RF Direction of Arrival Estimation Using a Linear Antenna Array

Final Report

O. Johnson, E. Pipe, L. Carslake and W. Sam

06 May 2019

Abstract

Modern communication and sensing systems are becoming increasingly dependant on knowing the position of a electromagnetic source. This information can be used to optimise data throughput of a mobile communications system, or successfully track and locate objects in an increasingly busy electromagnetic environment. Phased Array Antennas are becoming increasingly utilised in small, low cost systems incorporating higher levels of DSP in commercially available products.

This report details the design, construction and testing of a full phased array system which demonstrates the application of several different Direction of Arrival (DOA) calculation methods. The completed system includes a fully operational Radio Frequency (RF) data acquisition system and provides a platform for further research into array signal processing. Several different algorithms have been demonstrated to operate successfully with this system in a reflective environment.

Contents

1 Introduction . . . 3

2 Literature review . . . 3
2.1 Phased Antenna Arrays . . . 3
2.2 Direct Conversion IQ Radio Receiver . . . 4
2.3 Direction of Arrival (DOA) algorithms . . . 4
2.4 Digital Hardware . . . 7

3 Aims and Objectives . . . 8

4 Methodology . . . 9
4.1 System Design . . . 9
4.2 RF Front End . . . 9
4.2.1 Antennas . . . 9
4.2.2 Low Noise Amplifiers (LNAs) . . . 13
4.2.3 AD8348 Demodulator Stage . . . 17
4.2.4 Local Oscillator (LO) . . . 20
4.2.5 Anti-Aliasing Filters . . . 24
4.2.6 Analog to Digital Conversion . . . 27
4.2.7 Development Boards . . . 29
4.2.8 Final PCB Design . . . 31
4.3 FPGA Digital System . . . 32
4.3.1 Overview . . . 32
4.3.2 Microblaze . . . 34
4.4 ADC Interface . . . 35
4.4.1 FIFO Buffer . . . 37
4.4.2 SRAM Storage . . . 37
4.5 User Interface . . . 38
4.6 DOA Estimation . . . 39
4.6.1 Simulated receiver signals . . . 39
4.6.2 MATLAB DOA Estimation Implementation . . . 40
4.6.3 Simulink DOA Estimation Implementation . . . 40
4.6.4 System Integration . . . 43

4.6.5 Calibration of the receiver 43

5 Results and Discussion 45
5.1 Design and Construct Quadrature RF receiver 45
5.2 Design, Simulate and construct phased array antenna 47
5.3 Design and simulate DOA algorithm 48
5.3.1 Synthesise quadrature test signals 48
5.3.2 Simulate DOA algorithm with artificial phase difference and noise (MATLAB) . 51
5.3.3 Build and Implement Simulink model of DOA algorithm 53
5.4 Create GUI for presenting data to user 54
5.4.1 Read data from digital hardware to PC 54
5.4.2 Integration of DOA algorithm with GUI 54
5.4.3 Produce DOA result when requested by user 54
5.5 Overall System Results for DOA Estimation 55
5.5.1 Hardware Cascade Analysis 55

6 Conclusion 55

7 Recommendations and Future Work 55
7.1 Fine tune circuitry 55
7.2 Printed Antenna 56
7.3 Automatic Gain Controll (AGC) 56
7.4 Increase the number of elements in the array 56
7.5 Extra front end downconversion for higher RF operation - 4G/WiFi 56
7.6 Automatic Calibration for range of input signals 56
7.7 Use a faster method of off loading data from the FPGA 56
7.8 Add advanced triggering mechanisms to the FPGA 57
7.9 Implement Algorithm directly on FPGA 57

8 Achievement Evaluation 57

9 Appendices 61

1 Introduction

By measuring the difference in phase of an incoming signal from multiple receivers a distance apart, it is possible to calculate the Angle of Arrival (AOA) from which the signal came. There are various algorithms that can calculate the AOA, each applicable for different types of environments, but a constant through every method is the need to capture the signal. In order to capture AOA information about an electromagnetic signal, an antenna array is needed. This consists of multiple antennas in a known and specific geometry.

Knowing the direction from which a signal source is coming from is useful for tracking and locating objects. This sort of signal processing has been used extensively in Radar, however more recently with the advent of 5G, location and tracking of a transceiver is being used to optimise the transmission and reception of signals for mobile users.

In this project the design and construction of a 4-element antenna array and RF receiver, including digitization and signal processing, to calculate the AOA of an incoming wave is demonstrated. A custom RF front end passes incident signals to an ADC, where it is then sampled and passed through an FPGA into a Matlab Graphical User Interface (GUI) where a number of different algorithms including beamscan, Minimum Variance Distortionless Response (MVDR) and QR decomposition methods are used to successfully estimate the Direction of Arrival (DOA).

For this specific system, a receive chain composed of custom antennas, a transistorised LNA, a 50MHz-1GHz IQ demodulator IC, active filtering for anti-aliasing, diode clipper circuits for ADC protection, and digitisation in ADCs. Once digitised, an FPGA offers flexibility and performance to implement DOA algorithms, storage of samples and transmission to the GUI. From the GUI the user is able to control the system and view the received signals.

2 Literature review

2.1 Phased Antenna Arrays

For a source which is a distance away from the antenna array which is significantly larger than the element separation distance, the source is defined as being in the Far-Field [1]. This implies that the antennas do not perceive near-field coupling (only radiated fields) and the spherical wavefronts of the propagating waves become so large that they can be approximated as straight lines. This leads to the condition illustrated in Figure 1.

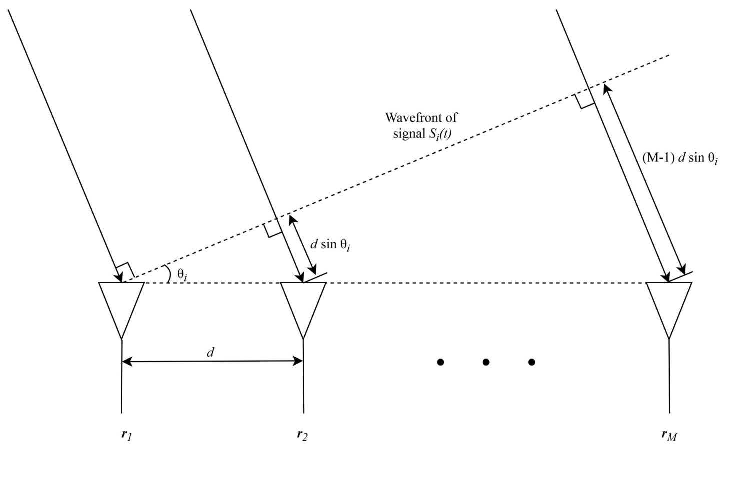


Figure 1: Uniform Linear Array (ULA).

The diagram in figure 1 shows the derivation of values for signals received by an M-element ULA with spacing of d between elements. The incoming signal wavefront is denoted as $S_i(t)$ and r_1, r_2 and r_M are the signals received at each sensor. A given signal located at θ_i from the on-axis position of the antennas results in a phase shift in signal received by each element of the array, as the signal at each subsequent sensor travels a further

$d\,sin\,\theta_i$. Hence the maximum extra distance travelled is $(M-1)\,d\,sin\,\theta_i$. From this diagram, it is possible to see that a signal placed perpendicular to the array will have no phase shift between the incoming signals at each sensor and that by manipulating the data received, it is possible to determine the direction of arrival (DOA) angle [3].

A simple method for determining the angle of arrival is the manipulation of the phased array geometry. With a known distance between elements, the time delay can be found:

$$s = d\,sin\theta \qquad s = \frac{v}{t} \qquad t = \frac{d\,sin\theta}{f\,\lambda} \qquad\qquad d = \frac{\lambda}{2} \quad \rightarrow \quad t = \frac{sin\theta}{2f}$$

If the first signal occurs at time $t_1 = 0$, The phase shift of a signal can be written as:

$$\phi = 2\pi\frac{t_2 - t_1}{T} = 2\pi f t$$

The the two equations can be related and rearranged for angle θ:

$$\phi = 2\pi f\left(\tfrac{sin\theta}{2f}\right) = \pi sin\theta \quad \rightarrow \qquad \theta = sin^{-1}\left(\tfrac{\phi}{\pi}\right) \tag{1}$$

2.2 Direct Conversion IQ Radio Receiver

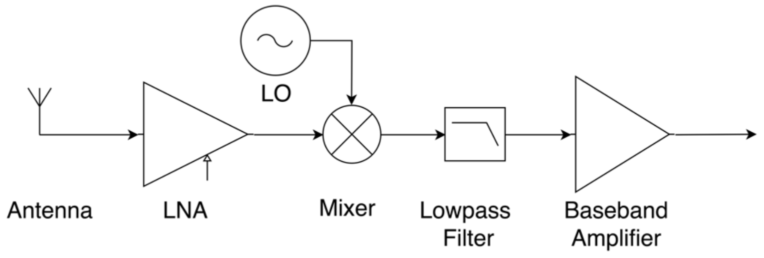


Figure 2: Example direct conversion receiver architecture. RF signal is amplified in a low noise amplifier (LNA), mixed to to baseband, low-pass filtered to prevent aliasing and then amplified and buffered at baseband to drive an ADC..

A direct conversion IQ receiver system is shown in figure 2. The receiver features a single mix down stage, digitising at IF or baseband. This provides a compromise between a standard superheterodyne receiver, and a fully digitised front end. Through the use of a single mix down stage, there is much less complicated analog circuitry than in a superhet, whilst being able to use a cheaper, and less power hungry ADC than in a directly sampling a signal at RF [2].

2.3 Direction of Arrival (DOA) algorithms

The received received signals at the ULA (as shown in figure 1) can be described through the equation [3, 4]:

$$\mathbf{r}(t) = \mathbf{A}\mathbf{S}_i(t) + \mathbf{n}(t) \tag{2}$$

Although, DOA angle estimation is reliant on sampling of incoming signals at selected time instants t_n ($n = 1, 2, ... N$), so this can be re-written as:

$$\mathbf{r}(n) = \mathbf{A}\mathbf{S}_i(n) + \mathbf{n}(n)$$

Where $\mathbf{S}_i(n)$ is the incoming signal, $\mathbf{n}(n)$ is the noise of the signal, M is the number of sensors or elements in the antenna array, and $\mathbf{A}$ is the array manifold made up of steering vectors which correspond to the L the number of sources:

$$\mathbf{A} = [\, a(\theta_1),\ a(\theta_2)\ ...\ a(\theta_L)\,] \tag{3}$$

Thus when $L = 1$:

$$\mathbf{A} = [\, a(\theta_1)\,]$$

The steering vector for a single source is:

$$\mathbf{a}(\theta) = [\, 1e^{jk}, 1e^{2jk}\ ...\ 1e^{(M-1)jk}\,] \tag{4}$$

where $k = \frac{2\pi d}{\lambda} sin\,\theta$. If $d = \frac{\lambda}{2}$ then $k = \pi sin\,\theta$. It should also be noted that this aplies for the evaluation of angles between $\frac{\pi}{2}$ and $\frac{-\pi}{2}$. Since $sin\,\theta = cos\,(\frac{\pi}{2} - \theta)$, $k = -\pi cos\,\theta$ for the evaluation of angles between 0 and π.

For a single source, the array response matrix is simply the transpose of the steering vectors:

$$\mathbf{a}(\theta) = [\, 1e^{jk}, 1e^{2jk} \;...\; 1e^{(M-1)jk} \,]^T$$

which can be calculated through π radians to characterise response of the array.

DOA algorithms often use higher order statistics to the within the processing of the signal data. This assumes that received signal and received noise are independent and have a mean of zero. This results in the spatial array correlation matrix given by:

$$\mathbf{R} = E\{\mathbf{r}(n)\mathbf{r}(n)^H\} = \mathbf{ASA}^H + \mathbf{R}_n \tag{5}$$

If $\mathbf{R}_n$ is considered to be spatially white, then $\mathbf{R}_n = \sigma^2\,\mathbf{I}$, with σ^2 representing the noise level, and $\mathbf{I}$ is the identity matrix with dimensions $M \times M$. Here $\mathbf{S}$ is the received incoming signal and H is the complex conjugate transpose or Hermitian transposition.

For a S number of snapshot instances, the correlation matrix of $\mathbf{R}$ can be simply estimated:

$$\hat{\mathbf{R}} = \frac{1}{S}\sum_{s=1}^{S} \mathbf{r}(n)\mathbf{r}(n)^H$$

The minimum number of snapshots corresponds to the number of elements M and the higher the number of snapshots used, the more accurate the estimation.

The most fundamental method of DOA angle estimation is a conventional (delay and sum) beamformer [4], which uses calculated weights to create a beam that can be steered through spatial spectrum and impart various values corresponding to the DOA angle [5].
The limitation of this method is the array aperture since the resolution of the beamformer is inversely proportional.
The beamformer power spectrum that shows the spectral density of the received signals in the spatial domain is given by:

$$P_{BS}(\theta) = \frac{1}{N}\sum_{n=1}^{N} |\mathbf{a}(\theta)^H \mathbf{r}(t)|^2 = \mathbf{a}(\theta)^H\,\hat{\mathbf{R}}\,\mathbf{a}(\theta) \tag{6}$$

Minimum Variance Distortionless Response (MVDR) estimation assumes that signal and noise data is uncorrelated, hence the variance of the system can be simplified to the sum of signal and noise variances and minimised to suppress noise [6]. This method improves on the resolution of the conventional beam former because of the sharper peak outputs[7, 8]. The MVDR spectrum is given by:

$$P_{MVDR}(\theta) = \frac{1}{\mathbf{a}(\theta)^H\,\hat{\mathbf{R}}^{-1}\,\mathbf{a}(\theta)} \tag{7}$$

Another method method is linear decomposition (QR decomposition) estimation which makes use of linear decomposition to produce the signal and noise sub spaces. Unlike MUSIC or ESPRIT algorithms, this method does not use computational complex eigenvalue decomposition (EVD) which makes calculation more feasable on limited processing hardware [9].

For covariance matrix estimate $\hat{\mathbf{R}}$:

$$\hat{\mathbf{R}} = Q\,R = [Q_s\; Q_n]\begin{bmatrix} R_L \\ O \end{bmatrix} \tag{8}$$

Where Q_s is the column vector of the signal space, and Q_n is the column vector of the noise space. The upper triangular matrix R_L represents the signal space which corresponds to the size (L x M). O is a null matrix of size (M - L) x M. The orthogonality of the Q matrix (in that $Q\,Q^H = Q^H\,Q = I$, and) results in:

$$Q^H\,\hat{\mathbf{R}} = Q^H\,Q\,R = \begin{bmatrix} Q_s \\ Q_n \end{bmatrix} [Q_s\;Q_n] \begin{bmatrix} R_L \\ O \end{bmatrix} \rightarrow \begin{bmatrix} Q_s^H\,\hat{\mathbf{R}} \\ Q_n^H\,\hat{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} R_L \\ O \end{bmatrix} \tag{9}$$

Substituting in equation (4) for $\hat{\mathbf{R}}$, and for uncorrelated sources, A is full rank and S is non-singular:

$$Q_n^H\,\hat{\mathbf{R}} = Q_n^H\,A\,S\,A^H = O \qquad\qquad Q_n^H\,A = O$$

The steering vectors $a(\theta)$ that are orthogonal to the noise space Q_n are used to reveal the direction of arrival. Therefore, the angle can be found using a peak search:

$$P_{QR}(\theta) = \frac{1}{||Q_n^H\,A(\theta)||} = \frac{1}{A^H(\theta)\,Q_n^H\,Q_n\,A(\theta)} \tag{10}$$

A similar method to the QR peak search method is the Multiple Signal Classification (MUSIC) technique. This requires eigenvalue decomposition of the covariance matrix, which can be calculated using the eigenvalues and eigenvectors [10] according to :

$$A^H v_i = 0 \qquad\qquad i = (L+1),\,(L+2)\,...\,M$$

These eigenvectors can then be concatenated to make up the the noise space E_n:

$$E_n = [V_{L+1}, V_{L+2}\,...\,V_M] \tag{11}$$

Then, perform a peak search for the angle θ:

$$P_{MUSIC}(\theta) = \frac{1}{||E_n^H\,A(\theta)||^2} = \frac{1}{A^H(\theta)\,E_n\,E_n^H\,A(\theta)} \tag{12}$$

The rotation invariance of a decomposed matrix can also be explored using the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT). This method uses the two subspaces of noise and space to solve eigenvalues to form a matrix that relates the two [3, 4]. A simple method of doing this is by splitting up the array into two which are shifted by one. The selection can be written as:

$$Y_1 = [i_1, i_2\,...\,i_{M-1}, 0] \qquad\qquad Y_2 = [0, i_2, i_3\,...\,i_M]$$

$$Y_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad\qquad Y_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the vector will correspond to the size of the array, with the first or last element being null. A_i will then be equal to 1. This selects the response matrices:

$$A_1(\theta) = Y_1 A(\theta) \qquad\qquad A_2(\theta) = Y_2 A(\theta)$$

The shift invariant property of the sub arrays can be written as:

$$A_2(\theta) = A_1(\theta)\,\Phi \tag{13}$$

The array responses and signal eigenvectors span the same subspace, related by a non-singular linear transform T, resulting in:

$$U_S = A(\theta)\,T \tag{14}$$

Combining with the selection matrices and using the two sub matrices together:

$$U_{S1} = A_1(\theta)T \qquad\qquad U_{S2} = A_2(\theta)T$$

$$T\Psi T^{-1} = \Phi \tag{15}$$

With Ψ being similar to the matrix Φ, the eigen values will be the same. Finding the least squares solution to equation (15) and calculating the eigenvalues ($e_i = e_1, e_2\,...$) of this solution results in the DOA estimate:

$$\theta_i = sin^{-1}\left(\frac{angle(e_i)}{2\pi\,d}\right) \tag{16}$$

The benefit of ESPRIT is that it does not require the searching over a range of θ which results in a less computationally intensive calculation [7, 11]. However, some of the operations required to estimate the angle of arrival could be more difficult to implement depending on which platform is being used.

2.4 Digital Hardware

The digital hardware within the receiver focuses on the acquisition and processing of the demodulated signal. Acquisition is achieved through the use of an analogue to digital converter (ADC) connected to specialised processors such as a CPU or FPGA. Within these processors, transformations and operations of the digitised signal are performed which implement the signals processing tasks required.

Various types of ADCs are available with the key parameters being speed, sampling rate and resolution. For communication system applications current ADC technology provides a base for sampling at resolutions between 8-12 bits, at sampling rates of Ks/s to 100's Ms/s for an acceptable price. The most common technology for the high-speed ADCs is successive approximation (SAR), moving towards pipelined and flash technologies for higher speed applications.

Current research has focused on improving these 3 parameters, with an example of the current state of the art being a 56Gs/s Time-interleaved ADC with an ENOB of 5.7 bits by K. Sun et al [12]. This design demonstrates the tradeoff between sampling rate and resolution encountered when designing ADCs. An application of high-speed ADCs is the direct sampling of the RF signal without down conversion, affording the advantages of DSP to more of the signal chain, however, designs using this technique are currently restricted due to the limited development and high price of high-speed ADCs.

Digital signals processing (DSP) offers many advantages over analogue processing. DSP can offer great accuracy, repeatability and flexibility as operations are performed mathematically compared to analogue processing which relies on the tolerance of circuit components to achieve these characteristics [13]. For DSP design two processors are focused on the single-core microprocessor architecture and the field programmable gate array (FPGA). The microprocessor performs a series of sequential operations chosen from a set of instructions determined during the design of the device. The FPGA however, uses an array of logic blocks that are connected at the time of programming allowing for greater flexibility and the use of real-time sequential operations.

The digital acquisition system must interface with further hardware that lets the user visualise the signals received and perform additional processing. Many communication protocols exist that allow the transfer between devices, the most common in order of increasing bandwidth are Serial over USB, USB (universal serial bus) and Ethernet (802.11 10/100BASE-TX, etc.). In general, as the bandwidth of the interface increases as does the complexity and overheads. For example, implementing serial over USB is easily achieved as many devices contain hardware and software libraries to implement this, whereas Ethernet requires the implementation of an IP stack to handle the state of connections and sent packets.

3 Aims and Objectives

The following Aims and objectives were devised using the initial project specification document [14]. The project will be deemed a success if these aims are achieved, the achievement of an aim is defined by completion of each objective within that aim. The aims and objectives are listed below:

1. Design and Construct Quadrature RF receiver with operational frequency range 50MHz - 1GHz.
    - Design of supporting circuitry for AD8348 IC.
    - Design a baseband filter network.
    - Characterise receiver properties on RF test equipment.
    - SMA connectors for RF inputs.
    - SPI control of ADF4351 oscillator.
    - Integration of ADF4351 into final PCB.
2. Design, Simulate and construct phased array antenna.
    - Design appropriate antennas.
    - Simulate antenna design.
    - Construct and test antennas with VNA.
    - Design LC antenna matching network.
    - Test antennas with receiver system.
3. Design and Simulate DOA algorithm.
    - Synthesise quadrature test signals
    - Simulate DOA algorithm with artificial phase difference and noise.
    - Build simulink model for chosen algorithm
4. Implement DOA algorithm on FPGA hardware
    - Convert Simulink model to device specific code
    - Test and debug hardware using synthetic signals
    - Interface with receiver
5. Create GUI for presenting data to user
    - Read data from digital hardware to PC
    - Produce DOA result when requested by user.

4 Methodology

4.1 System Design

Figure 3 describes the high-level system which has been implemented in this project. It is composed of a 4-element IQ receiver array which is sampled at 500KSPS before being read into an FPGA which processes and transfers the samples to a Matlab GUI over a UART bus where a user can then run several different DOA estimation algorithms.

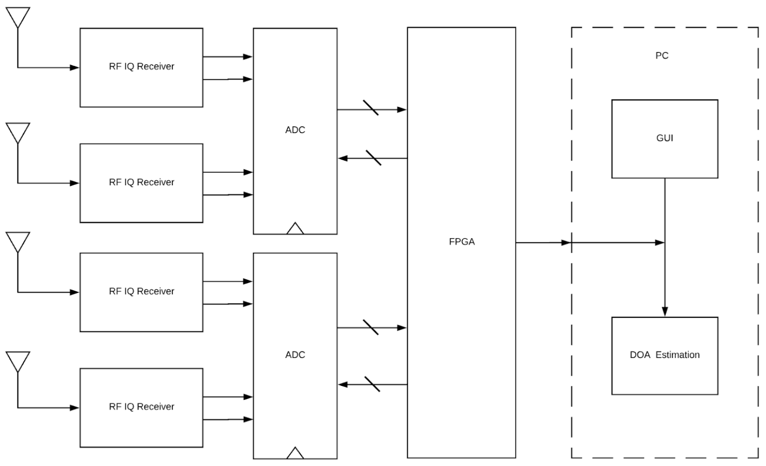


Figure 3: High-level system diagram showing the major function blocks of the system.

4.2 RF Front End

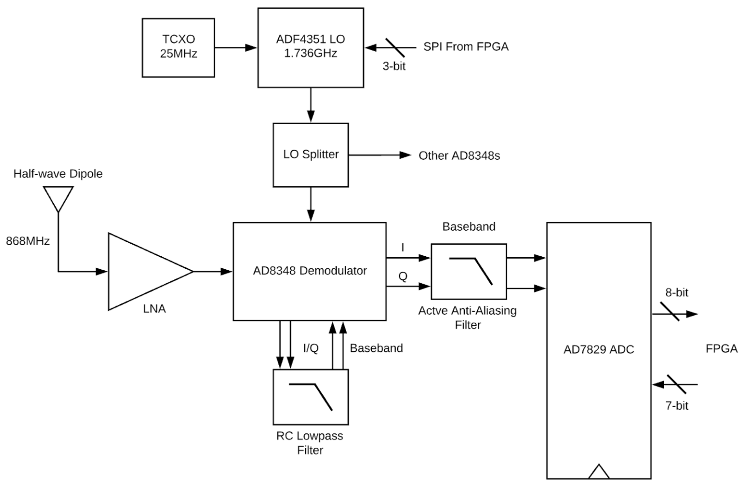


Figure 4: System diagram showing only one receiver module. The IQ signals at the input to the anti-aliasing filter are differential signals

A single RF IQ receiver block from figure 3 is then expanded in figure 4 which describes the systems within this block. It includes: an 868MHz half-wave dipole antenna for transmission reception, a Low Noise Amplifier (LNA), an AD8348 IQ demodulator, an IQ RC low-pass filter (to reduce the power flow into the Fixed Gain Amplifier (FGA) inside the IC), an IQ active anti-aliasing filter circuit and finally an AD7829 ADC. The ADF4351 drives the demodulators at 1.736GHz via a splitter network and this IC is controlled by the FPGA via SPI.

4.2.1 Antennas

To detect 868MHz transmissions effectively, the phased array needs to be composed of suitable antennas. The desired characteristics were omni-directionality (to give the array the potential for constant main beam intensity for any beam angle selected) and a small cross sectional area in the plane of interest (to enable the antennas to resemble point receivers).

Since up to 8 of these were required, it was concluded that they posed a significant cost when several antenna topologies are relatively simple to design and construct and it was decided to explore these as an alternative to purchasing antennas. To meet the desired characteristic above while maintaining simplicity and ease of implementation, two antenna topologies were investigated: a spiral monopole and dipole.

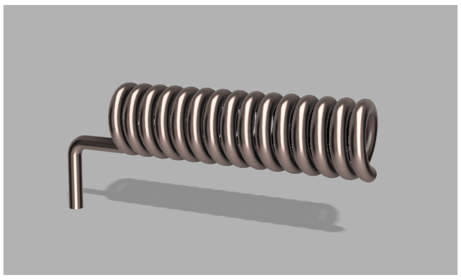


Figure 5: Fusion 360 render of the monopole design.

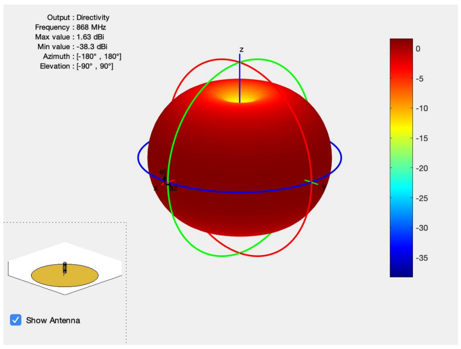

Output : Directivity
Frequency : 868 MHz
Max value : 1.63 dBi
Min value : -38.3 dBi
Azimuth : [-180° , 180°]
Elevation : [-90° , 90°]

Figure 6: Simulated monopole radiation pattern, showing normal mode operation as desired (onmi-directional).

4.2.1.1 Monopole Design,Simulation and Testing Since many commercial antennas are helical monopoles, it was concluded that this antenna topology was worth investigating. A well documented helical monopole design was then found and replicated [15]. The design features 13 and $\frac{3}{4}$ turns, a wire diameter of 1mm and a coil diameter of 5mm. Figure 5 illustrates the design used.

Using Matlab's Antenna Toolbox, the monopole was simulated to obtain its radiation pattern to ensure the design had the correct radiation pattern (normal mode operation not end-fire/axial). The result is reproduced in figure 6.

Once constructed, the prototype helical antenna was fitted onto an SMA connector with a large ground plane connected underneath and return loss measurements were taken using a VNA which revealed a peak of 6dB at the correct frequency of 868MHz. However, this was significantly lower than the return loss claimed in [15] (13.8dB). Since the dipole prototype performed better (see next section), and since this area of work was not in the specification, it was decided to continue forward with the dipole design only.

4.2.1.2 Dipole

4.2.1.2.1 Design

The length of an idealised hertzian half-wave dipole is given as $\frac{\lambda}{2}$ [1], which for the design frequency of 868MHz provided an antenna length of 17.3cm. The hertzian half-wave dipole is an approximation for the resonant frequency however, as it assumes an infinitesimally thin wire is used. Using Matlab's Antenna Toolbox, return loss simulations revealed the resonant frequency of the 17.3cm long half-wave dipole was closer to 810MHz when 1mm diameter copper wire was used (available in the University Stores). After iteratively tweaking the antenna length and running return loss simulations, the correct resonant frequency was achieved for a total wire length of 16.2cm. While physically much larger than the monopole design in length, the azimuth cross-sectional area is in fact less than the monopole design since it is equal only to the width of the copper wire. This makes it resemble the desired point source characteristic more closely.

4.2.1.2.2 Simulations

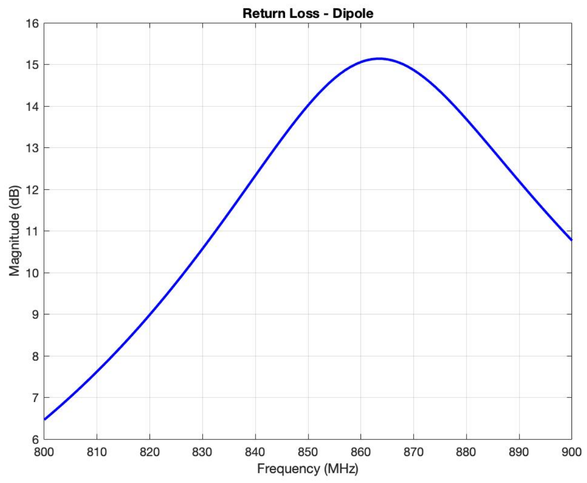


Figure 7: Return loss simulation result of the half-wave dipole antenna design.

The results of the return loss simulations conducted in Matlab for the 16.2cm long and 1mm thick copper wire dipole are provided in figure 7:

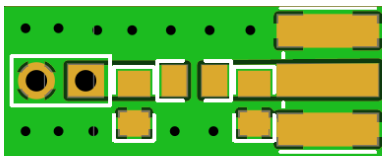


Figure 8: Antenna PCB design (2nd layer is a ground plane). The through hole pads on the left form the mounts for the copper wire dipole antenna. The output of the antenna is then fed through the matching π-network before ending with an SMA connector. Coplanar Waveguide was used for the positive trace

With a peak of over 15dB, this indicated that the dipole would outperform the monopole design in terms of its sensitivity. The radiation pattern was also simulated and verified to be omnidirectional with a peak directivity of 2.1dBi, which was expected since this is a characteristic feature of half-wave dipole antennas [1].

4.2.1.2.3 Prototype Testing

To verify the simulation results before, a prototype was constructed using the 1mm thick enameled copper wire from the University Stores directly onto an SMA connector. A VNA was then used to measure the return loss which was found to be -12dB which was significantly larger than that recorded for the monopole design. It was therefore decided that the dipole would be more suitable to be used in the array. The radiation pattern was not investigated since dipole radiation patterns are well categorised and simple and an anechoic chamber with a near field scanner for a full test were not readily available.

4.2.1.2.4 Dipole Mount Design

Since it is preferable to have a set of antennas with similar responses, a consistent construction method was investigated. Since University bought PCBs were free up to £80 (paid for by the University), it was decided that using a PCB for the copper wire mounting was a highly reproducible and cost effective method. In addition, this allowed for a design to incorporate pads for a π-network (on which an L-pad match can also be made) which could then be used for impedance matching. Figure 8 shows the PCB design.

4.2.1.2.5 Matching

Once the PCB antennas had been constructed and verified to have the correct resonant frequency, return loss measurements taken on a Vector Network Analyser (VNA) were taken in R+jX format to obtain the input impedance of the antennas. This was carried out with the antenna placed inside a small anechoic chamber constructed from spare Radio Absorbing Material (RAM) in the laboratory. Table 1 and figure 9 show the measured input impedances for the 4 antennas to be used in the array.

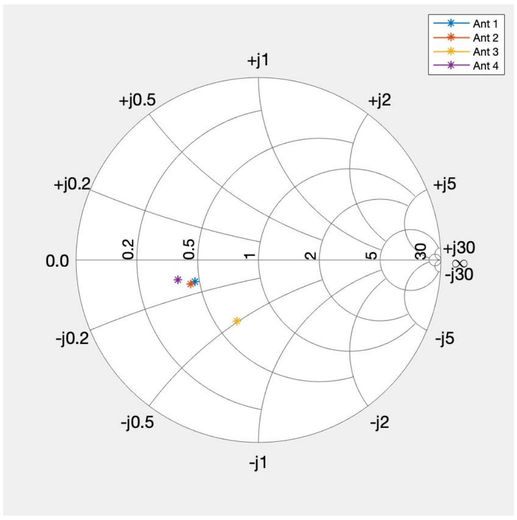


Figure 9: Smith chart plot of the unmatched antenna input impedances.

	Impedance	
	Re(Z)	Im (Z)
Antenna 1	23.6	-6.5
Antenna 2	22.3	-6.9
Antenna 3	32.1	-24.7
Antenna 4	19.0	-5.2

Table 1: Recorded input impedance of the half-wave dipole antennas.

Smith charts were then used to determine the required series/shunt capacitance/inductance required to match the antenna to a 50Ω system at 868MHz using an L-pad topology. Because the measured input impedances were relatively similar, the required matching networks were designed to be the same and applied to all the antennas. This reduced the need to buy specific components for each one but still meant the antennas would all still be matched to be close to 50Ωs. The optimal matching network was calculated to be a series inductance of 5.6nH and a shunt capacitance of 3.9pF (going from the antenna towards the SMA connector). This match was then verified using Telestrian (an online matching network calculator) [16]. It should be noted that LC matching circuits do not produce a broadband match [17] and thus provide early bandpass filtering on the received RF signal. Once the components were then placed on the antenna PCBs, more return loss measurements were obtained to verify the input impedance match. It was found however that the matching network had in fact worsened the impedance mismatch to 50Ωs instead of improving it, as shown in figure 10. It is unsure what caused this to happen as the VNA (with the cables used during testing) was fully calibrated during use. Equally, it is unsure why Antenna 3 shows significant impedance deviation from the other antennas as they were all of a similar construction and had close to identical lengths.

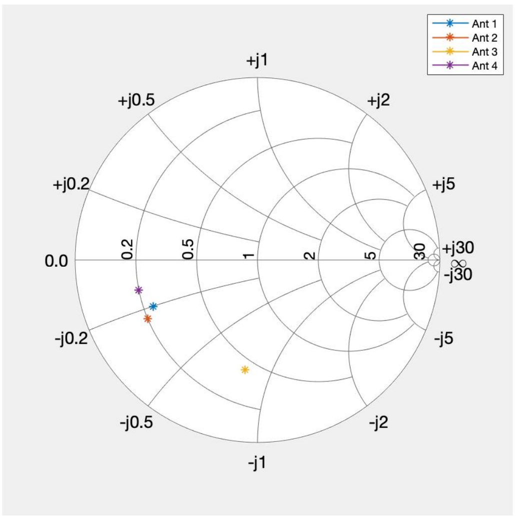


Figure 10: Measured input impedance after matching. The mismatch has been worsened by the network used. The points have spread out due to the use of constant component values for all the antennas.

Rather than spend a lot of time investigating the cause of these problems, it was decided to instead remove the matching networks and use the unmatched antennas in the final system. This is because antenna matching was not critical to the end functionality of the array. The only system effect this will have is reduced receiver sensitivity.

4.2.1.3 Casing

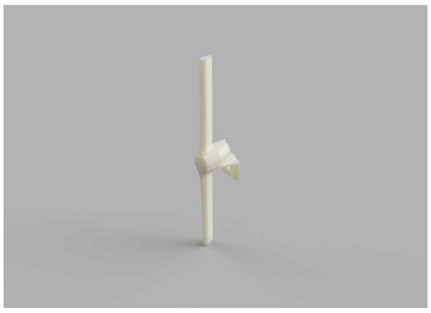


Figure 11: Fusion 360 render of the antenna designed antenna casings

Antenna casings were developed to contain the components of the antenna, including the copper wires, PCB and SMA connector. The case was designed to help keep the copper wires straight. The parts were designed in Autodesk Fusion 360, and 3D printed out of a red-coloured polylactic acid (PLA) bioplastic. A render of the model can be seen in figure 11. However, the water content of PLA and dyes used in the colouring process resulting in coupling between the copper wire and the plastic surrounding it. This greatly affected the response of the antenna, reducing the resonant frequency from 868MHz to 800Mhz. As a result of this, only the SMA end of the enclosure was used. This provided a method of securely mounting the antennas whilst inducing less interference with the response of the antennas.

4.2.2 Low Noise Amplifiers (LNAs)

The first iteration of the receiver circuit had no LNA before the demodulator IC; it was determined that in order to detect an input signal without added gain, the signal strength at the receiver must be must be greater than -55dBm. A 0dBm transmission across an indoor room (4x5m^2) with our constructed dipole antennas was found to provide approximately -70dBm of signal power at the input to the receiver. Thus LNA design was performed to add and extra 15dB of gain at the beginning of the signal chain. An LNA would also help to reduce the significance of the noise being added to an input signal by the demodulator IC which, at best, has a noise figure of 10dB [18].

4.2.2.1 Theory

At RF, Heterojunction Bipolar transistors (HBTs) are often used. These differ from standard Homojunction Bipolar Transistors (BJTs) in that they use different semiconductors for the emitter and base layers. This enables the doping to be arranged for a better high frequency response. Their operation and use is exactly the same of that of a BJT, however they can operate up to hundreds of GHz [19].

Bipolar transistors operate through current being injected into the transistor base. When a base current is injected, then a collector current is pulled through the device. In order to bias these transistors into the desired mode of operation, a constant d.c. current must be injected into the base. Bipolar transistors are also sensitive to thermal runaway due to increasing collector current, this can quickly destroy a device. Hence a circuit that delivers the same current, independent of temperature, is crucial. The bias network must keep the transistor $V_CE$and I_C constant, so that RF performance between different devices is approximately uniform. Both active and passive feedback networks can be used to control the biasing. A transistor datasheet often will specify a Vce against Ic graph with different plots, corresponding to base currents. To achieve a specific I_C, an I_B and V_{CE} must be chosen using this graph. A datasheet will normally specify a bias setting for minimum noise figure, and contain a graph of Ic against the transition frequency (f_t), for maximum gain, the device should be biased to correspond with the maximum f_t [20]. The simplest amplifier topology is the common emitter, this is shown in figure 12. The base acts as the input, and the collector terminal as an output. The circuit has medium voltage and current gains, giving a high overall power gain. The circuit is also inverting. This can be seen intuitively; when VBE swings high, the collector current increases, dropping voltage across the collector resistor (RC), reducing the output voltage. As the input voltage swings low, the current through RC is reduced, and so the voltage at the output increases. The gain of a common emitter transistor is predominantly defined by the internal resistance of the emitter terminal.

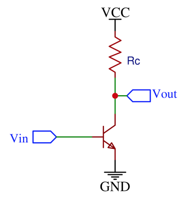


Figure 12: Common-Emitter amplifier topology. The emitter is 'common' as it is tied to ground acting as the common terminal for both the base and collector.

RF transistor amplifiers differ from standard low frequency designs as they often use an RF Choke (RFC) inductor instead of a collector resistor. This means that an output swing of 2Vcc is achievable, instead of Vcc. In order to swing the output beyond VCC, the amplifier relies on the energy stored in the RFC [21].

4.2.2.2 Transistor Choice

When choosing a device the optimum compromise between largest gain and lowest noise figure, whilst maintaining stability was desired.

The decision around which RF transistor to use was made by searching through available products and selecting one based upon the desired characteristics. Through this process, the BFP840 HBT, by Infineon [22], was selected as the datasheet states that at 900MHz, with a Vce = 1.8V and an Ic = 5mA, a maximum gain of 26.5dB and a typical noise figure of 0.6dB were specified.

4.2.2.3 Biasing

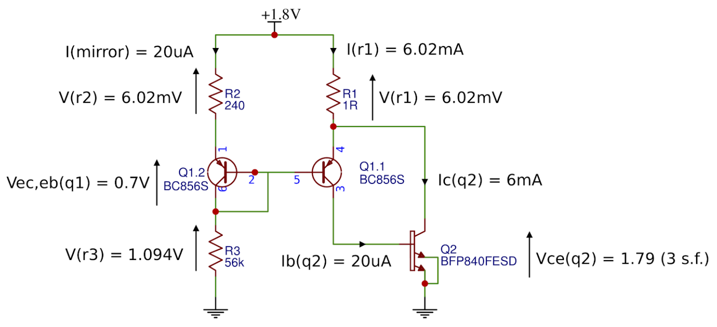


Figure 13: Current mirror biasing circuit for the BFP840 transistor.

The circuit used to bias the LNAs is shown in figure 13 From the datasheet of the BFP840, it can be found that in order to drive the RF transistor with a 6mA collector current, 20μuA of base current is necessary. Q1.1 is connected in a diode configuration, which then alongside R2 & R3 conducts 20μA of current. The base of Q1.1 is also connected to the base of Q1.2, such that Q1.2 is also biased sufficiently to drive 20uA of current. This is passed directly into the base of the RF transistor Q2, which in turn will pull 6mA through its collector.

Q1.1 and Q1.2 are named such as they are two seperate transistors housed in a single package, and therefore share near identical characteristics. With a collector current of 6mA through Q2, R1 develops a voltage of 6mV across it, as the two transistors are identical, an equal voltage is developed across R2, which has been designed to pull 20mA of current when 6mV is placed across it. If less current were to be pulled through the collector of Q2, then the voltage across R2 would decrease, increasing the voltage across Q1.1, therefore increasing the overall mirrored current in to the base of Q2, which would in turn increase the collector current of Q2. It can be seen that a negative feedback loop occurs, which ensures that the device remains stable and correctly biased.

4.2.2.4 Final Circuit

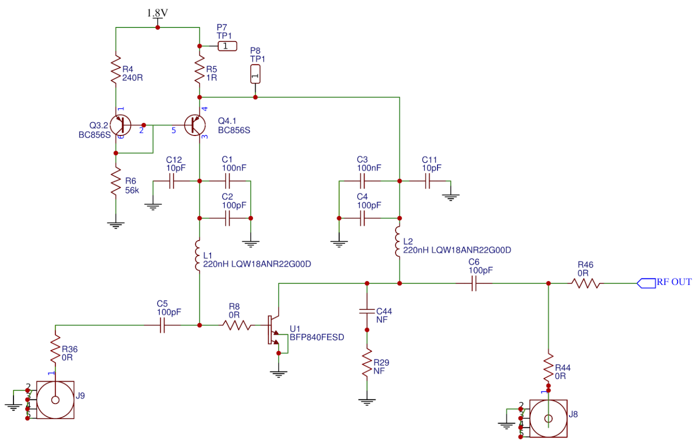


Figure 14: Final BFP840 LNA design.

The final amplifier circuit is shown in figure 14 L1 & L2 were selected to perform RF blocking, and were chosen as these presented the highest impedance at 868MHz from the known available components. It is worth noting that the amplifier also relies on L2, to provide output swing beyond the d.c. collector voltage on the transistor. C1,2,3,4,11&12 are in parallel to provide a.c. shorts to ground at various frequencies so that no RF leaks into the biasing arrangement. C5,6 provide a.c. coupling into and out of the circuit. R8 is a 0Ω link that can be used to limit the signal into the base of the transistor in order to reduce the gain if necessary. C44 and R29 are also no fit components in case some gain reduction or filtering is required. J9 is an SMA connector for RF input, and J8 is used to probe the output of the amplifier.

It was decided that as the transistor is stable and has sufficient gain with a 50Ω source and load impedance, no matching was necessary.

4.2.2.5 Simulation

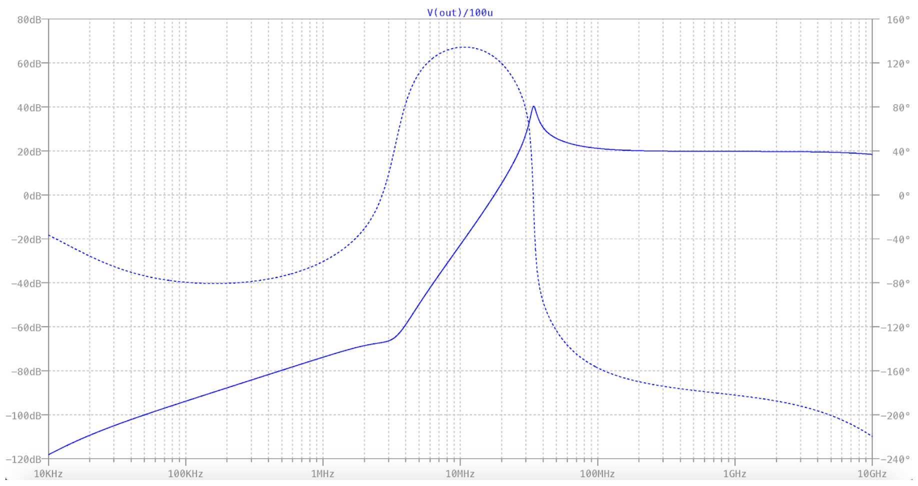


Figure 15: LTSpice Simulated LNA frequency response. Includes the BFP840 spice model and parallel resistors to represent the Q factor of the inductors.

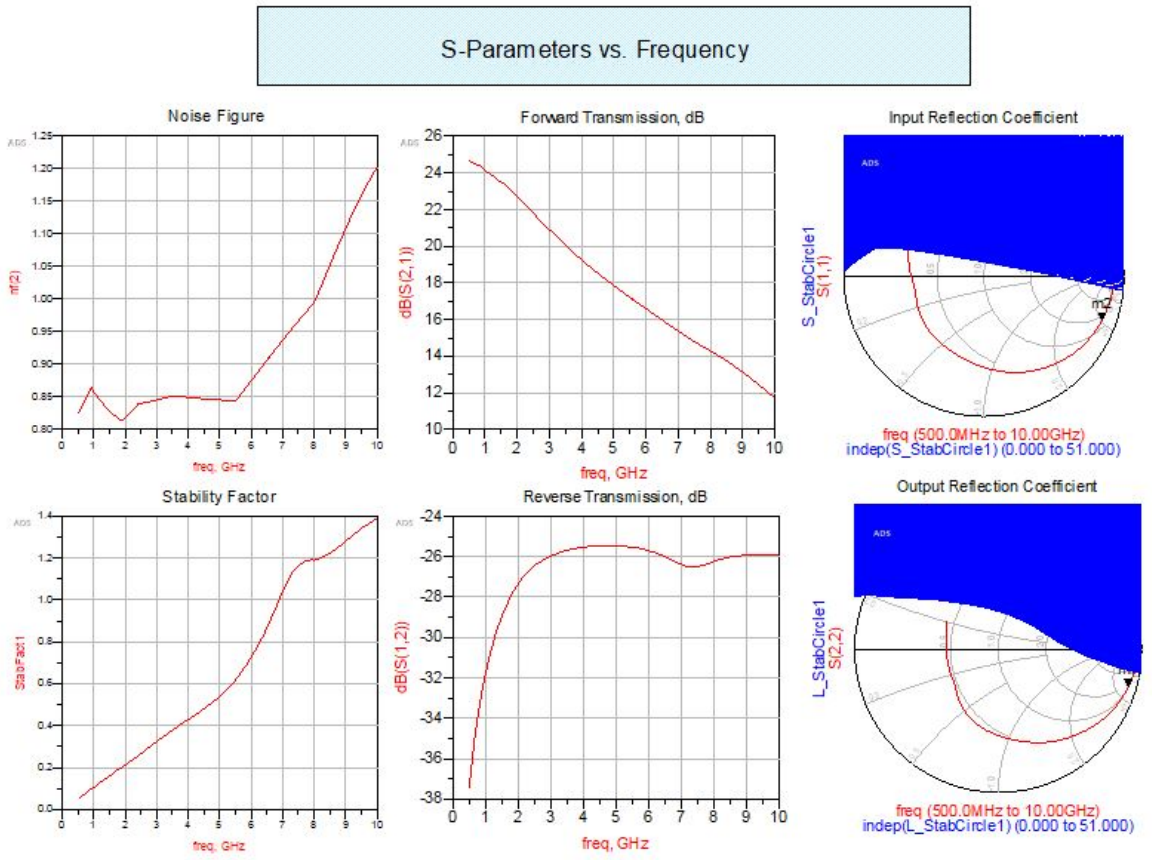


Figure 16: ADS simulation results showing the expected Noise Figure, Stability Factor, Forward Transmission, Reverse Transmission, Source Stability Circle and Load Stability Circle.

Simulation of the LNA circuit was first performed in LTSpice, and then in Keysight ADS. These verified that the d.c. and a.c. characteristics were as designed. The simulation results were used to debug the construction of each circuit, as they highlighted some small differences from the design such as slight voltage imbalances due to the use of standard component values. Figures 15 and 16 show the simulated frequency response in LTSpice and simulation results from ADS.

4.2.2.6 Verification

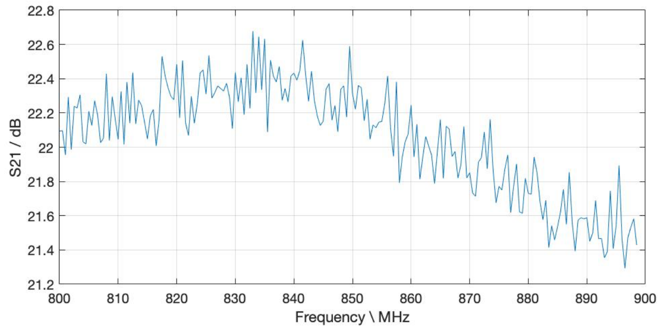


Figure 17: Development LNA S21 plot over the 800-900 MHz frequency range. The gain varies by 1dB over the frequency range.

An LNA was constructed as part of the second receiver development board. At 868MHz a gain of 21.7dB was measured using a signal generator and spectrum analyser. LNA stability was determined by looking over the full span when no input is applied. The gain method [23] was used to estimate the noise figure of the LNA, however this method is known to be unsuitable for measurement of low noise figures. The measured noise figure was 2.3dB, which is worse than the 0.6dB quoted in the transistor datasheet but still sufficient to significantly decrease the (minimum) 10dB noise figure of the demodulator. It is worth noting that this noise figure method also includes noise generated within the spectrum analyser and interconnects, so the amplifier noise figure will be a proportion of this measurement. Figure 17 shows the S21 plot of the development LNA. The gains shown here agree with the gains predicted by the LTSpice simulation.

4.2.3 AD8348 Demodulator Stage

The AD8348 is an IQ demodulator IC, which can downconvert and demodulate signals in the frequency rage 50MHz-1GHz [18]. This IC contains an inbuilt Variable Gain Amplifier (VGA), a gilbert cell mixer, and a 20dB Fixed Gain Amplifier (FGA) for baseband amplification. 868MHz RF signal is routed directly into the IC which handles the downconversion process. Although the IC can be configured in multiple ways, we have decided to configure it to use each circuit element in the order designed, described by the datasheet. Figure 18

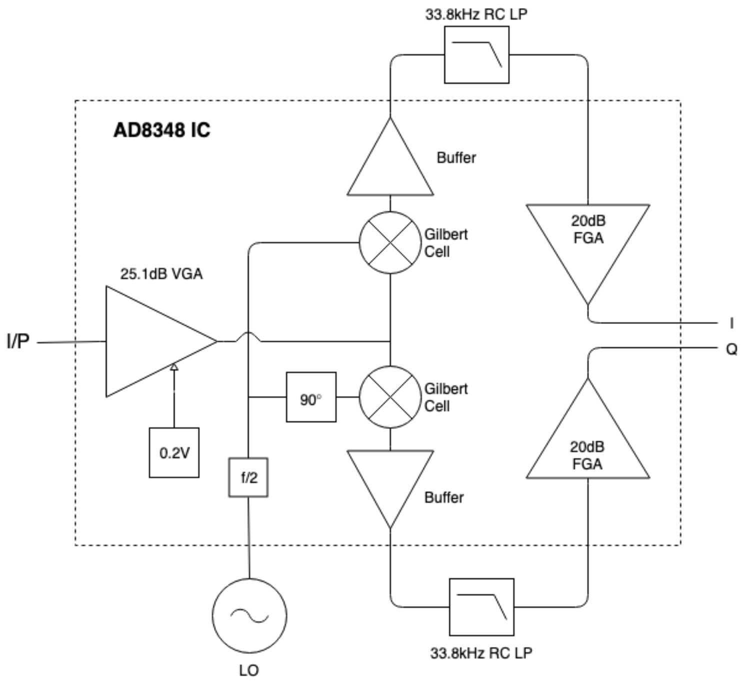


Figure 18: AD8348 Demodulator System. A Variable Gain amplifier is held at a maximum gain of 25.1dB, this is then fed into two seperate Gilbert Cell mixers, this is then buffered, and sent out of the IC to a 1st order RC filter stage, the signal then re-enters the IC to pass through the 20dB fixed gain amplifier (FGA). The frequency of the input LO signal is halved, and then an unshifted version is fed to the I mixer, and a 90 degree shifted version is passed to the Q mixer.

4.2.3.1 Frequency Mixing Basics

Low frequency signals are often encoded into an RF carrier for transmission in a particular frequency band. Frequency conversion is used to separate out baseband signals before and after transmission so multiple signals can be multiplexed in frequency. When two sinusoids are multiplied, a sum and difference frequency are produced (equation 17). The unwanted frequency can then be filtered out, and so a frequency conversion has occurred. In order to convert back to the original frequency, the modulated signal can be multiplied back with the LO, and the baseband message is retrieved.

$$sin(\omega_c t)\, sin(\omega_m t) = \frac{cos(\omega_c + \omega_m)t \, + \, cos(\omega_c - \omega_m)t}{2} \tag{17}$$

In hardware, the frequency mixing process is performed by a non-linear circuit component, in its simplest form this can be a diode, however an active switching element is more common. This replaces the carrier wave with a square wave (+/-1) of the same frequency. When the mixed signal is multiplied by 1 the input is transferred to the output with no change. When it is multiplied by -1 the input is phase shifted by 180 degrees, performing an inversion. In a switching mixer, the LO signal doesn't need high linearity as it is purely a switching signal. The linearity is defined by the input signal.

A Gilbert Cell (show in figure 19) is a double balanced RF mixer that utilizes its symmetric topology to remove the input signals from the output by cancellation. It acts as two differential pairs, with one input controlling the bias current through the pairs, and the other driving the base electrodes. The output signal is an accurate multiplication of the two input signals. The symmetry in the circuit ensures that the RF and LO signals don't leak out of the output. The LO input is driven by a square wave (or pre distorted sine wave) that multiplies the RF input level by +/-1, hence performing the frequency mixing [24].

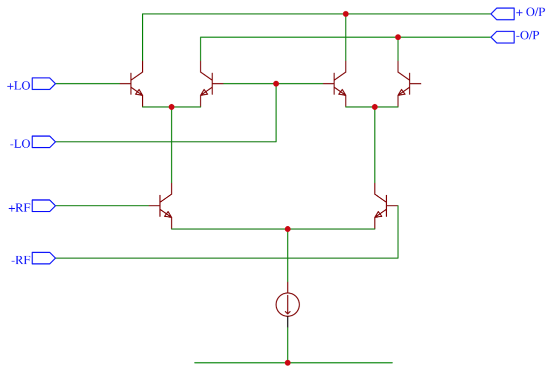


Figure 19: Gilbert Cell Mixer Schematic. The output is a multiplication of the two input signals. Since is mainly consists of semiconductor devices, it can be easily integrated into ICs.

4.2.3.2 Gain Setting

The inbuilt variable gain amplifier is controlled from an external voltage on pin 17 (VGIN), this can be set over the range of 0.2 to 1.2V, which correspond to gains of +25.5dB to -18.5dB respectively. On our first iteration of the circuit design we used a potentiometer to drive this pin and set the gain, however it quickly became apparent that we would need the maximum amount of gain that the IC could deliver. The potentiometer gave way to a fixed potential divider providing 0.2V, for maximum gain.

4.2.3.3 Reference Voltage and Biasing

Each demodulator IC contains a 1V reference voltage source output (pin 14, VREF) which is to be connected to the common mode voltage input (pin 5, VCMO), setting the d.c. level of the output pins. This voltage level is then used to hold d.c. values within the anti-aliasing filter stages further down the signal chain. The VREF pin of a single demodulator can only provide 2.5mA of current, so an opamp buffer was added to the output of a single demodulator providing a much higher current capability. This meant that every receiver in the array can be driven by a single reference voltage, to ensure that they are all held at the same value. The opamp buffer circuit is shown in figure 20.

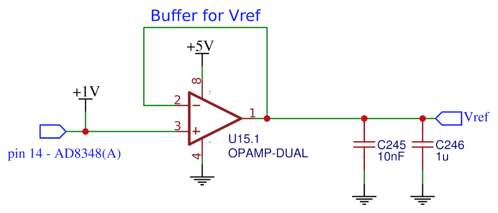
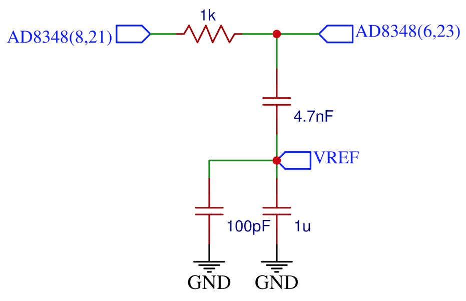


Figure 20: Buffer circuit used to provide a constant Vref across the entire array.

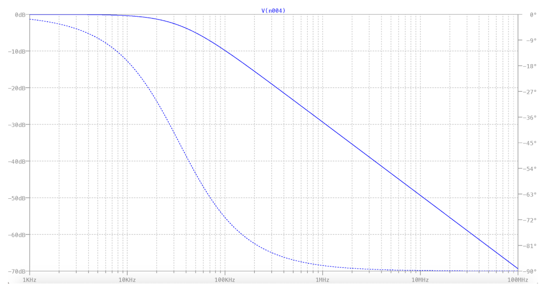
4.2.3.4 RC Filter

To ensure that minimal out of band signals were leaking into the FGA, 1st order RC filters were added directly after the mixer stage (pins 8, 21), before the input to the FGA (pins 6, 23). The circuit diagram and frequency response of these circuits is shown in figure 21 The frequency response was checked using LTSpice.

Figure 21: 33.8kHz RC filter: (a) Schematic, (b) Frequency response.



(a)



(b)

4.2.3.5 Balun Development

In order to eliminate noise within the demodulator IC the manufacturer suggests driving the inputs with balanced signals [18], which can then be converted back, post-downconversion, to unbalanced signals. Another advantage of the baluns, is that they can be used to impedance match the signals into and out of the demodulator. In order to perform this conversion two balun topologies were investigated; LC and transformer baluns.

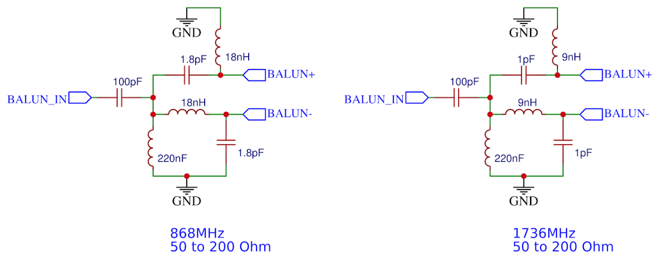


Figure 22: LC Balun Schematics

The LC baluns were designed using the method described in [25]. Two baluns were designed, one balun for 868MHz and one for 1736MHz. They were designed to convert from a 50Ω transmission line to a 200Ω balanced impedance. Their schematics are shown in figure 22.
An LTSpice simulation was performed, and showed that at their respective frequencies, the baluns would have an insertion loss of approximately 10dB and a phase shift of 180 degrees.
The performance of the ETC1-1-13 transformer balun was compared with the performance of the LC baluns The datasheet [26] claims a typical insertion loss of 3.32dB in the 4.5-1000MHz range.
Each balun design was constructed and then tested using a VNA. The input was connected to port 1, the positive output to port 2, and the negative to port 3. Plotted results are shown in appendix [balun plots], table 2 provides a summary of the LC and transformer results. It can be seen that the transformer balun has consistent power loss and phase shifts between ports. At 868MHz, the transformer has a much better power balance, but doesn't achieve as accurate a phase shift. At 1736, the transformer balun loses more power, but the phase shift between output ports is much closer to 180 degrees.

Balun Output	1736MHz LC	ETC1-1-13 (1736MHz)	868MHz LC	ETC1-1-13 (868MHz)
+ Magnitude / dB	-8.8	-6.35	-5.67	-4.04
− Magnitude / dB	-18.48	-4.1	-7.73	-3.77
Phase Difference / Degree	-159.11	184.63	184.71	171.12

Table 2: Summary of the magnitude and phase imbalance for each balun.

The transformer balun was chosen to be used in the final design, as the LC baluns couldn't deliver as much power to the receiver ICs. The power from the LO was being split down with no amplification, and the RF signal gain is to be maximised. In both cases, maximising power throughput was the most important factor.

4.2.3.6 AD8348 Final Schematic

The final schematic for the demodulator circuit is shown below in figure 23. The baluns, filters, gain setting potential divider, and Vref buffer can all be seen.

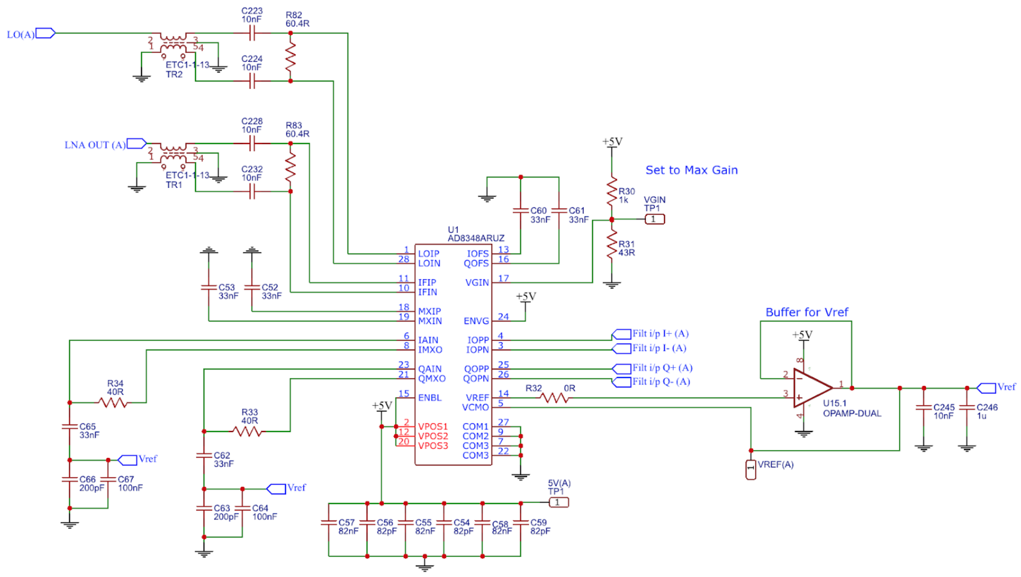


Figure 23: Final schematic of demodulator circuit.

4.2.4 Local Oscillator (LO)

To mix down from the RF input signal to baseband, the AD8348 demodulator requires an LO drive signal which is double the 868MHz RF carrier frequency (1.736GHz) and is in the power range -10 to 0 dBm. The ADF4351 was selected as a suitable IC to provide this functionality and has a programmable output frequency range of 35 to 4400 MHz for a programmable output power range of -4 to 5 dBm. The LO hardware development is discussed in sec. 6.2.4.8.

4.2.4.1 ADF4351 Development Board

To enable rapid overall system development, a development board was acquired which included pinouts for the ICs SPI and other digital I/Os as well as SMA outputs on the LO drive complementary pairs. Using this board, a program (and user interface) was developed to achieve frequency locking on the ADF4351 by setting register values within the IC. A custom circuit design was then developed for the ADF4351 to achieve improved performance and to enable the LOs integration onto the final PCB.

4.2.4.2 Programming

The ADF4351 uses six 32-bit programmable registers to control the on-chip functions. These are accessible through a 3 wire interface similar to the SPI protocol. This consists of master and slave devices, with data, clock, and slave select lines between them. Programming of the oscillator has been achieved through two methods; standalone with an external microcontroller (Arduino Uno/ Leonardo) and integrated, allowing for basic functionality to been included in the final FPGA design.
The standalone functionality implements full serial communication between the computer and the microcontroller. This allows for the oscillator to set to any frequency and power within the ADF4351's limits. When the user selects a frequency and power in the GUI they are transmitted to the microcontroller which calculates the register values, send these to the ADF4351 and returns the set frequency to the GUI. Within this, there is bounds checking of inputs and a warning to the user if the desired frequency is unobtainable due to the limitations of the available configuration registers.
The FPGA implements a similar functionally however to reduce the complexity of the design the SPI communication is limited to one frequency (1.736 GHz) as this allows for a predetermined set of register values to be

used 3. This is acceptable as the FPGA will only be used with the current 868MHz receiver, however, it would be possible to implement this if required.
The SPI protocol is implemented within the FPGA as a combination of soft hardware using the Xilinx AXI Quad SPI IP and it's driver functions to allow control by the soft microprocessor core. This has been configured to a clock frequency of 4 MHz and a packet length of 32 bits. The oscillator programming sequence is triggered when an ASCII "1" is received over serial, the AXI Quad SPI core is then initialised, configured and used to transmit the required registers to the oscillator.

	Bytes (HEX)			
Reg	1	2	3	4
0	00	45	00	B0
1	00	00	80	C9
2	00	00	4E	42
3	00	80	04	B3
4	00	95	00	3C
5	00	58	00	05

Table 3: Oscillator register values for 1736MHz.

4.2.4.3 Frequency Locking

Within most RF systems including this project, PLL's are used to generate and control the local oscillator frequency. Phase locked loops (PLL) are a control system that generate an output signal from a input signal through comparison of their phases. In this project it is used to generate an output frequency from a fixed reference source. When used as a programmable electronic oscillator they consist of a variable frequency oscillator, phase detector, dividers and loop filter.
The feedback loop is driven by a high stability reference frequency discussed in 4.2.4.4.2. In the ADF4351, the reference frequency input includes selectable doublers, a 10-bit R counter and divider, creating a user controllable comparison frequency. The phase frequency detector (PFD), creates a difference term from the comparison frequency and the divided output frequency which drives a charge pump. The charge pump creates pulses of current which pass through the loop filter and control the voltage controlled oscillator (VCO). The VCO is the main high frequency generating component of the PLL and is adjustable over a range of frequencies depending on the input voltage provided by the loop filter. The feedback loop is continued as the output frequency is then divided by fractional and integer dividers and input to the PFD. When there is a difference between the comparison frequency and the divided output frequency current pulses are created, adjusting the VCO until the two phases match.
When the PFD has determined that there is an insignificant difference between the signals, it will signal that the loop is locked, meaning that the output frequency is sufficiently close to the desired value. The ADF4351 allows for programming of this function, either as a digital or analogue lock detects. For fractional-N operation it is recommended for the number of PFD cycle monitored to be 40 and integer-N mode to use 5.

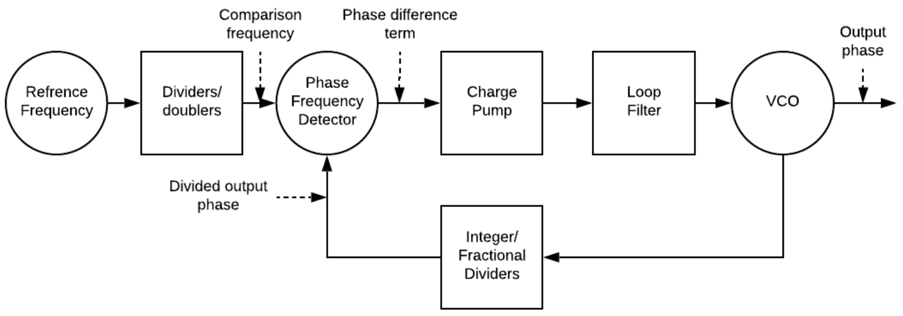


Figure 24: PLL system diagram.

4.2.4.4 Custom Design

The custom circuit design for the ADF4351 contains features from both the development board and the recommended application circuit provided in the IC's datasheet [27] . Taking advantage of the improved power supply topology, increased decoupling and an improved reference oscillator, the custom LO design aimed to show an improved performance over the development board LO in terms of the frequency lock accuracy, phase noise and spurious emissions.

4.2.4.4.1 Ultra-Low Noise Regulators

The bought development board uses a single standard 3.3V regulator to provide the ICs analog and digital power supplies (separated by a shunt capacitor and series inductor for high frequency noise suppression) and the RF output biasing. In the custom board, dedicated ultra-low noise regulators [28] were employed to provide separate power supplies for the analog and digital supply and the RF output biasing supply to reduce noise bleeding into the RF output of the IC.

4.2.4.4.2 Reference Oscillator

To reduce the output frequency error to the desired lock frequency, the SIT5000 [29] Temperature Controlled Crystal Oscillator (TCXO) was chosen as the reference 25MHz oscillator. With a frequency stability of less than 5ppm (nominal 1.5ppm), this provides a maximum reference frequency deviation of (+ -) 125Hz. This in turn translates to a maximum LO system frequency error of 8.7kHz or 2.6kHz using the nominal value. The exact deviation was identified during testing (see results).

4.2.4.4.3 Loop Filter The loop filter within the PLL is typically low pass to integrate the current pulses from the phase comparator, creating a voltage for the VCO [30]. Dependent on the loop filter are certain dynamic characteristics such as settling time and frequency overshoot. The stability determined by the loop filter is most important during times of disturbance for example, at startup and when changing channels.

In this design the loop filter has not been changed form the data sheet recommended values, as this provides a design with switchable fast lock elements and is designed for a loop bandwidth of 35KHz.

4.2.4.4.4 LO Drive Splitter

To reduce cost and provide synchronicity, a single LO was required to drive all the receiver modules in the system. A splitter was then required to ensure that each AD8348 demodulator IC received an LO drive of a similar power level and different splitter topologies were investigated to provide the LO signal to an 8-element receiver array. To begin, a circuit to divide a single LO output to all 8 AD8348s was developed. Resistive wye splitters were chosen for the splitting junctions for simplicity and wide bandwidth [ref] and, with 3 series sets of these alone (see figure 25), an output LO power of 5 dBm would have resulted in -13 dBm (minimum of -10 dBm is required) of power at each demodulator due to the 6 dB insertion loss of each wye splitter.

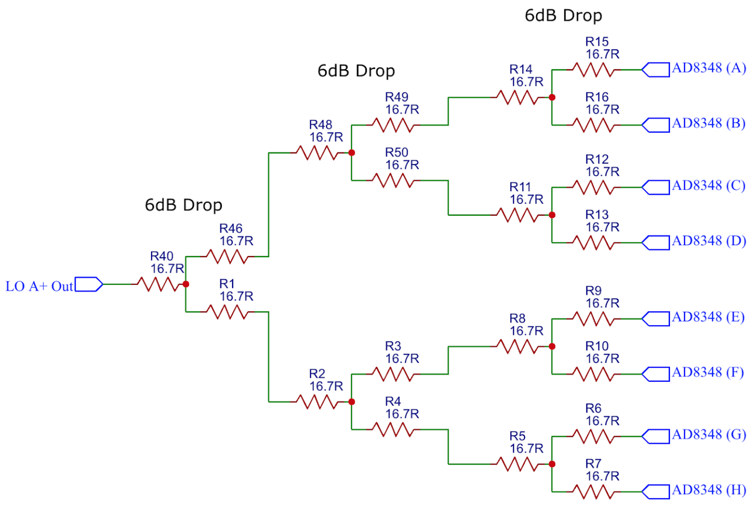


Figure 25: 8-way resistive splitter.

To compensate for this, an ADL5602 matched gain block amplifier [31] was included into the schematic to provide 20 dB of gain, resulting in a power level of 7dBm at the AD8348 ICs for a LO output power of 5dBm. Setting the LO output to less than -2dBm would then bring the splitter outputs to the correct range. However, since there is the potential for the ICs to be damaged, a π-attenuator of 7dB was then added before the gain block, giving the schematic shown in figure 26 below:

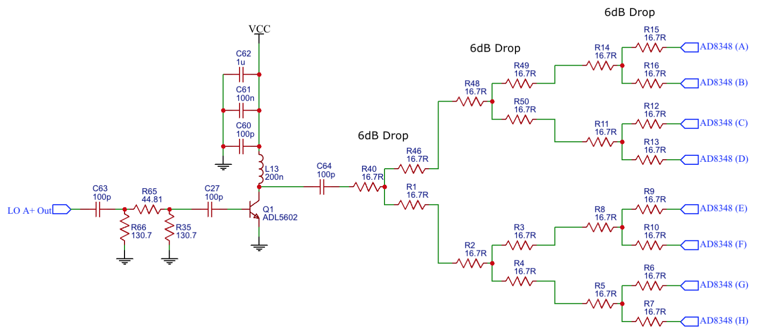


Figure 26: 8-way resistive wye splitter with a gain stage before it.

To investigate the feasibility of this gain block circuit, a 4-way splitter version (same schematic as figure 26 but with the last set of wye splitters removed) was designed and included on a development board (see 4.2.7.5). The gain of each output (with respect to the input) was then investigated using a signal generator and spectrum analyser and a VNA was then used to determine the phase response of each output. Table 4 and figure 27 give the results of this test.

	Measurement			
	S21	S31	S41	S51
Gain (dB)	3.61	1.31	1.46	3.55

Table 4: Measured port gains of the 4-way gain block splitter.

A superior solution was then explored in which the auxiliary complementary output of the ADF4351 (pins 14 and 15) are activated so that there are 4 LO output pins to drive the 8 demodulators. This then means each output needs splitting only once with only a single 6dB insertion loss for each. This therefore provided a solution which only required 4 wye splitters and would give a power range of -10 to -1 dBm at the demodulators, satisfying the whole AD8348 LO input range requirement. This makes this the most cost effective solution, but introduces the 180 degree phase imbalance between the positive and negative outputs of each LO output pair on the ADF4351 into the receiver system. Phase differences would exist in the baseband signals produced by the demodulators driven by the positive outputs (A+ and B+) and the demodulators driven by the negative outputs (A- and B-).
Two methods of compensation for this were then considered. First, the PCB track lengths could be tuned on the final board to bring the negative LO outputs in phase with the positive outputs. Therefore, if each track carrying the positive LO signal to a demodulator was of length x cm, then the traces carrying the negative LO signal would need to be of length $x + \frac{\lambda}{2}$, making them an additional 8.64cm long. A simpler alternative was to accept that identical channel phase responses would be impractical to achieve (since systems other than just the LO may introduce phase imbalances between channels) and instead perform a phase calibration on the final system. This would then involve measuring each overall channel phase response for a constant phase input to the LNAs and then applying a correction to all measured data from this. The latter method was chosen for the final system.
In the final system it was then decided to reduce the array element number down to 4 instead of 8 for ease of implementation. While all 4 LO outputs could be used to drive a single demodulator on their own, there would be the potential for over-driving the LO input of each AD8348 and therefore, would have required a 5dB π-attenuator for each LO output. To reduce component count, it was decided to instead use just the two LO outputs (A+ and A-) and then use a wye splitter on each.

4.2.4.4.5 Final Schematic

The final LO schematic is given below in figure 28:

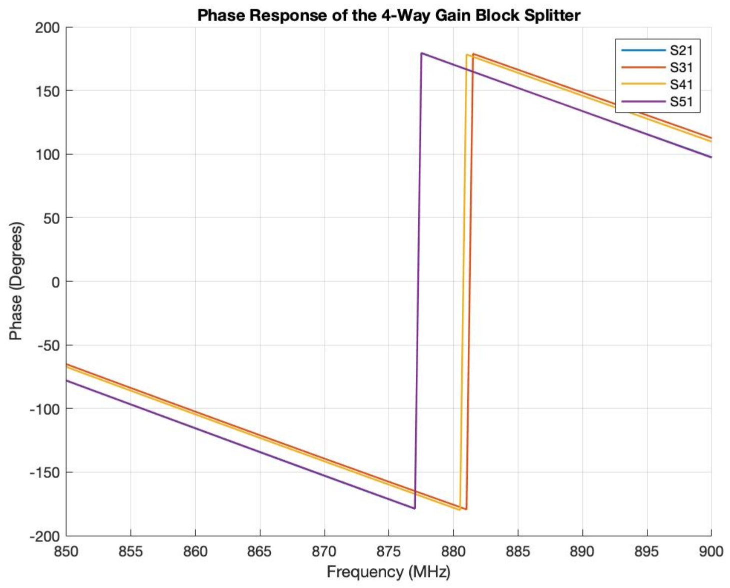


Figure 27: Phase response of the 4-way gain block splitter circuit. The S21 and S51 traces are over the top of each other. Port 1 is defined as the input and ports 2, 3, 4 and 5 are the outputs going down the circuit implemented on the development board (see 4.2.7.5). The differences in phase between the S21 and S51 measurements and the S31 and S41 measurements are due to the xfdifferences in track lengths for those ports (the centre two ports have the least track length from the gain block output to the SMA connector). At 868MHz, the phase shifts are -137 degrees (S31 and S41) and -150 degrees (S21 and S51).

4.2.5 Anti-Aliasing Filters

In order to prevent aliasing in the ADC process active filter circuits were designed. This consisted of a differentially driven filter circuit, followed by a standard 2nd order Sallen & Key Butterworth filter. The MAX4413EKA rail to rail op amp was used as the active component within the filters. The use of active op amp circuitry at baseband frequencies has some advantages over their passive counterparts, e.g. potential for gain, reduced component volume, ADC buffering and dc coupling. The AD8348 demodulator IC outputs a 1V reference level, which must be d.c. coupled to the ADCs, through these filters.

4.2.5.1 Theory

At low frequencies inductive coils become very large. Active filters eliminate the need for magnetic components through the use of opamp circuitry. The Voltage Controlled Voltage Source (VCVS) filter topology is based upon the inverting circuit invented by Sallen and Key and always has a gain of greater than 1. This topology is shown in figure 29.
The gain is set by the equation in 4.2.5.3. Each RC pair defines a 1st order pole in the frequency response. If R1 = R2 and C1 = C2, then the filter has the response of a second order Butterworth, which gives a maximally flat passband. Another reason that active filters are widely used, is that the opamp acts as a buffer, being able to drive following stages with no concern for the filter output impedance [21].
Whilst VCVS filters are useful for single input circuits, different techniques must be applied for multiple input circuits. Daisy's theorem, that the sum of all the gains into an opamp must equal to one, treats a connection to ground as an input with a gain (ground gain). Using this idea it is possible to design opamp circuits with multiple inputs of differing positive and negative gains whilst only using a single formula AV=RF/Ri where RF is the feedback resistor into the negative input and Ri is the resistor between an input and the signal source. Put simply, all the positive and negative gains included within the circuit, must be countered by an equal but opposite gain from ground into the opamp so that the sum of all gains equals 1. [32]
In order to evaluate the system cascade properties, a noise figure was required for each circuit element, including the active filters. The noise figure of an opamp can be approximated using the equation below. For the AD8348 source resistance of 200Ω and a voltage and current noise defined from the opamp datasheet, an approximate noise figure for each opamp circuit used was calculated to be 15.97dB for a bandwidth of 34kHz [42].

$$NF = 10\,log10\left(1 + \frac{V_{Noise}^2+(I_{Noise}\,R)^2}{4kTR_S}\right)$$

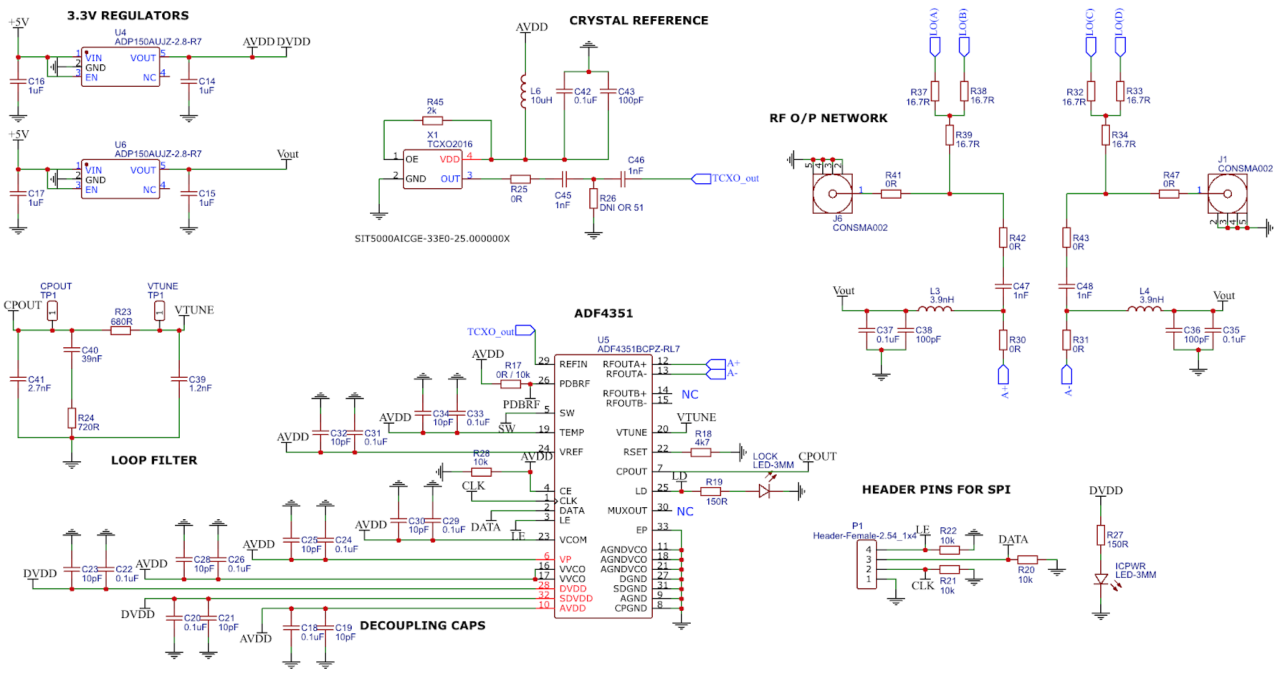


Figure 28: ADF4351 based local oscillator schematic. It includes the dual 3.3V ultra-low noise regulators, the high accuracy TCXO, loop filter, header pins for SPI communication from an Arduino or the FPGA and finally the RF output network with biasing and the resistive wye splitters.

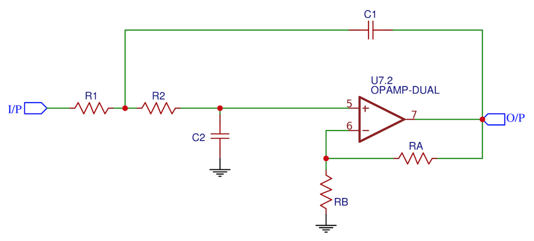


Figure 29: VCVS Low-Pass Filter Schematic. The Gain is set by RA & RB, the frequency response is determined by the pole pairs R1C1 & R2C2.

Where: V_{noise}, is the voltage noise, I_{Noise} is the current noise, R_S is the source resistance, K is boltzmans constant, T is the absolute temperature in kelvin.

4.2.5.2 Differential Filter

The first active filter is designed to perform two functions, active differential to single ended conversion, and a 1st order filter. Using Daisy's theorem, A differential to single ended amplifier was designed. The technique was then adapted into a filter through the use of complex impedances (in the form of resistor/ capacitor pairs) within the design formula. The cutoff frequency of the circuit was defined by equation below.

$$fc = \frac{1}{2\,\pi\,RC} = \frac{1}{2\,\pi\,1k\,4.7n} = 33.862kHz$$

1k resistors were used with 4.7nF capacitors. This Schematic is shown in figure 30

4.2.5.3 Active 2nd Order Butterworth

A VCVS topology was used to implement a 2nd order butterworth filter, the schematic of which is shown in figure 31. The gain of the circuit is defined by the equation below. The circuits were designed to allow a gain to be implemented however, in the final design no gain was necessary. To design for this RA was short circuited, and RB was selected to equal 100kΩ.

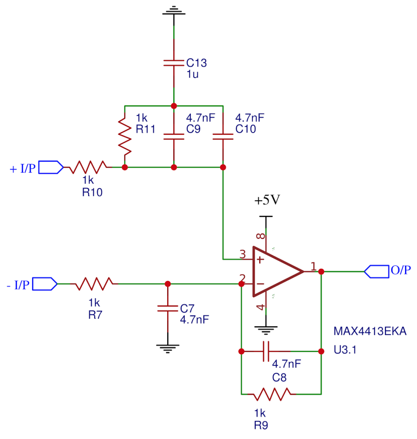


Figure 30: Differential to single ended LP filter schematic.

$$A_V = 1 + \frac{RA}{RB}$$

The frequency response follows a 40 dB/dec curve with a cutoff frequency defined by the equation. The selected values were 1kΩ and 4.7nF as for the differential to single ended filter which means that the active filters combine to form a 3rd order (60dB/dec) slope. The entire circuit between the mixer and the ADC forms a 33.8kHz 4th order low-pass filter with a gain of 26dB (20dB FGA, 6dB from differential to single ended conversion.).

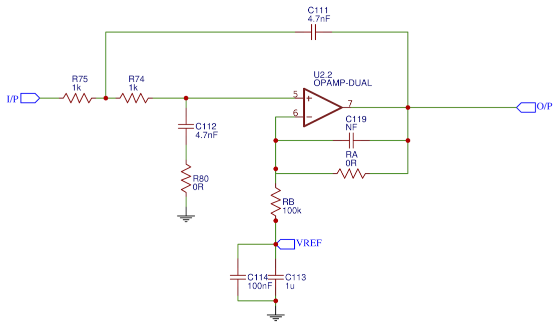


Figure 31: VCVS Butterworth LP filter design. f_C =33..8kHz, A_V = 1.

4.2.5.4 Simulation

Both active filters were simulated in LTSpice, first alone, then connected in series. The LTSpice filter response is shown in figure 32 From the graph, the -3dB frequency is 33.8kHz, and the slope is 3rd Order.

4.2.5.5 Verification

The filters were first constructed on a development board to check that the frequency response agreed with the LTSpice simulation.

NOTE: The butterworth filter constructed on the development board had a gain of 4 (RA = 2k, RB = 1k) however additional gain was later deemed to be unnecessary, with the addition of a front end LNA, and so was reduced to 0dB (RA = 0Ω, RB = 1k). The final response was checked and the trends aligned.

The filter was characterised using a signal generator and oscilloscope. An opamp inverter circuit was used to drive the differential filter, which then drove the butterworth. Figure 33 shows the results. It can be seen that the slope of the magnitude response corresponds to a 3rd order slope and that at 34kHz the magnitude is 3.024dB less than the maximum.

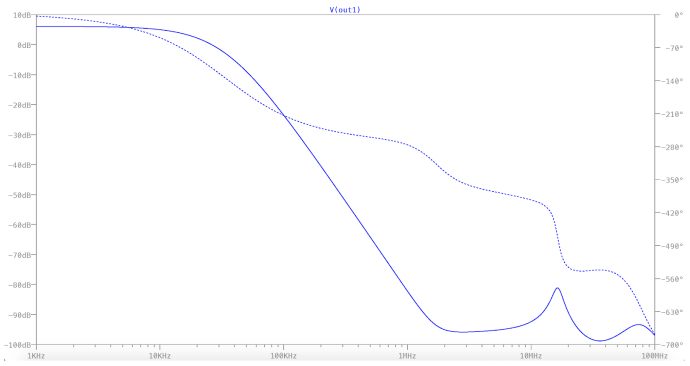


Figure 32: Simulated Anti-Aliasing filter response. A flat passband can be seen to descend into a 3rd order, 60dB/dec, slope. In order to achieve a flat passband, the phase response was not considered in the design.

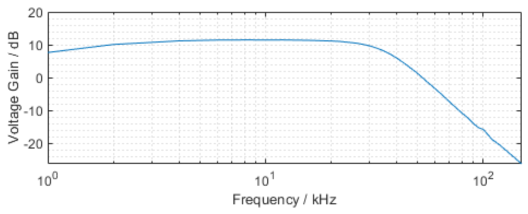


Figure 33: Measured Anti-aliasing Filter Response. 17.03dB of attenuation is measured between 50kHz and 100kHz, one octave. This varies only slightly from the 18dB/ct slope expected from a 3rd order filter. Measurements above 150kHz could not be taken as the signal magnitude was less than the noise level of the oscilloscope.

4.2.6 Analog to Digital Conversion

4.2.6.1 Selected ADC ICs

The AD7829 [33] was selected for the project to convert the filtered baseband signals to digital. The AD7829 has 8 analog input channels meaning two of them were required to allow for full sampling of an 8-element IQ receiver array. Having an overall sample rate of 2MSPS gives a per channel sample rate of 250KSPS when using all 8 channels or 500KSPS when using 4. One of the main reasons for this choice of ADC is that the AD7829 has 8-bit resolution. This is because there were concerns as to whether higher resolution DSP systems would be feasible to implement on the FPGA due to limited DSP slice resources [34] .

4.2.6.2 System Bandwidth Estimation

If the non-simultaneous sampling feature of the ADCs is unaccounted for then this poses an upper limit on the total system bandwidth. This is because the time delay between the channels translates into a phase error between channels which is frequency dependant. As this phase error is frequency dependant and not constant, it will be difficult to remove via calibration.
Calculations carried out in Matlab were then used to obtain an early estimate for the receiver array maximum bandwidth for a differing number of elements (since reducing the number of elements increases the sample rate per channel). Figure 34 below shows the phase error (between the first and last channels of a single ADC) introduced as a function of baseband frequency.
Taking a maximum tolerable phase error of 10% (36 degrees) produces the following maximum system bandwidths shown in table 5:
This indicated that a trade-off exists for system bandwidth and array directivity (a function of the number of

Element Count	Bandwidth (kHz)
8	29
4	67
2	200

Table 5: Bandwidth (less than 10% Error) with varying number of elements.

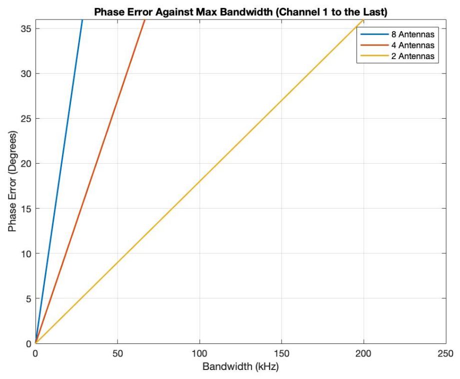


Figure 34: Phase error as a product of bandwidth as a result of the non-simultaneous sampling of the ADCs.

elements) when non-simultaneous sampling is unaccounted for. The results also highlighted that, for array sizes larger than 2, the maximum bandwidth is less than GSM (200kHz), which is the smallest bandwidth mobile communications system in large deployment. This carried the implication that the developed system would be unable to detect GSM transmissions unless either the non-simultaneous sampling was accounted for or new, faster ADCs were implemented.

4.2.6.3 ADC Input Requirements

From the AD7829 datasheet, the analog inputs should be designed to have a maximum voltage swing of 2Vpkpk and should be DC biased at 1V (swings between ground and +2V). For an arbitrary input signal with any bias, the correct 1V biasing can be achieved through the use of an op-amp buffer circuit, as shown in figure 35 :

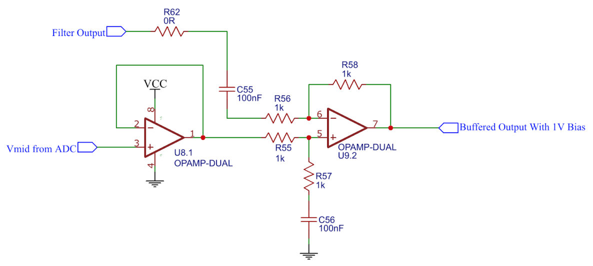


Figure 35: Unity gain buffer circuit which adjusts the DC bias of the input signal to 1V.

Since the AD8348 demodulator ICs also use a reference voltage of 1V, which the filters maintain and pass to their outputs, the signals input to the ADCs are already correctly biased. This circuit then, while still advantageous for its buffering properties, was not implemented in the final system to reduce the final system component count. Instead, a Vref buffer was implemented to ensure all demodulators had a matched bias (see section 4.2.4.6).

4.2.6.4 ADC and FPGA Schematic

To provide the 3V power supply required by the AD7829 ADCs, the low voltage drop-out LM217LD13TR variable voltage regulator [35] was used. Since the pins of the FPGA are configurable, a connection schematic was devised that would produce no trace crossover. The CMOD-A7 FPGA was connected to the ADCs as shown in figure 36:
The channel inputs to the ADCs were configured as A(I), A(Q), B(I), B(Q) for ADC 1 and so on as shown. While the final system was reduced to be a 4 element receiver array, the spare ADC analog input were connected by 0Ω links to the same channel inputs so that, if a 2nd board is constructed, the outputs from that board can be utilised by the same ADCs by connecting the header pin rows together on the two boards. Header pins were added along the FPGA to allow for a logic analyser to be connected for debugging.

4.2.6.5 Input Protection Circuits

During initial testing, it was found that the ADCs had become damaged. Upon further investigation it was determined that this had been caused by analog input overvoltages, caused by errors in the constructed filters

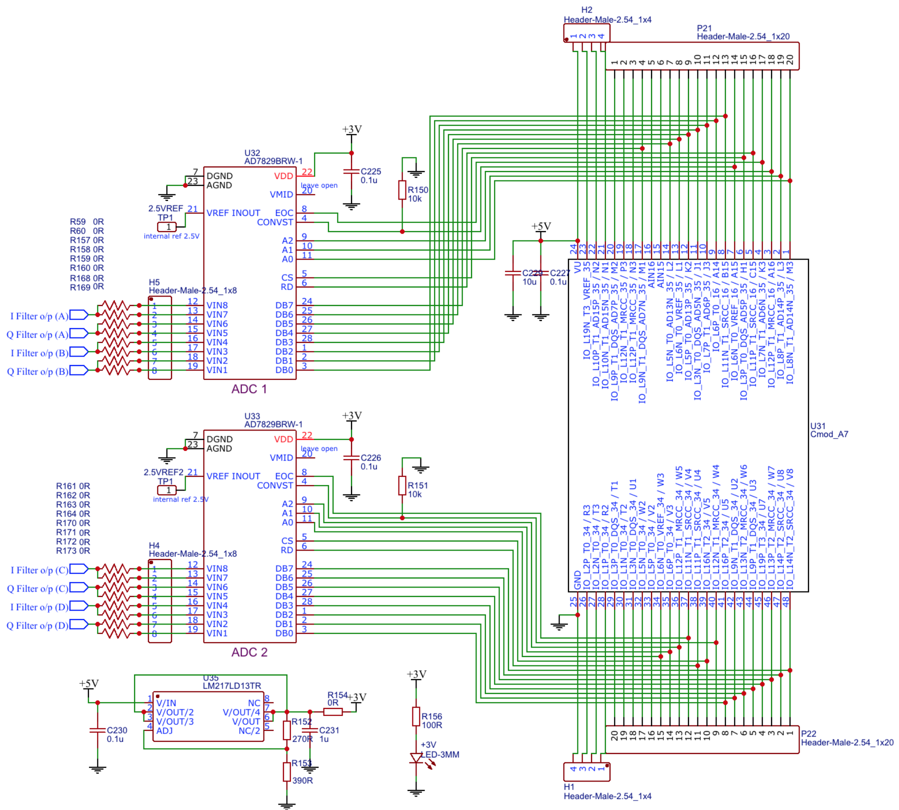


Figure 36: ADC/FPGA hardware interface schematic. The ADCs are powered by an adjustable regulator set to 3V.

(whose outputs were fixed at 5V). To prevent re-occurrence, basic protection circuits which could be feasible to implement onto the already constructed final system PCB were investigated. It was found that the Vishay VLMG1300-GS08 [36] LEDs already in use as power indicators had a forward voltage drop of 2V, which is the same as the desired ADCs maximum analog input voltage (the AD7829 can operate with inputs up to Vdd 3V before becoming damaged but these will clip internally and distort the data anyway). The LEDs were then connected across the filter outputs and ground as shown below in figure 37.

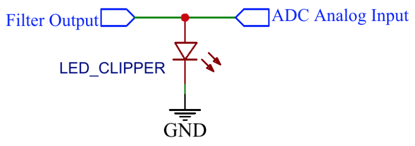


Figure 37: VLMG1300-GS08 LED 2V clipper circuit.

4.2.7 Development Boards

4.2.7.1 RF PCB Layout Considerations

At RF, standard circuit theory begins to break down. A number of reasons cause this, most significantly the high frequencies mean that parasitic circuit elements become increasingly large reactances, and the frequencies can ionise air easily so signals will radiate if due care isn't taken. To route RF signals, coplanar waveguide has been used. The self resonant frequency of each component used in the design has been verified to be larger than the frequency of operation.

4.2.7.2 AD8348 Demodulator Test Board

Since it was determined early into the project that the AD8348 demodulator IC was to be used, an early PCB was developed to allow for rapid testing of its functionality and characteristics. In addition, an early version of the balanced filter was included. A PCB design was produced and is shown below in figure 38.

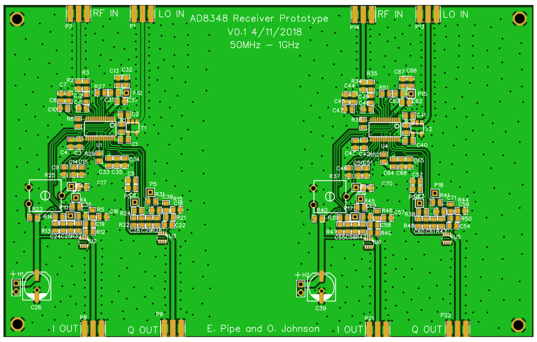


Figure 38: AD8348 demodulator and balanced filter development board with two circuit copies on. The design features Coplanar Waveguide for the RF traces. The bottom layer is a continuous ground plane and ground stitching to the top ground plane is achieved with vias.

This design used 0805 packages for the majority of the components (namely resistors and capacitors) but due to reduced cost and component footprints it was decided that future designs would be composed of 0603 packaged components.

4.2.7.3 ADC/FPGA Development Board

A PCB was designed for the ADCs, FPGA and associated circuitry to allow for: earlier testing of the ADC/FPGA schematic produced in figure 36, earlier development of the digital hardware interface between the ADCs and FPGA and to produce a generalised FPGA/ADC board for early data acquisition. Figure 39 below reproduces the top layer of the board produced which also features regulators with different output voltages to use as basic power supplies for other external test circuits (the bottom layer is a ground plane only). A holder is also featured to allow for the potential development of a dedicated ATTiny85 based ADF4351 programmer.

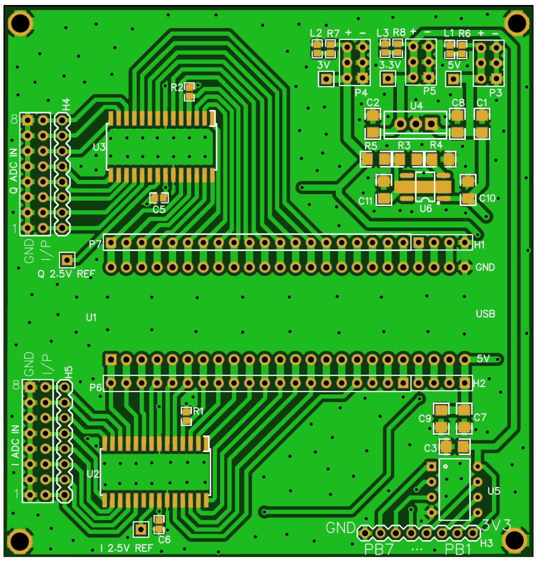


Figure 39: ADC/FPGA development board. It includes power supplies for +5V, +3.3V and +3V for powering external test circuits.

4.2.7.4 System Development Board

Following the design completion of the demodulator, balanced filter, the LNA, the custom LO and the LC baluns, a single receiver system design was produced and a PCB was manufactured which then allow for each individual system to be tested, verified and characterised before the whole receiver system chain could then also be investigated. The PCB produced is given below in figure 40.

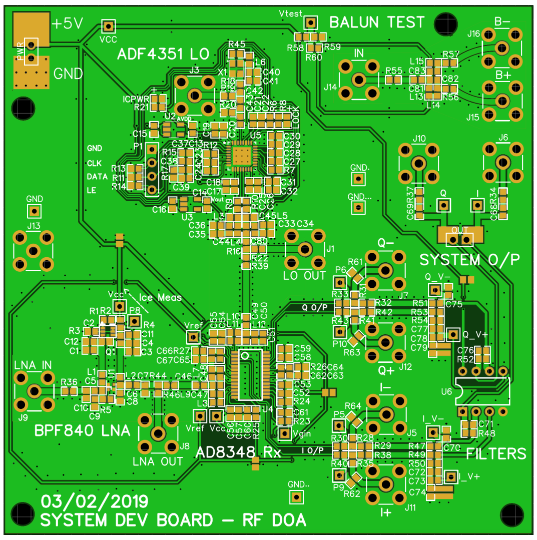


Figure 40: System development board. Includes test circuits for the BFP840 based LNA, AD8348 Demodulator, the ADF4351 LO, LC Baluns and the balanced filters. This is a 2 layer board, in which the bottom layer is a ground plane with as little breaks as possible. Coplanar Waveguide is used for RF traces.

4.2.7.5 Filter Development Board

A second development board (figure 41) was created to test and verify the opamp circuits; Differential to single ended, 2nd Order Butterworth, VREF buffer, and ADC input buffer. It also included a 4-way RF power splitter with inbuilt gain block.

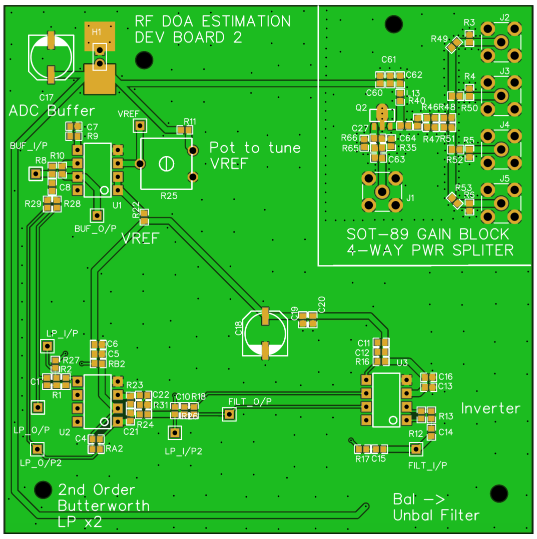


Figure 41: Filter development board which includes test circuits for the balanced filter, 2nd order Butterworth and bias voltage buffer circuit for the ADCs.

4.2.8 Final PCB Design

Once verification of each subsystem had been completed using the development boards, a final PCB (figure 42) was designed to produce a fully integrated system which included the 4 channel RF IQ receivers, LO, ADCs and the CMOD-A7 FPGA. This board featured 4 layers to allow for a dedicated ground plane and power plane, allowing the back layer to be an additional signal layer. The dimensions of the CPWG were altered from those used in the development boards to account for the reduced dielectric layer thickness between the ground plane and top layer (RF signal layer).
The board was designed such that each of the RF front ends for the different channels were close to identical in shape and configuration to reduce differences in their characteristics. The LO drive traces were designed to be the same length as mentioned previously and were divided to the demodulators using the resistive wye splitters.

The boards power supply features a standard 7805, meaning any DC power supply in the range +7 to +12V is sufficient to power the board.

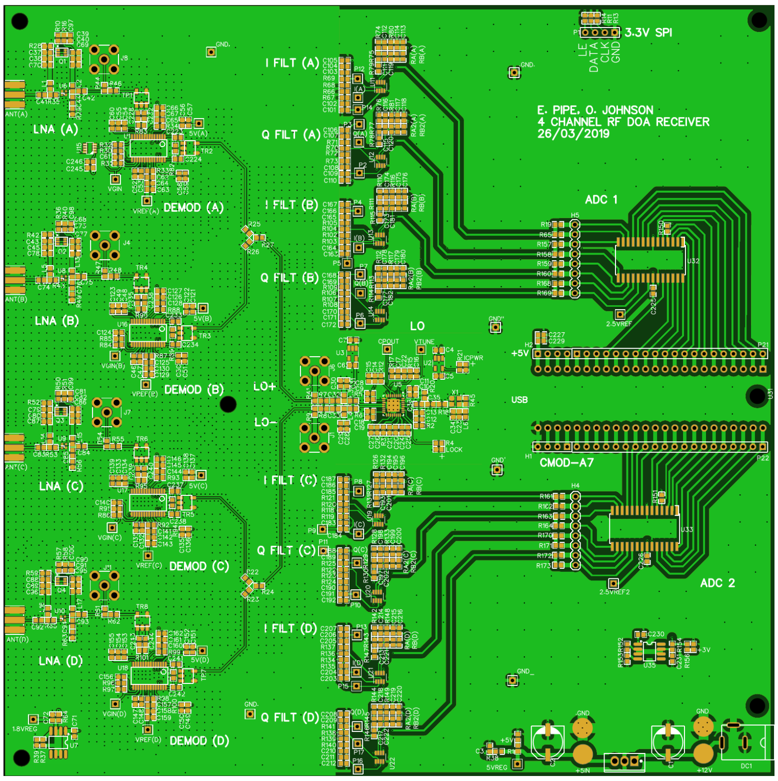


Figure 42: Final PCB design which includes 4 RF receivers, the LO in the centre, the anti-aliasing filters in the top and bottom centre, and the ADCs and FPGA to the right. CPWG is used for all RF traces and the RF section of the board features frequent ground stitching.

4.3 FPGA Digital System

4.3.1 Overview

The FPGA system performs the digital acquisition of the received signals, forming the interface between the receiver and GUI. For a modern receiver system the flexibility afforded by DSP is very important. In this case, the flexibilty allows for the implementation of multiple algorithms and to move the development process into a software environment.

4.3.1.1 Requirements

While there are limited explicit aims and objectives relating to the FPGA system, a list of requirements can be inferred to enable other systems to meet their aims and objectives.

- Have 15 I/O pins available per ADC
- Have 3 I/O pins available for SPI to oscillator programming
- Be able to communicate with the ADCs at 2MHz
- Be able to capture each ADC simultaneously and at a total data rate of 4 MSPS
- To be able to buffer at least 200 samples from each ADC channel
- To be able to process and send the captured data to the GUI
- Be able to program and control the local oscillator
- For the system to be “Plug and play”
- To be able to implement a DOA algorithm

4.3.1.2 Choice of FPGA vs DSPIC

To meet these requirements two feasible options were selected, a DSP specialised microcontroller and an FPGA. A brief investigation was performed into the suitability of each was conducted and the results are as summarised. Firstly flexibility, the FPGA being an array of digital logic blocks connected at the time of programming is much more flexible than the single core DSPIC. Both devices meet the requirement of a minimum of 33 I/O pins, with the Artix 7 FPGA available in packages with between 106 and 500 I/O pins and the DSPIC33e in packages between 28 and 144 pins [37] . The cost of each device has also been compared, an example FPGA development board the CMOD A7-35T for £68 [38] and the MIKROE-766 DSPIC development board for £23 [39] . While FPGAs are considerably more expensive, either device is within the budget. Both devices also meet the requirement for included communication protocols which the FPGA slightly leading over the DSPIC as Ethernet is included which may be used for future revisions.
The storage of samples relies on the memory available in the device, to meet the specification of 200 samples for 8 x 8-bit channels, a total of 3.2 Kilobytes is required. The CMOD A7 easily meets this both internally and there is a 512KB external SRAM if further storage is required. As DSPICs range from 0.25 - 20 KB of ram, some meet this requirement, however this is expandable if external ICs are used.
Most importantly, both options rely on specialised development tools to implement the design. The FPGA features a mature and well-documented development environment; with IP cores available to implement common functions required of the system. Furthermore, there is the option of multiple levels of abstraction within the process. Hardware description languages (HDL) can be used to design register level logic, high-level synthesis (HLS) allows programming in C and block diagrams for the greatest level of abstraction. The choice of abstraction is further extended to DSP design. Multiple tools are available to design, simulate and implement algorithms in Mathworks Simulink or HDL can be used to generate the same logic. The DSPIC tools are also mature, well documented and developed for however there is less choice of abstraction as most programming relied on register level control.
Overall it has been decided to proceed with using the FPGA for data processing in this project. This meets all aims, objectives and specifications required and provides the most likely chance of success due to the wide range of development tools and integration with familiar software.

4.3.1.3 Choice of FPGA

A wide variety of FPGAs are available, varying significantly in cost and number of digital logic resources. There is also the choice between a development board or single IC. Designing from a single IC provides the greatest flexibility and opportunity, however this comes with the greatest complexity and risk. Development/Evaluation boards provide an all in one solution but are less flexible and are harder to integrate into a final system.
As an appropriate development board, the CMOD A7-35T, is available, it has been selected to develop using this. This allows for faster development as it builds upon a verified platform, furthermore being an Xilinx 7 series device is has access to the Vivado design suite and other modern tools. The module is also in a package (DIP) suitable for integrating into the final design with the required number of I/O pins.

4.3.1.4 Dataflow

Overall control within the FPGA is achieved through the use of a Microblaze processor as a master to the AXI4 internal control bus. The Microblaze is a soft processor, and in general, is suited to applications requiring performing varied and sequential tasks. Furthermore, the C programming language lends itself to sequential operations such as the block level control for the sequence of this system.
This design uses a buffered sampling system. This is opposed to a streaming system, where data is continuously sent from the ADC through the processing system to the GUI. Such systems have additional complexity as they require precise timing of each component to avoid bottlenecks. In the design of this system, the data rate is limited by the UART serial connection chosen to interface with the GUI. When calculated the streaming data rate from 2x 8-bit ADCs at 2MSPS would be 32 Mbits/s however serial used at the maximum speed of 115200 baud is 115.2 Kbits/s. This is not a problem for this design as the samples are stored in SRAM before they are transmitted, allowing the estimated 1 second transfer time for a 2000 sample (1ms) capture.
Even though this is a buffered system it is good practice to reduce the latency as far as possible. This is achieved through using IP to avoid passing data through the slower processor, instead using direct memory operations and avoiding duplication of the data. The following illustration shows the data flow design. Each ADC is represented in the top of the diagram and the lower half represents the processing, combination and transfer of the data.

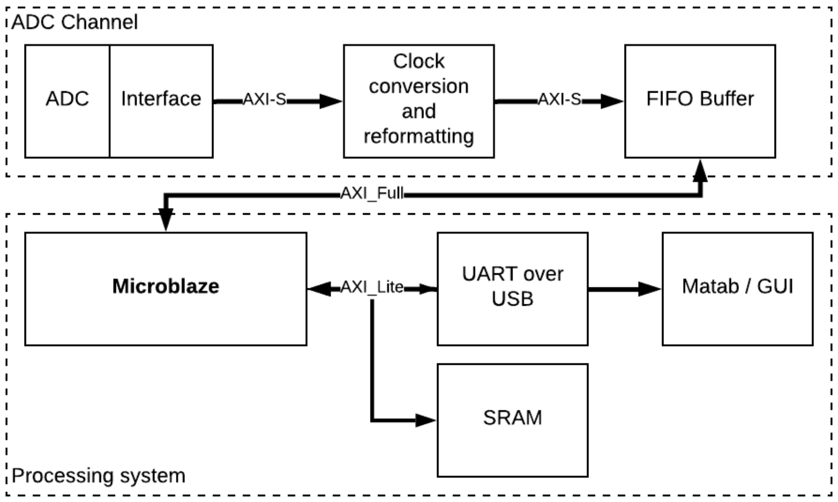


Figure 43: Dataflow within the FPGA.

4.3.2 Microblaze

4.3.2.1 Theory

As stated earlier the MicroBlaze soft processor has overall control of the FPGA system. The Microblaze is a 32-bit RISC soft processor [40] . A soft processor differs from regular processors in that it is wholly implemented using the internal logic of the FPGA. This increases the flexibility of the processor and in this design has been customised to suit the tasks required. Implementation within the FPGA also significantly increases the speed of transferring data from the internal logic to the processing system, only paralleled by SoC (System on chip) implementations such as the Xilinx Zynq.
Successful overall control of the sampling system requires the processor to:

- Interface with all peripherals and memory
- Display the current step in the sampling sequence to the user
- To interface with the GUI.

Control for peripherals is through the "Xilinx bare-metal drivers" these provide functions, documentation and examples for interfacing each IP core from within the processor using the AXI interface. Examples of the functions provided are configuration lookups for locating the base address of devices, configuration and initialization functions, and functions to control all functions.

4.3.2.2 Implementation and analysis

The MicroBlaze processor allows for user configuration of the design parameters to meet the users required specifications. In this case, these are:enabling the peripheral AXI connection, connecting to the 100MHz AXI clock so conversion is not required for interfacing and for implementation with optimisation towards minimum FPGA area utilisation. This results in a final utilisation of 632 (3%) lookup tables (LUTs) and 228 (0.5%) flip flops (FFs), overall this is a negligible amount of usage for such an important part of the system.
Programming the MicroBlaze is implemented through a C source file, which lends itself towards sequential operations and provides a rich debugging environment. For example, the Xilinx SDK tools provided allow for user breakpoints and reading of register and memory space as the MicroBlaze is running.
Overall the MicroBlaze system is successful in controlling the sampling sequence and does not introduce a significant delay into the transfer of samples. The use of LEDs and buttons allows the user to know the state of sampling and to start or reset the sequence. The design also retains flexibility for future implementation of faster internal transfer techniques such as DMA and faster external transfer protocols such as Ethernet if required.
A simplified final design of the software is represented below.

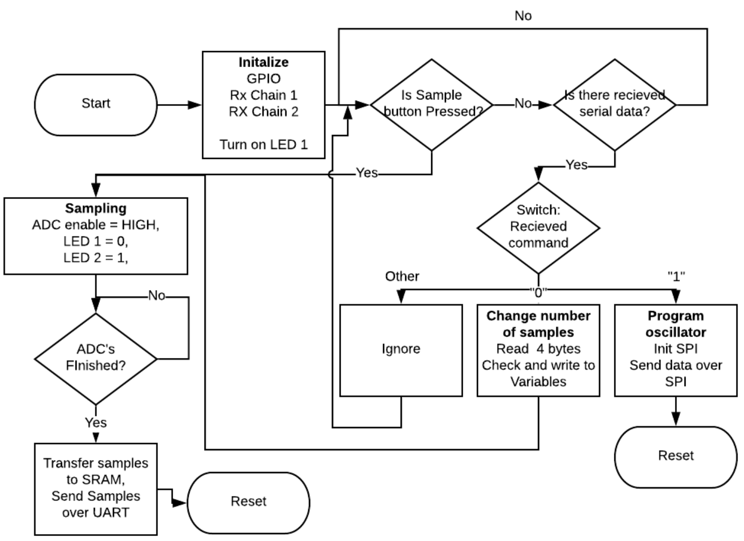


Figure 44: MicroBlaze program.

4.4 ADC Interface

4.4.0.1 Theory

The ADC interface encompasses generating the appropriate signals to enable sampling, to capture the sampled data and to process into an appropriate format for processing and storage. The ADC selected (AD7829) uses a parallel interface to output the data in an 8-bit wide bus with other signals used for the control of sampling [41] .
The format chose to transmit the received data within the FPGA is AXI-Stream. This is recommended by the chip vendor Xilinx, for applications where there is a continuous stream of data. Furthermore, there is a significant amount of IP available for manipulating and controlling this interface, destination addresses are not required reducing complexity and overheads, and it has data rates suitable for future faster ADCs.
The most important factor for the system performance determined by ADC characteristics, is the noise figure. This is determined by imperfections in the ADC design such as a finite number of bits causing quantisation error.
All parameters except the signal to noise ratio (SNR), are determined by the external system the ADC is placed in. The SNR can be taken from the ADC datasheet as 48 dB and is used in the system cascade analysis.

4.4.0.2 Implementation

Programming of the interface was completed fully within the HDL Verilog. This is a specialised language commonly used to describe the structure of digital logic. The biggest difference between HDLs and traditional languages is that HDLs allow for multiple simultaneous and synchronous operations. To first define the ADC interface a diagram was created to show the control flow of the interface from which the interface was subsequently created.
Starting with the clocking of the interface, an external 8MHz clock is created using a clock wizard within the FPGA. This is then divided twice to form the 4 MHz and 2 MHz clocks required for synchronising the interface. This method has been selected as it efficiently generates both the required frequencies and to reduce the number of clock tiles, as this is to 4 within the FPGA fabric.
For sampling to be triggered the core must have the enable input pulled low and the reset input pulled high. Enabling is controlled by the MicroBlaze processor and the reset is managed for the entire design by a clocking and reset system block such that it is synchronous to the design clock.
Looking at the external ADC interface, the specification for which is shown in the AD7829 datasheet. To start sampling the CONVST (Conversion start) needs to be pulled low, in the design this occurs on the falling edge of the 2 MHz Clock. At the same time, it is required to increment the address bus to the next channel to be sampled. The ADC will then signals that sampling is complete by pulling EOC (end of conversion) low. Within the FPGA these signals are connected to the ADC RD (read) and CS (chip select) pins to allow for the data to

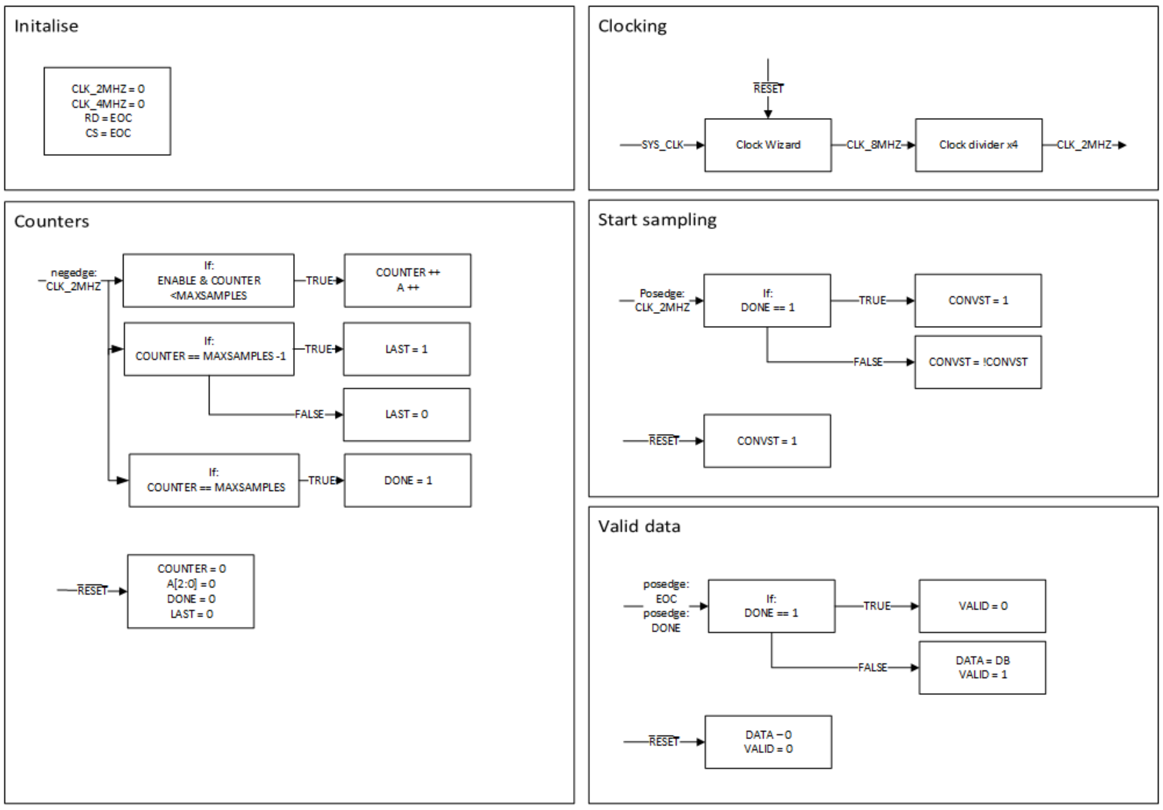


Figure 45: Overview of the ADC - FPGA interface

be read immediately. To allow for setup time, the data is read on the rising edge of the next 4MHz clock such that it is offset from the end of conversion. The sampling process is then repeated on the next falling edge of the 2 MHz clock.
The second interface of this design is the AXI-Stream bus for internal transmission within the FPGA. AXI-Stream consists of an 8-bit data bus, Clock, tvalid and tlast signals. The clock for this bus is the same as the 2 MHz sampling clock as this requires no buffering and the 100 MHz system clock is not present within this block. The tvalid signal is implemented such that it signals there is valid data on the bus so that the slave device knows to receive data. This is important as the clock within the bus is always active and this is the only method of signalling there is no new data. Tlast is controlled through the logic that counts the number of samples performed and compared to the number of samples parameter configured at the time of design.

4.4.0.3 Analysis

To verify the operation and tolerance of the design a functional and timing analysis was performed. This was achieved through test benches that generate/simulate the signals from the ADC. To ensure time accuracy of these the appropriate values were used from the ADC datasheet. From this functional simulations were conducted to ensure the interface was operating as expected. When these were successful, a further timing analysis was performed. A timing analysis requires full FPGA implementation and takes into account the propagation delays within the digital logic hence takes longer to perform than a functional simulation. The output of these simulations is shown
After the successful simulation, the interface was implemented on the FPGA to further verify successful operation. Using a logic analyzer the signals were read and verified to match the simulation.
To verify the compatibility of the internal AXI interface, and Xilinx ILA (Internal logic analyser) was used. The ILA is an IP core that implements a logic analyser within the FPGA logic, that can be read from the design tools within Vivado. The logic analyzer is designed to decode AXI signals providing they meet the specification for the bus. Decoding was successful however the implementation of the tready signal was incorrect. It was implemented as a signal enabling tvalid to be pulled high, starting the transmission, however, tready is only pulled high after tvalid goes high. To work around this, tready is disconnected from the bus and connected to a constant high signal. This is appropriate as within this system the next block would always be in a state ready to receive data, and the tready signal is not a mandatory part of the standard.

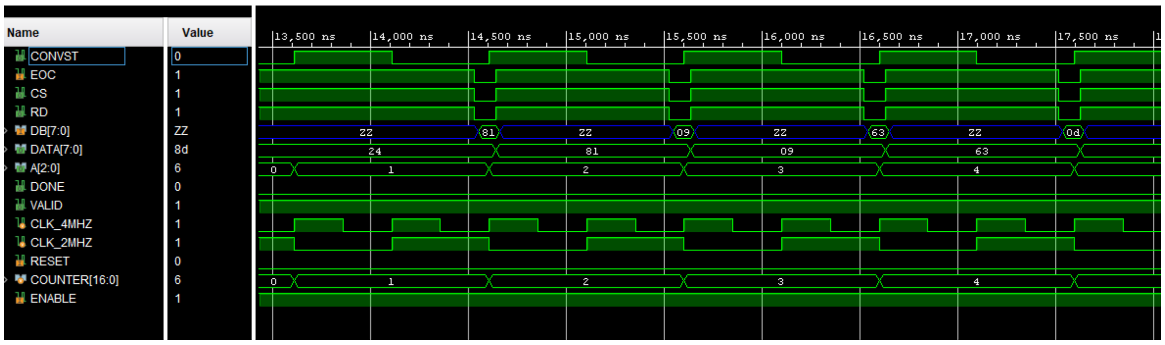


Figure 46: Timing simulation of the ADC interface signals

4.4.1 FIFO Buffer

4.4.1.1 Theory

A first in, first out buffer is a form of temporary storage commonly used to interface between systems working at different schedules. The FIFO forms a queue of streamed data from the ADC until is ready to be processed by the MicroBlaze. Processing of the signal is required to maximise the storage within the buffer. This involves converting the clock and data frequencies and then combining 4 samples to meet the minimum FIFO width.

4.4.1.2 Implementation

The FIFO chosen for this implementation is the Xilinx AXI FIFO: Memory Mapped to Stream. The AXI FIFO is design specifically for this task and contains many user controllable parameters. Over the default configuration, the transmit buffer has been disabled as this is not required and reduces FPGA utilisation, the depth has been increased to 8192 samples and the full signal threshold to 8000 samples. The FIFO is implemented in the internal block RAM of the FPGA and includes the logic required to control the flow of data in lookup tables.
As mentioned earlier clock conversion is required. This is because the FIFO core does not support asynchronous clocks between the Stream and AXI4 interfaces. Requiring conversion of the 2 MHz stream to the FPGA 100 MHz clock. The AXI-S Clock Converter IP is used for this, as it supports 1:N clock conversion between asynchronous clocks.
Further processing is then required to combine 4 packets into one 32-bit wide packet. This is since the minimum width of the FIFO is 32 bits. If a combination was not performed, the lower 24 bits of each FIFO frame would be unused.
The streaming chain between and including the ADC interface and FIFO is duplicated for each ADC. This is so that a successful system can be implemented for a single channel and then duplicated for the other ADC used, with the possibility of easily expanding the ADC count further.

4.4.1.3 Analysis

Verification was performed using the internal logic analyzers and output from the MicroBlaze serial terminal. This shows that the device is handling data correctly and is not losing samples. FPGA utilisation for each FIFO has also been analysed, this is 17% of BRAM and 7% of LUTs. For a total utilisation between both channels of 35% BRAM and 18% of LUTs. Though high this is not a problem as the usage when combined with the 25% BRAM utilisation from the MicroBlaze memory, does not exceed the total available. This would pose a problem if the number of samples captured was required to be significantly increased however this could be mitigated through the switch to direct memory access (DMA) allowing for direct storage in the larger external SRAM.

4.4.2 SRAM Storage

4.4.2.1 Theory

The 512KB external SRAM is used to store samples from the FIFO after sampling is complete and before they are transmitted to the GUI. This would allow for samples sizes larger than the size of the FIFO without the complexity of the DMA solution if data was transferred to SRAM while the sampling is occurring. Implementing SRAM storage focuses on the interface between the SRAM IC and the internal address space of the MicroBlaze processor.

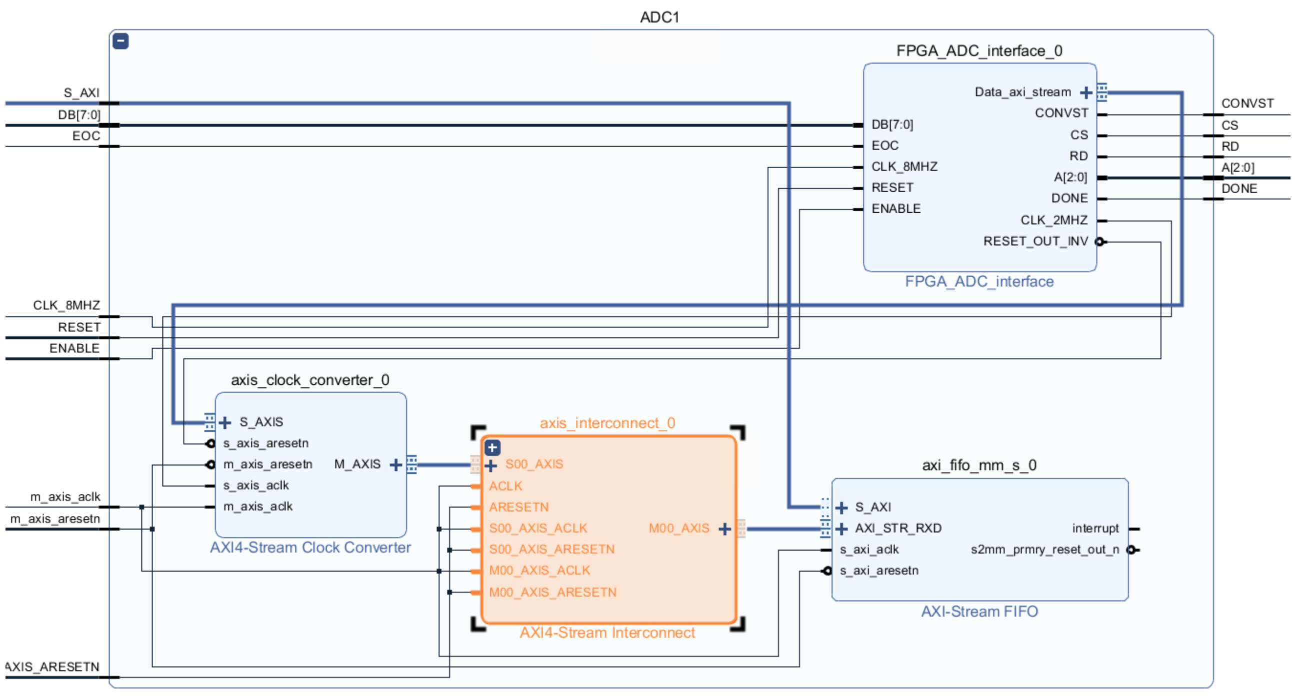


Figure 47: Final implementation of the ADC interface and storage

4.4.2.2 Implementation

The Xilinx external memory controller (EMC) is utilised to interface with the external SRAM IC. This is an all in one controller, adaptable to many different storage devices, creating an interface between the external memory connections and the internal FPGA AXI Bus. The SRAM is then presented as a memory device in the address space of the processor to allow for easy communication.
For the EMC to be configured successfully it is configured with timing parameters from SRAM datasheet, for example, this IC has a hold time of 8ns. Other parameters are automatically inferred such as the AXI bus parameters.
Confirmation that the device was functioning correctly was performed to ensure the timing characteristics were met. To do this the MicroBlaze was programmed to write to the SRAM then confirm that what was written is the same as what is being read back.

4.4.2.3 Analysis

The external memory controller is the simplest method of implementing an interface with an external memory IC and benefits from the testing and verification of professional IP. From verification, during the design process, it can be seen the core works as expected and does not use significant FPGA resources. There is, however, the concern that that 512Kb of SRAM is only able to store $\frac{1}{8}$ second long of samples, this is not currently a problem and can be mitigated through the use of further external SRAM devices.

4.5 User Interface

4.5.0.1 Theory and Aims

To visualise the captured results and DOA algorithms it is required to create a graphical user interface. The main aims of this are summarised:

- To visualize the raw data from the ADCs to confirm the signals are as expected
- To visualize the output of the DOA algorithms
- Including the power spectrum
- Visualise the estimated angle of arrival for each algorithm

The GUI must also handle communication with FPGA to trigger sampling and set user-defined parameters.

4.5.0.2 Implementation

The method selected to implement this was to use the MATLAB app designer software, chosen because is a commonly used program in research and teaching environments. The GUI design process is a combination of using a GUI to place tabs, buttons and plots then adding functionality to these by associating actions with coded functions.
The first step was to implement a serial connection monitor. This gives the user a choice of the serial port and then the options to connect, disconnect or refresh the available devices. This view also shows the current state of connection and performs basic checking to ensure the serial port is accessible, warning the user if it is not. When connected the user is able to program the oscillator to the LO frequency required to receive at 868 MHz.
The next view available to the user is the RAW data plots. This plots the voltage received at the ADCs as a function of time, the user is required to select the number of samples and clicked the "Get data" button. When this is pressed, the GUI sends a serial command to the FPGA triggering sampling and transmission of the data. To manipulate the data, controls for moving, zooming, and resetting the plots is implemented and there is a button to export the data to location of the users choice as a .csv file. This allows for calibration if the user overwrites the calibration data with the previously captured signal.
The final view, DOA, presents a plot of power density against angle and a table summarising the estimated arrival for each of the DOA algorithms. The "Get DOA" button on this view, performs the same function as the "Get Data" button but includes the processing of the received data using the DOA algorithms.

4.5.0.3 Analysis

A GUI was successfully created and further analysis with respect the aims of the project is available in the results section.

4.6 DOA Estimation

There are many methods for DOA angle estimation. In this project, various beamscan and non-parametric methods are explored. The aim of this was to develop an algorithm that was suitable for implementation on an FPGA.

4.6.1 Simulated receiver signals

The first stage in developing a DOA algorithm was the simulation of received signals. These would be used to verify the function of the algorithm.

4.6.1.1 MATLAB Phased Array System Toolbox

Simulink was used because of the Phased Array System Toolbox, which allowed for the simulation of received signals by specifying parameters of the system. This made it easy to generate signals at various angles, as a variable could be passed from the workspace in MATLAB to the Simulink model. Figure 48 shows the block model diagram of the system used to generate signals. The Narrowband RX array could be used to specify the array geometry, whilst the Receiver Preamp could be used to specify noise power.

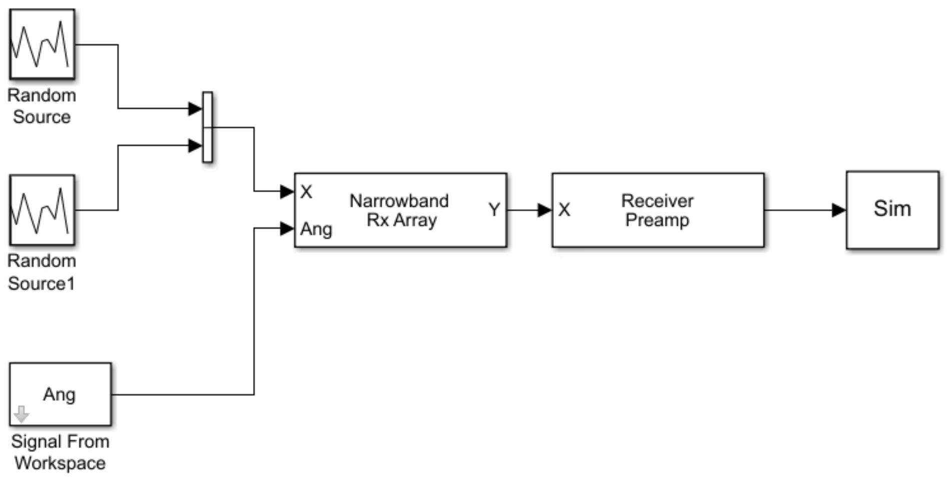


Figure 48: Simulink model of RF narrowband receiver with 4 element ULA

Further on in the project, a similar method to that described above was found. This allowed for the generation of received signals in a MATLAB script, making it easier to make changes to the system by simply changing a variable in the workspace. In this method, operating frequency, distance between elements, number of samples and noise power were specified. The angle of arrival could also be provided, which would be used to validate the results of the algorithms.

4.6.1.2 Phase Shifted Sinusoids

Finally, a set of phase shifted sinusoids were also used. This involved adding linear phase shift to a set of signals that would correspond to the array geometry and mimic the nature of delays caused by a signal positioned off axis. The signals also included additive white gaussian noise (AWGN), which could be varied to provided information about the level of signal to noise ratio (SNR) required for the prepared algorithms to function adequately. The added phase shift could then be related to the angle of arrive through equation (5), which could be used to confirm the function of the algorithms.

4.6.2 MATLAB DOA Estimation Implementation

For the implementation of DOA algorithms in MATLAB, a few beamscan-based non parametric methods were explored.

4.6.2.1 Beamscan methods

The basic Beamscan search algorithm as detailed in equation (6) was first implemented. This involved the design of a array manifold matrix that could be manipulated for a given angle θ. The limit of this search would be the increment of which this scan was performed. At least 180 points would need to be searched to provide a resolution of 1 degree. 18000 points were used in the implementation to provide resolution of 0.01 degree. Listing (3) in the appendix shows the code that was used to generate the array manifold.

After creating a suitable array manifold, an estimate for the received signals was required. This could be done using MATLABs own covariance matrix function ‘cov()’, but instead the estimation followed that outlined in equation (4). This calculation can be seen in listing (4), where the code iterates through the number of snapshots specified and multiplies each row of the input signal by its complex conjugate. This is then summed and then divided by the number of snapshots to provide the correlation matrix $\hat{\mathbf{R}}$.

Once these two matrices were calculated, a search was iterated through the angle θ through 18000 points and the power spectrum shown in equation (6) was found.

Similarly to the beamscan method, the MVDR method also required both the array manifold and correlation matrix of the signals. This algorithm then minimises the inverse of the correlation matrix multiplied by the array manifold matrices to find the peaks of the spectrum. This can be seen by equation (7).

4.6.2.2 Other methods

The QR decomposition method made use of the noise subspace of the correlation matrix of the input signals. The MATLAB 'qr' ([Q R] = qr(A)) function was used to do this. The Q matrix of the decomposition can be divided into noise and signal sub matrices, where the signal occupies the columns 1 to L, and the noise occupies the columns L+1 to M. For the array in this system, L = 1 and M = 4, so the Q_n matrix is specified as the lower three rows of the Q matrix. The orthogonal relationship of the Q_n matrix and the A manifold matrix, means that the DOA can be found through a peak search, as shown in equation (10).

4.6.3 Simulink DOA Estimation Implementation

After experimenting with various non-parametric methods in MATLAB, a parametric method was trialled in Simulink with the use of Xilinx Model Composer. The method to be implemented was a variation of the ESPRIT algorithm outlined in section 2.3, with the use of LU decomposition to provide the signal and noise space in the lower triangle matrix L and the upper triangle matrix U. After decomposing into L and U matrices, the noise and signal spaces could be separated using a selection matrix to create US1 and US2 (1x3) matrices.

4.6.3.1 Xilinx Model Composer

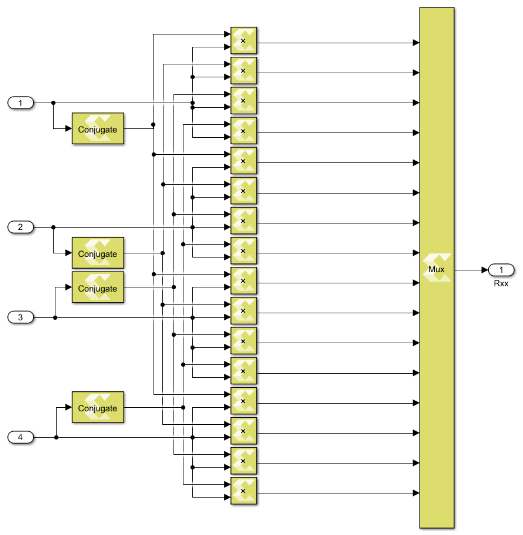


Figure 49: Covariance Estimation Sub-System

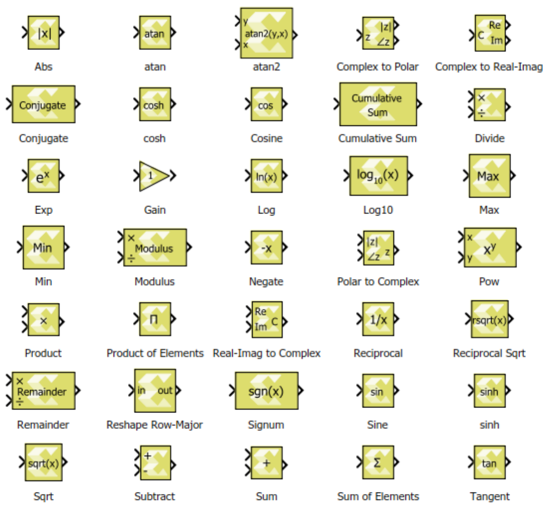
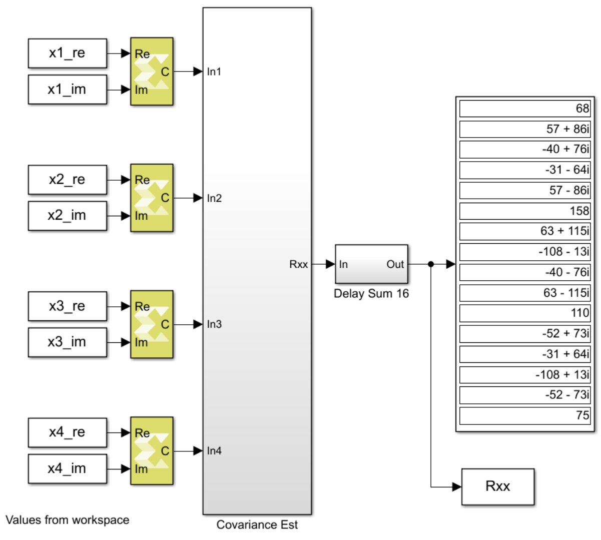


Figure 50: Maths blocks available through Xilinx Model Composer Library

For the implementation in Simulink using Model Composer blocks, the algorithm had to be broken down into smaller functions. For example the covariance estimation function seen in figure 49. required 16 multiply blocks. Other systems such as the inverse matrix block required more resources that usual because of the need to process complex signals. Whilst some Model Composer blocks were capable of handling complex data, functions such as square root were not able to accept complex inputs, so they also needed to be developed. Models created using these tools must use the blocks available through the Xilinx Model Composer library (figure 50) in order to be implemented properly.

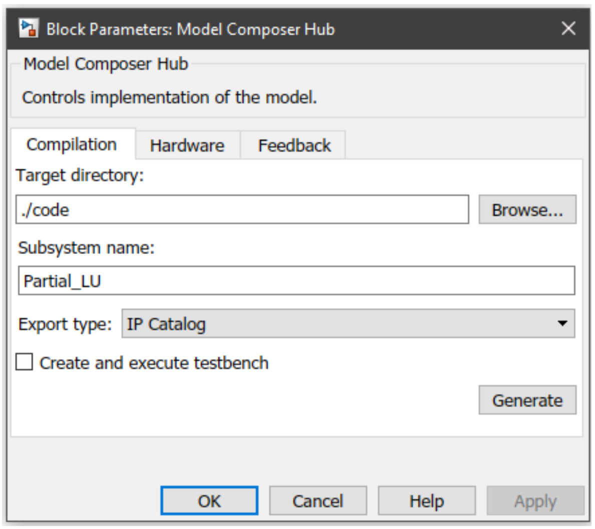
To test parts of the system, a known set of value were imported from the MATLAB workspace into the system. The data was then fed through the operation such as covariance estimation, and then exported back to the workspace for verification. A display block was also used in Simulink for a faster visual verification during development. This can be seen in figure 51.



```
68
57 + 86i
-40 + 76i
-31 - 64i
57 - 86i
158
63 + 115i
-108 - 13i
-40 - 76i
63 - 115i
110
-52 + 73i
-31 + 64i
-108 + 13i
-52 - 73i
75
```

Figure 51: Covariance calculation clock with workspace import/export and display.

Exporting the system as IP could be done via the Model Composer hub (figure 52). By selecting the hardware version, specifying clock frequency and throughput factor, subsystems could be exported as IP catalog to be used with Xilinx Vivado. This process automatically optimised processes for implementations onto Xilinx FPGAs and also automates the generation of code in hardware description languages (Verilog and VHDL) as well as utilisation estimates (figure 53).



Block Parameters: Model Composer Hub

Model Composer Hub

Controls implementation of the model.

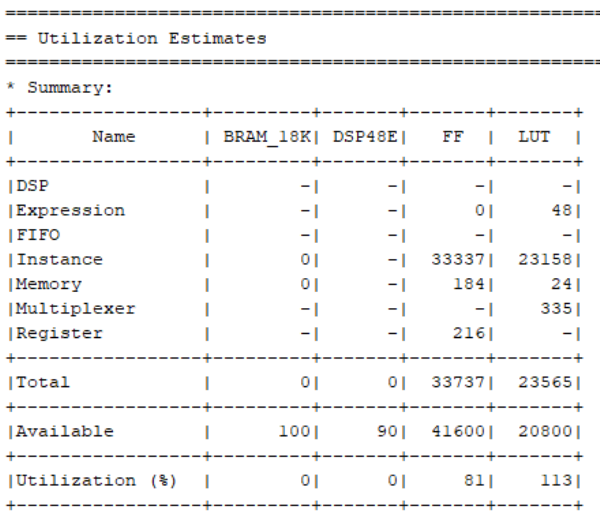
Target directory: ./code

Subsystem name: Partial_LU

Export type: IP Catalog

☐ Create and execute testbench

Figure 52: Model Composer Hub.



```
============================================================
== Utilization Estimates
============================================================
* Summary:
+-----------------+---------+-------+-------+-------+
|       Name      | BRAM_18K| DSP48E|   FF  |  LUT  |
+-----------------+---------+-------+-------+-------+
|DSP              |        -|      -|      -|      -|
|Expression       |        -|      -|      0|     48|
|FIFO             |        -|      -|      -|      -|
|Instance         |        0|      -|  33337|  23158|
|Memory           |        0|      -|    184|     24|
|Multiplexer      |        -|      -|      -|    335|
|Register         |        -|      -|    216|      -|
+-----------------+---------+-------+-------+-------+
|Total            |        0|      0|  33737|  23565|
+-----------------+---------+-------+-------+-------+
|Available        |      100|     90|  41600|  20800|
+-----------------+---------+-------+-------+-------+
|Utilization (%)  |        0|      0|     81|    113|
+-----------------+---------+-------+-------+-------+
```

Figure 53: Utilisation estimates for delay and sum block.

4.6.4 System Integration

4.6.5 Calibration of the receiver

Since it was discovered that each of the channels of the receiver produced different phase shifts in the outputs for a constant phase input, calibration was required in order to allow the system to accurately estimate the DOA.

To do this, the algorithm must have a method of calibration to account for the phase and amplitude discrepancies caused by both the receiver and the ADCs.

Firstly, the signal must be up-sampled to improve the resolution of the phase correction and alignment.

The sampling frequency of 500kHz and received signal frequency of 20kHz gives 25 samples per period of signal received, resulting in a minimum detectable phase difference of 14.4 degrees (360/25). This limits the DOA angle estimation to a minimum of 4.59 degrees resolution.

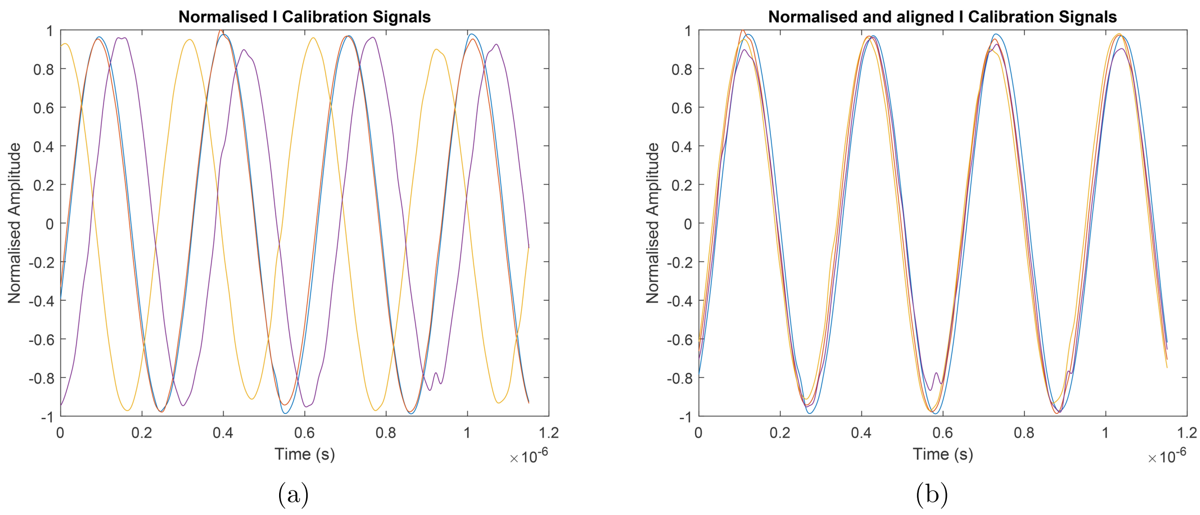
Up-sampling by a factor of 10 results in a effective sampling frequency of 5MHz, resulting in 250 samples per period and a minimum phase difference of 1.44 degrees (360/250). This improves the estimation of the algorithms by increasing the minimum detection to 0.459 degrees.

The up-sampling function used in the MATLAB implementation can be seen in listing (5), which requires the input signal and the scale factor desired. The MATLAB function spline() is a cubic spline date interpolation, which has smaller error than linear interpolation and results in a smoother curve. However, this method reduces the number of usable samples (so up-sampling 200 samples by a factor of 10 results in fewer than 2000 samples) because of the interpolation used.

The signals are then normalised to have amplitude between -1 and 1, which removes any differences in amplitudes between receiver channels.

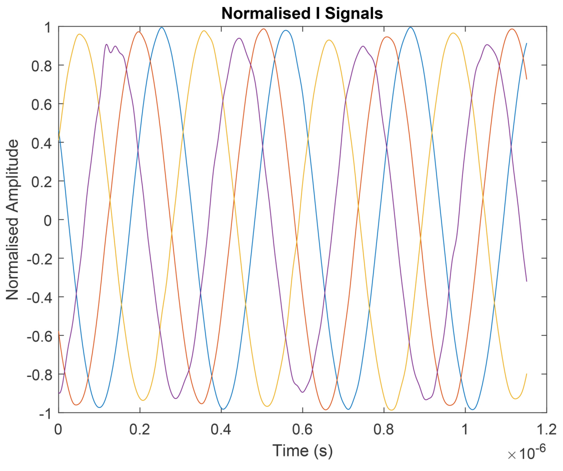
The phase difference is then calculated and converted into an integer number of samples that the misaligned signals can be shifted by. The calibration can be seen in figure 54, where the calibration signals are measurements taken in an anechoic chamber with a source positioned at 0 degrees 57. After calibration, the signals in phase, which is as expected with a source positioned perpendicular to the array. This can be confirmed by calculating the DOA with another set of data where the source is positioned in a similar way.

Figure 54: Calibration signals: (a) before alignment, (b) after alignment.

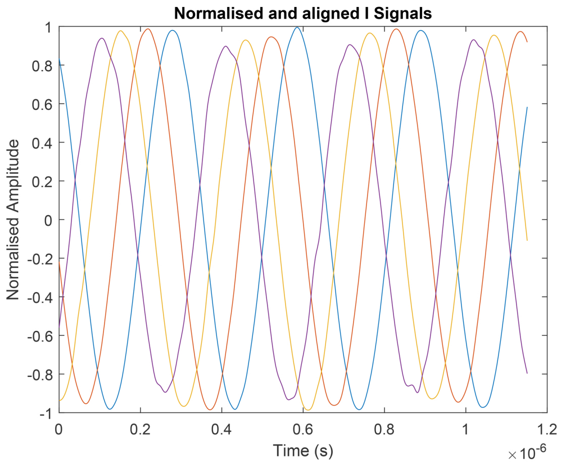


Once the algorithm has been calibrated, real data can be used to retrieve DOA angle spectra and estimates. For a signal positioned approximated 20 degrees off axis, the signals before and after alignment can be seen in figure **??**. Despite the fact that the two sets of signals look similar, the phase changed can be observed by the order of the signal colours.

Figure 55: Data signals: (a) before alignment, (b) after alignment.



(a)



(b)

The various spectra are then calculated with the newly aligned signals, as shown in figure 56. The peak searches result in angles of 22.34, 22.99 and 22.34 degrees for the respective DOA methods.

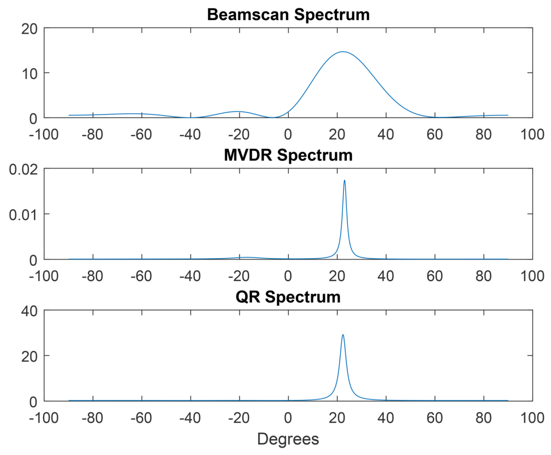


Figure 56: DOA spectra for real data with source positioned at 20 degrees.

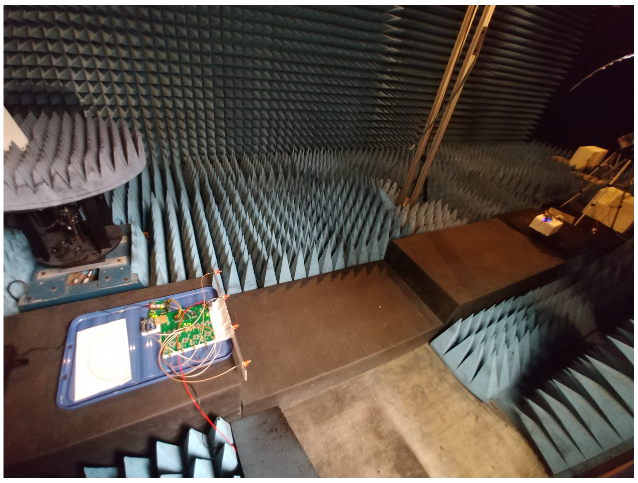


Figure 57: System testing inside the anechoic data.

5 Results and Discussion

5.1 Design and Construct Quadrature RF receiver with operational frequency range 50MHz - 1GHz.

Each circuit element was verified to work as designed with expected variation due to component tolerances. The constructed LNAs have slight variations in gain, which is reflected in gain differences at the output of each receiver. A plot of the S21 frequency response, obtained using a VNA, of the LNAs can be seen in figure 58, and the specific values measured at 868MHz are shown in table 6, the plot shows slight gain imbalances over the span, and highlights other frequencies at which the amplifier might also work well. Each AD8348 demodulator was verified to work as specified by the datasheet. For accurate measurements at 868MHz an RF signal generator and spectrum analyser were used. The gain of these LNAs is less than the gain of the development LNA.

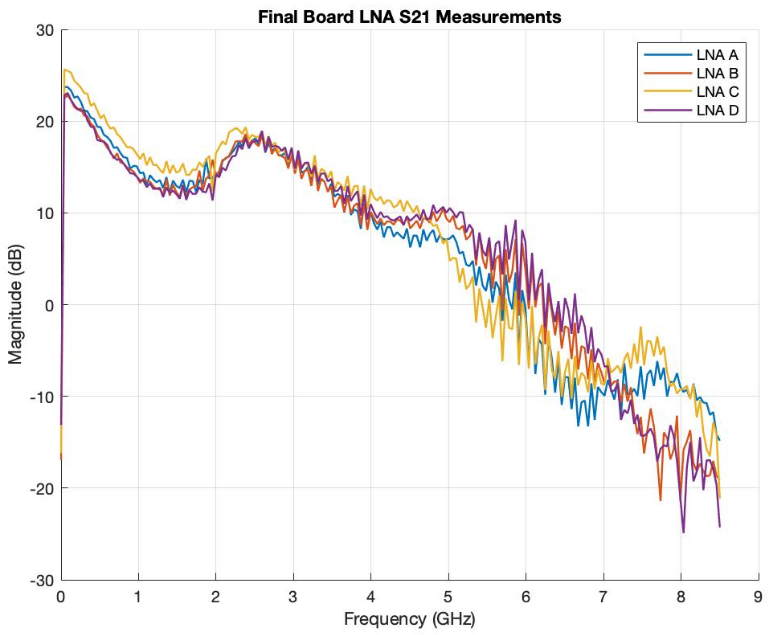


Figure 58: VNA measurement of the S21 (power gain) frequency response of each LNA on the final board.

LNA	A	B	C	D
I_C / mA	2.9	2.2	5.9	2.9
Gain at 868MHz / dB	15.25	14.51	16.95	14.38

Table 6: Measured gains and collector currents of each LNA. The maximum deviation between gains is 2.57dB.

Two filters were used for anti-aliasing at baseband; an RC lowpass, and a 3rd order active filter. The RC lowpass is a standard design. The active filters were simulated to perform a 3rd order response, and balanced to unbalanced conversion. This conversion give a 6dB gain from within the filter stages. Within the active filter block is a standard 2nd order butterworth VCVS design, to add extra gain to a channel, one resistor can be changed. The active filters were verified to have the same properties as the simulated versions.
SPI control of the ADF4351 was developed first from an Arduino Uno, then later from the FPGA itself. When using the Arduino uno, the oscillator frequency and output power available for the user to change and control from within the GUI. Further control of parameters such as charge pump current and reference frequency are available to change from the program file which is then uploaded to the microcontroller over UART. The FPGA uses a copy of the calculated registers from the microcontroller to perform this function. The values used are 1.736Ghz at minimum power, this is suitable as it decreases the complexity of the FPGA design and is not required to be changed further.
The customised ADF4351 local oscillator design was implemented successfully into the final system. To evaluate the system's performance, the output spectra of the bought LO, the test LO (from section 4.2.7.4) and the final

system LO were recorded using a spectrum analyser. Figure 59 shows the spectra recorded when each of the LOs were set to output 1.736GHz on an output power setting of +2dBm (the required frequency for the correct mix down of 868MHz inside the demodulators).

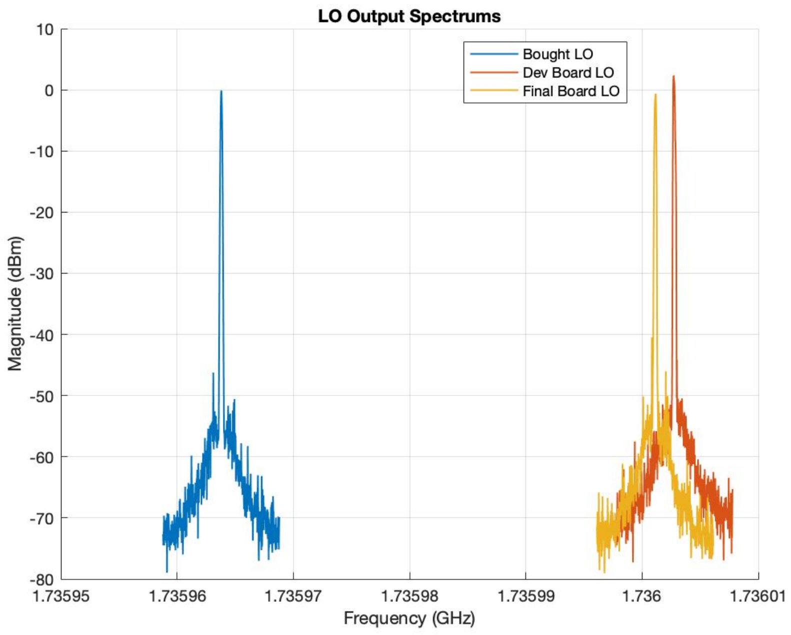


Figure 59: RF output spectrums recorded for each of the oscillators when set to the same lock frequency of 1.736GHz.

Analysing the position of the peaks in MATLAB and subtracting the lock frequency from the desired 1.736GHz, the frequency error in each LO was calculated and the results are summarised in table 7 below:

	Locking Frequency Error (kHz)	Peak Output Power (dBm)
Purchased LO	36.183	-0.1356
Development PCB LO	2.717	2.3509
Final System LO	1.180	-0.6343

Table 7: Frequency offset errors and measured output power for all the tested LO systems.

The drop in recorded output power from the setting of +2dBm is overall a product of the SMA cables used in the measurement (1m SMA cable insertion loss was measured to be 1.3dB) while the large discrepancy shown by the Development PCB LO can be explained by the discovery that this LO failed to show any power setting change when re-programmed. It is not understood what prevented this feature from working since the lock frequency setting still worked correctly and both power and frequency are set by the registers in the ADF4351 which are in turn adjusted via SPI. If one feature works it would imply the other should also work.

Looking at the lock frequency errors, it can be seen that the custom LO design produces a significantly reduced error. This will be a product of the high accuracy TCXO used. The discrepancy between the development PCB LO and final system LO are most likely be due to TCXO tolerance as both are within the specified limit. These results are significant as the bought LO's frequency error is comparable to the theoretical bandwidth of the whole system and, if unaccounted for, this would lead to the mix down products from the demodulator stage being significantly out of band (mixed down to much larger than 67kHz/34kHz instead of less (for a 4-element/8-element array)). Referring back to the discussion in section 4.2.6.2, this would then introduce phase errors outside of tolerance and would result in a reduction in the overall system's DOA estimation accuracy and would also cause the output signals to be attenuated at the anti-aliasing filters.

Overall a working 4-channel receiver was designed, constructed, and tested. Depending on the LO frequency, it can demodulate an RF signal over the specified frequency range with a bandwidth of 34kHz. The average sensitivity was measured as -90dBm (at the anti-aliasing filter output) for an SNR of 0dB (minimum SNR required by the QR algorithm in simulation). The final board was designed so that two boards could be connected together to form and 8 channel receiver, using only the ADCs from the first board. The conversion gain of each receiver is shown in table 8 Taking an average of the Gains, and using the maximum deviation between gains, it can be found that the final receivers have a gain of 73.52 +/-1.677 dB. With a maximum deviation of 3.354dB, the difference in the amplitude of C and D is a factor of 2.25 in linear.

Receiver	A		B		C		D	
	I	Q	I	Q	I	Q	I	Q
Gain / dB	74.34	74.34	72.79	72.79	75.19	75.06	71.84	71.84
Difference from C (I) dB	0.8550	0.8550	2.409	2.409	0	0.1368	3.354	3.354

Table 8: Conversion gain for each channel of the receiver.

5.2 Design, Simulate and construct phased array antenna

Two different antenna topologies were successfully designed, simulated and tested for a design frequency of 868MHz. It was concluded that the half-wave dipole was more suitable due to its larger return loss (both simulated and tested), ease of manufacture and reduced cross-sectional area in the azimuth plane. Figure 60shows 16.2cm long half-wave dipoles produced using the PCB housing (which included pads for matching), mounted onto a frame using the back part of the 3D printed casing only.

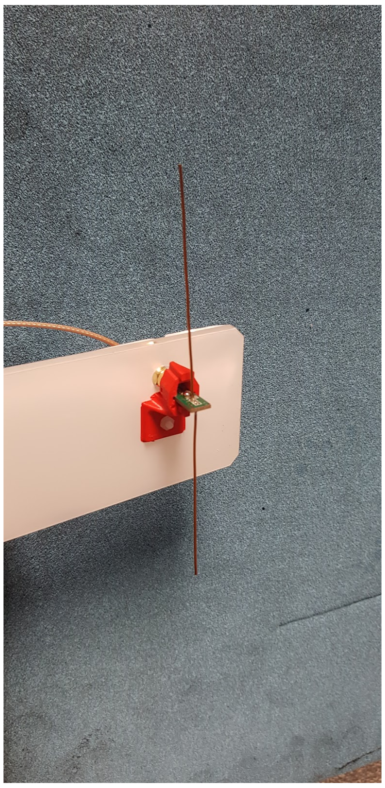


Figure 60: One of the 4 final system antennas which is shown mounted in the final system.

While the matching networks did not function as desired, this did not impact the antennas overall functionality. The return loss of the 4 final antennas were measured using a VNA and the results are given below in 61

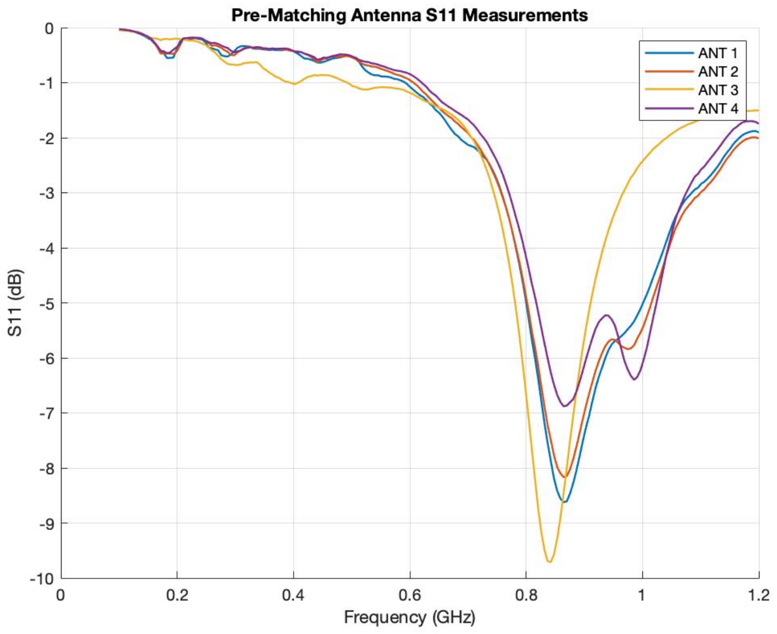


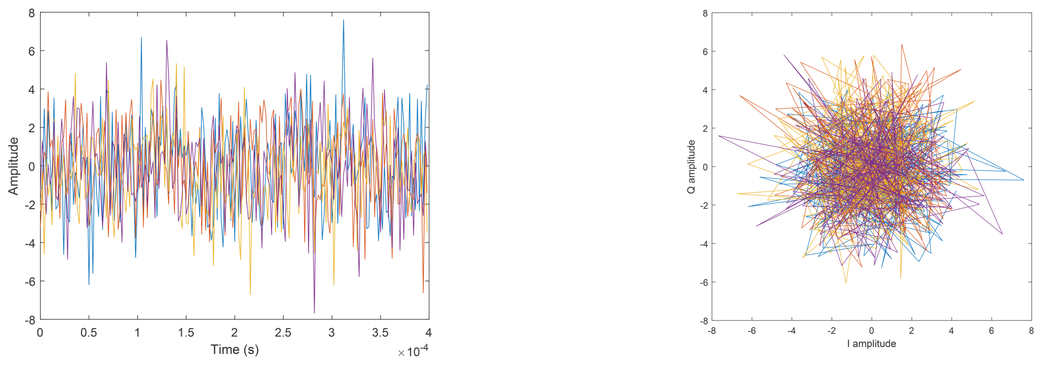
Figure 61: Measured S11 of the antennas used in the final system. The peaks are all close to 868MHz and have magnitudes between -7dB and -9.7dB.

5.3 Design and simulate DOA algorithm

5.3.1 Synthesise quadrature test signals

Using the Phased Array System Toolbox in MATLAB, the simulated received signals for a four-element ULA can be seen below in figures 62 and 63. The noise power for these simulations was set from 10W to 0.1W.

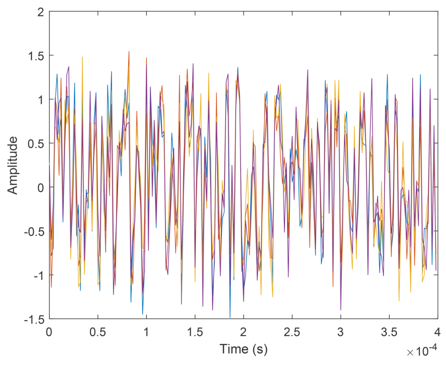
Figure 62: Simulated signal with $P_{Noise} = 10W$: (a) I signal plot, (b) I and Q signal plot.



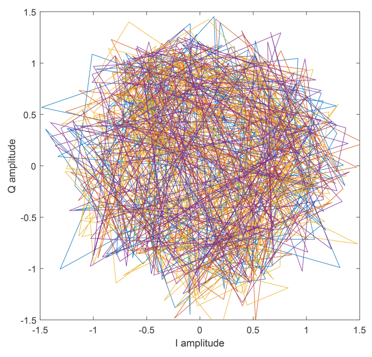
(a)

(b)

Figure 63: Simulated signal with $P_{Noise} = 0.1W$: (a) I signal plot, (b) I and Q signal plot.



(a)



(b)

The signals generated using this method appear to be random wideband noise as shown by the FFT of the signals in figure 64. Despite this, all channels appear to be in phase when the DOA angle is specified as 0 degrees. This can most clearly be seen in figure 63(a). The non-periodic nature of these signals, will provide better angle estimates because there is less correlation between them.

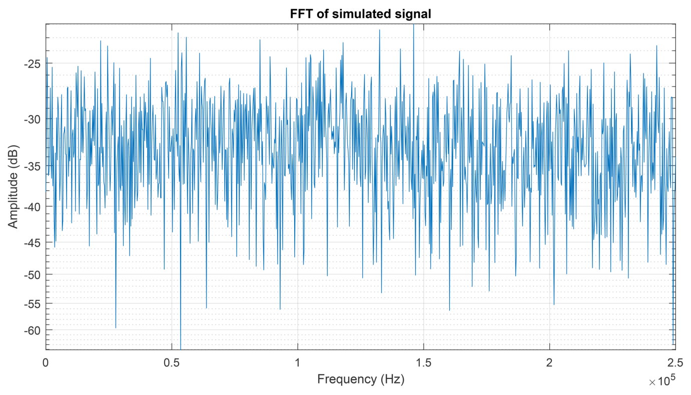
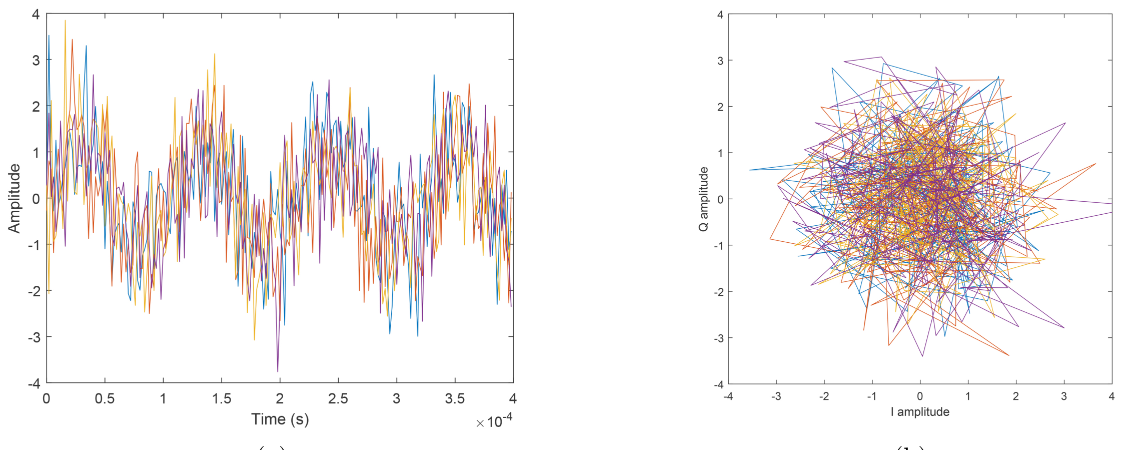


Figure 64: FFT of signal made with Phased Array Toolbox.

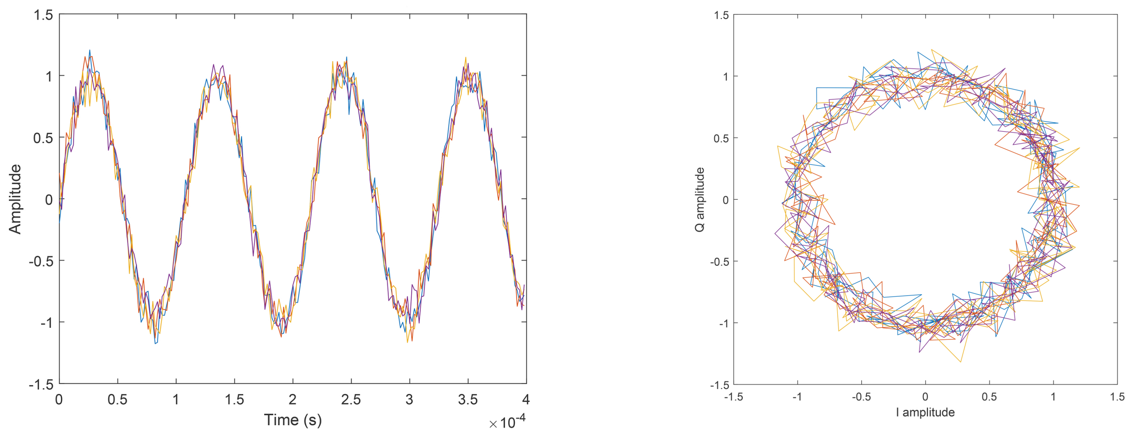
Figures 65 and 66 show signals generated using the phase shifted sinusoid method. From these images it is possible to see that the signals are very clearly in phase. These simulated signals were generated at a frequency of 20kHz and sampled at a frequency of 500kHz, which are the same as the expected signal frequency and sampling frequency of the signals at being fed into the FPGA.

Figure 65: Simulated signal with $SNR = 0dB$: (a) I signal plot, (b) I and Q signal plot.



(a) (b)

Figure 66: Simulated signal with $SNR = 20dB$: (a) I signal plot, (b) I and Q signal plot.



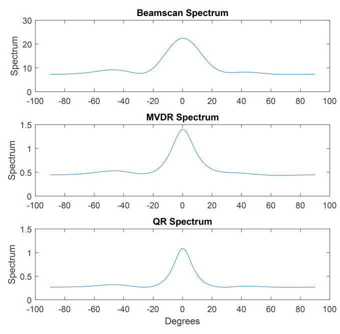
(a) (b)

5.3.2 Simulate DOA algorithm with artificial phase difference and noise (MATLAB)

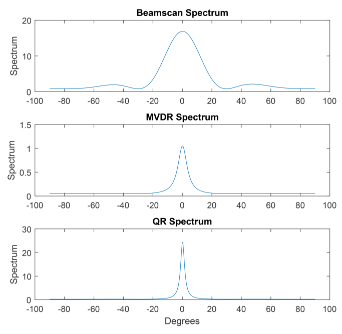
Once simulated signals were successfully generated, the DOA algorithms were tested. Figure 67 shows the spectra of different DOA algorithms calculated using phased sinusoid signals with different SNRs. The table 9 shows the located peaks of the spectra for each SNR tested.

As expected, the beamscan spectrum has the widest peaks and hence the lowest resolution of angles. However, the beamscan algorithm can identify peaks with similar accuracy to other techniques provided the input SNR is high enough. Larger errors tend to occur when the beamscan spectrum reaches extreme ranges angles, beyond 60 degrees from the axis. This is a result of two peaks forming in the spectrum with similar amplitudes.

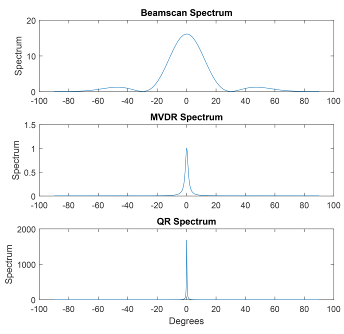
Figure 67: DOA Spectra for phased sinusoids with various SNR: (a) 0dB, (b) 10dB, (c) 20dB.



(a)



(b)



(a)

The results from the peaks search of the DOA spectra can be seen in table 9. This also confirms that there is approximately 1 degree of error when the SNR is at 0dB.

Method	SNR (dB)		
	20	10	0
Beamscan	-0.01	0.11	1.05
MVDR	-0.01	0.12	1.02
QR	0.01	0.10	1.11

Table 9: Angle estimates from different DOA algorithms.

Repeating the script over 200 iterations and averaging the result, the Root Mean Square Error (RMSE) of the algorithms was found. This can be seen in figure 68. From this graph, all three algorithms in MATLAB are seen to have less than 1 degree of error for an SNR of 0dB. However, this is for a single source placed at 0 degrees. At an angle of $\theta = -70$ degrees and an SNR of 0dB, the RMSE is double that of a source placed on-axis. This is a result of the phase shift between signals becoming more difficult for the algorithm to discern. The closer the angle of arrival to +/- 90 degrees, the greater the error.

Figure 68: RMSE of DOA algorithms: (a) $\theta = 0$ degrees, N = 200, (b) $\theta = -70$ degrees, N = 200.

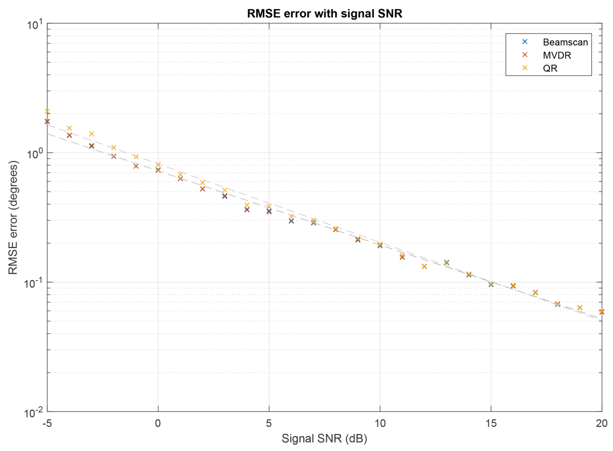


Figure 69: (a)

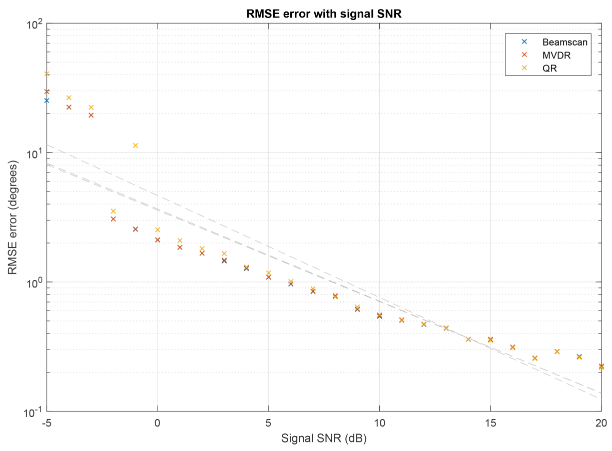
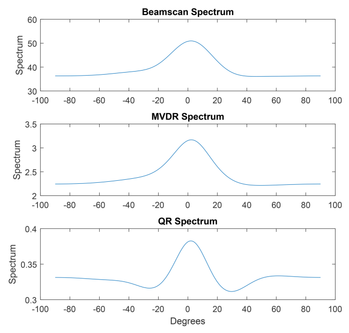


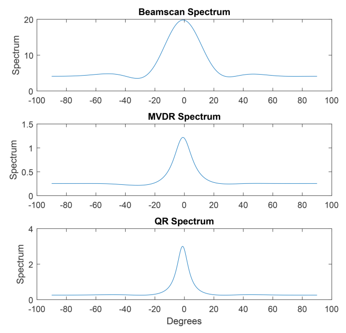
Figure 70: (b)

Figure 71 shows the DOA spectra using the signals simulated with the Phased Array Toolbox. These also follow the trend, where the lower the noise power, the sharper the peak in spectrum.

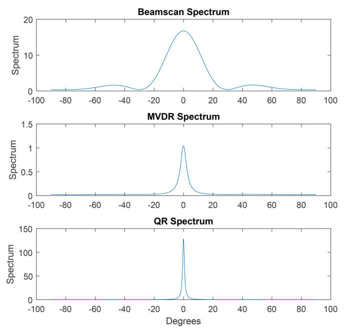
Figure 71: DOA Spectra for Phased Array Toolbox Signals with various P_{Noise}: (a) 10W, (b) 1W, (c) 0.1W.



(a)



(b)



(a)

5.3.3 Build and Implement Simulink model of DOA algorithm

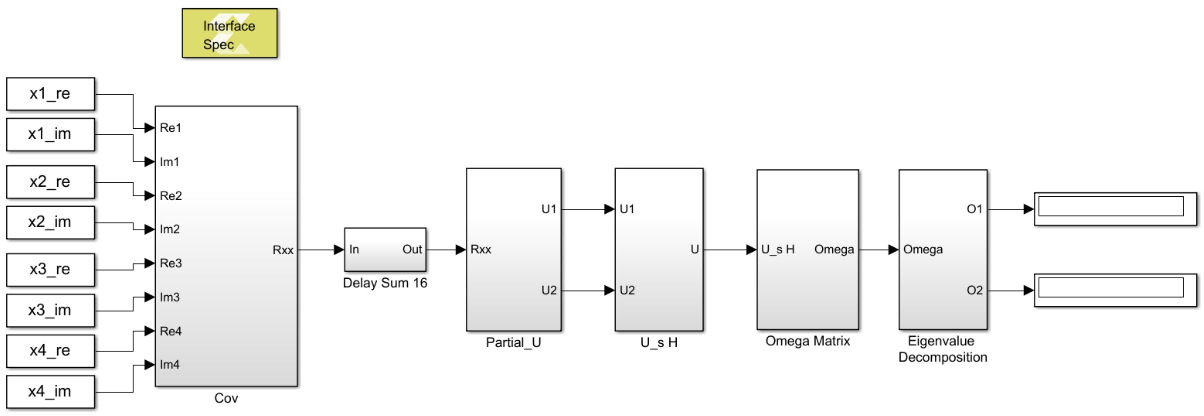


Figure 72: Top Level System of LU ESPRIT algorithm

The complete system model of the LU ESPRIT algorithm can be seen in figure 72.

After testing various parts of the system model in Simulink, sub-system blocks were exported via the Model Composer Hub as IP Catalog. This process created resource utilisation estimates as shown in table 10, where the data type format was specified as 8-bit integer. From this, it is show that it is possible to implement certain functions effectively without high utilisation of valuable DSP slices. Especially as the FPGA receives 8 bit unsigned data from the ADCs.

However, the delay and sum subsystem requires a high number of look-up-tables (LUTs), at over 110% utilisation of the available LUTs on the Artix A7. It should also be noted that this subsystem uses only 16 samples, which would only result in a poor covariance matrix estimation. Therefore, it may not be possible to implement this with a larger number of snapshots using Model Composer. A better implementation of this could use either the Block RAM or slower SRAM to stream data which would be less efficient and result in a less responsive algorithm, but there is no method of controlling hardware usage.

Overall the implementation of DOA algorithms on the FPGA through the use of Simulink and Model Composer seems possible, but may require more knowledge or experience with FPGAs. Model Composer may not be the best choice of design tool because of its lack of control in comparison to Xilinx System Generator which provides lower level design with greater control over registers and other hardware. Ultimately, it is a trade off between more abstraction and more control of the FPGA hardware.

Component	Resource			
	BRAM 18K	DPS48E	FF	LUT
Covariance Iteration	2	22	412	1815
16-Sample delay and sum	0	0	33737	23565
Partial U Decomposition	0	9	745	1619
Eigenvalue Decomposition	0	15	1444	3422
Total Available	100	90	41600	20800
Total Required	2	46	36338	30421

Table 10: Resource estimates from model synthesis.

5.4 Create GUI for presenting data to user

5.4.1 Read data from digital hardware to PC

This has been successfully implemented using the UART serial protocol, for two way communication between the FPGA and the GUI. The transfer can be performed at baud rates upto 115200 or 9600 samples/ second when accounting for overheads. While this is not fast enough to stream data from the ADCs to the GUI it is possible to transfer the hardware limited maximum number of samples in approximately 2 seconds. This time is comparable to the time required by the GUI to plot the data and rescale the axis.

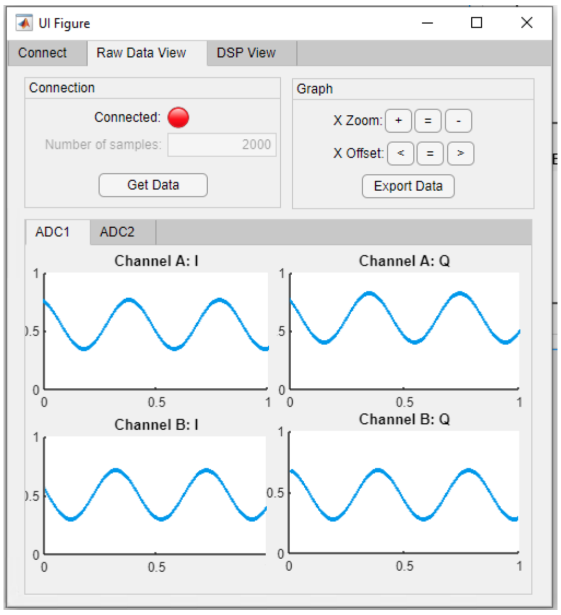
5.4.2 Integration of DOA algorithm with GUI

The DOA algorithms have been fully implemented within the GUI (shown in 73). It requires a single button press to sample, receive, process and plot the results from multiple DOA algorithms.

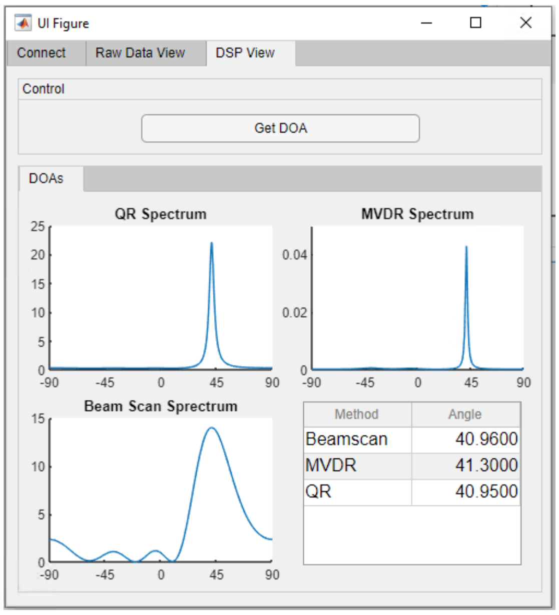
5.4.3 Produce DOA result when requested by user

The control of sampling is fully user controlled with an extension in that the number of samples can be adjusted up to 8000 samples per channel.

Figure 73: Examples of finished GUI: (a) Raw Data View, (b) DSP View.



(a)



Method	Angle
Beamscan	40.9600
MVDR	41.3000
QR	40.9500

(b)

5.5 Overall System Results for DOA Estimation

5.5.1 Hardware Cascade Analysis

To obtain an estimate for the sensitivity of the implemented receive chain, a cascade analysis was performed. Estimates for the Noise Figure (NF) of the op-amp filters and the ADC were calculated using [42] and [43]. Table x below shows the results of this cascade analysis.

Module Data				
	Gain (dB)	Gain (linear)	NF (dB)	F (linear)
LNA	16	39.81	2.30	1.70
BALUN	-0.32	0.93	0.32	1.08
Demodulator	55	316227.77	14.50	28.18
Balanced Filter	3	2.00	15.97	39.54E+00
Filter	0	1.00	15.97	39.54E+00
ADC	0	1.00	31.02	1.26E+03
Cascade Analysis				
	Gain (dB)	Gain (linear)	NF (dB)	F (linear)
LNA	16	39.81	2.30	1.70
BALUN	15.68	36.98	2.30	1.70
Demodulator	70.68	11.69E+06	3.87	2.44
Balanced Filter	73.68	23.33E+06	3.87	2.44
Filter	73.68	23.33E+06	3.87	2.44
ADC	73.68	23.33E+06	3.87	2.44
Sensitivity =	-124.82	dBm		

Table 11: The NF of the AD8348 was obtained from Figure 26 in [18] for 868MHz, the NF of the transformer Balun was taken to be the same as the insertion loss given in the datasheet since it is a passive device. The gain of the balanced filter is assumed to be 3dB due to the conversion from differential to single-ended.

The minimum input power required at the antenna for an SNR of 0dB (required by the algorithms) was then calculated to be -111dBm. This is a best case sensitivity since none of the trace losses are taken into account, none of the passive noise sources are taken into account in the active filtering stages and the ADC is assumed to have no insertion loss.

6 Conclusion

An RF system for array signal processing has been developed which features a high sensitivity receiver, a flexible data acquisition system and a Matlab GUI which enables DOA calculations to be performed on data accurately when calibrated. Each of the required subsystems have been designed, constructed and verified to function as required. The system is a ready-made platform for any generic array signal processing task, to be performed within the RF and bandwidth limits. These could be beamforming for jamming resistance and receiver flexibility, passive radar to locate and track an object, or as the receiver in a communications system.

7 Recommendations and Future Work

7.1 Fine tune circuitry

With more time it would be useful to fine tune component values, so that each receiver would have an almost identical performance. Currently a 3dB imbalance in maximum and minimum receiver gains prevent the full ADC swing to be used on every channel, meaning that some channels will have a much greater level of quantisation noise than others. Variations in the LNA gains could be used to account for this. A dedicated active biasing IC may be able to deliver more consistent results. Another improvement may be to add all-pass filters for phase correction over the frequency range. These could be in the analog, or digital domain. The diode clipping circuits used for ADC protection have a soft-clipping effect which distorts the signal when it approaches the maximum value. To ensure signal integrity is maintained, a hard clipping circuit would be advantageous.

7.2 Printed Antenna

Due to the casings effect on the antenna resonant frequencies, alternative dipole manufacturing techniques were considered, namely a printed dipole design. For this, the antenna is constructed from the traces on a PCB which have a specific width, length and shape. Since PCB etching processes are highly accurate and reproducible, this would ensure antennas with almost identical electrical properties are produced. In addition, being constructed onto FR4 material would provide them with rigidity, removing the need for a protective casing which could alter the frequency response.

7.3 Automatic Gain Controll (AGC)

To ensure that the ADC is always driven at the optimum level, i.e. not being overdriven, whilst maintaining maximum swing, AGC is often used. This is a feedback controller that adjusts the gain of an amplifier in the signal chain according to the magnitude seen at the ADC. The AD8348 IC has a variable gain amplifier built in for this specific purpose, the AD8348 datasheet suggests the AD8362 controller [18]. If AGC was implemented on each channel separately then small gain imbalances in each receive chain, such as have been experienced in the project, would be corrected for. This would remove any need to fine tune the circuits, and would also allow the receivers to operate over a much larger range of input signal strengths.

7.4 Increase the number of elements in the array

By increasing the number of antennas in the array, the phased arrays directivity can be increased. This can be used to increase the systems sensitivity and the resolution at which the sky can be scanned. With more antennas, different array geometries could also be explored, including circular arrays and 2D arrays, both of which would allow for the DOA to be calculated in altitude as well as azimuth. Such as a system could be implemented as an effective passive RADAR. The final PCB has been designed to allow a 2nd board to be connected to the unused inputs of the 8-channel ADCs on the first board, allowing for the easy extension of the system to an 8-element array.

7.5 Extra front end downconversion for higher RF operation - 4G/WiFi

To adapt the antenna array for the detection of higher frequency mobile communication systems such as 4G, Wi-Fi and other frequency bands, a 2nd RF front end stage could be implemented with the one currently developed. A suggested receiver architecture for this system would be a superheterodyne, which features 2 mix down stages whereby the first stage generates an intermediate frequency. For ease of implementation, it is suggested that this could be achieved by converting the current RF front end to an IF stage and running a new mix down stage before it. It has already been shown that both the BFP840 LNA design and ADF4351 LO designs can function at much higher frequencies than they are currently operating on (the LNA has a gain of 10dB at 5GHz and the ADF4351 can reach frequencies of 4.4GHz) so these would be suitable systems to be implemented as part of a 5GHz (IF of 600MHz) or 2.4GHz receiver for Wi-Fi. This however would require a new high performance mixer (for example, the Analog Devices HMC8193 [44] . This may also require bandpass filters for image rejection and other features of a superheterodyne receiver.

The conversion to a superheterodyne capable of detecting 5GHz/2.4GHz Wi-Fi would however require faster ADCs since the bandwidth required for Wi-Fi is 20/40 MHz which is significantly larger than the current system bandwidth. It should also be noted that for these frequencies, new antennas and array geometry is required (new spacing $d = \frac{\lambda}{2}$).

7.6 Automatic Calibration for range of input signals

It would be useful to further develop a rigorous calibration procedure, including a dedicated calibration circuit, based around an oscillator, gain block and wye Splitter, which could be directly connected to the antenna inputs. A calibration script could then be run from the GUI to automatically adjust the calibration values.

7.7 Use a faster method of off loading data from the FPGA

If there is a need to implement a streaming design it is required to first implement a faster method of transmitting data between the FPGA and GUI. This is a common task required of FPGAs and there a various options available, in increasing bandwidth order, USB2, Gigabit Ethernet, USB3 and PCI-e. Gigabit ethernet forms a good compromise between speed and complexity of implementation. There is IP available for this implementation and hardware implementation can be realised using an Ethernet PHY and RJ45 Socket. Consideration of this has been taken when designing the processing system, as this has sufficient bandwidth to maximise the faster connection.

7.8 Add advanced triggering mechanisms to the FPGA

For the system to be able to sample discontinuous signals such as protocols that transmit at low duty cycles, a mechanism for triggering the sampling process is required. This could be implemented as a simple check for if a set level was exceeded upto a decoder that check for specific data within a transmission.

7.9 Implement Algorithm directly on FPGA

As the implementation of the DOA algorithms was limited due to the choice tools selected, future work could be performed to evaluate the merits of different abstraction level tools for DSP design. This would enable implementation of the DOA algorithm within the FPGA meeting the aim for this project. The tools to evaluate would be Xilinx model composer, XIlinx system generator and HDL level code.

8 Achievement Evaluation

The project aims are re-stated below:

1. Design and Construct Quadrature RF receiver with operational frequency range 50MHz - 1GHz.
2. Design, Simulate and construct phased array antenna
3. Design and simulate DOA algorithm
4. Implement DOA algorithm on FPGA hardware
5. Create GUI for presenting data to user

Overall, the project has been a success. Aims 1,2,3,5 have been achieved. Aim 4 was not achieved as the developed algorithm wasn't implemented to perform processing on the FPGA. This issue came about due to a problem that was not foreseen at the time. Although the algorithm hasn't been implemented into the FPGA logic, it was integrated with the computer GUI and so from a user perspective the end result is the same. Despite this aim not being completed, the overall system functionality and user experience are unaffected, which makes this only a minor problem in terms of the overall project deliverable. It can therefore be concluded that the project was a success.

References

[1] Antenna Theory: Analysis and Design, Constantine A. Balanis 3rd edition (2005) Ch. 2 p. 34

[2] Mashhour, A., W. Domino, and N. Beamish (2001). “On the Direct Conversion Receiver — A Tutorial.” Microwave Journal, June, pp. 114 – 128.

[3] D. Muñoz, Position location techniques and applications. London: Academic, 2009, pp. 23-35.

[4] A. Zoubir, Academic Press Library in signal processing. Oxford, U.K.: Academic Press, 2014, pp. 599-614.

[5] M. Amin, Ahmed-Ur-Rahman, Saabah-Bin-Mahbub, K. Ahmed and Z. Chowdhury, “Estimation of direction of arrival (DOA) using real-time Array Signal Processing”, 2008 International Conference on Electrical and Computer Engineering, 2008. Available: https://ieeexplore.ieee.org/document/4769244. [Accessed 6 May 2019].

[6] L. Godara, “Application of antenna arrays to mobile communications. II. Beam-forming and direction-of-arrival considerations”, Proceedings of the IEEE, vol. 85, no. 8, pp. 1195-1245, 1997. Available: https://ieeexplore.ieee.org/document/622504. [Accessed 6 May 2019].

[7] V. Krishnaveni, T. Kesavamurthy and A. B, “Beamforming for Direction-of-Arrival (DOA) Estimation-A Survey”, International Journal of Computer Applications, vol. 61, no. 11, pp. 4-11, 2013. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.303.7100&rep=rep1&type=pdf. [Accessed 6 May 2019].

[8] VOCAL Technologies, L. (2018). Minimum Variance Distortionless Response (MVDR) Beam former. [online] Vocal.com. Available at: https://www.vocal.com/beamforming/minimum-variance distortionless-response-mvdr-beamformer/ [Accessed 18 Oct. 2018]

[9] N. Tayem, “Real Time Implementation for DOA Estimation Methods on NI-PXI Platform”, Progress In Electromagnetics Research B, vol. 59, pp. 103-121, 2014. Available: http://www.jpier.org/PIERB/pierb59/08.14012307.pdf. [Accessed 6 May 2019].

[10] GIRD Systems, “An Introduction to MUSIC and ESPRIT”, Girdsystems.com, 2019. [Online]. Available: http://www.girdsystems.com/pdf/GIRD_Systems_Intro_to_MUSIC_ESPRIT.pdf. [Accessed: 06 May 2019].

[11] A. Hussain, N. Tayem, M. Butt, A. Soliman, A. Alhamed and S. Alshebeili, “FPGA Hardware Implementation of DOA Estimation Algorithm Employing LU Decomposition”, IEEE Access, vol. 6, pp. 17666-17680, 2018. Available: https://ieeexplore.ieee.org/document/8327576. [Accessed 6 May 2019].

[12] Sun, K., Wang, G., Zhang, Q., Elahmadi, S. and Gui, P. (2019). A 56-GS/s 8-bit Time-Interleaved ADC With ENOB and BW Enhancement Techniques in 28-nm CMOS. IEEE Journal of Solid-State Circuits, 54(3), pp.821-833.

[13] Mackay, S. (2019). Digital Signal Processing. IDC Technologies.

[14] O. Johnson, E. Pipe, L. Carslake, W. Sam, ”RF Direction of Arrival Estimation Using a Linear Antenna Array - Project Initialisation Document”, University of Sheffield, 19 October 2018.

[15] Professor Technology Ltd, ”CPA-868MHz, Coil Antenna Design”. [Online]. Available: https://www.mobilefish.com/download/antenna_designs/coil_antenna_868mhz.pdf

[16] , Telestrian: RF and wireless design, “Interactive Smith Chart Impedance Matching”. [Online]. Available: http://cgi.www.telestrian.co.uk/cgi-bin/www.telestrian.co.uk/smiths.pl

[17] G. Breed, ”Improving the Bandwidth of Simple Matching Networks”, High Frequency Electronics, pp. 56 - 60, 2008

[18] Analog Devices, '50 MHz to 1000 MHz Quadrature Demodulator' AD8348 datasheet. 2006. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8348.pdf [Accessed: 03/05/19]

[19] D.G. Hill, 'P-N-P Heterojunction Bipolar Transistors in AlGsAs/InGaAs/GaAs', Stanford University, 1990. [Online]. Available: http://luciano.stanford.edu/thesis/Hill.pdf [Accessed: 05/05/19]

[20] I. Rosu, 'Bias Circuits for RF devices,' YO3DAC - VA3IUL. 26/05/2009.

[21] P. Horowitz, and W Hill, "The Art of Electronics." Cambridge Univ. Press, 1989.

[22] Infineon, 'Robust Low Noise Silicon Germanium Bipolar RF Transistor,' BFP840 datasheet. 03/04/13 Available: https://docs-emea.rs-online.com/webdocs/1620/0900766b81620d89.pdf [Accessed: 5/05/19]

[23] Maxim Integrated (Nov 21, 2003), Tutorial 2875: Three Methods of Noise Figure Measurement. [Online]. Available: https://www.maximintegrated.com/en/app-notes/index.mvp/id/2875

[24] Gilbert, B. (December 1968). "A precision four-quadrant multiplier with subnanosecond response". IEEE Journal of Solid-State Circuits. SC-3 (4): 353–365.

[25] Linear Technology, '0.01MHz to 4GHz Low Power Active Mixer' LT5560 datasheet. 2006. Available: https://docs-emea.rs-online.com/webdocs/1252/0900766b812525d8.pdf [Accessed: 03/05/19]

[26] Macom, 'E-Series Transformer, RF 1:1 Transmission Line' ETC1-1-13TR datasheet. [Online]. Available: https://cdn.macom.com/datasheets/ETC1-1-13.pdf [Accessed: 05/05/19]

[27] Analog Devices, "ADF4351 Datasheet, Wideband Synthesizer with Integrated VCO", Revision A. 2017. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data- sheets/ADF4351.pdf

[28] Analog Devices "ADP150 Datasheet, Ultralow Noise 150 mA CMOS Linear Regulator", Revision C. 2015. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data- sheets/adp150.pdf

[29] SiTime Corporation, "SiT5000: Standard Frequency MEMS (VC)TCXO", Revision 1.0, November 2015. [Online]. Available: http://www.farnell.com/datasheets/2581498.pdf?_ga=2.243501574.2063331806.1557091560-366759672.1542048276

[30] Banerjee, D. (2006). PLL performance, simulation and design. 4th ed. Indianapolis: Dog Ear Publishers.

[31] Analog Devices, "ADL5602 Datasheet: 50 MHz to 4.0 GHz RF/IF Gain Block", Revision A, 2013. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADL5602.pdf

[32] D. Knollman, "Designing with op amps: Single-formula technique keeps it simple." [Online]. 02/03/1999. [Available: https://www.edn.com/design/analog/4341150/Designing-with-op-amps-Single-formula-technique-keeps-it-simple]

[33] Analog Devices, "AD7829-1 Datasheet, 3 V/5 V, 2 MSPS, 8-Bit, 8-Channel ADC", Revision 0. 2006. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data- sheets/AD7829-1.pdf

[34] Zynq-7000 SoC Data Sheet: Overview. (2019). 1st ed. [ebook] Xilinx. Available at: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf [Accessed May 2019].

[35] STMicroelectronics, "LM217LD13TR Datasheet, Low current 1.2 to 37 V adjustable voltage regulators", Revision 7, September 2015. [Online]. Available: http://www.farnell.com/datasheets/1973122.pdf?_ga=2.179356071.2063331806.1557091560-366759672.1542048276

[36] Vishay Semiconductors, "VLMG1300-GS08 Datasheet: Highbright 0603 ChipLED", Revision 1.4, March 2018. [Online]. Available: https://www.vishay.com/docs/82437/vlmo1300.pdf

[37] 7 Series FPGAs Data Sheet: Overview. (2018). [pdf] Xilinx. Available at: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf.

[38] Farnell. (2019). DIGILENT, Development Board CMOD A7. [online] Available at: https://uk.farnell.com/digilent/410-328-35t/development-board-artix-7-fpga/dp/2614574?MER=bn_level5_4NP_LastViewed_2 [Accessed 1 May 2019].

[39] Farnell. (2019). MIKROE-766 - Development Board. [online] Available at: https://uk.farnell.com/mikroelektronika/mikroe-766/dev-board-ready-for-pic-dspic-dip40/dp/2281682 [Accessed 1 May 2019].

[40] Seely, J., Erusalagandi, S. and Bethurem, J. (2017). The MicroBlaze Soft Processor: Flexibility and Performance for Cost-Sensitive Embedded Designs. 1st ed. [pdf] Xilinx. Available at: https://www.xilinx.com/support/documentation/white_papers/wp501-microblaze.pdf.

[41] AD7829 - 3 V/5 V, 2 MSPS, 8-Bit, 8-Channel ADC. (2019). 0 ed. [pdf] Norwood, MA: Analog Devices. Available at: https://www.analog.com/media/en/technical-documentation/data-sheets/AD7829-1.pdf.

[42] Texas Instruments, 'OA-11 A Tutorial on Applying Op Amps to RF Applications.' 03/2013. [Online]. Available: http://www.ti.com/lit/an/snoa390b/snoa390b.pdf' [Accessed: 05/05/19]

[43] E. A. Ball, "EEE6239 Lecture 26: Data Converters for SDR", The University of Sheffield, 2019.

[44] Analog Devices, "HMC8193 Datasheet, 2.5 GHz to 8.5 GHz, I/Q Mixer", Revision B, May 2018. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/hmc8193.pdf

9 Appendices

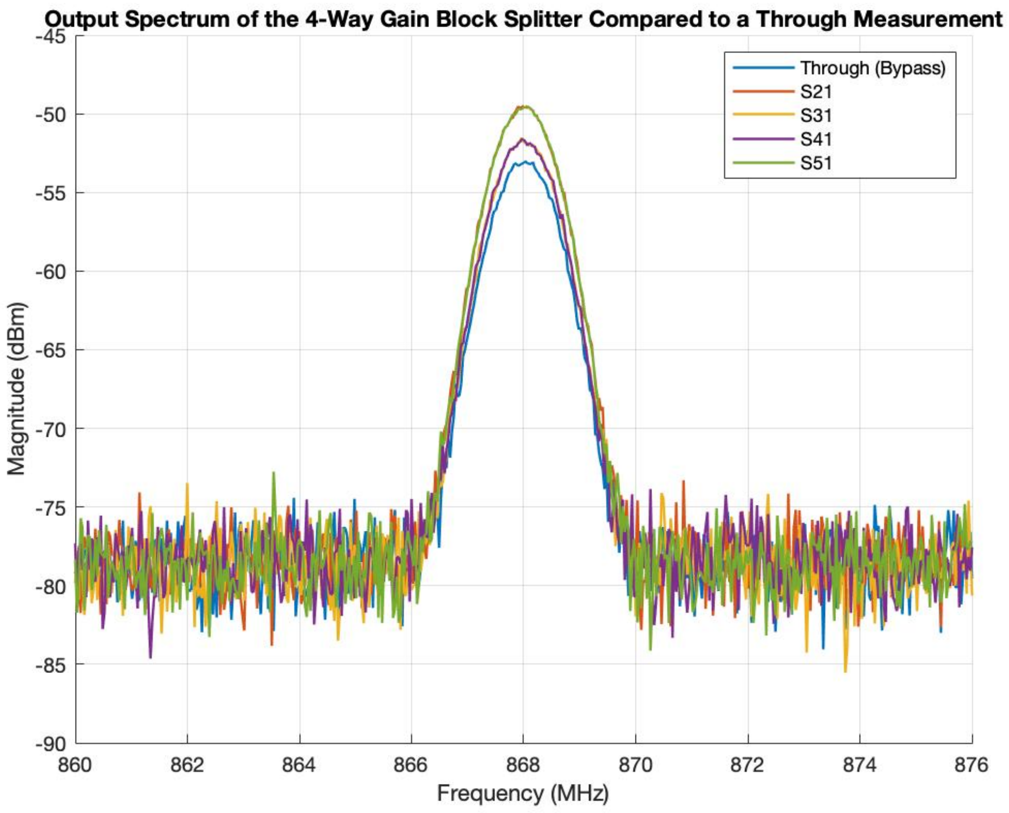


Figure 74: Output Spectrum of the 4-way gain block splitter compared to a through measurement

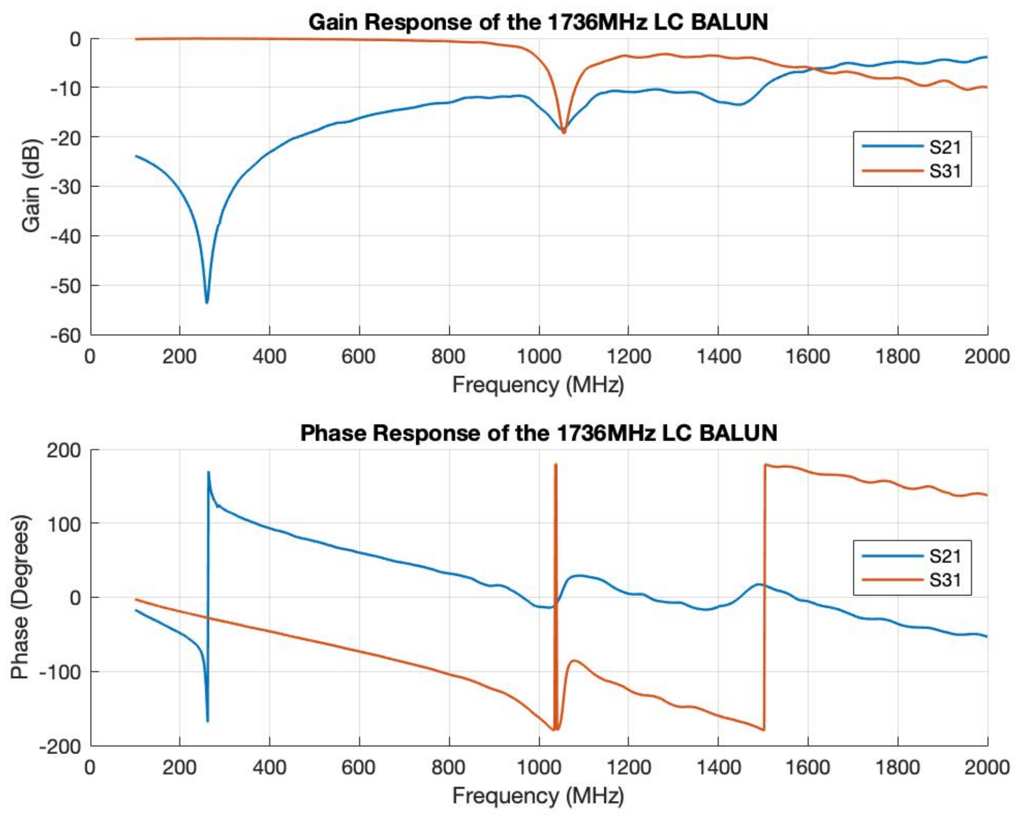


Figure 75: Gain and Phase response of the 1736MHz LC Balun

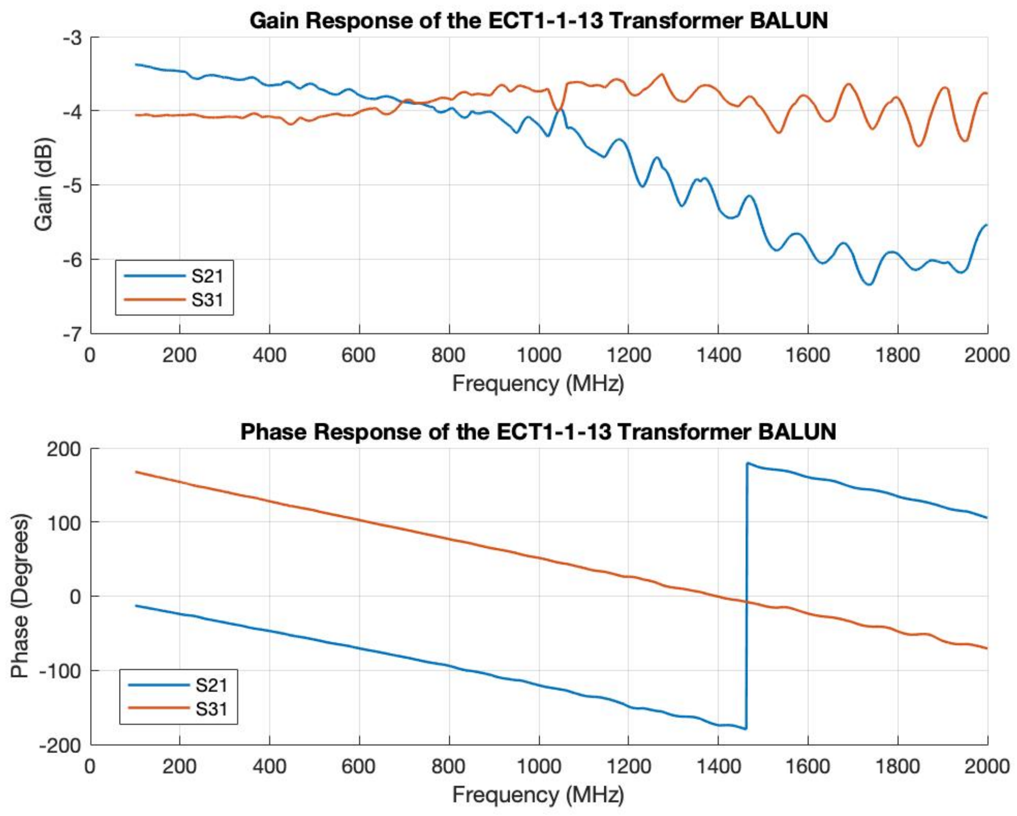


Figure 76: Gain and Phase response of the ETC1-1-13 Balun

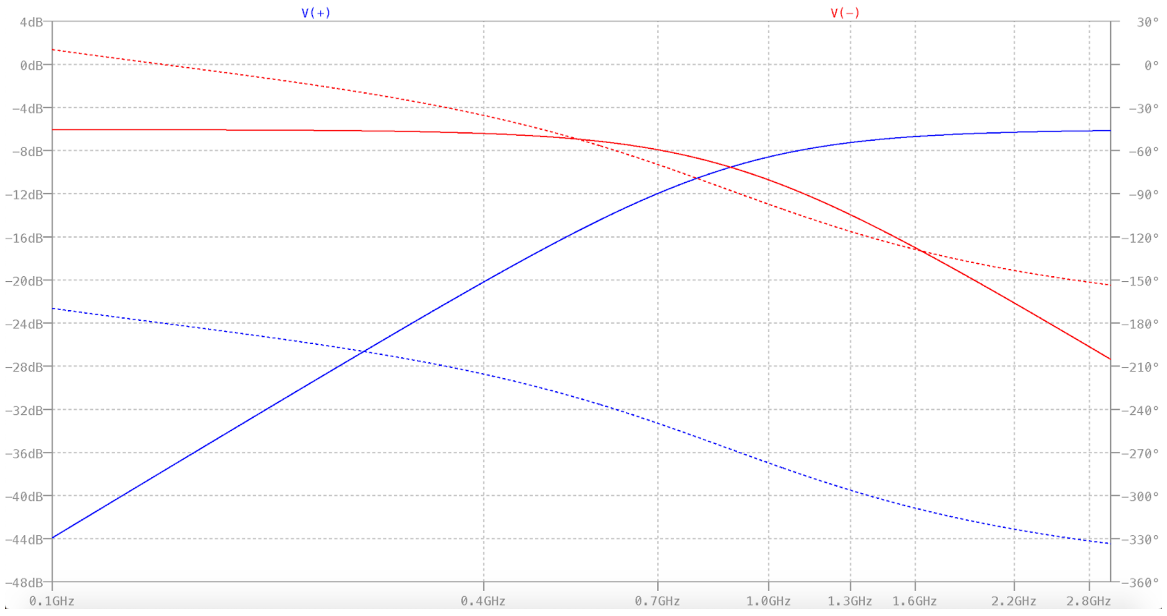


Figure 77: Simulation of LC balun at 868MHz

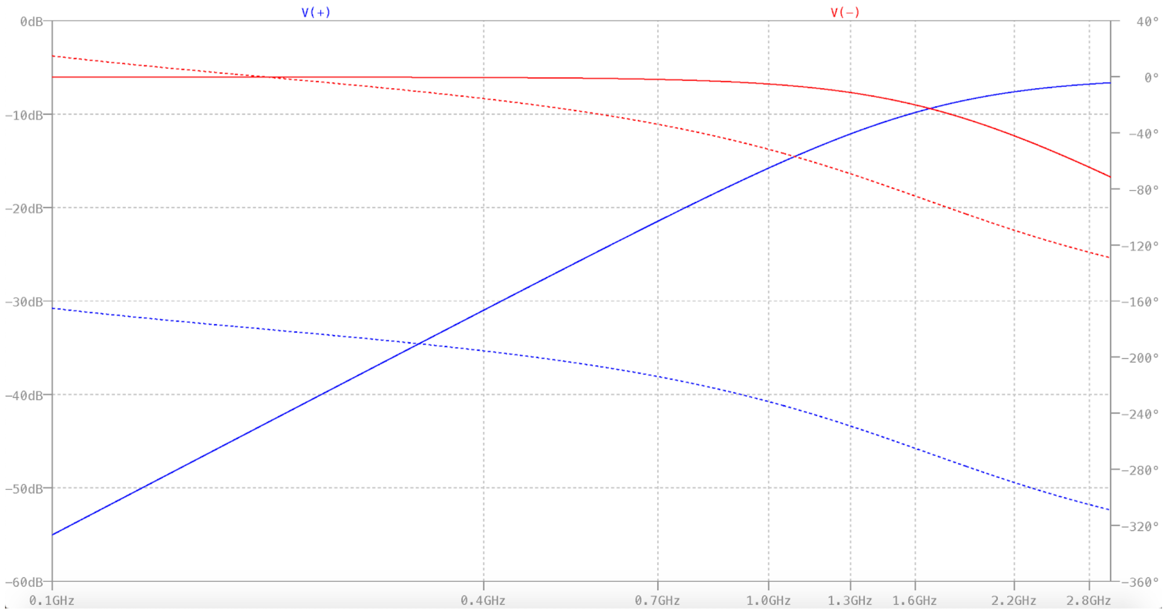


Figure 78: Simulation of LC balun at 1736MHz

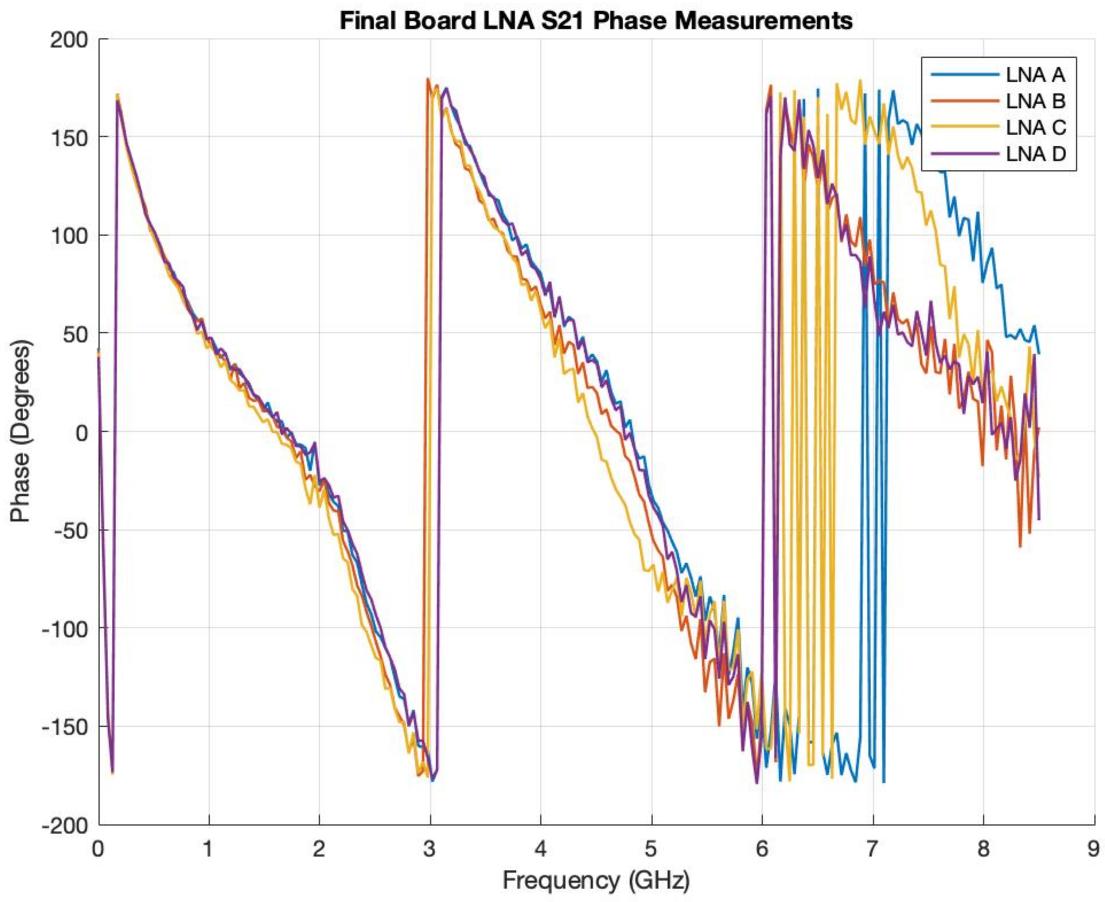


Figure 79: S21 Phase measurements from final board.

Listing 1: Code used to synthesize received signals using Phased Array System Toolbox.

```
fc = 8.68e8;                    % Operating frequency
fs = 500e3;                     % Sampling frequency
c = physconst('LightSpeed');    % Propagation velocity
wl = c/fc;                      % Wavelength

d = wl/2;                       % Distance between antennas
M = 4;                          % Number of elements
L = 1;                          % Number of sources

ang1 = [given_angle;0];         % [azimuth, elevation] angles
angs = [ang1];                  % Concatenate angles


Nsamp = 200;                    % Number of snapshots
noisePwr = n_power;             % Noise power in watts

ula = phased.ULA('NumElements',M, 'ElementSpacing',d);
pos = getElementPosition(ula)/wl;

signal = sensorsig(pos,Nsamp,angs,noisePwr);     %generate signal
```

Listing 2: Generation of phased sinusoid received signals.

```
samples = 200;                                      % Number of samples
F_s = 500e3;                                        % Sampling frequency
time_vect = 0 : 1/(500e3) : samples/(500e3);        % Create time vector
phase_shift_deg = 0;                                % Phase shift to add
phase_shift = deg2rad(phase_shift_deg);             % Convert to radians
fc = 20e3;

theoretical_ang = rad2deg(asin(phase_shift/pi)) % Angle of arrival from array geometry

SNR = 10;                                           % Specify SNR of signal

for k = 1:4

   SigI(:,k) = awgn(sin((2 * pi * fc * time_vect) + ((k-1)*phase_shift)), SNR);
   SigQ(:,k) = awgn(sin((2 * pi * fc * time_vect) + ((k-1)*phase_shift) + deg2rad(90)), SNR);

end

signal = (SigI(1:200,:) + (1j*SigQ(1:200,:)));  % Combine I and Q signals
```

Listing 3: Generation of array manifold matrix.

```
for j = 1:18000                          % Scanning through 180 degrees in 0.01 degree increments
        eta = ((j-1)*pi)/18000;          % Angle from 0 to pi radians
        Y(j) = -(j-9001)/100;            % Translating angle range to -90 and +90 range

            for x=1:M                    % define steering vector
               for k=1:L
                  u(k, x) = exp((-2*1j*x*pi*d*cos(eta))/wl);
               end
            end

        a(:,:,j) = circshift(u, L);
        a(1,1,j) = 1;                    % First element = 1
        A(:,:,j) = a(:,:,j).';           % Resultant Array manifold

end
```

Listing 4: Generation of correlation matrix.

```
Rxx = zeros(4,4);                          % Pre allocate correlation matrix
snapshots = 200;                           % Number of snapshots

for j=1:snapshots                          % Calculate correlation matrix

   temp = (signal(j, 1:4)') * (signal(j, 1:4));
   Rxx = Rxx + temp;                       % Summation of matrix iterations

end

Rxx = Rxx/snapshots;                       % Divide by Number of snapshots
```

Listing 5: Upsample function.

```
function upsig = UPsample(signal,n)

   for k=1:8
      x(:,k) = 0:1:(size(signal(:,k))-1);        % Original X scale
      xq(:,k) = 0:(1/n):(size(signal(:,k))-1);   % Upscaled X scale
      temp = spline(x(:,k), signal(:,k), xq);    % Upsampled Signal
      upsig(:,k)= temp(:,k);
   end

end
```