(V_{x-} - V_b) + C\frac{d}{dt}(V_{y-} - V_b) = I_s(exp(a(V_b - V_{out+}) - 1) \tag{4.11}$$

$$C\frac{d}{dt}(V_{x+} - V_c) + C\frac{d}{dt}(V_{y-} - V_c) = -I_s(exp(a(V_{out-} - V_c) - 1) \tag{4.12}$$

$$C\frac{d}{dt}(V_{x-} - V_d) + C\frac{d}{dt}(V_{y+} - V_d) = -I_s(exp(a(V_{out-} - V_d) - 1) \tag{4.13}$$

This shows that the diode currents, node potentials, and the capacitor currents will change dynamically with the inputs into the circuit. If we assume that the inputs themselves are periodic then we can assume that the system will be constantly switching between the "on" branches and will also eventually saturate since current can only travel in one direction throughout the design. This means that by providing more stages we can essentially create more paths for current to flow and to build up to a higher potential between stages. The following equations express the current through the branch diodes for the design when a non-grounded bias is applied (such as a cascaded stage) to express the general current through the diodes which flow to the output nodes of each stage.

$$C\frac{d}{dt}(V_{x+}-V_a)+C\frac{d}{dt}(V_{y+}-V_a)+I_s(exp(a(V_{bias+}-V_a)))=I_s(exp(a(V_a-V_{out+})-1) \quad (4.14)$$

$$C\frac{d}{dt}(V_{x-}-V_b)+C\frac{d}{dt}(V_{y-}-V_b)+I_s(exp(a(V_{bias+}-V_b))-1)=I_s(exp(a(V_b-V_{out+})-1) \quad (4.15)$$

$$C\frac{d}{dt}(V_{x+}-V_c)+C\frac{d}{dt}(V_{y-}-V_c)-I_s(exp(a(V_c-V_{bias-}))-1)=-I_s(exp(a(V_{out-}-V_c)-1) \quad (4.16)$$

$$C\frac{d}{dt}(V_{x-}-V_d)+C\frac{d}{dt}(V_{y+}-V_d)-I_s(exp(a(V_d-V_{bias-}))-1)=-I_s(exp(a(V_{out-}-V_d)-1) \quad (4.17)$$

This can be further simplified to express in terms of change in the node voltages along with the current dependent with time.

$$Cd(V_{x-}-V_d)+Cd(V_{y+}-V_d)-I_s(exp(a(V_d-V_{bias-}))-1)dt=-I_s(exp(a(V_{out-}-V_d)-1)dt \quad (4.18)$$

these equations show that additional current will flow through through the bias points when the bias potential is greater than the node voltage at each branch for the left-side nodes. This means that we should overall receive an amplification between stages since the current is continuing to build between the stages. These equations also show that for the right-side nodes that the current will flow to a previous branch when the node voltages are greater than the bias potential. This will decrease this node voltage as current flows and will bring down the negative output potential between the stages. These results conclude that when a positive correlation is detected, we should receive a more positive output common-mode result and when anti-correlated a more negative output common-mode result will occur. An interesting case is where there is no correlation detected. For this case since the input

Parameter	Value
V_{pp}	2 V
Input Frequency	915 MHz

Table 4.1: Common Mode Diode Simulation Parameters

voltages are out-of-phase across all nodes, these inputs should effectively cancel each other out which ensures that none of the diodes will activate and no output potential will be present.

This equation represents the single stage output of the designed correlator. This equation can be applied to each stage with the overall amplitudes boosted due to the cascading of stages which brings the potential "up". A major issue with this design is that diodes have a quite high forward voltage (around 0.7 Volts) and restricts our operating range of the circuit. This can be easily addressed with a CMOS design alternative which can both increase our operating range and decrease the size of each stage so more stages can be integrated on-chip. Another alternative is to use schottky diodes which while they decrease the forward-voltage needed to switch the diode, are typically larger than MOS devices and have greater leakage current which is not ideal for accurate correlation calculations especially if minimum power consumption is desired for no correlation or anti-correlation for specific branches.

Next the designed circuit will be simulated within Cadence Virtuoso using the Spectre simulator. This design will be simulated using the Skywater 130nm Process Development Kit (PDK). For the initial simulation, the inputs to the design will be sinusoidal with a phase swept input for one of the input vectors while the other will remain constant. Since the correlation is the desired metric for this design, we expect to see a monotonically increasing output waveform that appears to be linear when being swept from -180 degrees to 0 degrees. This setup was simulated using two sinusoidal sources with 2V peak-to-peak sinusoudal inputs. Fig. 4.2 displays the recorded output voltage of the simulation which displays the positive and negative output terminal results and its common mode. Fig. 4.3 also shows the recorded input power into the design which can be used to display the power consumed by the system at different correlation values. For the ran simulation table **??** displays the input source parameters.

We can see that the output waveform is not entirely linear within the simulation and seems to begin to saturate at the correlation extremes, but from the range of approximately -160 to -20 degrees a roughly linear common-mode result is found and is also monotonic. This shows

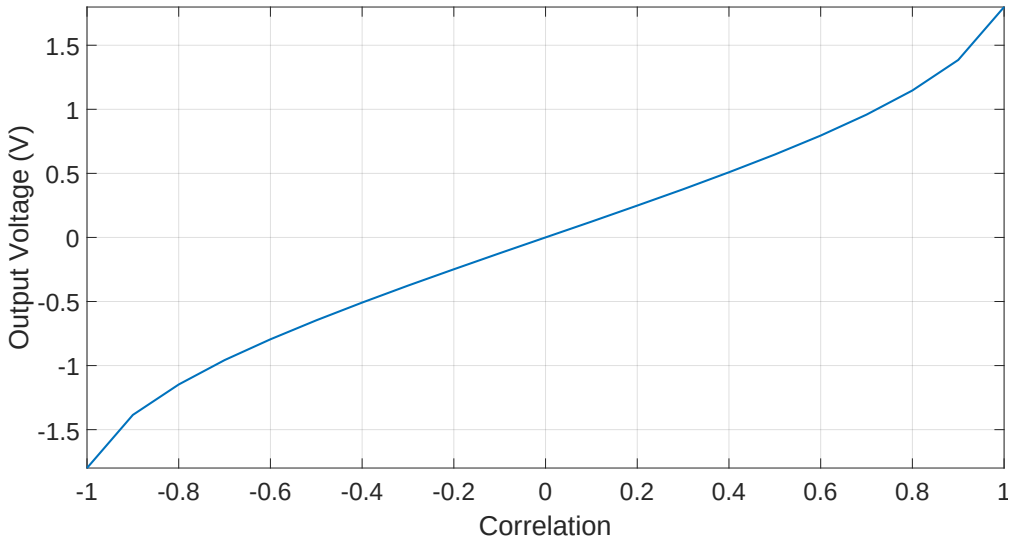

Figure 4.2: Output Voltage Simulation of Common-Mode Design within Cadence Virtuoso with a sinusoidal sweep

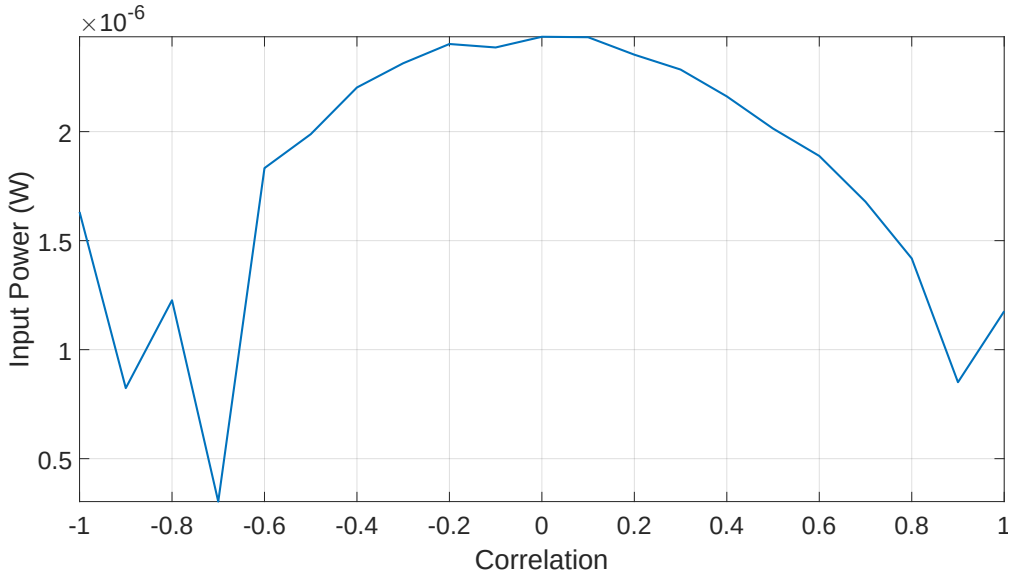

Figure 4.3: Input Power Simulation of Common-Mode Design within Cadence Virtuoso with a sinusoidal sweep

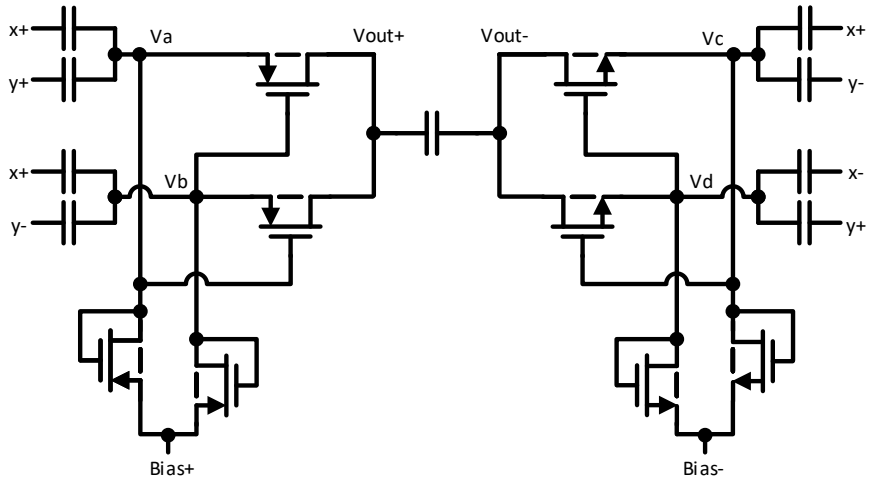

Figure 4.4: Single-Stage Common-Mode CMOS Correlator

that we can in fact calculate the correlation even with a single stage using the common-mode design. However, the input power consumed seems to provide a different result. We can see for this case that with a load attached we actually consume the most input power when the circuit is not correlated which is not ideal behavior.

4.2 Common-Mode CMOS Design

Next the Common-Mode CMOS Design will be analyzed. This structure essentially has the same type of configuration as the diode common-mode configuration but uses MOSFET-configured diodes with the added variation of using the voltage from the across nodes for it's gate voltage. Utilizing MOSFETs can allow for a greater amount of control over threshold voltages of the switches to allow for smaller input voltage ranges to be used. We will look at the left (positive correlation) side first. Fig. 4.4 can be used as a reference for node labels for the differential diode correlator.

First we can estimate the operating region of the MOSFETs connected to the positive output. For the device connected from node A to the positive output terminal

$$V_a - V_{out+} \leq V_a - V_b - |V_{th}| \tag{4.19}$$

This can be simplified to:

$$V_{out+} \geq V_b + |V_{th}| \tag{4.20}$$

This condition will be true since if V_b is considered to be equivalent to V_a then for the transistor to be on, V_b must have a negative potential. This means that we can consider that while on, the transistor should be operating within the linear region. We can then write the current through this node as

$$I_a = k_p^{'} \frac{W}{L} \left((V_a - V_b - |V_{th}|)(V_a - V_{out+} - \frac{1}{2}(V_a - V_{out+})^2 \right) \tag{4.21}$$

Which can be simplified to

$$I_a = k_p^{'} \frac{W}{L} \left(\frac{1}{2} V_a^2 - \frac{1}{2} V_{out+}^2 - V_a V_b + V_b V_{out+} - V_a |V_{th}| + V_{out+} |V_{th}| \right) \tag{4.22}$$

The same logic can be applied to node B. The current through the PMOS connected between node B to the positive output terminal can also be expressed as.

$$k_p^{'} \frac{W}{L} \left(\frac{1}{2} V_b^2 - \frac{1}{2} V_{out+}^2 - V_a V_b + V_a V_{out+} - V_b |V_{th}| + V_{out+} |V_{th}| \right) \tag{4.23}$$

Using KCL we can see that the current through the center capacitor with respect to the positive output terminal node can be expressed as.

$$I_z = \frac{k_p^{'} W}{L} \left(3V_a^2 - V_{out+}^2 + 2V_{out+} |V_{th}| \right) \tag{4.24}$$

By using the current equation through the center capacitor an open-form solution can be found which can be expressed as

$$dV_z = \frac{k_p^{'} W}{C_c L} \left(3V_a^2 - V_{out+}^2 + 2V_{out+}|V_{th}|\right) dt \tag{4.25}$$

Where V_z is equivalent to $V_{out+} + V_{out-} = V_{out}$ and we can assume that $V_{out+} = V_{out-}$ then $V_{out+} = \frac{1}{2}V_{out}$. We can then express the output voltage as.

$$dV_{out} = \frac{k_p^{'} W}{C_c L} \left(3V_a^2 - \frac{1}{4}V_{out}^2 + V_{out}|V_{th}|\right) dt \tag{4.26}$$

Then for the right-side (negative correlation) the operating region of the NMOS can be expressed as.

$$V_{out-} - V_c \leq V_d - V_c - V_{th} \tag{4.27}$$

Which is true since for this case for the NMOS to be on, V_d should be positive and will drive the output voltage to an even lower (more negative) potential. Thus the NMOS transistors will also be in linear. The currents through these transistors can be expressed after simplification as.

$$I_c = k_n^{'} \frac{W}{L} \left(\frac{1}{2}V_c^2 - \frac{1}{2}V_{out-}^2 + V_d V_{out-} - V_c V_d - V_{out-} V_{th} + V_c V_{th}\right) \tag{4.28}$$

$$I_d = k_n^{'} \frac{W}{L} \left(\frac{1}{2}V_d^2 - \frac{1}{2}V_{out-}^2 + V_c V_{out-} - V_c V_d - V_{out-} V_{th} + V_d V_{th}\right) \tag{4.29}$$

These can be summed together to also express the current through the center capacitor.

$$I_z = k_n^{'} \frac{W}{L} \left(\frac{1}{2}(V_c^2 + V_d^2) - V_{out-}^2 + V_{out-}(V_c + V_d) - 2V_c V_d + V_{th}(V_b + V_c - 2V_{out-})\right) \tag{4.30}$$

If we assume that $V_c = -V_d$.

Parameter	Value
V_{pp}	1 V
Input Frequency	2 GHz

Table 4.2: Common Mode CMOS Simulation Parameters

$$I_z = k_n^{'} \frac{W}{L} \left(3V_c^2 - V_{out-}^2 - 2V_{out-}V_{th}\right) \tag{4.31}$$

We can then use the capacitor current equation to express an open-form solution for the output voltage.

$$dV_{out} = \frac{k_n^{'} W}{C_c L} \left(3V_c^2 - V_{out-}^2 - 2V_{out-}V_{th}\right) dt \tag{4.32}$$

Which can be used to represent the output voltage in terms of the right-side of the circuit. All of these derived equations do not include a solution for the node voltage since they will change between stages if cascoded. This means that the derived equations should be general across all configurations. It is also important to remember that these equations will only be true if the transistors are on. It is important to note that the effects of transistor biasing is not included in the derived equations.

Next this design will be simulated in Cadence Virtuoso to show the operation of the design. Due to CMOS devices typically having a lower threshold voltage with the cost of a lower operating range than diodes, the CMOS devices will be utilized at a lower peak-to-peak voltage. For this simulation the following parameters will be used and can be read in table 4.2.

These input parameters will be simulated with a correlation sweep which controls the phase of one of the sinusoudal inputs. Fig. 4.5 displays the average output voltage per correlation and 4.6 displays the input power per correlation for the designed correlator.

With this design we can see that the output voltage is definitely quite high and roughly linear, however, the input power seems to actually be at a maximum when no correlation is present and roughly minimum when correlated or anti-correlated. This isn't the ideal result since we would ideally want the most power to be consumed when correlated and at

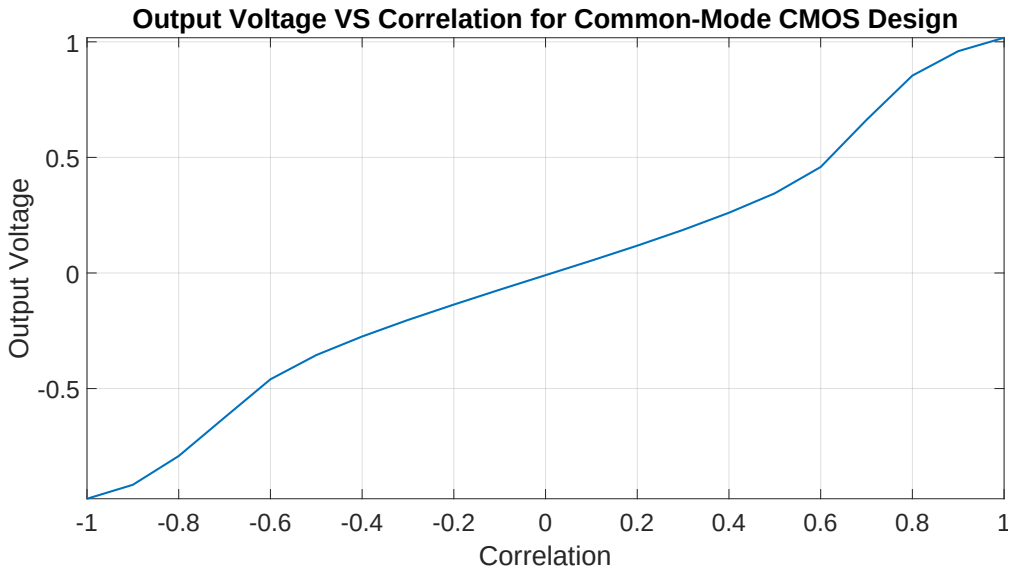

Figure 4.5: Simulated Output Voltage of Common Mode CMOS Design

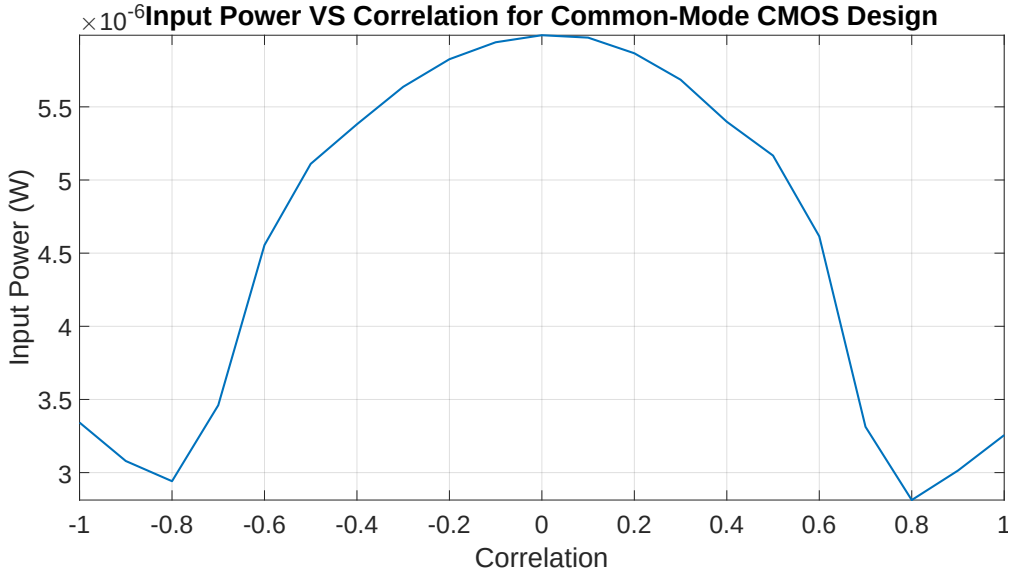

Figure 4.6: Simulated Input Power of Common Mode CMOS Design

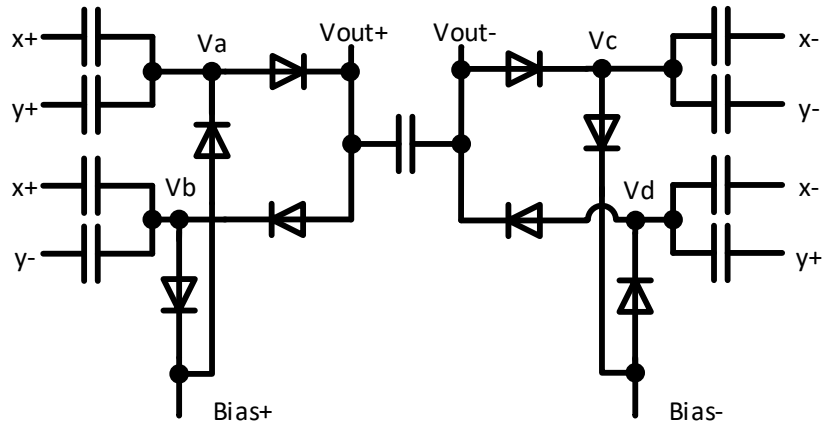

Figure 4.7: Single-Stage Differential Diode Correlator

a minimum when no correlation is present. To do this we will explore designs which are modified version of the initial common mode design to address these issues.

4.3 Differential Diode Design

An additional variation of this design was also explored to address a few potential shortcomings of the original design. One of these issues is to remove the need of an adder to compute the common-mode result of the circuit. Removing the need for this adder can save area on-chip and allow for simpler computation since the output voltage will be with respect to the load. Another additional benefit could be that differential results typically do not result in as much noise due to the subtraction instead of addition of the output terminals which cancels instead of amplifying the noise present. Fig. 4.7 can be used as a reference for node labels for the differential diode correlator.

Similar to the common-mode design, the following equations can be derived by using KCL with respect to the center capacitor current.

$$I_s(exp(a(V_a - V_{out+})) - exp(a(V_{out+} - V_b))) = I_z \tag{4.33}$$

$$I_s(exp(a(V_{out-} - V_c)) - exp(a(V_d - V_{out-}))) = I_z \tag{4.34}$$

By equating we can solve for an approximate solution for the output voltage

$$I_s(exp(a(V_a - V_{out+})) - exp(a(V_{out+} - V_b))) = I_s(exp(a(V_{out-} - V_c)) - exp(a(V_d - V_{out-}))) \tag{4.35}$$

$$exp(a(V_a - V_{out+})) - exp(a(V_{out+} - V_b)) = exp(a(V_{out-} - V_c)) - exp(a(V_d - V_{out-})) \tag{4.36}$$

$$exp(a(V_a - V_{out+})) + exp(a(V_d - V_{out-})) = exp(a(V_{out-} - V_c)) + exp(a(V_{out+} - V_b)) \tag{4.37}$$

$$a(V_a - V_{out+}) + a(V_d - V_{out-}) = a(V_{out-} - V_c) + a(V_{out+} - V_b) \tag{4.38}$$

$$V_a - V_{out+} + V_d - V_{out-} = V_{out-} - V_c + V_{out+} - V_b \tag{4.39}$$

$$\frac{1}{2}(V_a + V_b + V_c + V_d) = V_{out+} + V_{out-} \tag{4.40}$$

Which at first appears to actually be the same result as the common mode design. However, if we include the bias connections for the node voltages of V_a, V_b, V_c, and V_d. For example the left end nodes (V_a and V_b) will have the following current equations following KCL with a bias input.

$$C(\frac{d}{dt}(V_{x+}-V_a)+\frac{d}{dt}(V_{y+}-V_a))+I_s(exp(a(V_{bias+}-V_a))-1) = I_s(exp(a(V_a-V_{out+}))-1) \quad (4.41)$$

$$C(\frac{d}{dt}(V_{x+}-V_b)+\frac{d}{dt}(V_{y-}-V_b))-I_s(exp(a(V_b-V_{bias+}))-1) = -I_s(exp(a(V_{out+}-V_b))-1) \quad (4.42)$$

If we then substitute these values into our original center capacitor current equation we have the following result.

$$\begin{aligned} C(\frac{d}{dt}(V_{x+}-V_a)+\frac{d}{dt}(V_{y+}-V_a)+\frac{d}{dt}(V_{x+}-V_b)+\frac{d}{dt}(V_{y-}-V_b)) \\ +I_s(exp(a(V_{bias+}-V_a))-1-exp(a(V_b-V_{bias+})+1)) = I_z \end{aligned} \quad (4.43)$$

Which can then be simplified to

$$\begin{aligned} C(d(V_{x+}-V_a)+d(V_{y+}-V_a)+d(V_{x+}-V_b)+d(V_{y-}-V_b)) \\ +I_s(exp(a(V_{bias+}-V_a))-exp(a(V_b-V_{bias+})))dt = I_z dt \end{aligned} \quad (4.44)$$

This derived non-linear equation shows that current will either propagate "up" the chain if the bias of circuit has a positive voltage or "down" the chain if the bias is negative. This also shows that the differential design can have either a positive or negative current unlike the common-mode design. The same is true for the right-side of the circuit which is displayed below.

$$C(\frac{d}{dt}(V_{x-}-V_c)+\frac{d}{dt}(V_{y-}-V_c))-I_s(exp(a(V_c-V_{bias-}))-1) = -I_s(exp(a(V_{out-}-V_c)-1) \quad (4.45)$$

Parameter	Value
V_{pp}	2 V
Input Frequency	915 MHz

Table 4.3: Differential Diode Simulation Parameters

$$C(\frac{d}{dt}(V_{x-}-V_d)+\frac{d}{dt}(V_{y+}-V_d))+I_s(exp(a(V_{bias-}-V_d))-1) = I_s(exp(a(V_d-V_{out-}))-1) \quad (4.46)$$

And substituted into our original equation to find I_z.

$$\begin{aligned} -C(\frac{d}{dt}(V_{x-}-V_c)+\frac{d}{dt}(V_{y-}-V_c)+\frac{d}{dt}(V_{x-}-V_d)+\frac{d}{dt}(V_{x-}-V_d)) \\ +I_s(exp(a(V_{bias-}-V_d))-exp(a(V_c-V_{bias-}))) = I_z \end{aligned} \quad (4.47)$$

Then by rearrangement we can express the current incident from the right-side of the circuit with the by the following equation.

$$\begin{aligned} -C(d(V_{x-}-V_c)+d(V_{y-}-V_c)+d(V_{x-}-V_d)+d(V_{x-}-V_d)) \\ +(I_s(exp(a(V_{bias-}-V_d))-exp(a(V_c-V_{bias-})))\,dt = I_z dt \end{aligned} \quad (4.48)$$

Similar to the left-side the negative bias output voltage can either allow current to propagate "up" or "down" the chain relative to the correlation of the input voltages. This also further shows that a current will flow towards the negative output terminal if the signals are anti-correlated and will discharge the negative output terminal potential if the signal is positively correlated.

After deriving the equation for this correlator variation, the design was then implemented in Cadence and simulated to determine functionality of the correlator. This simulation is run using a 65nm process with input vectors of a 1 Vpp input voltage source ran at 2 GHz a 20 kω output resistor. Fig. 4.8 displays the simulated output voltage. Table 4.3 displays the input source simulation parameters.

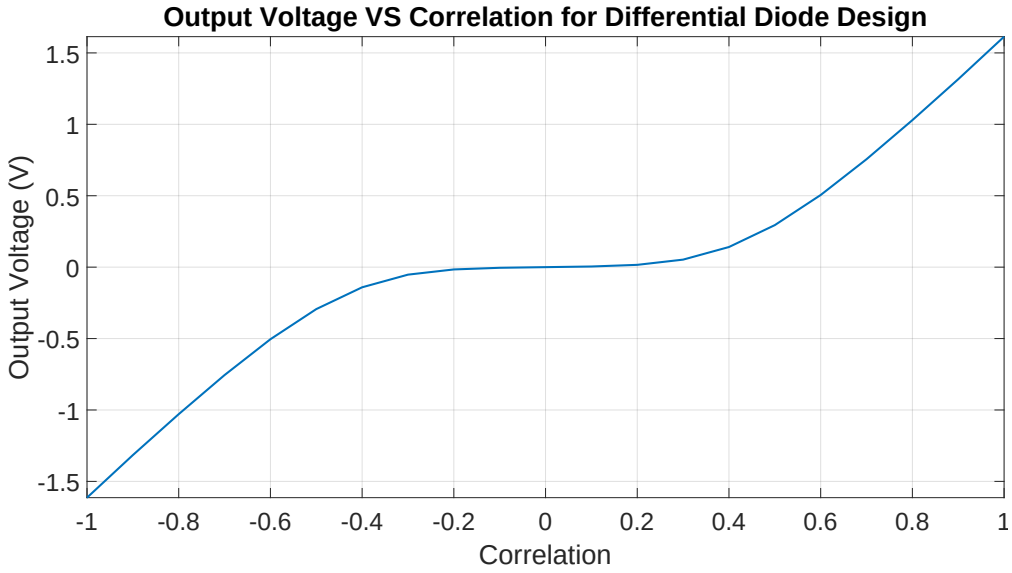

Figure 4.8: Simulated Output Voltage of Differential Diode Design

Next the input voltage of the design is measured to further characterize the operation of this design. Fig. 4.9 displays the input power of the design with the same parameters as the output voltage simulation.

We can see that the ideal situation of the highest amount of power is consumed when there is correlation present in the inputs (either correlated or anti-correlated) and at a minimum when there is zero correlation. This shows the desired measurement characteristics of maximum power consumption upon correlation.

4.4 Differential CMOS Design

Lastly, a CMOS variant of the differential diode design was created to address the threshold voltage issue with the differential design, reduce area on chip, and to add an additional control through the bulk pin for additional control over the threshold voltage of the MOSFETs. First the single stage with both bias pins grounded will be analyzed. For this initial step we will ignore cases where the transistors between the analyzed voltage pins and bias pins are activated since they will be a short circuit. First we will determine the approximate operating region of the MOSFETs. Fig. 4.10 can be used as a reference for node labels for the differential CMOS correlator.

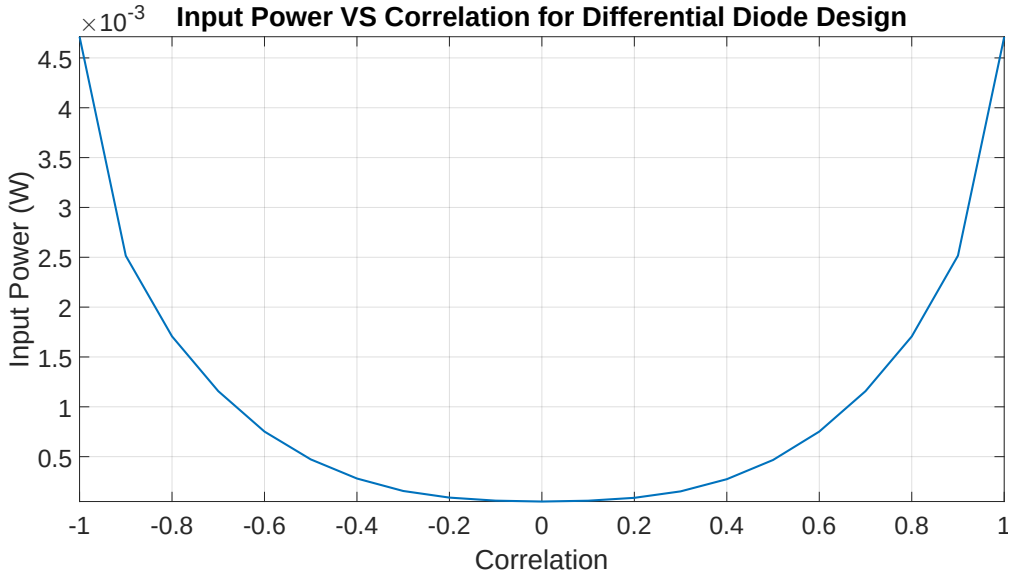

Figure 4.9: Simulated Input Power of Differential Diode Design

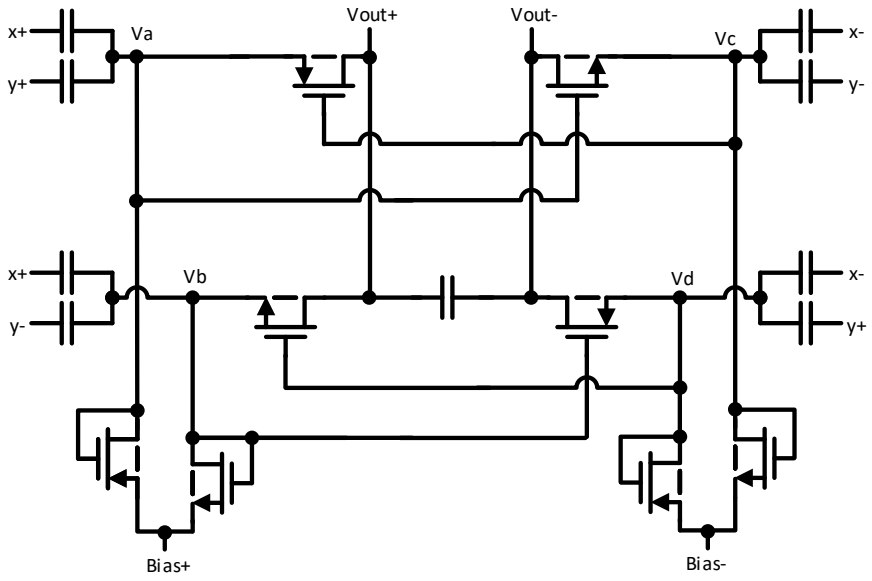

Figure 4.10: Single-Stage Differential CMOS Correlator

For node A we can assume due to symmetry, V_a is equivalent to the negative of V_c since the nodes will have opposite polarity.

$$V_{sg} = V_a - V_c = V_a - -(V_a) = 2V_a \tag{4.49}$$

$$V_{sd} = V_a - V_{out+} \tag{4.50}$$

For a PMOS when assumed on, to determine if operating in the linear region is

$$V_{sd} \leq V_{sg} - |V_{th}|, V_a - V_{out+} \leq 2V_a - |V_{th}| \tag{4.51}$$

$$-V_a \leq V_{out+} - |V_{th}| \tag{4.52}$$

This means that all cases that satisfy this equation (assuming the MOSFET is on). When the transistor is activated in this case the only current conducting path on the side of V_{out+} will be through V_a meaning that the magnitude of this node will always be equal. This means that the transistor will always be in the linear region. This condition will also remain for cascading of stages. We can then use this condition to solve for the current through this transistor.

$$I_a = k_p^{'} \frac{W}{L} \left((2V_a - |V_t h|)(V_a - V_{out+}) - \frac{1}{2}(V_a - V_{out+})^2 \right) \tag{4.53}$$

After simplification the current through this branch can be expressed as.

$$I_a = k_p^{'} \frac{W}{L} \left(\frac{3}{2}V_a^2 - V_a V_{out+} - \frac{1}{2}V_{out+}^2 - V_a|V_{th}| + V_{out+}|V_{th}| \right) \tag{4.54}$$

Similar analysis can be performed for the transistor connected to V_b but instead using an NMOS to find the current through that branch. This branch will also ideally only activate upon detection of anti-correlation and should not conduct at the same time as V_a.

$$I_b = k_n^{'}\frac{W}{L}\left(\frac{3}{2}V_b^2 - V_bV_{out+} - \frac{1}{2}V_{out+}^2 - V_{out+}V_{th} + V_bV_{th}\right) \tag{4.55}$$

Our total current equation for the left side can similarly be expressed as.

$$I_z = I_a - I_b \tag{4.56}$$

And by substituting in previously derived equations can be expressed as (assuming W and L are the same for all PMOS and NMOS devices).

$$I_c = \frac{W}{L}\left(\frac{3}{2}(k_p^{'}V_a^2 - k_n^{'}V_b^2) - V_{out+}(k_p^{'}V_a - k_n^{'}V_b) - V_a|V_{th}|k_p^{'} - V_bV_{th}k_n^{'} + V_{out+}(|V_{th}|k_p^{'} + V_{th}k_n^{'})\right) \tag{4.57}$$

Similar equations can be solved for the right-side of the design which can be expressed as.

$$I_c = k_n^{'}\frac{W}{L}\left(\frac{3}{2}V_a^2 + V_aV_{out-} - \frac{1}{2}V_{out-}^2 - V_aV_{th} - V_{out-}V_{th}\right) \tag{4.58}$$

$$I_d = k_p^{'}\frac{W}{L}\left(\frac{3}{2}V_b^2 + V_bV_{out-} - \frac{1}{2}V_{out-}^2 + V_b|V_{th}| + V_{out-}|V_{th}|\right) \tag{4.59}$$

Since it is assumed that $V_c = -V_a$ and $V_d = -V_b$. Expressed in terms of the center capacitor current.

$$I_z = \frac{W}{L}\left(\frac{3}{2}(k_n^{'}V_a^2 - k_p^{'}V_b^2) + V_{out-}(k_n^{'}V_a - k_p^{'}V_b) - V_aV_{th}k_n^{'} - V_b|V_{th}|k_p^{'} - V_{out-}(V_{th}k_n^{'} + |V_{th}|k_p^{'})\right) \tag{4.60}$$

We can then set the two current equations equal to each other and solve to reach an approximate solution which can be used to estimate the output voltage of the design. After simplification we can reach the following result for the current

Parameter	Value
V_{pp}	1 V
Input Frequency	2 GHz

Table 4.4: Differential CMOS Simulation Parameters

$$\begin{aligned} \frac{3}{2}\left(V_a^2+V_b^2\right)\left(k_p^{'}-k_n^{'}\right)+(V_a-V_b)\left(V_{th}k_n^{'}-|V_{th}|k_p^{'}\right)= \\ V_{out+}\left(k_p^{'}(V_a-|V_{th}|)-k_n^{'}(V_b+V_{th})\right)-V_{out-}\left(k_p^{'}(V_b+|V_{th}|)-k_n^{'}(V_a-V_{th})\right) \end{aligned} \tag{4.61}$$

Due to symmetry we can assume that $V_{out+}=V_{out-}$. By substituting this condition into the above equation we now have the result

$$V_{out+}=\frac{3}{2}(V_a+V_b)-3\frac{V_aV_b}{V_a+V_b}+\frac{(V_a-V_b)(V_{th}k_n^{'}-|V_{th}|k_p^{'})}{(k_p^{'}-k_n^{'})(V_a+V_b)} \tag{4.62}$$

Since the outputs are assumed symmetrical, then the following can also be assumed.

$$V_{out}=V_{out+}-V_{out-}=2V_{out+} \tag{4.63}$$

Assuming this our final equation for output voltage of this design comes out as.

$$V_{out}=3(V_a+V_b)-6\frac{V_aV_b}{V_a+V_b}+2\frac{(V_a-V_b)(V_{th}k_n^{'}-|V_{th}|k_p^{'})}{(k_p^{'}-k_n^{'})(V_a+V_b)} \tag{4.64}$$

The main benefit of this design is that the operating voltage can be significantly lower than those of conventional diode designs (less than approximately 0.7V). Fig. 4.11 shows the output voltage versus correlation for a simulation within Cadence Virtuoso. Table 4.4 displays the input source simulation parameters.

We can see that the design used allows for a very linear and monotonic output voltage value over swept correlation, this is very close to the desired signal shape from calculating

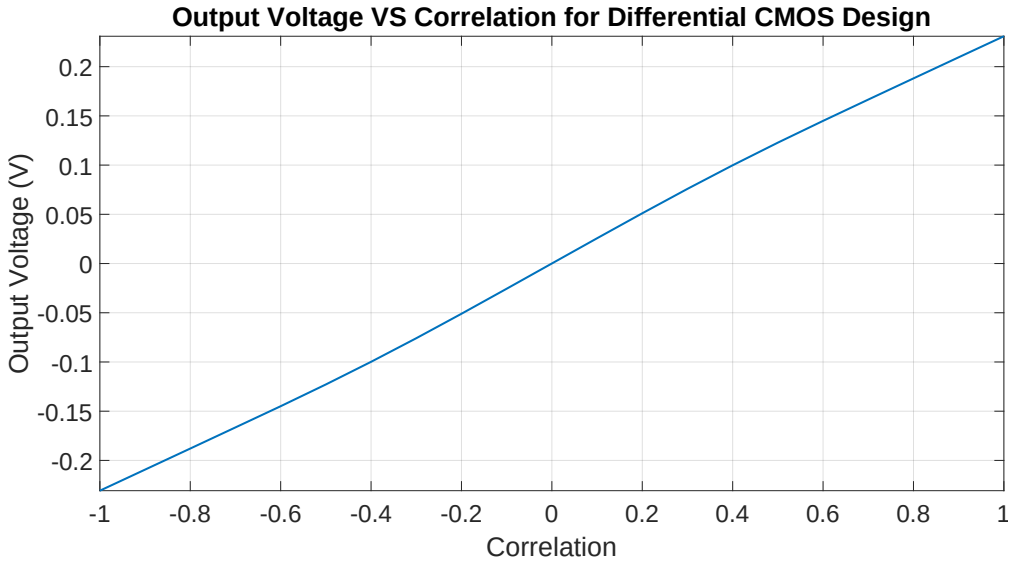

Figure 4.11: Simulated Output Voltage of Differential CMOS Design

correlation. Next the input power is measured for the designed circuit. Fig. 4.12 shows the input power of the simulated design.

Though with this design we can see a power consumption offset, the overall power usage is significantly lower than the diode design (around 0.45% for the peak and 11.9% for minimum power consumption).

Another benefit of using the CMOS design variation is the high control over parameters of the MOSFETs. By changing dimensions of the design we can increase the amount of amplification per stage but this modification will come at the cost of decreasing the linearity of the correlation voltage.

4.5 Multi-Stage Simulations

To further increase the boosting element of the designed correlator we can increase the number of stages connected in series to boost the voltage further. For this case (especially for high amplification) the diode structures may provide the most advantageous to use since typically diodes have higher operating voltage ranges. For an example of an output with multiple stages, the differential diode structure will be used to show a comparison of output

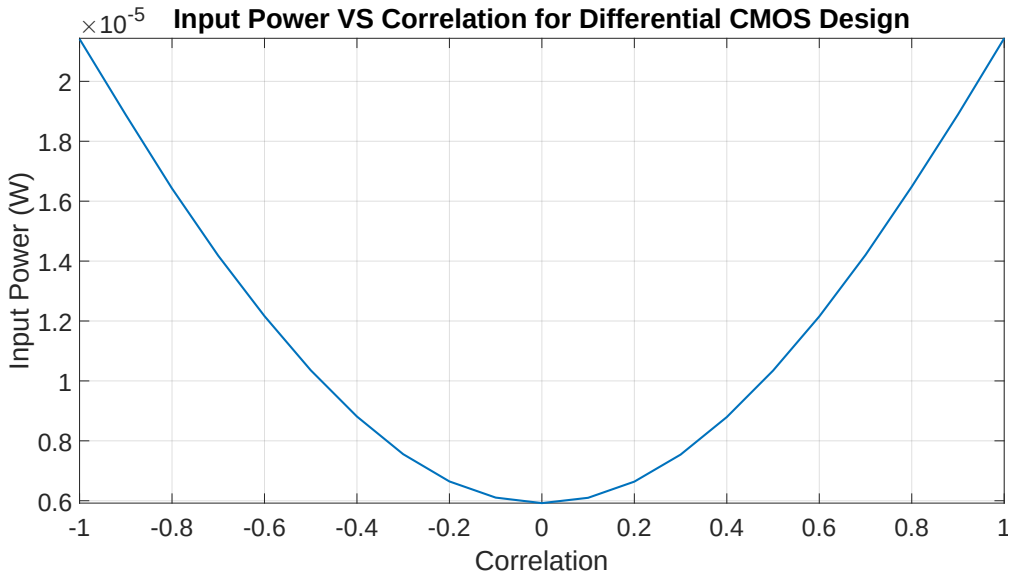

Figure 4.12: Simulated Input Power of Differential CMOS Design

voltage with a different number of stages with correlated input signals to display the voltage amplification. Fig. 4.13 displays the comparison of output voltage for 1 to 4 stages.

From this we can see that the output voltage receives quite a high voltage amplification for even just a few stages connected. We can also see an increase in the convergence time for the number of stages added. This means that adding stages will result in a higher delay before converging to an output value. For this case there is quite a bit of oscillation once converged which can be adjusted based on device characteristics and parameters. Since this will likely be smoothed from any circuits attached to the ASIC's load, we will not worry about adding these within this work.

We can further extend the amount of stages to an arbitrary amount to continue to achieve a higher output voltage range at the cost of increased power consumption. For example, for many of the designs put onto an IC within this work, a common number of 256 stages will be used. For this simulation, it will only be ran for around 5 μs due to the high amount of memory running a 256 stage requires, meaning that there is likely a higher voltage value that the correlator will converge to. Fig. 4.14 displays the output voltage and Fig. 4.15 displays the input voltage of the 256-stage correlator.

This amount of stages shows very similar characteristics to the previously explored differential diode structure but with just a higher power consumption and voltage range.

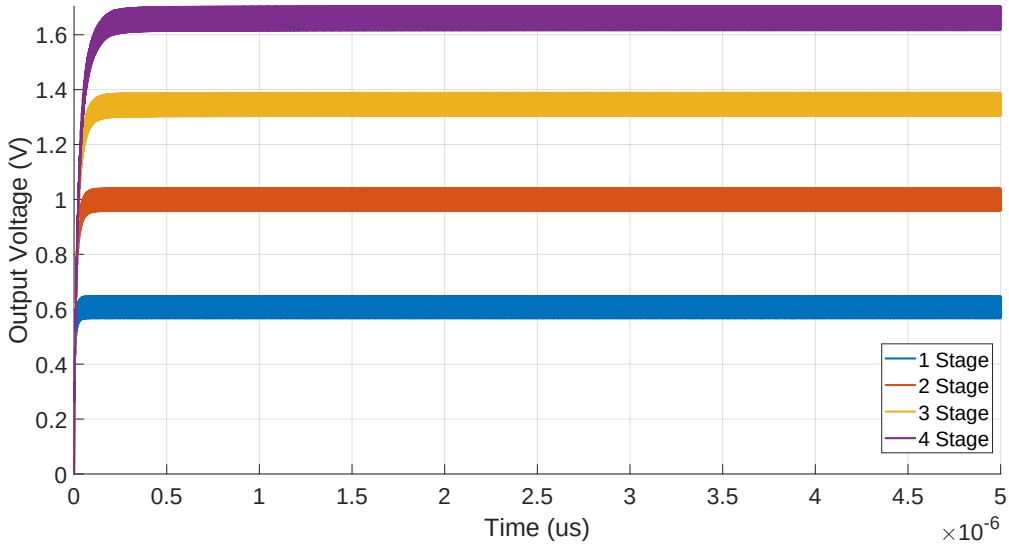

Figure 4.13: Simulated Output Voltage of differential diode correlator with differing number of stages

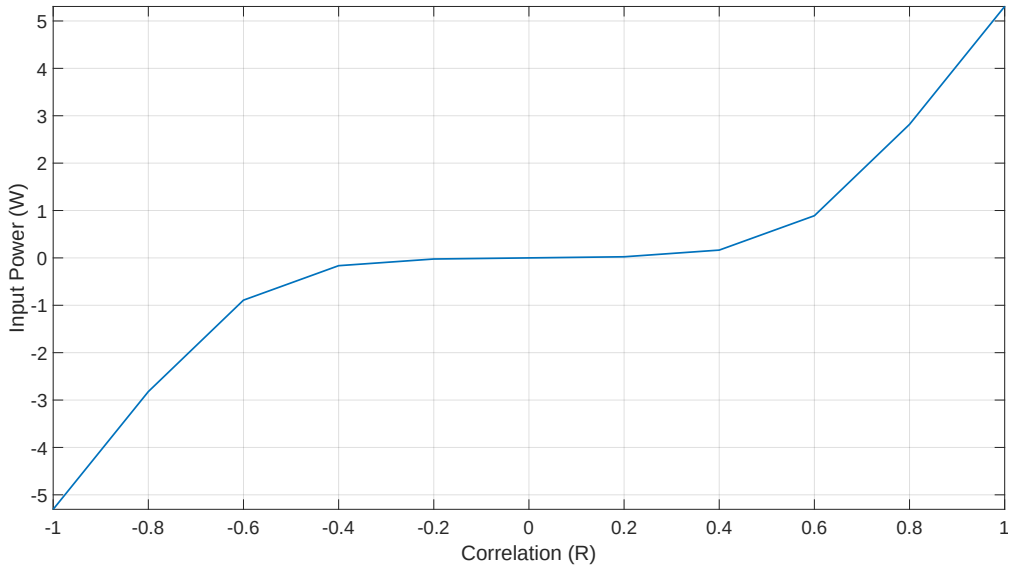

Figure 4.14: Simulated Output Voltage of 256-stage differential diode correlator

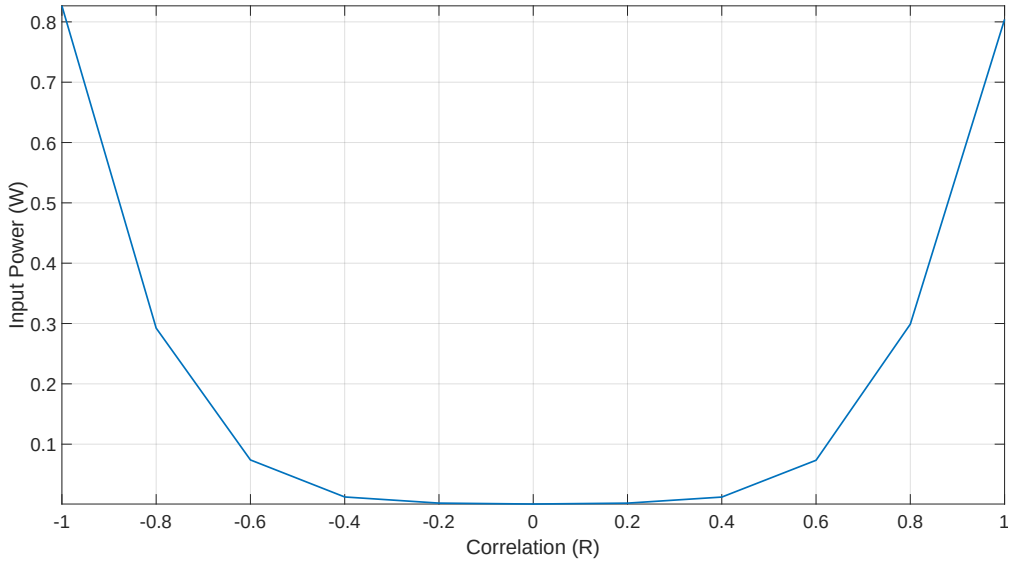

Figure 4.15: Simulated Input Power of 256-stage differential diode correlator

4.6 Simulated Metric Comparison

The performance of the designed correlators were then measured using a suite of different metrics to help characterize the components. These metrics include Hardware Dynamic Range, which is a metric of the feasible operating range of the design, Tera-OPerations per Watt (TOPs/W), which determines the number of trillion operations the design can compute given a watt of power, Bandwidth, which is the operating frequency range of the device, and RF power which is the average RF power consumed. All of these metrics were recorded with added noise Table 4.5 displays the table of the recorded metrics for both the common-mode and differential resonant correlator.

Plots of these data types were also created to show the non-averaged results shown in the above table. Fig. 4.16 displays the HDR of the system over time for both correlators.

Metric	Common Mode Resonant Correlator	Differential Resonant Correlator
Correlator Size	256	256
Hardware Dynamic Range (dB)	57	60
TOPs/W	2.7	10.59
Bandwidth	915M	915M
RF Power	5.5mW	7.5mW
Technology	TSMC65nm	TSMC65nm

Table 4.5: Simulated Metrics for Designed Correlators

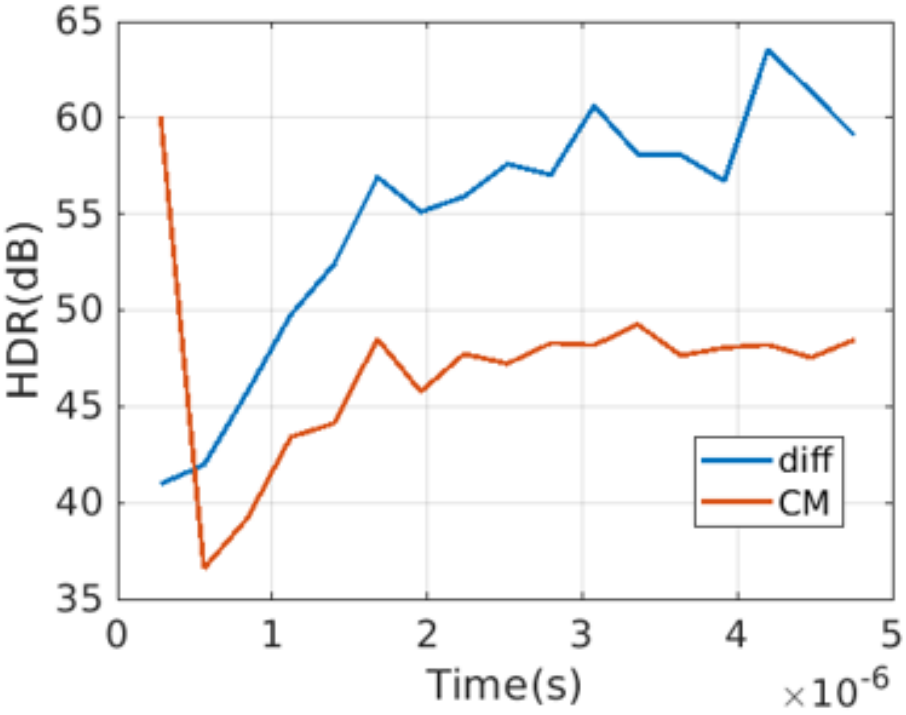

Figure 4.16: Plot of HDR for simulated system

Next the plot for energy per bit was also recorded to show the amount of energy needed to calculate each bit within the correlation calculation. Fig. 4.17 shows the recorded energy per bit.

The Effective Number Of Bits (ENOBs) were also calculated to display the number of bits estimated for the designed hardware to show the level of precision of the analog calculation. Fig. 4.18 displays the ENOBs recorded by the system.

Lastly, the TOPs per Watt are also plotted to show the amount of tera-operations possible for a given watt. Fig. 4.19 displays the recorded data for the TOPs per Watt.

From these simulations we can see that in most cases, the differential correlator seems to have a better long-term result for most of the recorded metrics except for energy-per-bit while for initial states the common-mode design seems to have better results. This means that depending on the operating time desired along with the desired metric to optimize, the baseline configurations of these designs could potentially be interchangeably used. Next a layout of these designs will be created for fabrication on silicon.

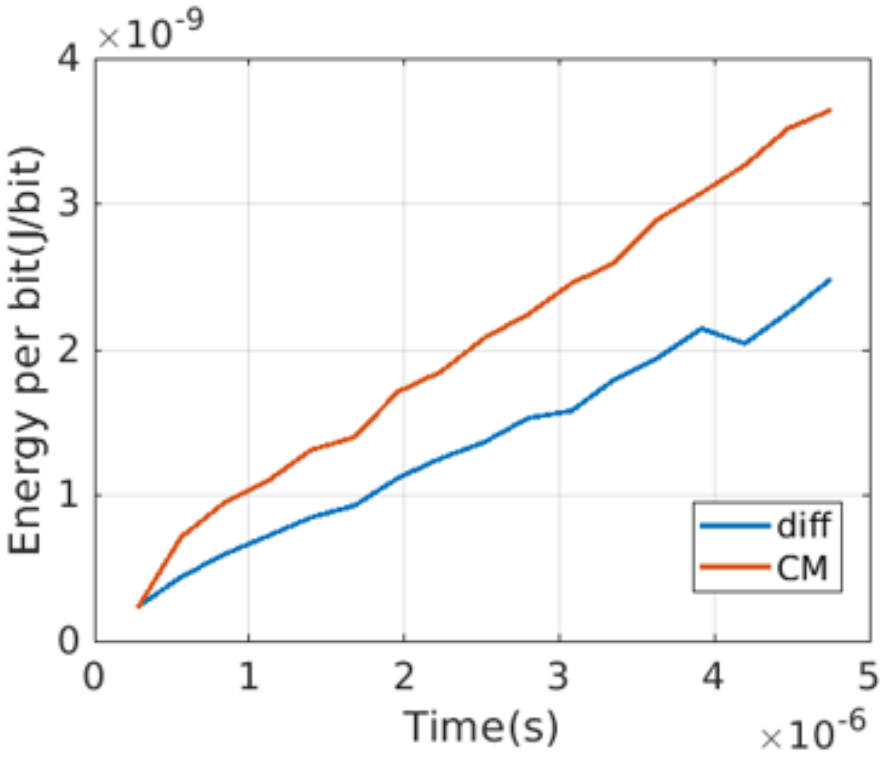

Figure 4.17: Plot of Energy per Bit for simulated system

4.7 Conclusions of Analyzed Designs

Overall it can be seen that the CMOS designs can operate at a lower threshold, have lower power consumption, and can allow for more dimensions of freedom for the designs. One major trade-off for these designs is that the operating range of the designs will be significantly lower due to process limitations. This means that amplification between stages must be relatively controlled to ensure that the output voltages do not greatly exceed the operating capability of the devices. A design consideration for the future could be to use high-voltage processes for any high-power applications. This sort of design constraint should not matter for RFID applications since they relatively operate with lower voltages (typically 3-4 V). This means that for deployment of this ASIC it is important to know the transmitted potential of the signal to adjust the number of stages and any additional peripherals for voltage regulation for interfacing with these devices.

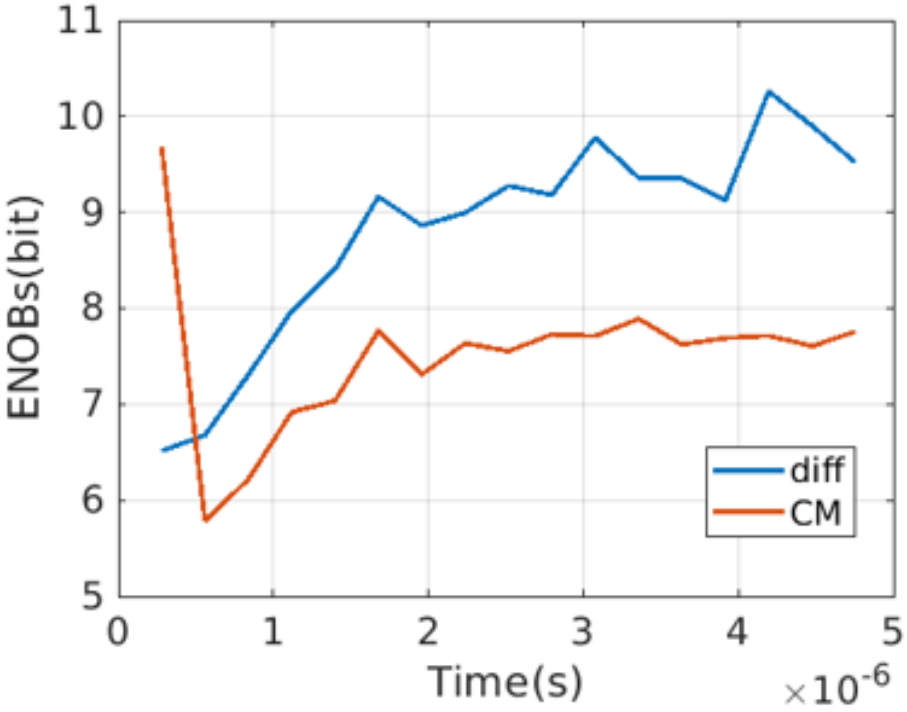

Figure 4.18: Plot of ENOBs for simulated system

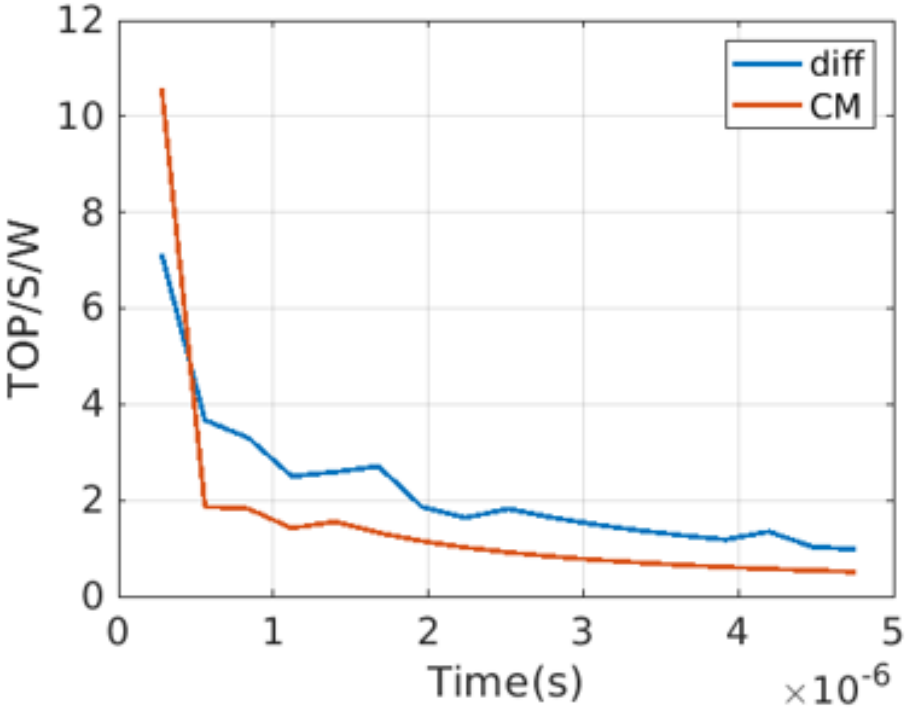

Figure 4.19: Plot of TOPs per Watt for simulated system

Chapter 5

Layout of Designed Correlators

Multiple variations of the designed correlators were implemented on-chip within multiple processes to determine their functionality and performance on hardware. These processes include 130mn, 65nm, and 22nm which each have different versions of the design created. All of these different designs had their own challenges and unique design processes.

5.1 Common-Mode Diode Design

The first design implemented on chip was using the Skywater 130nm processes node. Due to the large area size allocated for the node multiple test structures were implemented onto the chip to determine functionality. Fig. 5.1 displays the created layout for the correlator test-structures within the Skywater 130nm Process. The structures included are a single-stage test structure, 256-stage test structure, and a chip fill structure of around 400 stages which contain larger capacitors to increase charge storage for larger amplification but has a larger delay per stage.

One issue encountered during the design process was having a too-high metal density of one of the process-required Metal-Insulator-Metal Capacitor (MIM Cap) layers. This means to successfully submit the design we needed to remove some instances of the correlator to ensure the metal limit was not exceeded. This severely limited the amount of possible stages within the design since the process layers used for the MIM Caps utilizes metal layers required to pass the density checks, but the functionality of the included stages could still be tested. Another issue with the process used is that it is not designed for higher frequency applications due to its large design process (relative to modern closed-source processes used). This means that designs within this process are frequency limited but is sufficient for our application (915 MHz) with the use of custom Printed Circuit Boards (PCBs).

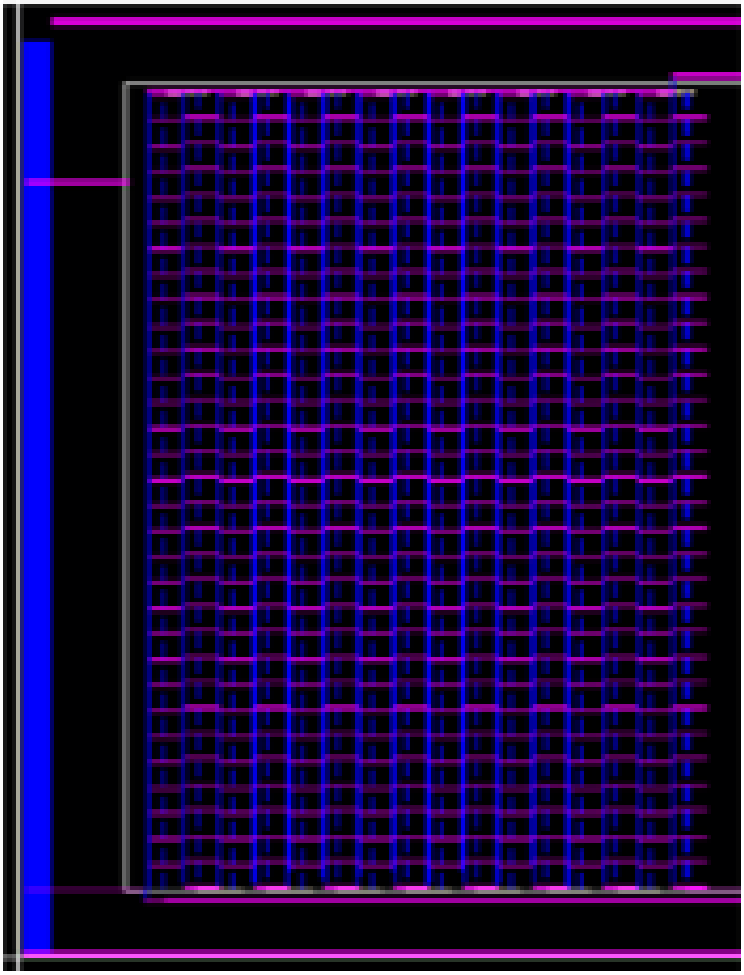

Figure 5.1: Layout of Common-Mode Correlator Structure in Skywater 130nm

This design was created utilizing the Efabless design platform which allows for both closed and open-source designs to be taped out using the Skywater 130nm Process and also comes with an included testing board. This design utilizes a "harness" which contains a subsection for the user-design (our designs in this case) and includes peripheral circuits such as an embedded RISC-V Processor and management circuits. This design also includes configurable General Purpose Input Output (GPIO) ports which allows for different analog and digital port configurations so many of the terminals are multi-use. In our case we only use analog ports meaning that we could not use the full functionality of the GPIOs implemented on board but must be accounted for when designing the circuit layouts. However, an interesting exercise taken during this design process is using Cadence Virtuoso to implement the layouts. There is not much documentation included with the PDK on how to utilize the Skywater 130nm PDK within Cadence and how to export and upload this design into the Efabless harness for generation of the entire silicon design. This process essentially bypassed the usual synthesis within the Efabless upload cycle and skipped straight into verification of the user project area. This allows us to use our existing toolset within Cadence and not need to relearn new tools to tapeout using this PDK.

5.2 Differential Diode Design

The second design put onto an IC is the Differential Diode Design. This was performed using a 65nm process and also was limited on the number of stages able to be added since another design was also implemented on-chip which took a majority of the available area. Fig. 5.2 displays the designed chip layout and highlights the section of the chip which contains the designed correlator.

The production of this design was the most unique out of the three designs within this work. This design was handled through Efabless which embeds a user's project within a custom area on a pre-built chip which contains a RISC-V Processor. This allows for additional functionality for digital designs and allows for configurations such as different General Purpose Input Output (GPIO) configurations so that multiple types of operations can be perferred on the single chip. This design did not use all of the functionality as the design is purely analog but has provided an interesting design process.

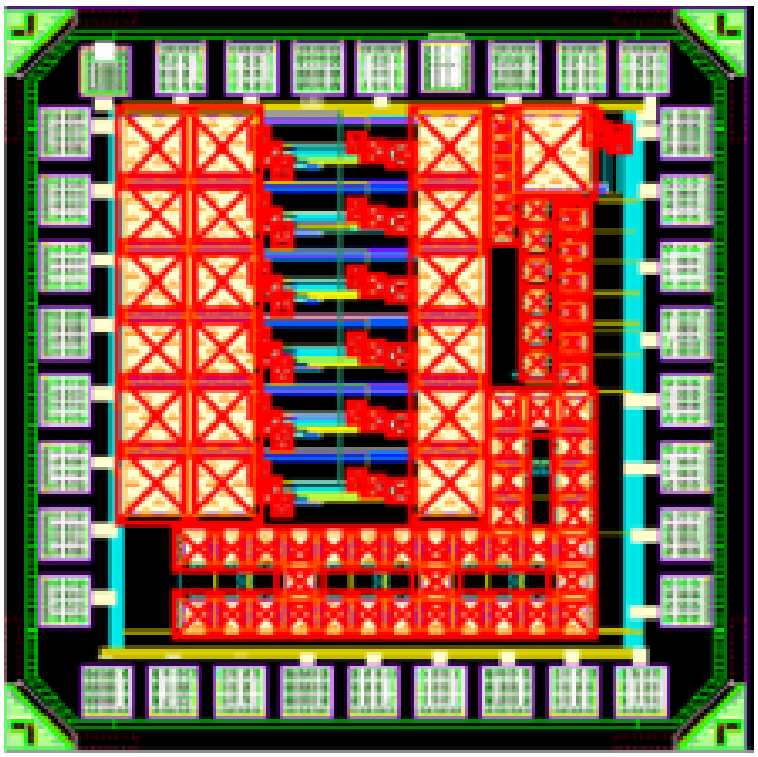

Figure 5.2: Layout of Differential Diode Structure in 65nm process

This design does not experience the design limitations present with the Skywater 130nm design and can be ran at 2 GHz or higher input frequency. This means that this design can be used to test both the basic RFID frequency of 915 MHz and higher frequency for the project.

Something important to note with this design is that the chip area is shared between multiple designs. Due to one of the other designs containing a bug, the performance of the correlator circuit could be degraded. The bug in our case seems to be that the VDD pin draws a large amount of current. This shouldn't affect the functionality, but could increase the power consumption of the design due to the chip substrate being connected to the chip VDD and the design being used containing multiple PN junctions.

5.3 Differential CMOS Design

The last design put on-chip that has been explored in this work is the Differential CMOS Variant of the Resonant Correlator. This design was implemented on a 22nm design using low voltage threshold MOSFETs. This essentially has roughly the same threshold as a schottky diode (approximately 0.3 Volts) but has the added functionality of a usable forward and backgate to further control the threshold and attempt to adjust for any leakage through the transistors. This design is also targeted to operate at 2 GHz minimum. Fig. 5.3 displays the layout of the generated chip. The left-hand design contains a 256-stage and 4-stage testing structure with fully decoupled inputs to fully test the effectiveness of the design.

The pads within this design take an unconventional design. The submitted design contained multiple sub-chips (around 5 in total) and requires additional dicing of the silicon. This greatly reduced the allocated area within the design and created a pad limitation. To address this, staggered inner pads were added to insert additional IOs without needing the increase the area used by each sub-chip. These can still be easily wirebonded for packaging due to the staggered nature of the pins. However, adding more pins also decreases the usable area by the implemented designs.

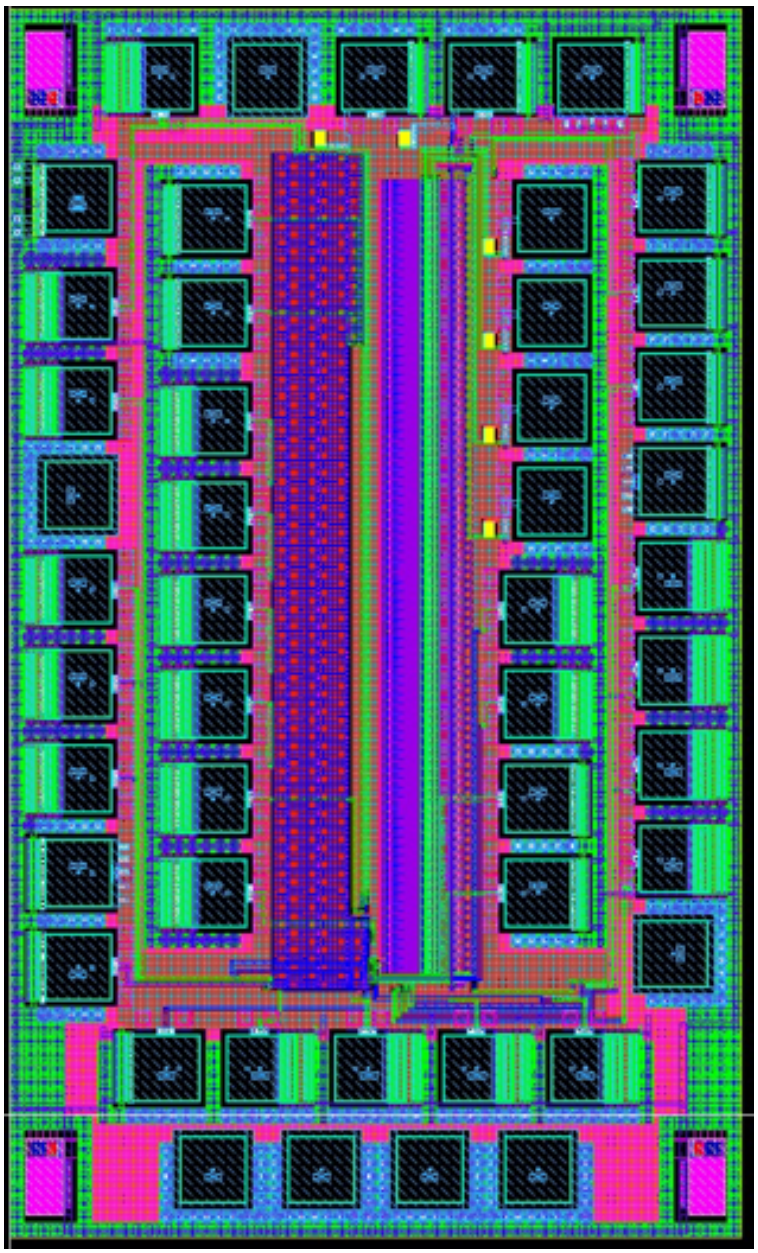

Figure 5.3: Layout of Differential CMOS Structure in 22nm process

Chapter 6

Testing of Designed Correlators

The different taped-out designs were then received and tested to analyze the performance of each design. Each of the chips needed to be packaged and integrated into some testing medium, whether that be a mounting socket with an individual pin-out or a PCB to feed in more advanced inputs to the design. The Skywater 130nm design came with its own testing board, meaning that firmware needed to be developed to test the design while the 65nm design was not provided with a testing board and required a custom PCB. Each of these designs will be analyzed then compared for the most efficient for a set of input vectors. Both of these designs will then be tested using a Software Defined Radio (SDR) to test the wireless functionality and high-frequency response of the designed circuits.

6.1 Design Challenges Faced

Due to the high-frequency design of the developed ICs multiple PCBs needed to be designed to fully test the designs. These involved multiple different aspects which needed to be tackled to ensure accurate results could be recorded from the PCB. For example, to ensure maximum power transfer can occur from the input generator to the designed chip, impedance matching needed to be considered to ensure that power isn't reflected from the transmission line between the board connector and the chip itself. This involved designing a 50 ω impedance line using on-board traces. These line wwere designed using Keysight Advanced Design System (ADS) which allows for full input of the intended substrate for the PCB along with the desired operating frequency of the design to create parameters of width of the line to ensure correct matching. Fig. 6.1 displays an example of microstrip single-ended line matching within ADS for a 2GHz line to 50ω.

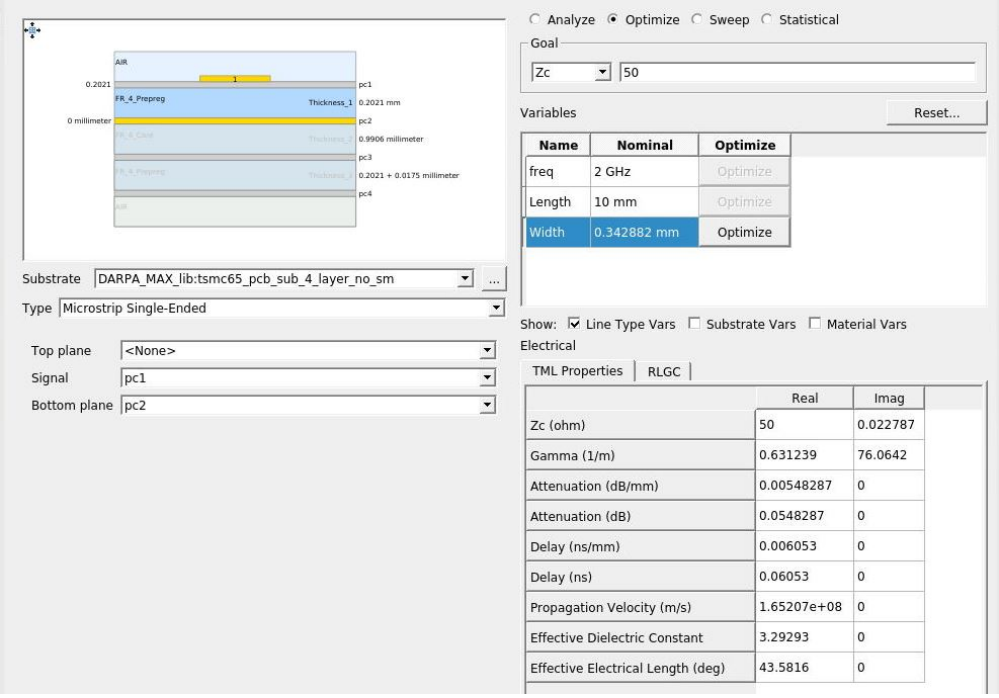

Figure 6.1: Example of Line Matching within ADS

For this design a coplanar-single ended design is used for matching. This means that the signal line is surrounded by a ground-plane within the same layer. The benefit of using this sort of matching method over microstrip single-ended matching is that line width can be thinner due to the surrounding grounding planes. This allows for more space efficient PCB designs which can reduce the cost per board and make interfacing with smaller components simpler (such as an IC). Fig. 6.2 displays the cross section of a coplanar impedance line. The substrate in our designs will be a standard FR4 material which has a dielectric constant of about 3.61.

Another important aspect to consider within the designed PCBs for this process is the physical stackup of the boards. Since the design is a RF design the signal integrity of traces on the board are important to consider. To ensure the best integrity possible a four-layer process is used. Four-layer processes generally can have better isolation between signal lines due to middle layers increasing the shielding. The PCBs designed within this work will have the following layer stack from top to bottom: signal, ground, power, signal. The top signal layer in this design will contain all RF traces since the top layer contains the coplanar matching lines. Fig. 6.3 shows the 4-layer PCB stackup used within all designs.

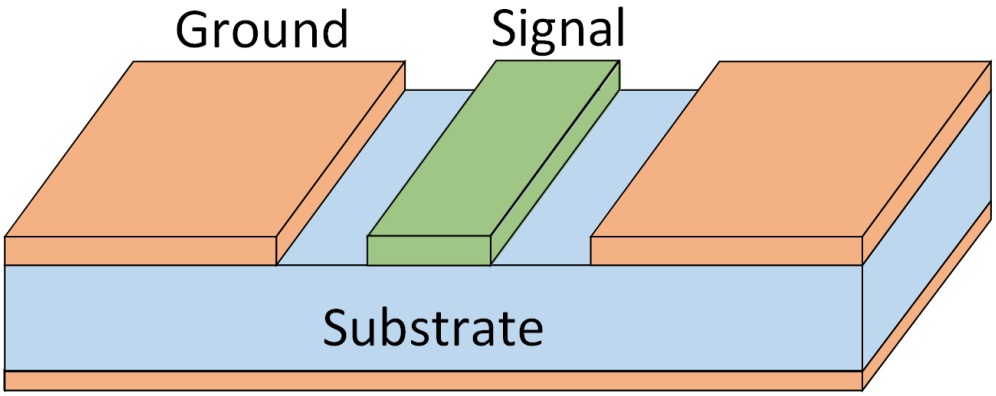

Figure 6.2: Coplanar Matching Line Cross-Section

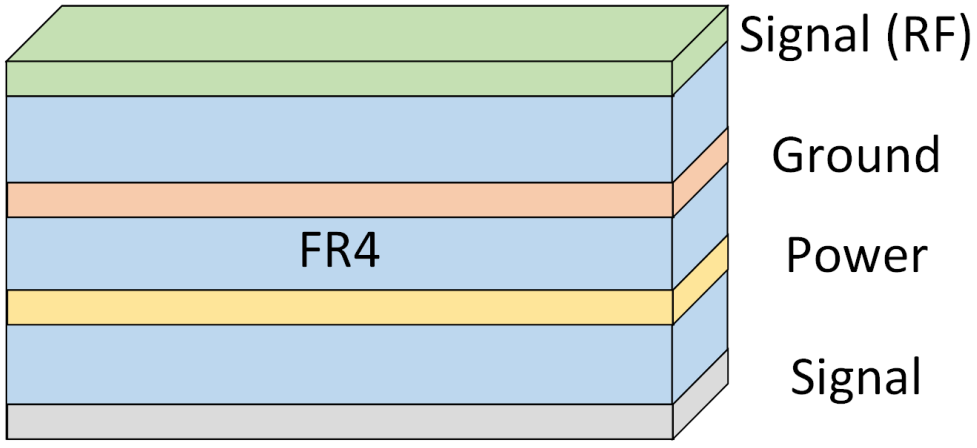

Figure 6.3: Four-Layer PCB Stackup

Another important aspect to consider when designing these PCBs is the soldermask on the RF traces. To ensure that the transmission lines are matched correctly, soldermask is omitted on these traces to ensure that there is no added capacitance caused by coupling between the traces and the soldermask which could change the line impedance. A tradeoff from omitting the soldermask is that the lines will be less protected and more susceptible to damages such as oxidation. This is not an issue in our case since the designed PCBs are purely for prototyping, but this is an important aspect to consider if designing for usage in non-lab environments.

Lastly, the input signal type needed to be considered for the designs. Since most RF systems do not inherently transmit a differential signal, these differential inputs to the chip needed to be performed on board. This is completed using a balun which converts an unbalanced (or single-ended) input into a balanced output (differential in our case). This means that additional matching and components are needed for the designed PCBs to ensure a correct input can be generated for our design from standard RF waveform generators. This also means that the tests using high-frequency inputs must be frequency limited since a finite bandwidth is associated with the utilized baluns.

6.2 Common-Mode Diode Correlator

The Skywater 130nm chip which included the Common-Mode Diode Correlator included a pre-made open-source testing board which was used for analyzing the performance of the design. This board comes with a RISC-V core on board which can be used for controlling all on-board functionality including GPIO pin configurations and performing additional house-keeping tasks. These GPIOs have multiple pin configurations possible including multiple digital and analog modes. Fig. 6.4 shows the pad connections possible to the testboard, and could be helpful for finding any issues with the design, whether it be from functionality or performance.

There are also specialized IO pins for purely analog connections The board also comes with a M.2 connector which allows for different chips to be easily slotted into the board. Fig. 6.5 displays the provided Chipignyte Board and the slotted Skywater 130nm chip.

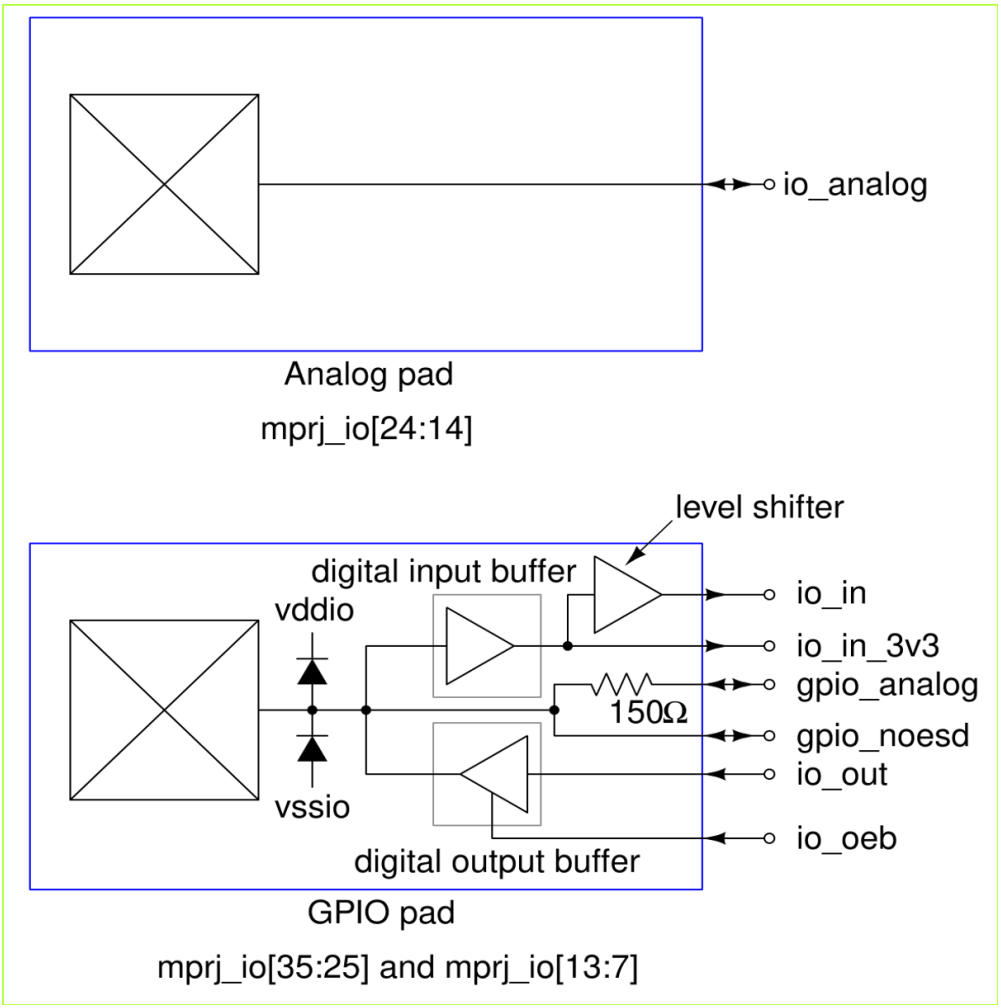

Figure 6.4: Efabless Chipignyte pin Configurations

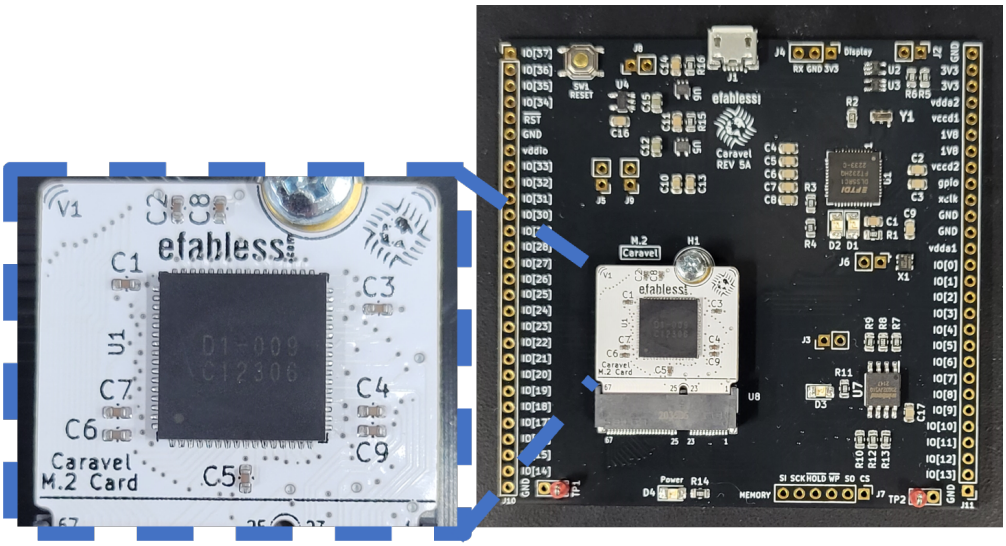

Figure 6.5: Efabless Chipignyte Testboard for Skywater 130nm Chip

Due to our design being predominately analog, a lot of the functionality on this board is not exactly needed such as the RISC-V on the developed chip. We also didn't require custom firmware to be needed for some of our test structures due to some analog pads being connected directly to the design area. An interesting result of this is the board for the single stage design does not actually need to be powered on to be tested due to not using the GPIOs. This board was then tested using the single-stage test structure to test for diode formation and to test functionality of just a single stage. This setup was verified using a DC input (of around or greater than 1.4 Volts) to turn on both diodes from the bias and to the output to check if a DC shifted voltage is outputted on the output terminal. This was verified on the hardware and the next step is to check the functionality of the design. Fig. 6.6 displays the connected board with the single stage configuration.

Data was then recorded of different sinusoidal correlated inputs to determine basic functionality. This means the inputs to the circuit were of the same frequency and peak-to-peak voltage but with a swept phase. This allows us to see the correlation of similar signals that are just shifted for ideal cases of a correlation calculation.

Next a 256-stage correlator design was tested to determine the functionality of a longer correlator now that the 1-stage design was verified. However, this design does require the board to be powered since the GPIO pins are being used. This means firmware must be written, compiled, and flashed onto the board for functionality of the board. It is important

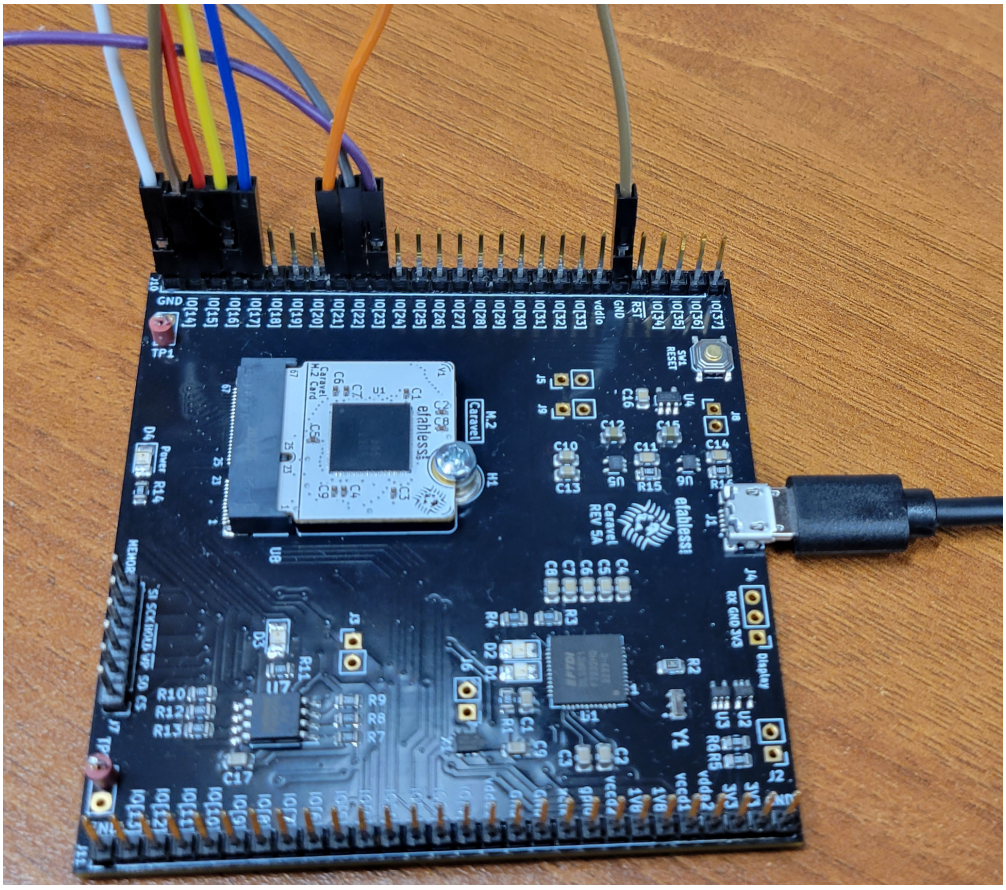

Figure 6.6: Single Stage Correlator Connection on Chipignite Board

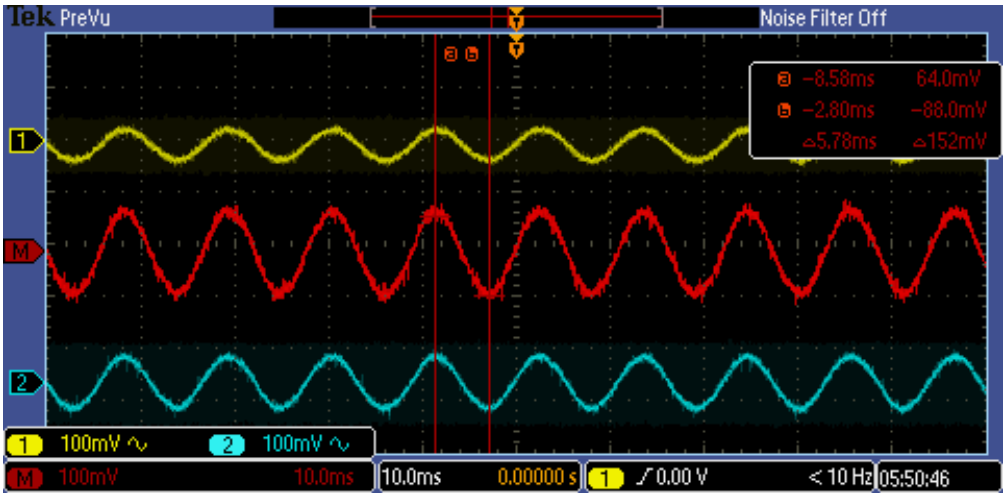

Figure 6.7: Output Waveform of in-phase, correlated signal

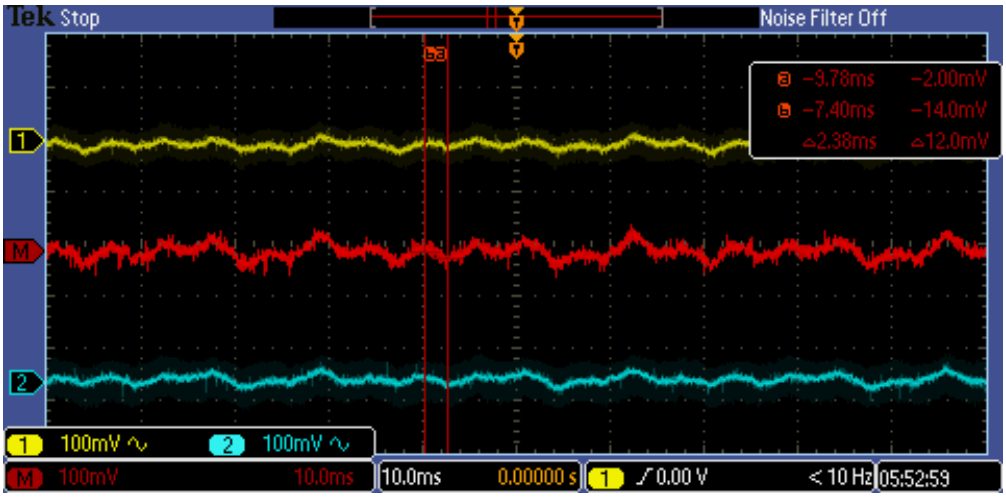

Figure 6.8: Output Waveform of out-of-phase signal

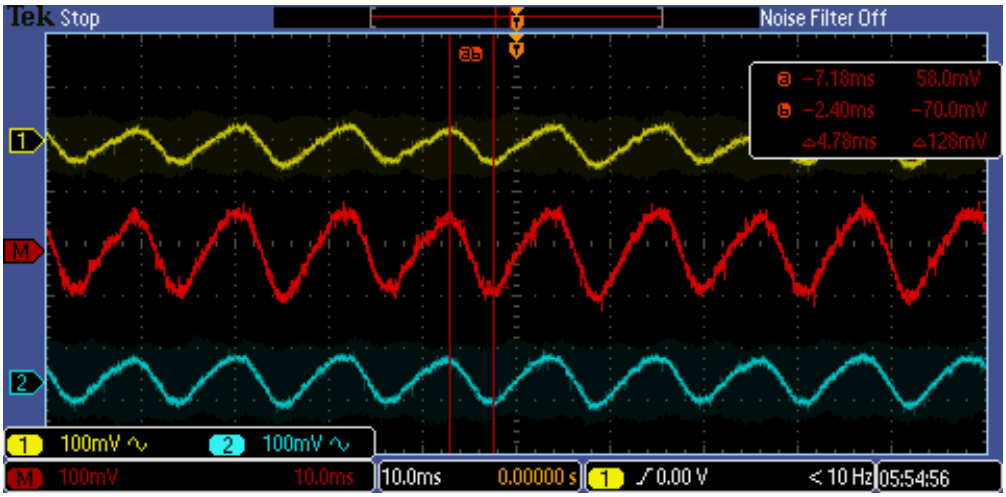

Figure 6.9: Output Waveform of anti-correlated signal

to note that no other functionality is designed using this firmware other than setting the port configurations. Due to the number of stages within the design a higher frequency was needed to ensure we would actually see a signal at the output when a pulse is applied. In this case we used an input of 100 MHz with a smaller peak-to-peak voltage (around 1V per each signal) and swept the phase of one of the input signals to again test the basic functionality. Another interesting note that we found is that the signal seemed to function more similarly to the original single-stage design if the bias pin is left floating. One issue found when testing this design is that the leakage of the capacitors is very high with our input frequency. In order to get an accurate reading of this design to measure additional parameters a much higher frequency is desired to test the design.

Lastly, the fill design was tested (around 400 stages) which also includes larger capacitors was tested for functionality. This design was created to test an extreme case of the design with a significantly higher number of stages while also having the potential to hold more charge per-stage for a wider output range.

Due to the need to collect data from higher frequency sources, pin-headers on the provided Chipignyte board were not sufficient and a different connector type is needed to collect data. For this a custom PCB is designed with SMA connections to RF signals along with RF baluns to split the single-ended RF input waveforms into fully differential waveforms. All of the work to develop this PCB is performed in KiCAD for the designing of the board along with ADS for calculating the trace parameters for impedance matching. Fig. 6.10 displays the schematic of the designed PCB which reuses the footprint and schematic of the Chipignyte design along with including SMA and balun connectors.

Next the layout of the PCB was performed. It is important to note that the designed board uses a 4-layer stackup and RF traces were generally routed on the top layer without soldermask covering. Fig. **??** displays the layout of the designed PCB within KiCAD.

Testing of this PCB will be explored with higher waveform generators in a later section.

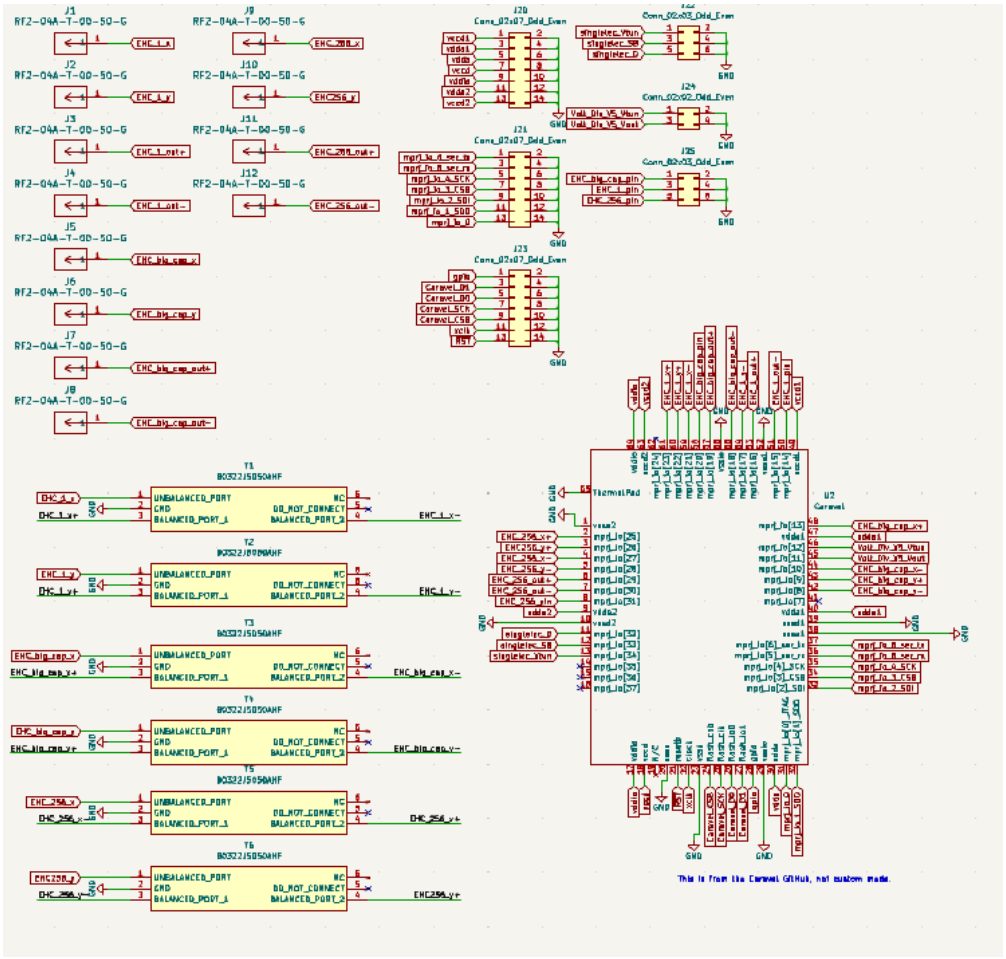

Figure 6.10: Schematic of Skywater 130nm Testing PCB in KiCAD

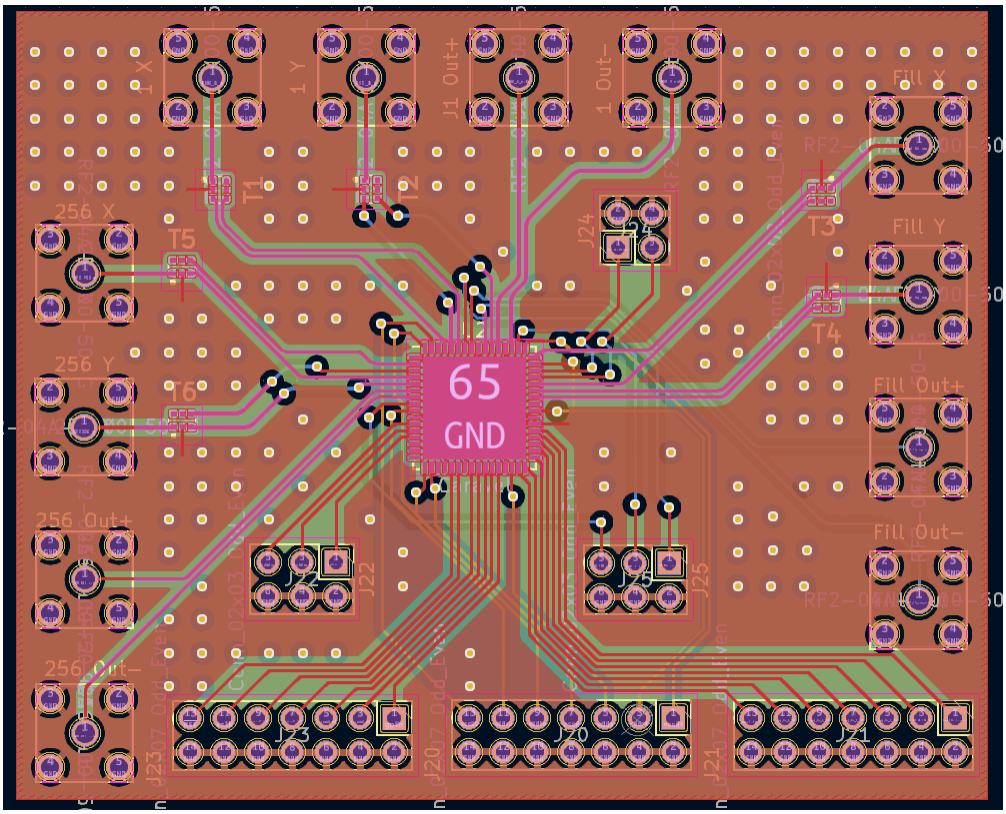

Figure 6.11: Layout of Skywater 130nm Testing PCB in KiCAD

6.3 Alternative Waveform Generation Methods Explored

Due to the restriction of testing equipment present in the testing environment not having a high enough operating frequency, additional methods of waveform generation were explored. These include using an FPGA, Software Defined Radio (SDR), high-frequency Vector Network Analyzer (VNA), and a Phase-Locked-Loop (PLL) generator board. All of these options can provide a much higher testing frequency but all come with their own design challenges. For example, the FPGA being used (a Digilent PYNQ-Z1 in our case) requires additional peripherals using PMOD connectors to implement a Digital to Analog (DAC) on hardware along with the complexity of creating additional software to generate the needed data whether it be through a Hardware Description Language (HDL) or Python. The SDR requires a software frontend to be created so we can control the inputs along with a means of visualizing reflected or collected waveforms through the same design.

Some other desired parameters desired may also require additional hardware such as directional couplers to measure the scattering parameters of the input signals to determine if correlation can also be measured through the input ports of the design. This means the finalized waveform generation method should be chosen with the option to add additional hardware to measure the desired parameters. Many high frequency signals utilize SMA connectors due to their ability to transmit high frequency waveforms without much attenuation and so do many high-frequency peripherals needed.

After working on all of these different testing methodologies, the SDR was decided to be the optimal means of testing the hardware due to the ability to add or change features through software along with the simplicity of generating controllable signals that can be transmitted to the testing PCBs. This also could be useful for testing any future applications of this correlator since it is designed for a wireless RF design.

6.4 Differential Correlator

Unlike the 130nm Chip, the 65nm design did not come with a testing board. This means that we needed to either purchase or create a medium which allowed for the design to be tested. Two different methods for testing the design were performed: using a chip slot holder

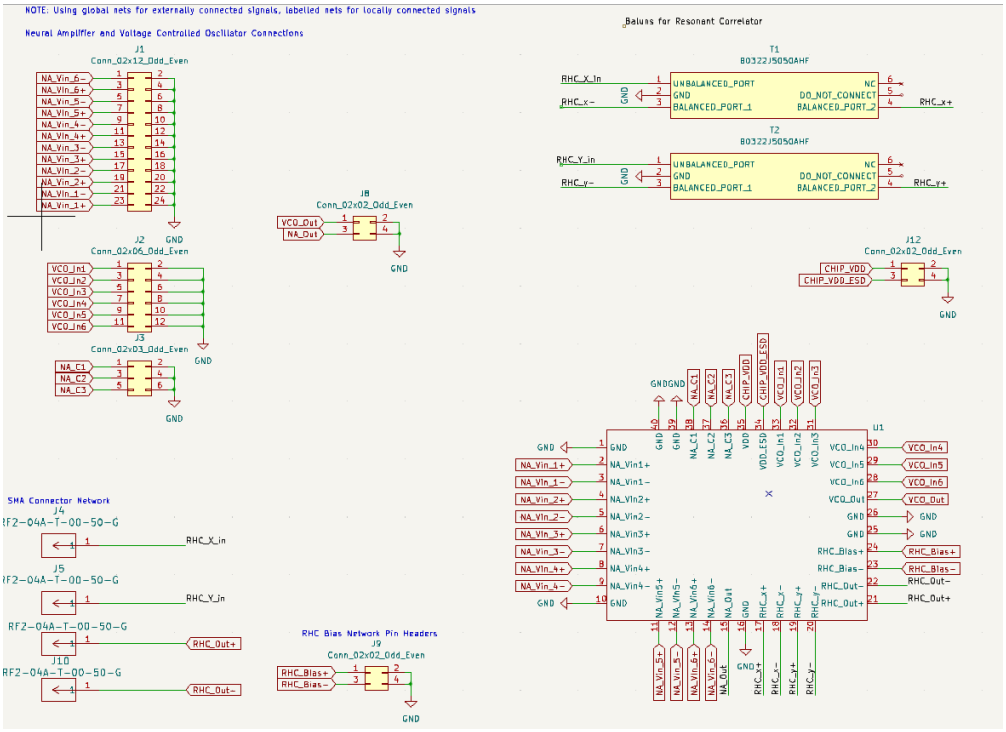

Figure 6.12: Schematic of Differential Correlator Testing PCB in KiCAD

with an exposed pinout and developing a custom PCB which allows for multiple input types to be used to test the chip.

The chip slot holder was tested using our custom in-lab testbench which allows for custom programming of I/Os through MATLAB. One major issue encountered in the testing process is that the custom testbench could not supply a negative voltage which is an issue for a design which generally requires fully-differential inputs. This was alleviated by using an external source generator to create these arbitrary input waveforms.

A similar process was taken to test the design of the 65nm chip. First the diode formation was tested using DC input signals at the bias pins to determine if a DC-offset signal is generated. After verification simple sinusoidal inputs were tested to determine basic functionality.

A custom PCB was also designed to test the chip more robustly. This design involved needing to add RF baluns to the design on board so that single-ended RF inputs from wavefrom generators can be split into balanced inputs to the design. This allows for full-system tests to be completed using conventional RF testing hardware. Fig. 6.12 displays the schematic of the designed PCB within KiCAD.

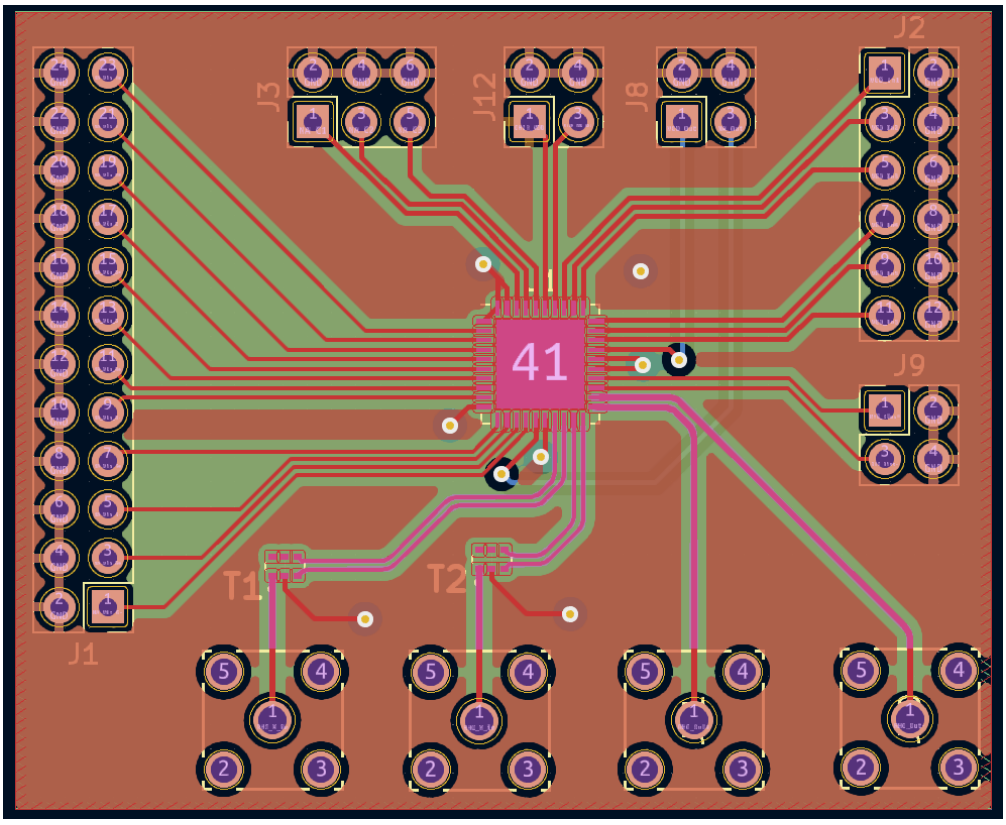

Figure 6.13: Layout of Testing PCB in KiCAD

Important aspects such as matching must be considered when designing these PCBs to ensure that a minimal amount of signal is reflected back into the input. ADS was further used to determine trace widths for both differential and single-ended coplanar matching designs due to their smaller size. Fig. 6.13 displays the layout of the designed PCB in KiCAD.

Careful consideration was taken for this design to ensure that traces are as short as possible and kept tight so that no signal degradation occurs due to the high-frequency designs being tested. Fig. 6.14 shows the fabricated PCB designed for testing.

6.5 Differential CMOS Correlator

The 22nm design which includes the differential CMOS Correlator had not arrived back by the submission of this work. It will be tested as a future work once the IC arrives. To

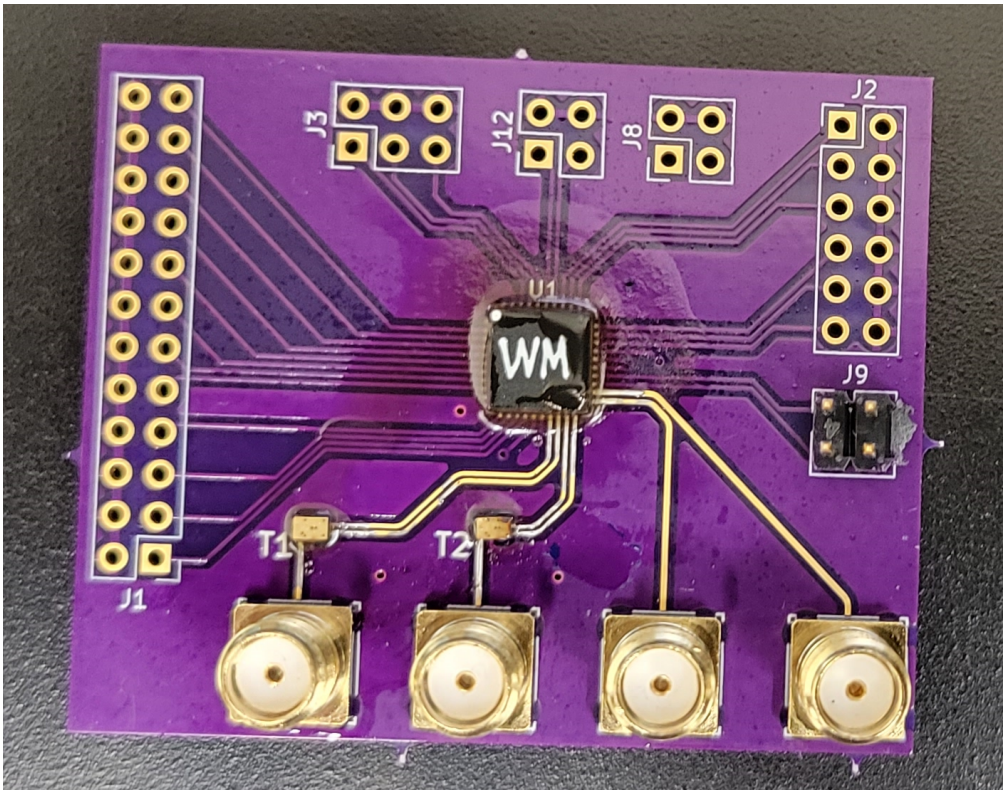

Figure 6.14: Fabricated Differential Correlator Testing PCB

prepare for the testing of this design a new PCB was created which can be used to test the design.

Due to the lower operating voltage for these diodes and their potential susceptibility to noise bypass capacitors were added to ensure no AC signals are coupled into the DC inputs of the circuit. This can further ensure that we have correct biasing for the design without any additional artificats of our testing equipment being inserted.

6.6 Software Defined Radio

After determining verification of the design of the Skywater 130nm and 65nm chips, a Software Defined Radio (SDR) was used to implement more complex waveforms to further analyze the performance of the produced designs. The software used for this SDR is GNU Radio which is an open-source flowchart-based design software for SDRs which synthesizes all developed modules into Python or C++ which makes the framework very flexible for any application and inherently provides User Interface (UI) control using the tool QT. This means that modular testing environments can be easily developed and used for testing the designs. Something that is important to note is that while this work does not explore the testing of the 22nm hardware designed, the test suites developed within this work can be extended to testing these designs.

The hardware used within the testing environment for the SDR is the Ettus USRP N210 which inherently supports GNU Radio and makes programming the radio for testing very simple. Due to the two sets of differential inputs needed for our designs a total of two SDRs of the same model are used. These are interfaced using a Gigabit Unmanaged Ethernet Switch so our computer which hosts the code can be interfaced simultaneously between the devices More testing hardware includes the Copper Mountain S5048 Network Analyzer for analyzing RF signals. An RF chamber is also used for testing the wireless capability of the correlator if connected to external antennas with two RF transmitters.

Within the SDR used includes a daughterboard which controls the transmit and receive functionality. Though the SDRs have their own internal parameter, these daughterboards are the transceivers which provide the actual functionality for delivering information to and

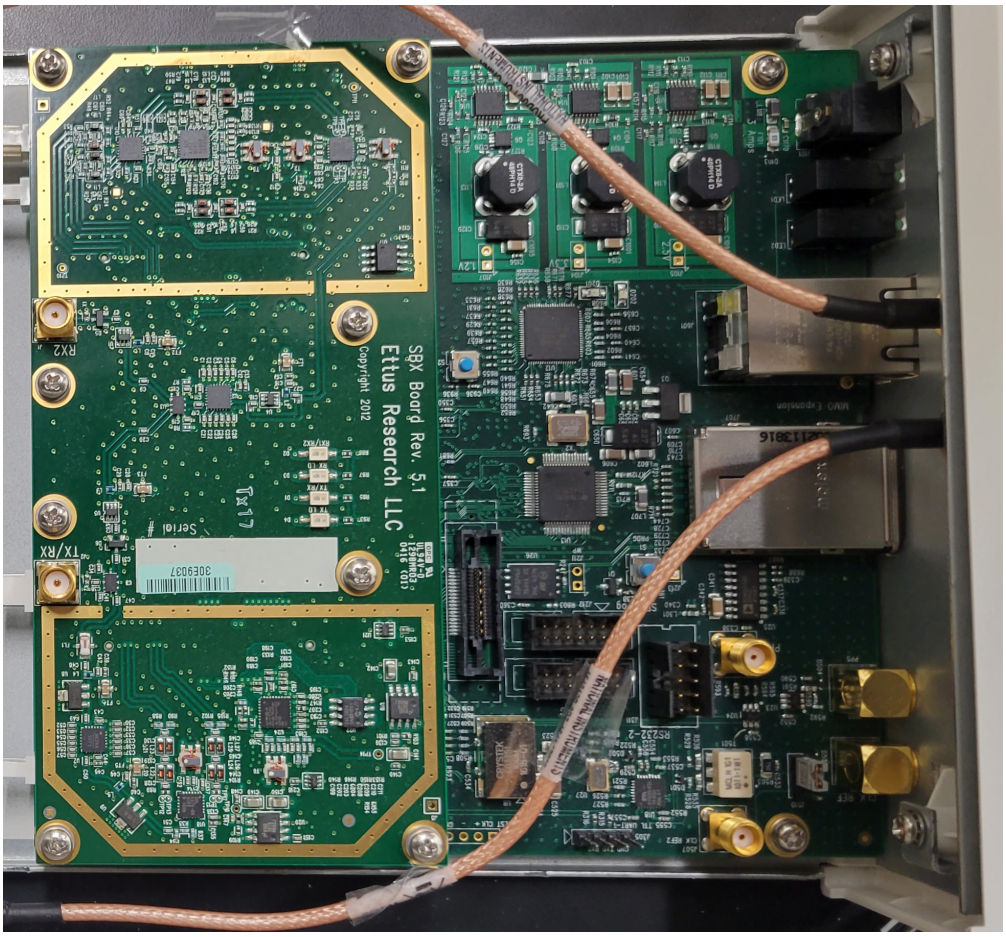

Figure 6.15: Internals of Ettus N210 with SBX Daughterboard

from the SDR. Fig. 6.15 shows the internals of the Ettus N210 with the daughterboard attached.

Alongside providing the RF setup (which this board includes one RX and one TX/RX configuration) this board also provides constraints of the operating parameters of the transmitted or received signal. Table 6.1 displays the SBX daughterboard parameters [20].

Table 6.1: SBX Daughterboard Device Parameters

Carrier Frequency Range	400MHz - 4.4GHz
Bandwidth	40 MHz
Output Power	Up to 100 mW
Noise Figure	5 dB

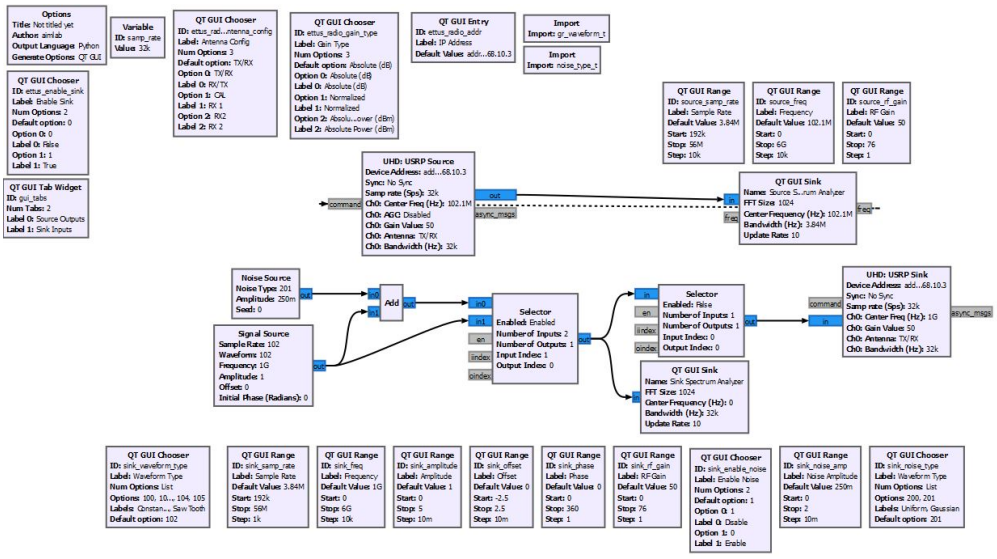

Figure 6.16: Diagram of SDR Setup in GNU Radio

These parameters mean that tranmitted signals need to be with the required carrier frequency band (this has no implication on the baseband signal). Meaning that the software that we develop to control the SDR needs to be controlled directly by the carrier since there is not a downconverter within the designed IC. This means that our actual input signal can have a lower frequency but with a phase shift to provide the different correlation parameters. We are also limited to a single TX signal per SDR but can adjust for this by using baluns to create differential signals at the outputs of the SDR. The carrier restriction set for this board has more than enough range for our testing parameters.

Using GNU Radio a modular application was created which allows for us to interactively decide whether the SDR will be used as a source or a sink for testing. Along with these parameters, sub-configurations exist to further decide the input waveforms to be used when utilized as a sink and the amount of noise to be appended onto the signal. This can be visualized using many different waveform visualizers such as an Fourier Transform or within the time domain. We can also do the same with the source configuration which allows for visualization of the received signal. This interface was designed with simplicity and functionality in mind to avoid the need to change the fundamental operation of the SDR configuration and can be entirely done on the application. Fig. 6.16 displays the block diagram of the developed application within GNURadio.

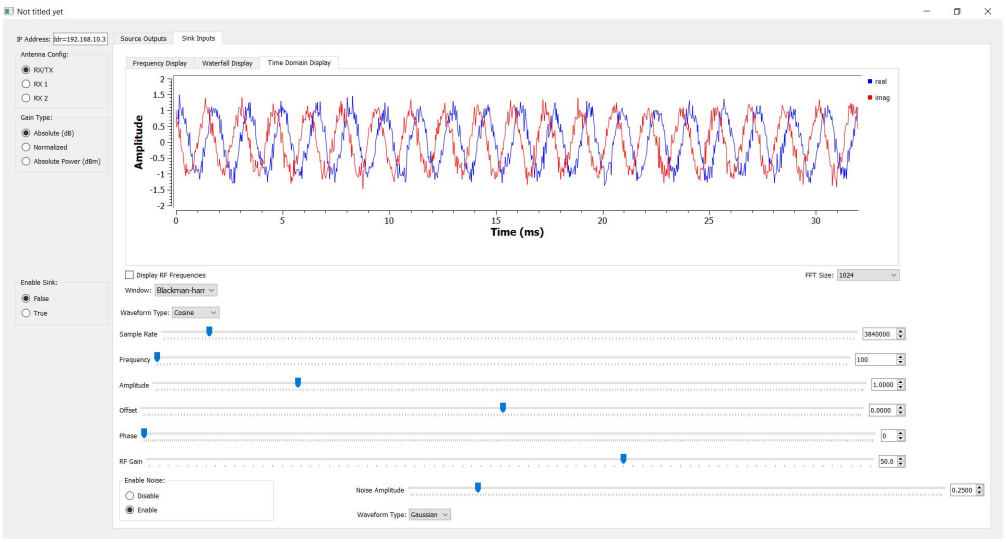

Figure 6.17: GNU Radio Application UI

Fig. 6.17 shows the developed UI for the block diagram. It is important to note that this application compiles using Python and should be able to be ran on most operating systems.

One crucial component when attempting to utilize this software is clock synchronization between the different SDRs. On the hardware used within this work there are multiple ways to achieve full synchronization. One includes using a proprietary MIMO cable to synchronize a pair of SDRs and another uses a reference clock of 10 MHz and a pulse per second (PPS) signal to synchronize the clocks. the reference and PPS method is preferred in our case for scalability in the future in case more SDRs are required to be connected together. For initial testing however since functionality was the desired metric the SDRs were manually calibrated using an oscilloscope to adjust the phases of the baseband signals.

A significant issue with this testing methodology is the fact that the carrier waveform's phases cannot be directly controlled using GNURadio. This means that full characterization of our chips will be very difficult for this system since correlation data can only be directly encoded within the baseband signals. This means that for full characterization a different system must be used which allows for full phase control of the carrier signals. This sort of issue is made irrelevant if the target system has on-board demodulation stages to remove the carrier frequency and the devices being tested can operate at lower frequencies. For the case of the correlators within this work, we would like to test the operation of the device at RFID frequencies without demodulation.

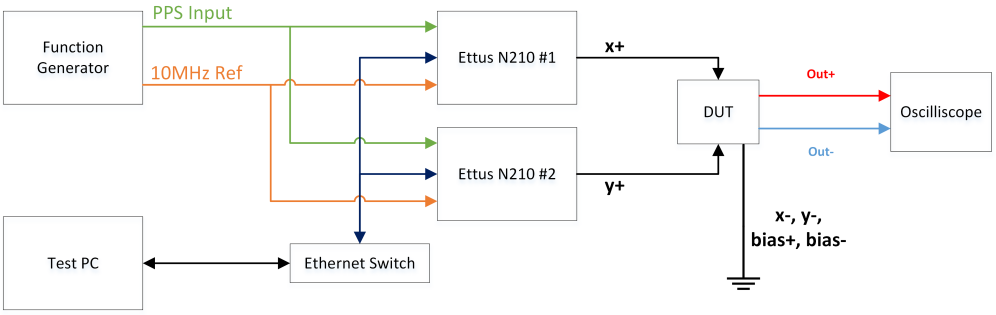

Figure 6.18: Block Diagram of Test Setup for Functionality

6.7 Testing of Designs using SDR

The next step after developing the GNURadio program to utilize the SDR is to test the designed chips using the application. Due to the need of having multiple differential inputs multiple SDRs are utilized to provide multiple input signals. Couplers are placed between the SDR inputs and the SMA connectors which are directly connected to the PCB input. A network analyzer is then connected to the coupled port to visualize the scattering parameters while providing little loss. The scattering parameters of interest in our case is S_{11} since we want to see the signal which is being reflected back into an input.

First we would like to test the functionality of the 65nm chip to determine the full functionality of the correlator using real RF signals. Now that the SDRs are set up, there is valid equipment that can be used to provide such an output. Fig. 6.18 displays a block diagram of the testing setup used to measure the functionality of the chip. It is important to note that these inputs will be constant sine waves, thus no demodulation is needed at the output to process any data.

The clock synchronization which is used within this testing environment is using the waveform generator for synchronization due to the availability of the hardware to generate such signals. These were generated using a Rigol DG4102 two-channel waveform generator. The inputs used are a 3.5V square pulse with a period of 1 second to emulate the PPS signal and a 10 MHz square wave at 1V to emulate the reference clock. Fig. 6.19 displays the hardware configuration using the SDR which was used for collected samples

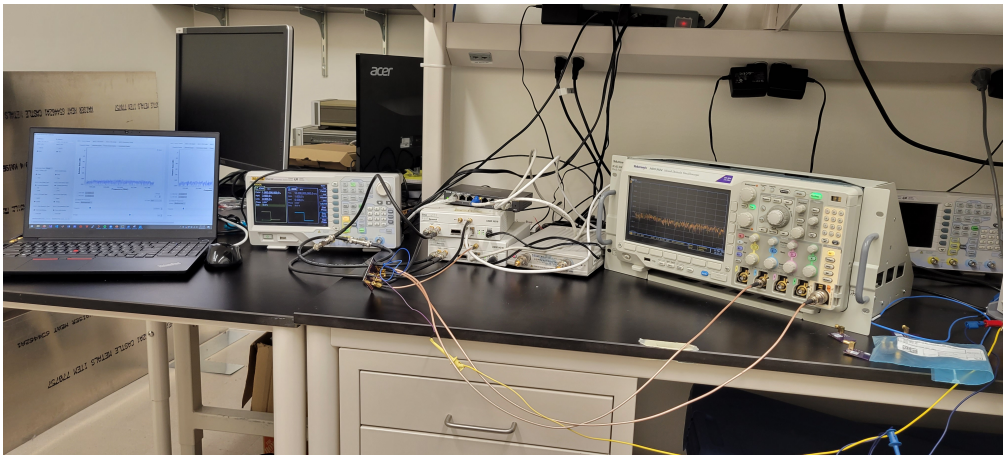

Figure 6.19: Hardware Testing Setup using SDR

6.8 Testing of Differential Correlator using SDR

The first test metric for testing the differential correlator is to test a single-ended configuration. This involves grounding the negative terminals of the inputs and only generating inputs on the positive inputs of the design. This can be used to test basic correlation for further proving functionality of the designs. Due to limitations for visualization with our testing setup, the spectral view of the output was visualised to see the power consumption of the design over different correlation values. Since the SDR testing configuration does not allow for carrier phase control, the correlation information is encoded within a baseband sinusoid meaning the correlation is encoded in a much lower frequency band than the transmitted waveform and is implemented within an amplitude modulation scheme. This is not the most ideal method of testing, since there is no demodulation stage on board, however for the current setup we can use this to provide an estimation of correlation functionality. Fig. 6.20 - 6.22 displays discrete correlation metrics for the single ended inputs (x+ and y+) at 90 degree phase difference offsets.

We can see that the measured dBm power is at its highest when the inputs are correlated and at its smallest when the inputs are anti-correlated at the positive output terminal. Table 6.2 displays the measured peak powers for the swept test.

We can see when converting this power to units of watts a significant power decrease is experienced as the correlation between the inputs decreases which shows that the differential design is fully working as intended at RFID frequencies for this input set.

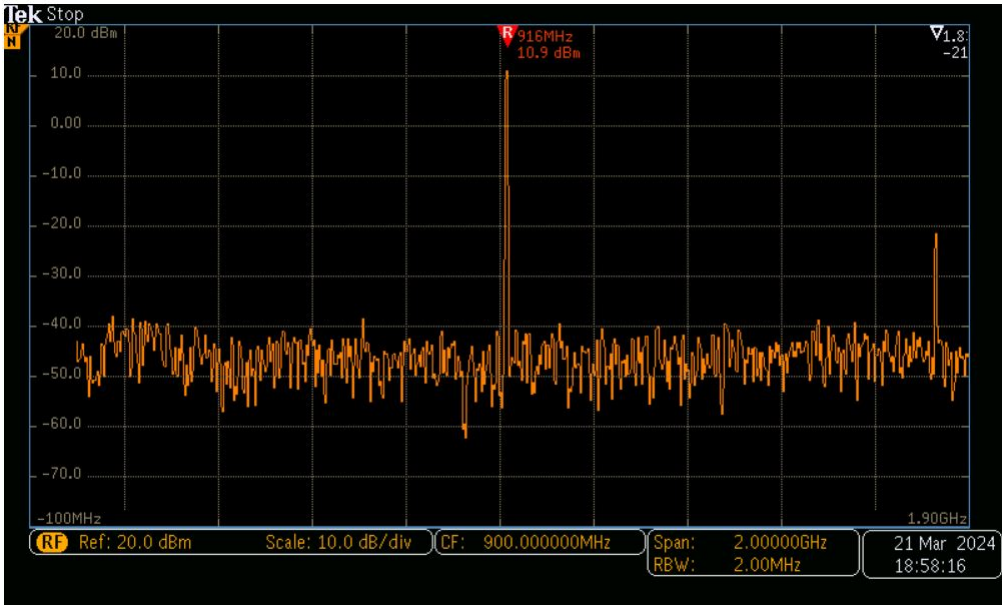

Figure 6.20: Scope Screenshot of single-input spectral output with 0 degree phase difference, 2Vpp Inputs at 915 MHz, 20 dB RF Gain

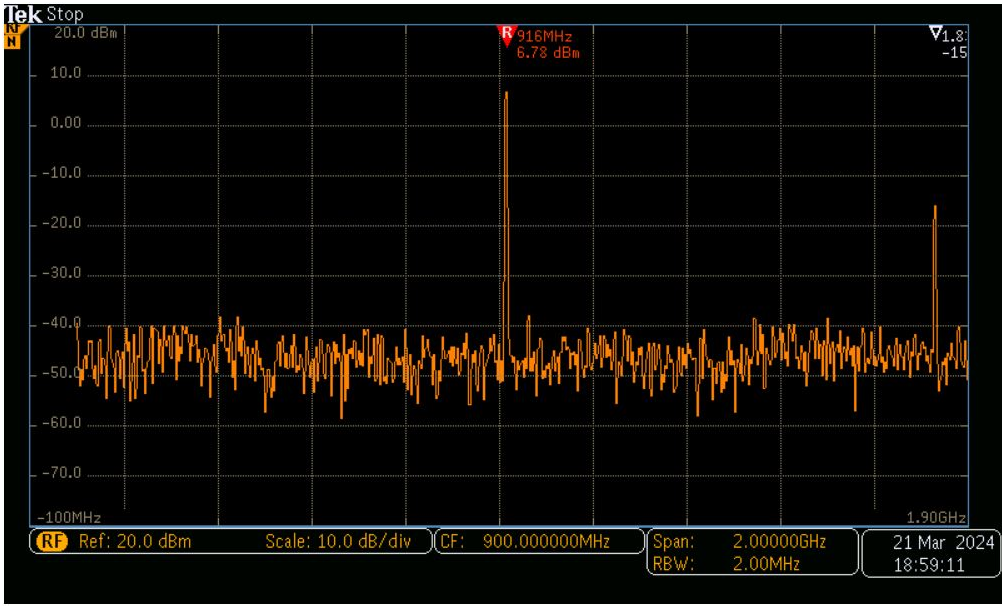

Figure 6.21: Scope Screenshot of single-input spectral output with 90 degree phase difference, 2Vpp Inputs at 915 MHz, 20 dB RF Gain

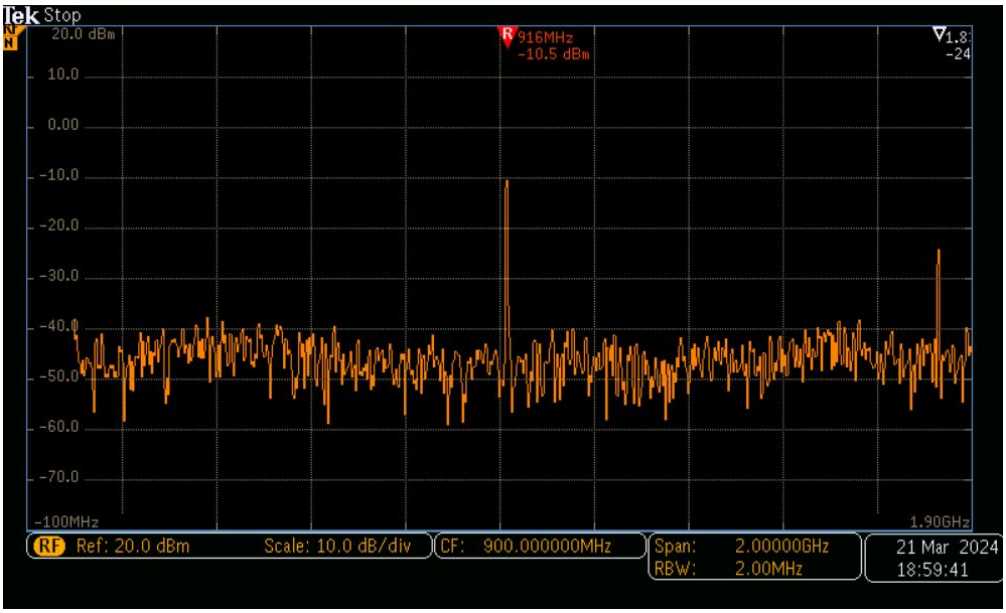

Figure 6.22: Scope Screenshot of single-input spectral output with 180 degree phase difference, 2Vpp Inputs at 915 MHz, 20 dB RF Gain

6.9 Arbitrary Waveform Generator System

Another testing device obtained later in the testing process is a Proteus P9484D Arbitrary Waveform Generator (AWG). This system allows for high frequency pulses to be generated with a greater amount of control than the SDR system. For example, with the AWG device we can control almost all characteristics of the carrier waveform, such as phase, which was not possible for the SDR system. This means with this testing device we can instead directly connect the DUT to the AWG and not need a demodulation stage unlike the SDR system. The AWG system used also allows for differential outputs on each channel. This means that for our devices we do not require baluns for testing. Table 6.3 displays the parameters of the utilized AWG. Fig. 6.23 displays the AWG used for testing.

Table 6.2: dBm Power of Single-Ended Test with discrete phase differences

Input Phase Difference (Deg)	Output Power (dBm)	Output Power (W)
0	10.9	12.3 mW
90	6.78	4.76 mW
180	-10.5	89.1 μW

Figure 6.23: Proteus P9484D AWG

Parameter	Value
Sampling Rate	9 GS/s
DAC Resolution	16-bit
Number of Channels	4 (Differential)
Maximum Output Channel Voltage	550 mVpp
Waveform Granularity	32 points

Table 6.3: Parameters of the Proteus P9484D

With this AWG we can also use a Digital Up Converter (DUC) with different levels of interpolation of our input signal to the channel to increase the highest possible carrier frequency on the output channels. The carrier frequency can be at highest 40% of the sampling rate of the AWG. This means with the highest level of signal interpolation (X8) the highest DUC frequency for the carrier is 3.6 GHz which is more than sufficient for the designs within this work.

The AWG used within this work also includes a digitizer. This is an input which can sample some signal inputted into the channel for visualization through software such as MATLAB. The digitizer for our case can be used to visualize the generated channel waveforms to measure and ensure a correct phase state for input sinusoids instead of requiring an oscilloscope for verification.

An issue that comes up with this system is that the output voltage range of the AWG is not sufficient for testing the diode designs on the received ICs. This means that additional hardware is required to generate a high enough amplitude waveform to power on the device. This will be addressed using an LNA which has an average gain of around 21 dB in the 500 MHz through 6 GHz band when supplied a +12V source. We can visualize the setup required for this AWG system to test our designs in Fig. 6.24.

To use the AWG multiple MATLAB Scripts were written to program the device. This was completed by sending commands using the Standard Command for Programmable Instruments (SCPI) syntax. The main tests needed to run to create a testing system for the designed chips is to generate high frequency carriers with controllable phase and to find operating points of AWG channel amplitude which can be used to provide a sufficient voltage range to test the designs. This was performed by creating a script to slowly ramp up the device voltage over a small range. Since our testing PCBs contain baluns, the inverted output of each channel will directly connected to a 50 ω terminator.

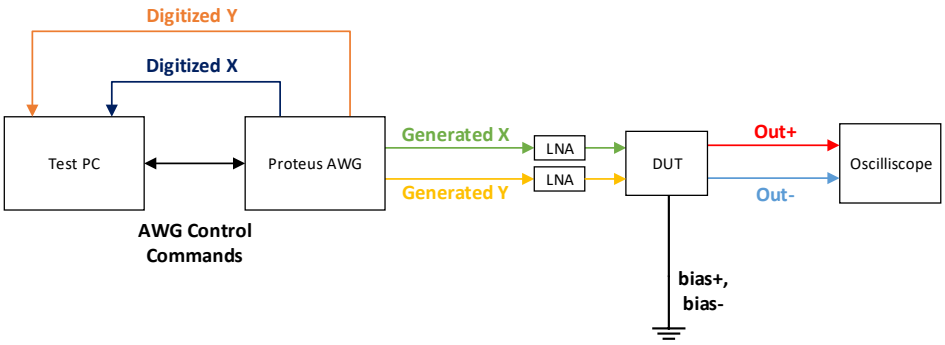

Figure 6.24: AWG Test Setup Block Diagram

To ensure that safe output voltages are being used from the LNA, MATLAB scripts were created to slowly ramp up the channel output voltage to determine operating points for testing the designs. For example, a peak-to-peak voltage of around 1.4V is desired for the diode correlator. These waveforms were visualized using an oscilliscope attached directly to the LNA output terminal. An example of this is at a carrier frequency of 500 MHz, a channel amplitude of around 400 mV was found sufficient to power on the diode device. This process can also be repeated for different frequencies and different desired operating points for all desired designs to be tested.

After finding these safe operating points of the AWG utilizing the LNAs the designed chips will be able to be fully characterized. We can further use the AWG once the chips are verified to not only test operation with basic sinusoidal carriers, but with data encoded in other waveform types such as square waves. All of the correlation encoding can be visualized using the AWG digitizer (provided the port into the digitizer does not contain an amplifier). We can also further use the spectrum analyzer functionality of the utilized oscilloscope to determine operations over a large band of frequencies. All of these aspects can be utilized to fully characterize the explored designs to determine their feasibility to be used on a custom RFID tag.

Chapter 7

Conclusions and Future Work

7.0.1 Accomplishments

Within a unique method of RFID tag tracking for high-precision data was explored. While analyzing data for this localization it was found that using correlation could be a potential method to correct errors found within the raw processed data. Though this seems like a promising metric to use to correct this phase data, some tradeoffs come from this implementation. A hardware design which can alleviate some of these tradeoffs, such as tag limitation and processing time, was explored in the form of a correlator circuit.

The hardware developed in this work explores a novel fully-passive analog correlator was designed, analyzed, taped-out, and initially characterized. Multiple versions of this design were explored to improve the performance and better suite applications such as in a low-voltage environment. Multiple testing setups were also developed for different methods of encoding correlation, one which involved baseband encoding through the SDR system and the other using the AWG which allows for encoding of correlation within either the carrier and baseband signals.

7.0.2 Future Work

Some future work includes more robust testing of the created design to determine the where the design can be improved. Another improvement could be to explore alternative storage methods (instead of using a capacitor) to reduce the area of this design. This is because for large voltages a bigger capacitor may be needed to store the charge between stages. Another possible future work could be to improve the operating range of the correlator. The frequency of the design could be pushed much higher by experimenting with different components such

as BJTs. Another potential area which could be explored for the design is with the addition of a oscillator connected to the bias point. In theory, this would allow for the circuit to act as an oscillator when no correlation is present at the inputs. This could be useful for different logic modes. For example, positive and negative correlation could result in different functionality for the connected peripheral circuit while a no-correlation input could act as a hold state or different input correlation voltages could activate different functionality for the designs. Another future step is to utilize the SDR testing setup developed on a system where the topologies explored is embedded on an actual RFID tag. This could be useful if we include a demodulation stage on tag which could allow for correlation calculation on baseband signals.

The RFID correlators designed can also be implemented on an actual tag for use with the explored tracking methodology. This will allow for correction of issues explained with the designed system and can be deployed into a real-tracking environment that requires high precision, such as tracking electrode patches.

References

[1] S. Zhang, C. Yang, X. Kui, J. Wang, X. Liu, S. Gui, "Reactor: Real-time and accurate contactless gesture recognition with rfid," *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2019.

[2] X. Tao, T.B. Shaik, N. Higgins, R. Gururajan, X. Zhou, "Remote patient monitoring using radio frequency identification (rfid) technology and machine learning for early detection of suicidal behaviour in mental health facilities," *Sensors*, 2021.

[3] Z. Xiao, A. Kilgore, S. Chakrabartty, "A framework for analyzing analog correlators using price's theorem and piecewise-linear decomposition," *Arxiv*, 2024.

[4] S. Chakrabartty, G. Cauwenberghs, "Margin propogation and forward decoding in analog vlsi," 2003.

[5] V. Lee, A. Alaghi, L. Ceze, "Correlation manipulating circuits for stochastic computing," *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018.

[6] V. Mangal, P. Kinget, "Clockless, continuous-time analog correlator using time-encoded signal processing demonstrating asynchronous cdma for wake-up receivers," *IEEE Journal of Solid-State Circuits*, vol. 55, Aug. 2020.

[7] R. Nayak, A. Singh, R. Padhye, et al, "Rfid in textile and clothing manufacturing: technology and challenges," *Fashion and Textiles*, 2015.

[8] P.V. Nikitin, K.V.S. Rao, "Theory and measurement of backscattering from rfid tags," 2006.

[9] M.M. Al-Azawy, F. Sari, "Analysis of dickson voltage multiplier for rf energy harvesting," *Global Power, Energy and Communication Conference*, 2019.

[10] Y.C. Wong, P.C. Tan, M.M. Ibrahim, A.R. Syafeeza, N.A. Hamid, "Dickson charge pump rectifier using ultra-low power (ulp) diode for ban applications," *Journal of Telecommunication, Electronic and Computer Engineering*, Dec. 2016.

[11] A. Diallo, Z. Lu, X. Zhao, "Wireless Indoor Localization Using Passive RFID Tags," *The International Conference on Mobile Systems and Pervasive Computing (MobiSPC)*, Aug. 2019.

[12] L. Qiu, X. Liang, Z. Huang, "Patl: A rfid tag localization based on phased array antenna," *Sci Rep*, vol. 7, 2017.

[13] M. Li, G. Shengxi, C. Guangyu, Z. Zhu, "A rfid-based intelligent warehouse management system design and implementation," *2011 Eighth IEEE International Conference on e-Business Engineering*, 2011.

[14] T. Tsukiyama, "Rfid based navigation system for indoor mobile robots," *Proceedings of the 18th World Congress The International Federation of Automatic Control*, 2011.

[15] S. Cheng, S. Wang, W. Guan, H. Xu, P. Li, "3dlra: An rfid 3d indoor localization method based on deep learning," *Sensors*, 2020.

[16] *Impinj Speedway Revolution Reader Application Note Low Level User Data Support.*

[17] H. Park, S. Jung, H.J. Chung, "An analog correlator based cmos analog front end with digital gain control circuit for hearing aid devices," *Analog Integr Circ Sig Process*, 2020.

[18] H. Uluşan, K. Gharehbaghi, Ö. Zorlu, A. Muhtaroğlu, H. Külah, "A self-powered rectifier circuit for low-voltage energy harvesting applications," *International Conference on Energy Aware Computing*, 2012.

[19] C. Peters, J. Handwerker, D. Maurath, Y. Manoli, "An ultra-low-voltage active rectifier for energy harvesting applications," *IEEE International Symposium on Circuits and Systems*, 2010.

[20] *Ettus SBX Manual.*

Appendix A

Software Defined Radio (SDR) Code

The code developed for the SDR testing system was developed in a hierarchical manner for code reuse since there are two identical SDRs used within the testing system. This means that most options for the SDRs and their associated input waveforms are parameterized for user control. This includes visualization for both inputs and outputs from the SDR along with the ability to use the systems with functionality such as an oscilloscope.

This software was created utilizing GNURadio which is an open-source software kit allowing for SDR application development. This software uses a block-diagram style of development and compiles the created blocks into Python or C code upon execution.

A.1 Front-End Code

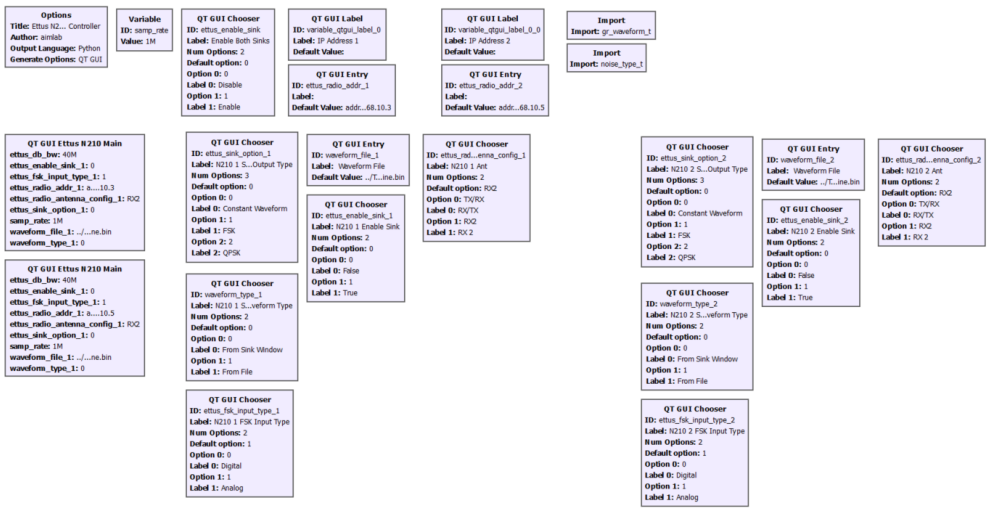

Figure A.1: SDR Front-End Block Code

A.2 Ettus USRP N210 Control Unit

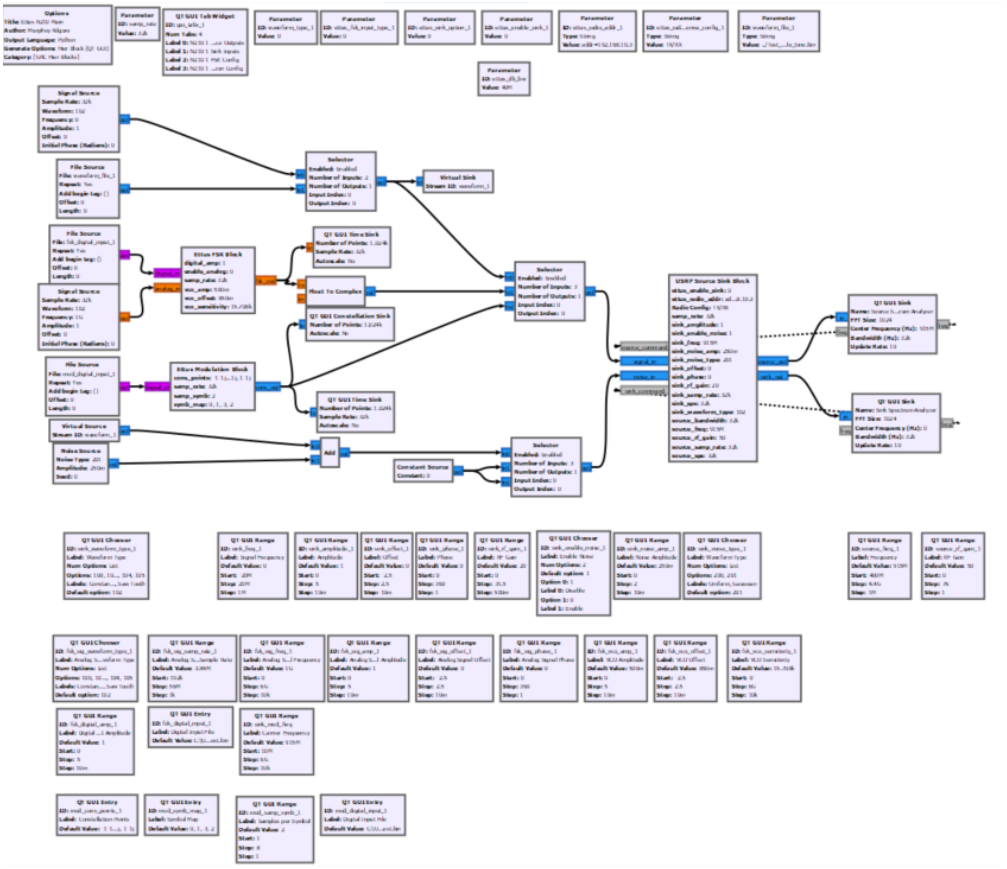

Figure A.2: SDR Ettus Controller

A.3 Ettus USRP Source Sink Controller

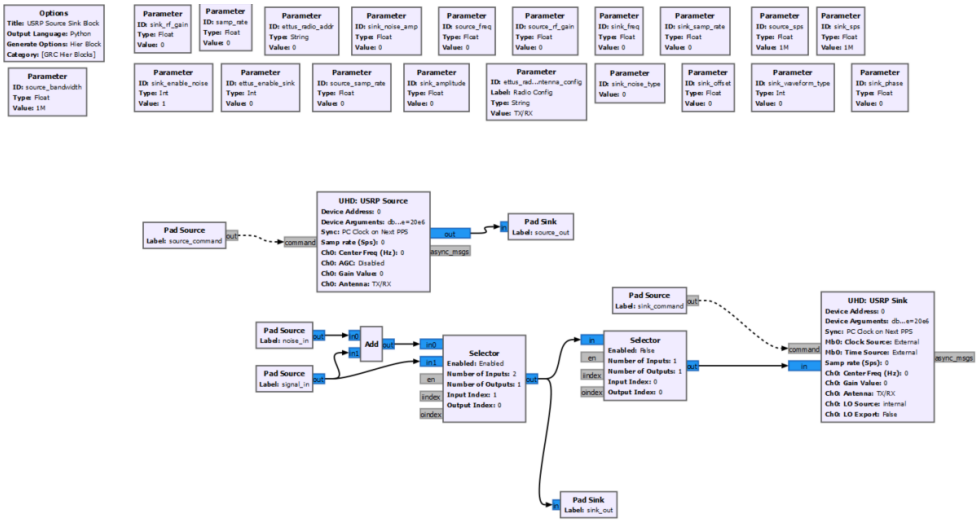

Figure A.3: SDR Ettus Source Sink Controller

A.4 Ettus FSK Block

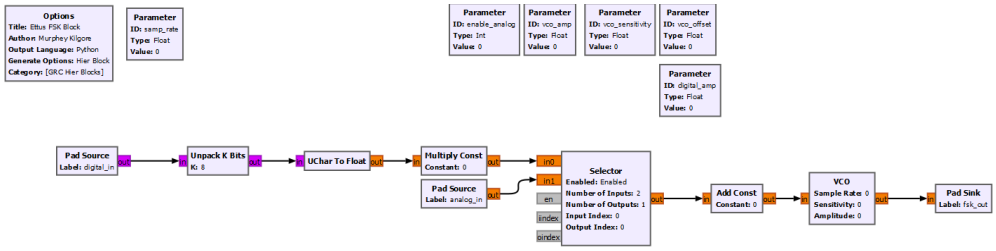

Figure A.4: SDR Ettus FSK Block

A.5 Ettus Modulation Block

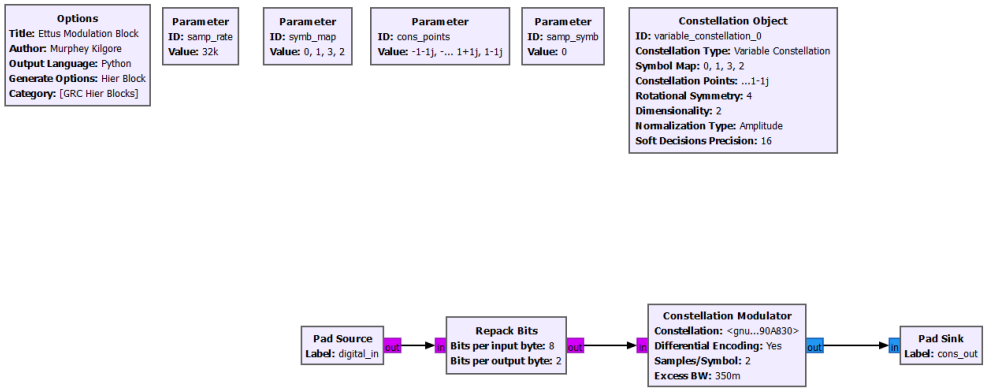

Figure A.5: SDR Ettus Modulation Block

Appendix B

Arbitrary Waveform Generator Codes

This appendix includes developed code to be used with the AWG utilized within this work. All of these scripts are written in MATLAB. These scripts do not use a proprietary interface to interact with the AWG but instead utilize the SCPI syntax for communicating with the device. Using MATLAB allows for simple scripts to be executed without having to manually input each command.

B.1 Convert Waveform to Binary

Input waveform data must be converted into a binary value (range determined by the number of DAC bits being used).

```
%% Script to send a sine wave using the DUC

%% Parameters for function
% Waveform Parameters
base_f = 10E3;
base_vpp = 500E-3;
base_samples = 1024; % This must be at least 64
base_phase = 0;

% Create a single period of the baseband sine
period = 1/base_f;
time = 0:(period/(base_samples-1)):period;
baseband = (base_vpp/2)*sin(2*pi*base_f*time + base_phase);
```

```
% Proteus connection parameters
ip = '192.168.20.5';
conn = 'LAN';
paranoia = 2;

% AWG options
samp_rate = 1000E6;
dac_bits = 16;
interp = 1;
chan_num = 1;
seg_num = 1;
v_pp = 0.5;
v_offset = 0.0;

%% Convert baseband to bit value
[baseband_inp, baseband_len] = convert_wfm_to_bin(baseband, v_pp,
   dac_bits, 'uint16');

%% Connect to device

[inst, admin, idnstr] = ConnecToProteus(conn, ip, paranoia);
fprintf('Connected to: %s\n', idnstr);

% Reset AWG to be safe
fprintf('Reseting AWG. \n');
inst.SendScpi('*CLS;*RST');

%% Set up AWG parameters
fprintf('Setting up AWG\n');
inst.SendScpi([':FREQ:RAST ' num2str(samp_rate)]);
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
% DAC Mode set to 'DIRECT" (Default)
inst.SendScpi(':SOUR:MODE DIRECT');
inst.SendScpi(':TRAC:DEL:ALL');
```

```
inst.SendScpi(':TRAC:FORM U16');

% Channel / Segment Parmaeters
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':TRAC:DEF %d, %d', seg_num, baseband_len));
inst.SendScpi(sprintf(':TRAC:SEL %d', seg_num));

fprintf('Done!\n');

%% Write waveform to selected channel
fprintf('Downloading waveform to device.\n');
prefix = ':TRAC:DATA 0,';
res = inst.WriteBinaryData(prefix, baseband_inp);
assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Set voltage characteristics of channel
fprintf('Setting Voltage characteristics for CH %d.\n', chan_num);
inst.SendScpi(sprintf(':SOUR:FUNC:MODE:SEGM %d', seg_num));
inst.SendScpi(sprintf(':SOUR:VOLT %s', num2str(v_pp)));
inst.SendScpi(sprintf(':SOUR:VOLT:OFFS %s', num2str(v_offset))); %
    NOTE: This is not supported in DIRECT, AC Coupled

assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Use segment mode for baseband, assuming segment table is empty
   on startup
inst.SendScpi('FUNC:MODE:SEGM 1');
inst.SendScpi(':SOUR:VOLT MAX');

%% Turn on the output
```

```
inst.SendScpi(':OUTP ON');
fprintf('CH %d Output Enabled.\n', chan_num);

%% Turn off the output
inst.SendScpi(':OUTP OFF');
fprintf('CH %d Output Disabled.\n', chan_num);

%% Disconnect when done
fprintf('Disconnecting from Proteus\n');
inst.Disconnect();
```

B.2 Basic Sine Transmission

```
%% Script to send a sine wave using the DUC

%% Parameters for function
% Waveform Parameters
base_f = 10E3;
base_vpp = 500E-3;
base_samples = 1024; % This must be at least 64
base_phase = 0;

% Create a single period of the baseband sine
period = 1/base_f;
time = 0:(period/(base_samples-1)):period;
baseband = (base_vpp/2)*sin(2*pi*base_f*time + base_phase);

% Proteus connection parameters
ip = '192.168.20.5';
conn = 'LAN';
paranoia = 2;

% AWG options
```

```
samp_rate = 1000E6;
dac_bits = 16;
interp = 1;
chan_num = 1;
seg_num = 1;
v_pp = 0.5;
v_offset = 0.0;

%% Convert baseband to bit value
[baseband_inp, baseband_len] = convert_wfm_to_bin(baseband, v_pp,
   dac_bits, 'uint16');

%% Connect to device

[inst, admin, idnstr] = ConnecToProteus(conn, ip, paranoia);
fprintf('Connected to: %s\n', idnstr);

% Reset AWG to be safe
fprintf('Reseting AWG. \n');
inst.SendScpi('*CLS;*RST');

%% Set up AWG parameters
fprintf('Setting up AWG\n');
inst.SendScpi([':FREQ:RAST ' num2str(samp_rate)]);
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
% DAC Mode set to 'DIRECT" (Default)
inst.SendScpi(':SOUR:MODE DIRECT');
inst.SendScpi(':TRAC:DEL:ALL');
inst.SendScpi(':TRAC:FORM U16');

% Channel / Segment Parmaeters
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':TRAC:DEF %d, %d', seg_num, baseband_len));
inst.SendScpi(sprintf(':TRAC:SEL %d', seg_num));
```

```
fprintf('Done!\n');

%% Write waveform to selected channel
fprintf('Downloading waveform to device.\n');
prefix = ':TRAC:DATA 0,';
res = inst.WriteBinaryData(prefix, baseband_inp);
assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Set voltage characteristics of channel
fprintf('Setting Voltage characteristics for CH %d.\n', chan_num);
inst.SendScpi(sprintf(':SOUR:FUNC:MODE:SEGM %d', seg_num));
inst.SendScpi(sprintf(':SOUR:VOLT %s', num2str(v_pp)));
inst.SendScpi(sprintf(':SOUR:VOLT:OFFS %s', num2str(v_offset))); %
    NOTE: This is not supported in DIRECT, AC Coupled

assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Use segment mode for baseband, assuming segment table is empty
   on startup
inst.SendScpi('FUNC:MODE:SEGM 1');
inst.SendScpi(':SOUR:VOLT MAX');

%% Turn on the output
inst.SendScpi(':OUTP ON');
fprintf('CH %d Output Enabled.\n', chan_num);

%% Turn off the output
inst.SendScpi(':OUTP OFF');
fprintf('CH %d Output Disabled.\n', chan_num);
```

```
%% Disconnect when done
fprintf('Disconnecting from Proteus\n');
inst.Disconnect();
```

B.3 Sinusoid with AM Modulated DUC

```
%% Script to utilize the DUC as a AM modulation b/w carrier
    and baseband

%% Parameters for function
% Waveform Parameters
base_f = 10E3;
base_vpp = 500-3;
base_samples = 1024; % This must be at least 64
base_phase = 0;

% Create a single period of the baseband sine
period = 1/base_f;
time = 0:(period/(base_samples-1)):period;
% Sine baseband
%baseband = (base_vpp/2)*sin(2*pi*base_f*time + base_phase);
baseband = zeros([1, length(time)]);
baseband(:) = base_vpp/2;
baseband_q = zeros([1, length(time)]);
baseband_q(:) = 0;

% Proteus connection parameters
ip = '192.168.20.5';
conn = 'LAN';
paranoia = 2;

% AWG options
```

```
samp_rate = 2.5E9;
dac_bits = 16;
interp = 4;
chan_num = 3;
seg_num = 1;
v_pp = 0.5;
v_offset = 0.0;

% DUC Parameters
NCO_f = 1.0E9;
NCO_phase = 0;
IQ_Mode = 1;

%% Convert baseband to bit value
[baseband_inp_I, baseband_len_I] = convert_wfm_to_bin(baseband,
    v_pp, dac_bits, 'uint16');

%% Create IQ waveform to transmit, Q1 will just be inverted I1
%[baseband_inp_Q, baseband_len_Q] = convert_wfm_to_bin(-1*baseband
    , v_pp, dac_bits, 'uint16');
[baseband_inp_Q, baseband_len_Q] = convert_wfm_to_bin(baseband_q,
    v_pp, dac_bits, 'uint16');

baseband_len = baseband_len_I+baseband_len_Q;
baseband_inp = zeros([1, length(baseband_inp_I)+length(
    baseband_inp_Q)], 'uint8');
idx = 1;
% This needs to be split by IMSB, QMSB, ILSB, QLSB, ...
for i = 1 : 2 : length(baseband_inp_I)
    baseband_inp(1, idx) = baseband_inp_I(1, i);
    baseband_inp(1, idx+1) = baseband_inp_Q(1, i);
    baseband_inp(1, idx+2) = baseband_inp_I(1, i+1);
    baseband_inp(1, idx+3) = baseband_inp_Q(1, i+1);
    idx = idx + 4;
```

```
end

%% Connect to device

[inst, admin, idnstr] = ConnecToProteus(conn, ip, paranoia);
fprintf('Connected to: %s\n', idnstr);

% Reset AWG to be safe
fprintf('Reseting AWG. \n');
inst.SendScpi('*CLS;*RST');

%% Set up AWG parameters
fprintf('Setting up AWG\n');
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(':TRAC:DEL:ALL');
inst.SendScpi(':TRAC:FORM U16');

% Channel / Segment Parmaeters
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':TRAC:DEF %d, %d', seg_num, baseband_len));
inst.SendScpi(sprintf(':TRAC:SEL %d', seg_num));

fprintf('Done!\n');

%% Write IQM Waveform
fprintf('Downloading IQM Information to device.\n');

inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':FREQ:RAST %s', num2str(2.5E9)));
inst.SendScpi(sprintf(':SOUR:INT X%s', num2str(interp)));
inst.SendScpi(':MODE DUC');
inst.SendScpi(':IQM ONE');
inst.SendScpi(sprintf(':NCO:CFR1 %d', NCO_f));
inst.SendScpi(sprintf(':NCO:PHAS1 %d', NCO_phase));
```

```
% Add 6dB modulation gain here
inst.SendScpi(':NCO:SIXD1 ON');

fprintf('Done!\n');

%% Write IQ waveforms to selected channel
fprintf('Downloading waveform to device.\n');
inst.SendScpi([':FREQ:RAST ' num2str(samp_rate)]);
prefix = ':TRAC:DATA 0,';
res = inst.WriteBinaryData(prefix, baseband_inp);
assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Set voltage characteristics of channel (in IQM 1 Channel & Seg
   3 is for baseband)
fprintf('Setting Voltage characteristics for CH %d.\n', 3);
inst.SendScpi(sprintf(':SOUR:FUNC:MODE:SEGM %d', 3));
inst.SendScpi(sprintf(':SOUR:VOLT %s', num2str(v_pp)));
inst.SendScpi(sprintf(':SOUR:VOLT:OFFS %s', num2str(v_offset))); %
    NOTE: This is not supported in DIRECT, AC Coupled

assert(res.ErrCode == 0);

fprintf('Done!\n');

% Use segment mode for baseband, assuming segment table is empty
   on startup
inst.SendScpi('FUNC:MODE:SEGM 1');
inst.SendScpi(':SOUR:VOLT MAX');

%% Turn on the output
inst.SendScpi(':OUTP ON');
```

```
fprintf('CH %d Output Enabled.\n', chan_num);

%% Turn off the output
inst.SendScpi(':OUTP OFF');
fprintf('CH %d Output Disabled.\n', chan_num);

%% Disconnect when done
fprintf('Disconnecting from Proteus\n');
inst.Disconnect();
```

B.4 Digitizer Visualization using DUC

```
    %% Script to use IQM Mode 1 to transmit an AM modulated
       waveform and visualize using the

%% Parameters for function
clear
% Waveform Parameters
base_f = 10E3;
base_vpp = 500E-3;
base_samples = 1024; % This must be at least 64
base_phase = 0;

% Create a single period of the baseband sine
period = 1/base_f;
time = 0:(period/(base_samples-1)):period;
baseband = zeros([1, length(time)]);
baseband(:) = base_vpp/2;
baseband_q = zeros([1, length(time)]);
baseband_q(:) = 0;

% Proteus connection parameters
ip = '192.168.20.5';
```

```
conn = 'LAN';
paranoia = 2;

% AWG options
samp_rate = 1E9;
dac_bits = 16;
interp = 4;
chan_num = 1;
seg_num = 1;
v_pp = 0.5;
v_offset = 0.0;

% DUC Parameters
NCO_f = 2.0E9;
NCO_phase = 0;
IQ_Mode = 1;

% Digitizer Options
dig_dev = 1;  % Options for this are 'DIG1' & 'DIG2'
dig_ch = 1; % Digitizer channel on hardware (CH1IN or CH2IN)
dig_samp_rate = samp_rate*2;  % This must be an integer
   submultiple of AWG sample rate
    % NOTE: This must be [1.6E9, 5.4E9] single mode, [800E6, 2.7E9
       ] dual
    % mode
dig_capture_length = 9600;   % Length of each frame
dig_num_frames = 1; % Number of frames to capture

%% Convert baseband to bit value
[baseband_inp_I, baseband_len_I] = convert_wfm_to_bin(baseband,
   v_pp, dac_bits, 'uint16');

%% Create IQ waveform to transmit, Q1 will just be inverted I1
```

```
[baseband_inp_Q, baseband_len_Q] = convert_wfm_to_bin(baseband_q,
    v_pp, dac_bits, 'uint16');

baseband_len = baseband_len_I+baseband_len_Q;
baseband_inp = zeros([1, length(baseband_inp_I)+length(
    baseband_inp_Q)], 'uint8');
idx = 1;
% This needs to be split by IMSB, QMSB, ILSB, QLSB, ...
for i = 1 : 2 : length(baseband_inp_I)
    baseband_inp(1, idx) = baseband_inp_I(1, i);
    baseband_inp(1, idx+1) = baseband_inp_Q(1, i);
    baseband_inp(1, idx+2) = baseband_inp_I(1, i+1);
    baseband_inp(1, idx+3) = baseband_inp_Q(1, i+1);
    idx = idx + 4;
end

%% Connect to device

[inst, admin, idnstr] = ConnecToProteus(conn, ip, paranoia);
fprintf('Connected to: %s\n', idnstr);

% Reset AWG to be safe
fprintf('Reseting AWG. \n');
inst.SendScpi('*CLS;*RST');

fprintf('Done!\n')

%% Set up AWG parameters
fprintf('Setting up AWG\n');
inst.SendScpi([':FREQ:RAST ' num2str(samp_rate)]);
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(':TRAC:DEL:ALL');
inst.SendScpi(':TRAC:FORM U16');
```

```
% Channel / Segment Parmaeters
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':TRAC:DEF %d, %d', seg_num, baseband_len));
inst.SendScpi(sprintf(':TRAC:SEL %d', seg_num));

fprintf('Done!\n');

%% Digitizer setup
fprintf('Setting up Digitizer to DIG%s and CH%s\n', num2str(
    dig_dev), num2str(dig_ch));

%inst.SendScpi(sprintf(':DIG DIG%s', num2str(dig_dev))); % Set
    digitzer device
inst.SendScpi(':DIG:MODE SINGLE');     % Assuming single mode for
    now
inst.SendScpi(':DIG:ACQ:FREE'); % Free memory
inst.SendScpi(sprintf('DIG:FREQ:RAST %s', num2str(dig_samp_rate)))
    ;  % Set digitizer sampling rate
inst.SendScpi(sprintf(':DIG:CHAN CH%s', num2str(dig_ch)));  % Set
    digitizer channel (corresponds to CH IN on hardware)
inst.SendScpi('DIG:CHAN:RANG HIGH');     % Assume digitizer set to
    500mVpp range
inst.SendScpi('DIG:CHAN:OFFS 0');    % Assume no DC offset

inst.SendScpi(sprintf(':DIG:TRIG:SOUR CH%s', num2str(dig_ch)));  %
     Set activate trigger to digitizer channel

fprintf('Done!\n')

%% Write IQM Waveform
fprintf('Downloading IQM Information to device.\n');

inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':FREQ:RAST %s', num2str(samp_rate)));
```

```
inst.SendScpi(sprintf(':SOUR:INT X%s', num2str(interp)));
inst.SendScpi(':MODE DUC');
inst.SendScpi(':IQM ONE');
inst.SendScpi(sprintf(':NCO:CFR1 %d', NCO_f));
inst.SendScpi(sprintf(':NCO:PHAS1 %d', NCO_phase));

% Add 6dB modulation gain here
inst.SendScpi(':NCO:SIXD1 ON');

fprintf('Done!\n');

%% Write IQ waveforms to selected channel
fprintf('Downloading waveform to device.\n');
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi([':FREQ:RAST ' num2str(samp_rate)]);
prefix = ':TRAC:DATA 0,';
res = inst.WriteBinaryData(prefix, baseband_inp);
assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Set voltage characteristics of channel (in IQM 1 Channel & Seg
   3 is for baseband)
fprintf('Setting Voltage characteristics for CH %d.\n', 1);
inst.SendScpi(sprintf(':SOUR:FUNC:MODE:SEGM %d', 1));
inst.SendScpi(sprintf(':SOUR:VOLT %s', num2str(v_pp)));
inst.SendScpi(sprintf(':SOUR:VOLT:OFFS %s', num2str(v_offset))); %
    NOTE: This is not supported in DIRECT, AC Coupled

assert(res.ErrCode == 0);

fprintf('Done!\n');
```

```
% Use segment mode for baseband, assuming segment table is empty
    on startup
inst.SendScpi('FUNC:MODE:SEGM 1');
inst.SendScpi(':SOUR:VOLT MAX');

%% Turn on the output
inst.SendScpi(':OUTP ON');
fprintf('CH %d Output Enabled.\n', chan_num);

%% Sample data through Digitizer
fprintf('Capturing frames through Digitizer\n')

inst.SendScpi(':DIG:CHAN:STATE ENAB'); % Enable digitizer channel
inst.SendScpi(sprintf(':DIG:ACQ:DEF %s, %s', num2str(
    dig_num_frames), num2str(dig_capture_length))); % Create memory
     space for digitizer
     % NOTE: This will create a total frame number of
         dig_capture_length *
     % dig_capture_length
inst.SendScpi(':DIG:ACQ:FRAM:CAPT:ALL');     % Capture all frames
    in sample
inst.SendScpi('DIG:ACQ:ZERO:ALL'); % Make sure memory is cleared
    before capture

inst.SendScpi(':DIG:INIT OFF'); % Make sure digitizer is off
read_data = zeros([1, dig_num_frames*dig_capture_length]); %
    Storage for read data
inst.SendScpi(':DIG:INIT ON'); % Turn on sampling of digitizer

% Read cycle
for i = 1 : 2*dig_num_frames*dig_capture_length
     res = inst.SendScpi(':DIG:ACQ:FRAM:STAT?');
     res = strtrim(res.RespStr);
     item = split(res, ',');
```

```
    item = str2double(item);
    % Item 2 being 1 signifies we are done
    if (length(item) >= 3 && item(2) == 1)
        break;
    end
    pause(0.1)
end

inst.SendScpi(':DIG:DATA:TYPE FRAM');     % Set capture type to
   frame data
inst.SendScpi(':DIG:DATA:SEL ALL');  % Select all frame data

res = inst.SendScpi(':DIG:DATA:SIZE?');  % Get size of digitizer
   read data in bytes
n_bytes = strtrim(res.RespStr); % Get rid of any whitespace

samp = inst.ReadBinaryData(':DIG:DATA:READ?', 'uint16');     % Read
    binary data NOT USING DDC
    % NOTE: This will be 12-bit samples in a 16-bit size
samp = int16(samp) - 2048;  % Zero level

inst.SendScpi(':DIG:INIT OFF');

fprintf('Done!\n');

%% Plot sampled data in terms of ADC value (add FFT here also)
figure;
plot(samp);
xlabel('Sample number')
ylabel('ADC value');

%% Turn off the output
inst.SendScpi(':OUTP OFF');
fprintf('CH %d Output Disabled.\n', chan_num);
```

```
%% Disconnect when done
fprintf('Disconnecting from Proteus\n');
inst.Disconnect();
```

B.5 Amplifier AWG Ramping

```
    %% This project is designed to test the ZX60-83LN12+ LNA
% Some notes:
    % This is NOT tunable through input voltage (requires 12V DC)
    % Gain will be around 22 dB (if testing between 0.5 to 2 GHz)
    % We need to find an operating point that is okay for this
        % Testing design dependent
    % Please see the 600mhz_... images to see an example of lna
       effect on a
        % pll generated waveform
    % Also please see the ZX60-83LN12+_SPARAMS image to get an
       idea of the
        % differnet s-param values

% For this code we can just manually set the output voltage of the
    AWG to
% safely check the output on the oscilloscope. For this let's just
    focus on
% using the DUC with an AM modulation (IQ Mode 1)

%% AWG Parameters
ip = '192.168.20.5';
conn = 'LAN';
paranoia = 2;

samp_rate = 2.5E9;
dac_bits = 16;
```

```
interp = 4;
chan_num = 2;
seg_num = 1;
awg_amp = 0.5;  % vpp
awg_offset = 0.0;

% DUC Parameters
NCO_f = 500E6;
NCO_phase = 0;
IQ_Mode = 1;

%% Input wave setup
num_samples = 1024;
baseband = zeros([1, 1024]);
baseband_q = zeros([1 1024]);

% Setup baseband (DC)
baseband(:) = awg_amp;
baseband_q(:) = awg_amp;

%% Convert baseband to bit value
[baseband_inp_I, baseband_len_I] = convert_wfm_to_bin(baseband,
    awg_amp, dac_bits, 'uint16');

%% Create IQ waveform to transmit, Q1 will just be inverted I1
%[baseband_inp_Q, baseband_len_Q] = convert_wfm_to_bin(-1*baseband
    , v_pp, dac_bits, 'uint16');
[baseband_inp_Q, baseband_len_Q] = convert_wfm_to_bin(baseband_q,
    awg_amp, dac_bits, 'uint16');

baseband_len = baseband_len_I+baseband_len_Q;
baseband_inp = zeros([1, length(baseband_inp_I)+length(
    baseband_inp_Q)], 'uint8');
idx = 1;
```

```
% This needs to be split by IMSB, QMSB, ILSB, QLSB, ...
for i = 1 : 2 : length(baseband_inp_I)
    baseband_inp(1, idx) = baseband_inp_I(1, i);
    baseband_inp(1, idx+1) = baseband_inp_Q(1, i);
    baseband_inp(1, idx+2) = baseband_inp_I(1, i+1);
    baseband_inp(1, idx+3) = baseband_inp_Q(1, i+1);
    idx = idx + 4;
end

%% Connect to the AWG
[inst, admin, idnstr] = ConnecToProteus(conn, ip, paranoia);
fprintf('Connected to: %s\n', idnstr);

% Reset AWG to be safe
fprintf('Reseting AWG. \n');
inst.SendScpi('*CLS;*RST');
fprintf('Done!\n');

%% Set up AWG parameters
fprintf('Setting up AWG\n');
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(':TRAC:DEL:ALL');
inst.SendScpi(':TRAC:FORM U16');

% Channel / Segment Parmaeters
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':TRAC:DEF %d, %d', seg_num, baseband_len));
inst.SendScpi(sprintf(':TRAC:SEL %d', seg_num));

fprintf('Done!\n');

%% Write IQM Waveform
fprintf('Downloading IQM Information to device.\n');
```

```
inst.SendScpi(sprintf(':INST:CHAN %d', chan_num));
inst.SendScpi(sprintf(':FREQ:RAST %s', num2str(2.5E9)));
inst.SendScpi(sprintf(':SOUR:INT X%s', num2str(interp)));
inst.SendScpi(':MODE DUC');
inst.SendScpi(':IQM ONE');
inst.SendScpi(sprintf(':NCO:CFR1 %d', NCO_f));
inst.SendScpi(sprintf(':NCO:PHAS1 %d', NCO_phase));

% Add 6dB modulation gain here
inst.SendScpi(':NCO:SIXD1 ON');

fprintf('Done!\n');

%% Write IQ waveforms to selected channel
fprintf('Downloading waveform to device.\n');
inst.SendScpi([':FREQ:RAST ' num2str(samp_rate)]);
prefix = ':TRAC:DATA 0,';
res = inst.WriteBinaryData(prefix, baseband_inp);
assert(res.ErrCode == 0);

fprintf('Done!\n');

%% Set voltage characteristics of channel (in IQM 1 Channel & Seg
   3 is for baseband)
fprintf('Setting Voltage characteristics for CH %d.\n', 3);
inst.SendScpi(sprintf(':SOUR:FUNC:MODE:SEGM %d', 3));
%inst.SendScpi(sprintf(':SOUR:VOLT %s', num2str(v_pp)));
%inst.SendScpi(sprintf(':SOUR:VOLT:OFFS %s', num2str(v_offset)));
   % NOTE: This is not supported in DIRECT, AC Coupled

assert(res.ErrCode == 0);

fprintf('Done!\n');
```

```
% Use segment mode for baseband, assuming segment table is empty
   on startup
inst.SendScpi('FUNC:MODE:SEGM 1');
inst.SendScpi(':SOUR:VOLT MAX');

%% Ramp the channel voltage within specified range
% NOTE: This is Vpp in Volts
awg_ramp_start = 10E-3;
awg_ramp_change = 10E-3;
awg_ramp_end = 400E-3;
wait_time = 1;  % in seconds

fprintf('Turning on CH %d Output for ramp\n', chan_num);
inst.SendScpi(':OUTP ON');

for volt = awg_ramp_start : awg_ramp_change : awg_ramp_end
    fprintf('Channel Vpp: %f Volts\n', volt)
    inst.SendScpi(sprintf(':SOUR:VOLT %s', num2str(volt)));
    pause(wait_time);
end

fprintf('Turning off CH %d Output, ramp completed\n', chan_num);
inst.SendScpi(':OUTP OFF');

%% Turn on the output
inst.SendScpi(':OUTP ON');
fprintf('CH %d Output Enabled.\n', chan_num);

%% Turn off the output
inst.SendScpi(':OUTP OFF');
fprintf('CH %d Output Disabled.\n', chan_num);

%% Disconnect when done
fprintf('Disconnecting from Proteus\n');
```

```
inst.Disconnect();
```