Development of an RF Listening Mode on the TIGER-3 FPGA Platform

Presented by:
Willem Francois van Zyl

Supervisor:
A. J. Wilkinson
Dept. of Electrical and Electronics Engineering
University of Cape Town

Dissertation submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for an MSc in Electrical Engineering specialising in Radar

October 17, 2020

This page is intentionally left blank

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signature:
W. F. van Zyl

Date: 17 October 2020

Acknowledgements

I would like to thank the following people for their support:

My supervisor, Prof A. Wilkinson, for always being enthusiastic about sharing his vast stores of knowledge while leaving me enough room to experiment.

Dr Willem Schonken for regularly checking up on my project and proving large amounts of invaluable advice.

My SANSA co-supervisors, Prof Mike Kosch and Mr Jonathan Ward, for getting me excited about working on the project and providing me with all the undocumented background information concerning the project. A special thanks to Prof Mike for taking the time to review my final work.

My desk neighbours; Katy, Luka and Othniel, with whom I spend many hours debating with about solutions, complaining to when solutions fail, and celebrating with when things work even though we don't know why.

My family and friends: For endless love and support when it's needed most. Specifically, I would like to thank my mother, Carna: You were the best teacher I could ever have asked for growing up and you invested hundreds of hours into shaping the person I am today. My father, Bertus: You taught me how to carry myself, how to take pride in what I do, and most importantly, not to be afraid to try things out of the ordinary. My sister, Berna: You have been my best friend since you were born. Thank you for reminding me to be creative with whatever I do. Finally, my class mate Darren: Undergrad and all the other things we got up to in those four years would have been a whole lot less fun without you. Thanks for all the study help and laughs.

Abstract

High frequency (HF) radars have many critical applications due to the effects that physical media have on the wave's propagation. The diffraction of HF radio waves in the ionosphere allows for long range communication and radar operation. Waves travel over the horizon where they may be reflected off large scatterers such as ships, or monitor sea states over large oceanic surface areas. Furthermore, the ionosphere provides key information on solar weather. Monitoring RF reflections from the ionosphere (specifically at the polar regions) is of great importance to the scientific community.

The use of the HF (3-30 Mhz) has been greatly simplified for radar transceivers in recent decades. Digital hardware can sample and process information fast enough to eliminate the need for conventional analogue down-converters. The result is an increase in sensitivity, signal to noise ratio and design simplicity. The primary advantage of digital radars is versatility. The ability to change parameters and even modes of operation means that digital radars have become more common, and have replaced or been partly integrated into most of their analogue counterparts.

The SuperDARN is a network of ionosphere monitoring radars that have been in operation since the 1980s. Since its inception it has undergone multiple improvements and served the scientific community well. The 4^{th} South African National Arctic Expedition (SANAE IV) makes use of a digital radar platform based on the third generation TIGER-3 FPGA boards. The highly adaptable nature of the transceivers provide a host of secondary applications and improvements to its analogue predecessors. The system is however not in a state that supports further development. Currently the system is programmed for a set mode of operation without access to the source software.

This work details the design and implementation process followed to bring the TIGER-3 system to a state that will support further development. In this state, peripheral interfaces are designed and implemented to allow for a *listening mode* of operation. In this mode, the radar samples a signal from an antenna and effectively communicates the data to a personal computer via an Ethernet link.

To achieve these outcomes; FPGA code (written in Verilog) was developed to implement IQ downconversion, digital filtering, and a client interface for the Ethernet link. The features were tested by recording and analysing digital outputs from the platform, and finally, by recording signal information obtained through the Ethernet interface.

Supporting literature will lay the groundwork for future projects to build on the base layer implementation; with the hope of redesigning the current SuperDARN implementation in the future. Further improvements to the current system could include a range of scanning patters and multi-frequency operation.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

ACF	**A**uto-**C**orrelation **F**unction
ASIC	**A**pplication-**S**pecific **I**ntegrated **C**ircuit
BRAM	**B**lock **R**andom **A**ccess **M**emory
CIC	**C**ascaded **I**ntegrator-**C**omb
CMOS	**C**omplementary **M**etal-**O**xide-**S**emiconductor
CMT	**C**lock **M**anagement **T**ile
CRI	**C**ross-**R**ange **I**nterference
DCM	**D**igital **C**lock **M**anager
DDS	**D**irect **D**igital **S**ynthesiser
DFT	**D**iscrete **F**ourier **T**ransform
DSP	**D**igital **S**ignal **P**rocessing
EMAC	**E**thernet **M**edia **A**ccess **C**ontrol
ESD	**E**nergy **S**pectral **D**ensity
FFT	**F**ast **F**ourier **T**ransform
FIFO	**F**irst **I**n, **F**irst **O**ut
FIR	**F**inite-**I**mpulse **R**esponse
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
GCL	**G**lobal **C**lock **L**ine
GMII	**G**igabit **M**edia **I**ndependent **I**nterface
HDL	**H**ardware **D**escription **L**anguage
HF	**H**igh **F**requency
IMF	**I**nterplanetary **M**agnetic **F**ield
IP	**I**ntellectual **P**roperty
IPv4	**I**nternet **P**rotocol **v**ersion **4**
IQ	**I**n-phase and **Q**uadrature
ISE	**I**ntegrated **S**ynthesis

Environment	
LB	**L**ogic **B**lock
LPF	**L**ow-**P**ass **F**ilter
LUT	**L**ook-**U**p **T**able
LVDS	**L**ow-**V**oltage **D**ifferential **S**ignaling
MAC	**M**edia **A**ccess **C**ontrol
MDIO	**M**anagement **D**ata **I**nput/**O**utput
MII	**M**edia-**I**ndependant **I**nterface
NCO	**N**umerically **C**ontrolled **O**scillator
NIC	**N**etwork **I**nterface **C**ontroller
ODDR	**O**utput **D**ouble **D**ata **R**ate
PHY	**Phy**sical Layer
PLL	**P**hase-**L**ocked **L**oop
PRF	**P**ulse **R**epetition **F**requency
ROS	**R**adar **O**perating **S**ystem
RLC	Resistor (**R**), Inductor (**L**), and Capacitor (**C**)
SANAE	**S**outh **A**frican **N**ational **A**gency
SM	**S**witch **M**atrix
SNR	**S**ignal-to-**N**oise **R**atio
SoC	**S**ystem-**o**n-**C**hip
SuperDARN	**S**uper **D**ual **A**uroral **R**adar **N**etwork
TIGER	**T**asman **I**nternational **G**eospace **E**nvironment **R**adar
TTFD	**T**win **T**erminated **F**olded **D**ipole
UDP	**U**ser **D**atagram **P**rotocol
ULF	**U**ltra **L**ow **F**requency

1 Introduction

RAdio **D**etection **A**nd **R**anging (radar) is a remote sensing technology that utilises electromagnetic radio waves. By precise time measurements, the time between transmit and receive can be measured and converted into range information. Furthermore, frequency information can be analysed to provide information on a target's radial velocity, by the Doppler effect. Radar became a popular topic of research in the 1930s, when detecting enemies remotely during World War 2 was a priority for many countries. Modern day radars make use of sophisticated hardware and signal processing to power advances in self driving cars, landing space crafts, etc [1]. Furthermore, it is integral to research areas such as studying the ionosphere [2] - which is a primary subject of this study.

Analogue radars traditionally make use of hardware demodulation and filtering techniques to limit and decrease the information within a signal to an amount that allows for sampling and processing. Recent developments in digital hardware has opened the door to digital radars that can synthesise and sample radar signals directly at the transmitted frequency. Digital radars can sample at frequencies in the gigahertz ranges [3], but the cost of hardware capable of doing so is still high. Field Programmable Gate Arrays (FPGAs) find their place between integrated circuit hardware and traditional CPU processors, and have become a preferred choice for providing re-configurable, high performance processing to radar systems [1].

1.1 Background

The Super Dual Auroral Radar Network (SuperDARN) is a global-scale network of phased array, HF radars used for measuring backscatter from the high-latitude regions of the ionosphere. The presence of high density plasma irregularities result in strong coherent-target radar return power. The motion of these irregularities cause a Doppler shift in the return signal, which is of particular interest to space weather analysts. The SuperDARN makes use of a network of coherent HF radars to resolve two dimensional plasma vector maps, illustrating the motion of large irregularities [4][5].

The network consists of 36 operational radars at the time of writing, of which 23 radars are located in the northern hemisphere and 13 in the southern hemisphere [6]. The locations and bearings of the radars are shown in Figure 1.1. The radars are classified according to their latitudes.

1.1.1 Preceding Technologies

The most successful early studies of the dynamic activity of the earth's magnetosphere, ionosphere and neutral atmosphere came from the Scandinavian Twin Auroral Radar Experiment (STARE), in the 1970s and 1980s [5]. These were VHF coherent-scatterer radars operating in the region of 140 MHz. STARE was used to study plasma instabilities in

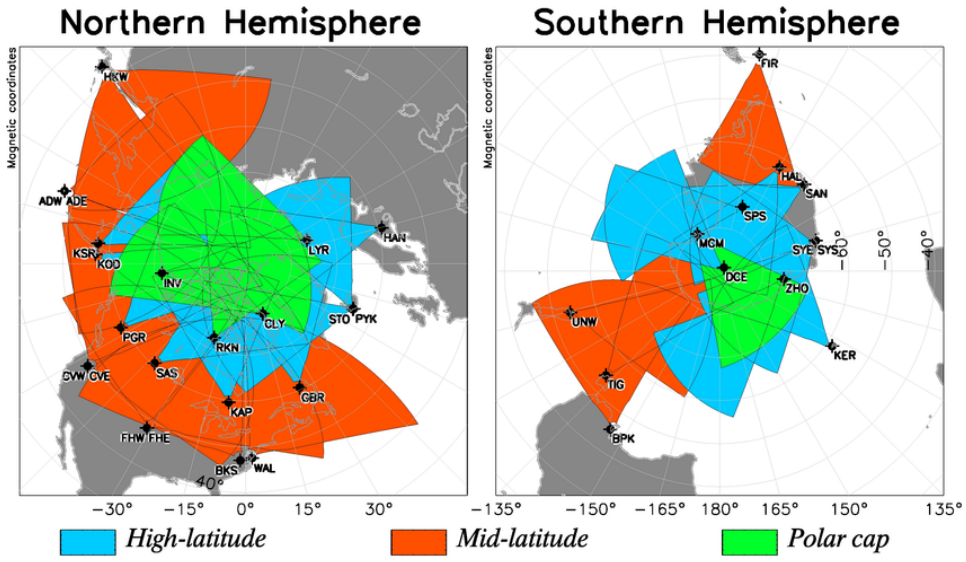

Figure 1.1: Map of the SuperDARN's nominal coverage [6].

the ionosphere - particularly the E-region of the ionosphere (between 90 km and 140 km altitude).

Coherently spaced ionospheric irregularities are the targets for these radars. Physical atmospheric effects result in a host of irregularities that appear across all the layers of the ionosphere with the majority appearing at relatively low altitudes (E-region). At the equatorial and high latitude regions, the F-region (between 140 km and 400 km altitude) contain an abundance of irregularities of particular interest [7]. The irregularities primarily appear perpendicular to the earth's magnetic field lines [2]. Therefore, in order to maximise backscatter, it is critical that the incident wave is perpendicular to the irregularities and magnetic field lines.

The STARE project was limited by the near straight-line propagation paths of the radio waves through the ionosphere at these frequencies. The orthogonality constraint limits its F-region detection abilities to low to mid-latitude regions, where the magnetic field lines are near perpendicular to the earth.

In 1983, an electronically steered, narrow beam, HF radar (operating at 8 MHz to 20 MHz) was developed at Goose Bay, Labrador and is still in operation today as the oldest member of the SuperDARN. A second HF radar was built at Schefferville, Quebec in 1989, which then lead to the concept of the SuperDARN [5].

HF radio waves undergo diffraction in the lower altitudes of the ionosphere which produce strong vector components orthogonal to the F-region of the ionosphere, as illustrated in Figure 1.2.

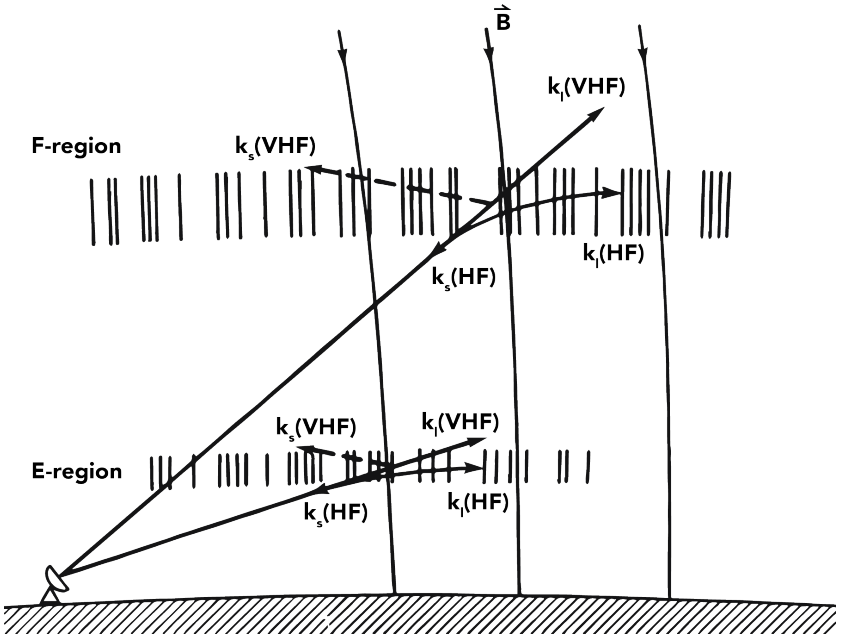

Figure 1.2: HF and VHF ionospheric propagation and backscatter [2].

This development opened the door to the first HF studies of magnetospheric ULF waves and atmospheric gravity waves. It was so successful that more radars followed in the northern and southern hemispheres, operating in pairs to eventually provide F-region plasma convection data [5]. All subsequent SuperDARN radars followed the same base mode of operation as the original analogue concepts from the 1970s, but improvements were being made on the radar hardware and software in order to meet the new demands from the scientific community. To facilitate these changes, the original SuperDARN investigators drafted an agreement to make use of a common radar operating system (ROS). The ROS would ensure that all the radars generate the same end-user data packets irrespective of the underlying technology used to collect the data [8].

1.1.2 The Birth of the Digital Radar

In 2003, the Cooperative UK Twin-Located Auroral Sounding System (CUTLASS) introduced stereo receiver channels allowing for two experiments to run simultaneously. During this time many of the radars migrated to digital receivers [5]. Another big development was the introduction of the new economic antenna array, which makes use of twin-terminated folded dipole (TTFD) antennas [9]. The most recent digital radar transceiver, named TIGER-3, was developed by La Trobe University and was first used in Buckland Park, Australia [10]. All these developments lead to eventual implementation of a radar at SANAE IV, Antarctica in 2003 [11]. SANAE makes use of TIGER-3 transceiver connected to an array of 16 (20 including the auxiliary arrays) TTFD antennas. More information regarding the SANAE IV radar is given in Section 2.2.

1.1.3 Preceding Work

This work is related to work done formerly by other authors. Their work is briefly detailed and referenced here.

Implementation of a Low Cost Demonstrator HF Riometer on a Flexible FPGA Backend - Thomas Dusterwald [12].
The aim of Thomas's work was to determine the feasibility of using the TIGER-3 to implement a riometer mode to run simultaneously with its main mode of operation. Thomas made use of a Red Pitaya FPGA development board in his experiments and successfully demonstrated its use as a processing unit for a riometer.

Modelling and Resolving of the Ambiguous Angle of Arrival Measurements of the SANAE IV SuperDARN Radar - Tighe Barris [13].
Tighe set out to resolve the angle of arrival (AOA) for the SuperDARN radar at SANAE, by using the auxiliary antenna arrays to measure phase differences. Like Thomas, his work also made use of the Red Pitaya FPGA platform. Tighe's results showed that (for the SANAE SuperDARN) an ambiguous AOA can be unwrapped to determine an absolute AOA for up to 70 degrees in elevation. These results were obtained for a single point source by adapting the carrier frequency - an ideal scenario which sets the groundwork for determining the AOA for ionospheric radiation.

High Frequency Surface Wave Radar Demonstrator - Johann Burger [14].
Johann's project focused on a secondary application for the Red Pitaya platform used by Tighe and Thomas. He set out to build a front-end for the Red Pitaya that would allow for the detection of ocean clutter and monitoring of the exclusive economic zone. From Johann's work, a fully functional HFSWR radar was developed. It was shown that the radar could feasibly be used to detect large stationary oceanic objects over the horizon and provide information on ocean features.

1.2 Problem Formation

This project is aimed at developing new software for the TIGER-3 FPGA transceiver platform as it is currently implemented for the SuperDARN, located at SANAE. The radar underwent substantial development since its original analogue implementation in 1999 [6]. The development of the digital version of the radar is, however, not well documented and only parts of the software for the SANAE implementation is currently available for further development.

A radar was first developed for SANAE in 1997 and upgraded in 2013 to the 3rd version of the TIGER digital radar (TIGER-3) [11]. The South African National Space Agency (SANSA) is responsible for the SuperDARN HF radar at SANAE. The agency is interested in further development of the platform, as its use in the research of the ionosphere

and HF propagation studies continues to grow.

The digitisation of the TIGER platform was consistently undergoing development by La Trobe University until around 2015, when its developments were last presented at the annual SuperDARN workshop [15]. Before 2015, La Trobe was planning a fourth implementation (TIGER-4) to improve on the existing system by localising the transceiver racks into a single rack, illuminating a larger area and incorporating the concept of a "self adapting radar". To the author's knowledge, none of these features were ever implemented and no further development was published.

The problem is therefore that SANSA currently has no support from any of the original developers of the radar. When the TIGER-3 was put into effect at SANAE in 2013, SANSA did not gain access to the editable software for the transceivers and was only provided with the capability to load the then-current version onto the transceivers. When it became apparent that La Trobe was ending their support, SANSA requested the software from La Trobe. Due to intellectual property (IP) constraints concerning certain components of the software, La Trobe could only provide SANSA with parts of the transceiver software that was not under IP protection.

All IP generated modules such as numerically controlled oscillators, filters, clock management modules, FIFOs and the Ethernet MAC are unavailable and need to be designed and built. The module responsible for creating the Ethernet packets also appears to be missing. The interfaces for the modules are given which may in some cases be used to determine the operation of the IP generated modules.

The C++ software used to receive the data from the FPGA is available but little is known about the generated Ethernet packets.

It is therefore the aim of this study to research the current implementation of the radar at SANAE with a focus on developing new software for the platform. It involves gaining an advanced understanding of the transceiver architecture and the reconfiguration of FPGAs. The new software will be in line with SANSA's interests, listed below:

- Develop base implementation software for the transceiver to support further development.
- Implement a listening mode for the SuperDARN, for which a user defined range of frequencies can be measured by the transceiver and transferred to a computer.
- Extend the range of peripheral interfaces to support transmission, and configuration from secondary device such as a computer.
- Develop software for the "timing box" - that is an additional transceiver unit used to control timing in the radar system.

- Make use of the TIGER platform for secondary applications such as a sea state over-the-horizon radar or a coastal monitoring radar.

Not all of the above goals are achieved in this project (the limitations are provided in Section 1.5). In order to work towards the above interests, design aspects include; digital circuit design, radar signal processing (such as filtering and down-conversion) and Ethernet interface design. An in-depth understanding of the transceiver box and its peripherals need to be obtained and the relevant peripheral interfaces need to be designed for the FPGA. A basic understanding of networking theory is developed to complete the PC receiver interface.

1.3 Motivation

As mentioned in Section 1.2, only part of the transceiver software was made available (in editable form) to SANSA, for further development. When referring to the complete system software, the following is included:

ROS: The radar operating system software is written in C, runs on the main computer and is responsible for receiving and processing the transceiver data into the standard SuperDARN data product. It is also used to configure the transceiver from the main computer. The ROS version is titled "ROS.3.1.x" and was modified for the TIGER-3 by La Trobe [16]. The software is currently in use at SANAE and is available for development [6].

Transceiver Software: The transceivers are reconfigured through HDL (written in VHDL). The HDL is converted into a bitstream, which is loaded onto the transceiver hardware [17]. The transceivers are responsible for sampling the received signals from the antennas, down converting it, and transmitting the resultant data to the ROS [15]. The TIGER-3 bitstream is available to SANSA but only a limited set of the VHDL modules are available.

There are therefore two identified approaches that could be taken in meeting SANSA's requirements. The first is to keep the transceiver software as is, and to modify sections of the ROS to control the transceiver and request the necessary data from it. Although this approach could be a simple solution to the problem, there are disadvantages: The limitations of the transceiver software is largely unknown and the operation of it is undocumented. Limitations may include receiver bandwidth, limited transmitter waveform control and modes of operation. The second approach is the complete redesign of the system. This is likely to result in an extended time-line for SANSA's interests due to the work load, but the result is a versatile system that is well understood from the ground up.

In agreement with SANSA, the second approach was taken for this project. This ap-

proach was identified as a more holistic approach, and if the basic requirements are met; the outcome can better support versatile development environment for future work.

1.3.1 HF Radar Applications

HF radar is generally considered an ageing technology. It requires large antennas, generally offers coarse range resolutions and is located in a well occupied spectral range (3-30 MHz). It is however uniquely useful for a specific range of applications. HF radio waves are largely unaffected by weather and have unique propagation effects when interacting with the earth's atmosphere. Two primary use cases are given below.

OTH (Skywave and Surface wave): HF is used for over-the-horizon radars which can illuminate regions over the earth's horizon [18]. This serves military interests in the form of coastal monitoring and aircraft tracking. It serves the scientific community through the measuring of sea states and ocean winds [14]. Furthermore, it also serves a commercial purposes in the form of long range radio communications.

Geospace Monitoring: HF is also the frequency of choice by scientists studying the ionosphere for plasma convection, atmospheric gravity waves, ULF waves, etc [19]. Furthermore, space weather analysts make use of the radar to predict the impact of solar storms on the earth.

1.4 Research Statement

This work details the operation of the TIGER-3 transceiver as part of the SuperDARN. The radar's operation was researched and detailed from an engineering and scientific point of view. It details the work done on implementing a listening mode on the platform. In this mode, the transceiver must sample the RF spectrum up to 30 MHz and perform the necessary signal processing to transfer relevant information to a personal computer. Externally variable user parameters should define the centre frequency and bandwidth of the transferred signal band.

After the design is complete; measurements were taken and compared to simulated results. The performance of the platform was thus characterised. The simulations take all the digital signal processing performed by the FPGA into account, to accurately compare the measured data to the expected outputs. Considering the results, this work lists considerations to be taken into account for future development.

1.5 Scope and Limitations

The current SuperDARN implementation is a complex integration of multiple transceivers in two-way communication with a central processing unit through network switches.

Clocking is synchronised in order to perform beam steering and ensure receive data coherency, which is just one example of the complexity introduced in working with multiple transceivers.

The scope of this work is largely limited by the use of a single transceiver, and therefore beam steering and clock synchronisation is not within the scope of this work. Furthermore, no consideration was given to the Ethernet receive-side, and all other multi-client networking operations of the SuperDARN system are out of the scope of this project.

The transmission of data from a single transceiver over a set Ethernet standard was implemented as a base proof of concept for future development. The client (personal computer) receive-side implementation is processed with minimal effort. The design of an advanced client receive interface is not included. The effective capturing and storage of the Ethernet data, for further analysis of results, is within the scope of this project.

The TIGER-3 sports an extensive list of peripherals which are used for the full implementation of the SuperDARN. Not all peripherals fall within the scope of this work. Flash memory can be used to store operational parameters and bitstreams, to support changes in the mode of operation of the FPGA - based on received data from the system computer over Ethernet. No interface was developed for the use of flash memory. Furthermore, peripherals DAC were explored briefly, but considering the main outcomes of this work, it is excluded from the scope. Other out-of-scope peripherals include; all additional ADCs (apart from the 16-bit ADC used for RF sampling) and all FireWire and VGA related peripherals (dated technology).

This work therefore makes use of the primary RF sampling ADC and the Ethernet PHY peripherals as its main interfaces. Switches and push-buttons are used to simulate variable user input parameters as a proof of concept. Other in-scope peripherals include; clocking peripherals, digital and analogue IO peripherals, and RF-side interface peripherals (such as receiver step attenuators and switches).

The literature reviewed in this report is aimed and setting a basis for future work and to provide context to the work as a whole. It lists considerations for future work that is not within the scope of implementation. It does however not go into detail on the SuperDARN data product and other signal processing done post-capture. Furthermore, the scientific background of the operation is summarised from an engineering perspective and does not include the advanced theory of the interaction between RF signals and the earth's atmosphere, nor does it detail the complex nature of the atmosphere with respect to plasma.

1.6 Software and Hardware

Primarily, this project is based on a single transceiver box, closely based to the transceivers in use at SANAE at the time of writing. The transceiver includes a Virtex-5 XC5VSX50T FPGA, and other supporting interface peripherals. The hardware is further detailed within the main body of this work.

Verilog is used to reconfigure the FPGA and was written and synthesised using ISE Project Navigator Version 14.7, on Windows 10. ISE is officially only supported for Windows 7/XP/Server and limited Linux distributions. Apart from small bugs (such as occasional crashes), the installation of Windows 10 was functional. Xilinx currently supports a newer design suite which is not compatible with the Virtex-5 family.

A signal generator (Agilent 33250A) was used to produce test signals up to 80 MHz in frequency. A digital oscilloscope (Agilent MSO9104A) was used to measure up to 16 digital channels and up to 4 analogue channels, rated at 1 GHz and 20 GSa/s.

The capturing of data transferred over Ethernet from the transceiver was done using Python 3.7, and the processing of the stored data was done using MATLAB 2019. Wireshark was used to analyse the header information of the data packets, the transmission parameters (such as data rate), and for general debugging of the Ethernet interface.

The hardware specifications for the PC in use are given below:

System Model:	Dell XPS 13 9370 Laptop
Processor:	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz
Memory:	8192MB RAM
OS:	Windows 10 Home 64-bit (10.0, Build 18362) (18362.19h1_release.190318-1202)
GPU:	Intel(R) UHD Graphics 620
Storage:	SSDPEKKF256G8 NVMe INTEL 256GB

1.7 Technical Requirements

A considerable part of this thesis concerns research of the TIGER-3 radar in its current state. The thesis also concerns the design of a listening mode. The design aspect of this project requires a clear set of technical requirements. For this project to be deemed a success, the results obtained and detailed in this thesis need to show that all the requirements were met. The requirements are detailed here.

External Peripherals

External to the FPGA, peripherals are used to provide and alter inputs and outputs to the FPGA according to the requirements of the radar.

- The relevant switches must be programmed to enable the RF input path and the input attenuation must be adjustable to ensure that the ADC's full dynamic range is used. The switching and attenuation abilities of the system must be fully characterised for future work.
- The ADC must be programmed to operate given the predefined pin configuration on the TIGER-3 board.
- The above requirement also applies to the Ethernet physical layer. The configuration must allow for the full gigabit Ethernet data rates to be achieved.

FPGA resources

There are FPGA hardware limitations in the form of available FPGA fabric space and dedicated primitives.

- Appropriate primitives must be used in the design to improve results.
- Resource usage must be detailed for the complete design.

Clocking

Implement a digital clock manager (DCM) to make effective use of the clocking peripherals.

- The required clocking frequencies need to be generated.
- Generated clocks must be forwarded to the relevant modules.
- Clock errors need to by minimised to ensure that phase errors are suppressed in synthesised waveforms.

Numerically Controlled Oscillator

A numerically controlled oscillator (NCO) is responsible for the synthesis of sinusoidal waveform samples.

- It is a requirement that the synthesised signal will not be constant in frequency, but rather be adjustable by the user.
- The tuning of the output frequency must account for the base banding of the full range of transmit frequencies with applied doppler shifts. This approximates to 8 MHz to 17 MHz.

Digital Filtering and Basebanding

A digital filter must be implemented to to sufficiently allow for clean basebanding and sample decimation by the suppression of stop-band frequencies. The specific stop-band suppression required for the SuperDARN in unknown due to the lack of measured data. Therefore, the requirements will not contain specific values.

- A narrow-band filter must be designed to maximise sample decimation while obtaining information for the expected Doppler shift range of 100 Hz.
- In addition to the narrow-band filter, a wide-band filter must be designed and tested for potential wide bandwidth applications. The requirement is to determine the maximum bandwidth that could be measured by the system.

System Noise

The two primary sources of noise are introduced into the receive chain are; analogue noise introduced by components such as the radar's power supply, and digital noise introduced by the digitisation of the signals and the basebanding process.

- The analogue noise in the system must be measured and the source of the noise be identified.
- Where possible, noise must be suppressed and the effects of the noise on the final output must be analysed.

Ethernet Interface

The capabilities of Ethernet interface determines the maximum data rates available for the system output.

- An Ethernet protocol should be chosen and implemented to maximise data rates.
- The true maximum data rate should be determined for future work.

Supporting Software

Receiving data from the radar is critical for confirming the working of the Ethernet interface and many other aspects of the system.

- Software must be developed to run on a personal computer in order to capture and log data received from the TIGER-3 board.

1.8 Plan of Development

This report contains seven chapters to detail the supporting literature, the design process and the final results.

Chapter 2 presents supporting literature to provide context to the project from a scientific and engineering perspective. Propagation theory and an overview of the digital version of the SuperDARN transceiver is detailed.

Chapter 3 builds on the literature by introducing design models and technical theory. The mathematical radar model is derived with special attention given to the IQ demodulation receiver architecture. Related processing techniques such as Doppler processing and ACF functions are detailed, and is followed by an explanation of the SuperDARN-specific processing techniques. The mathematical models are followed by the communication models used for the Ethernet interface - detailing each network layer used. The chapter ends with further design considerations concerning the FPGA hardware and the reconfiguration thereof.

Chapter 4 details the design implementation process. It presents each primary Verilog module written for the system to operate as a listening mode. A system overview is provided, followed by a sub-chapter detailing each primary task of the design. The chapter concludes with a description of the PC receiver interface.

Chapter 5 presents the results that were obtained for each of the design elements. The measured results are accompanied (where applicable) with simulations to confirm the validity of the results. The chapter concludes by detailing the FPGA resource usage for this implementation.

Chapter 6 details the discussion of the results obtained in Chapter 5. Finally Chapter 7 concludes the project by drawing conclusions from the aforementioned discussions and providing recommendations for future work.

2 Supporting Literature

This section focuses on the theoretical background of this study. It is divided into two subsections; the first will detail the physical composition of the ionosphere and the interaction of HF radio waves with it. The second subsection details the working of the SuperDARN from a scientific point of view, and stating some of its primary use cases.

2.1 HF Propagation and Backscatter in the Ionosphere

Ionospheric propagation, although important for the operation on the SuperDARN, does not affect the outcome of this study. This section therefore exists to provide context for the project from a scientific point of view. Backscatter depends on the physical properties of the transmitted wave and the ionosphere, which is discussed here.

2.1.1 The Ionosphere

The ionosphere forms part of the earth's atmosphere between 75 km and 1000 km in altitude. High energy particle collisions and solar radiation causes the ionisation of the gasses in the ionosphere [12]. The free electrons, being much lighter than the ions, characterise the interaction properties of the ionosphere with HF radio waves. Counter mechanisms cause recombination of the electrons resulting in varying electron densities in the ionosphere. The resulting density layers and their respective altitudes are shown in Figure 2.1.

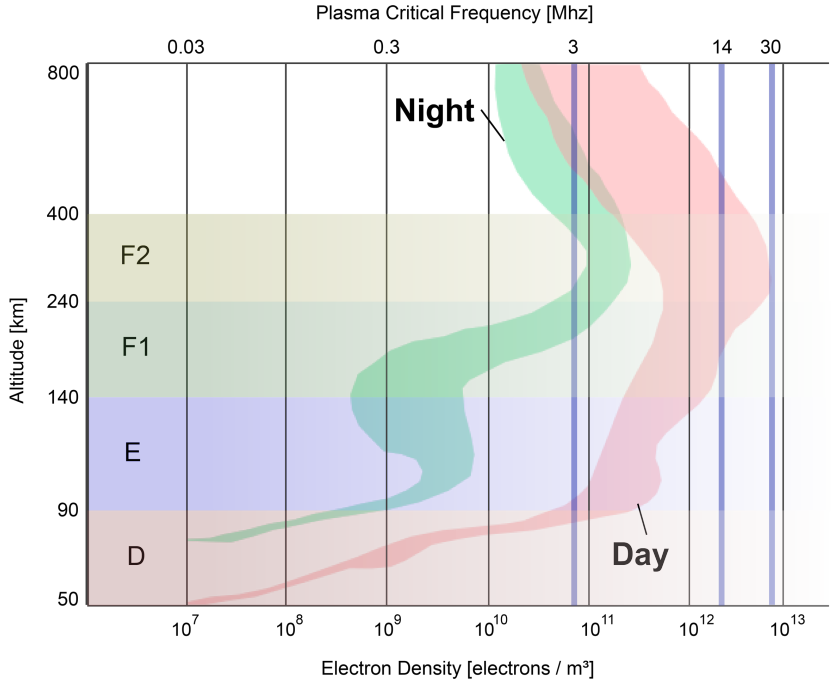

Figure 2.1: Ionospheric electron density height profile.

During the night, less ionisation occurs and only the E and F (combined F1 and F2) layers are observed. During the day, more ionisation occurs and all layers are observed. During the day the F layer divides into two parts [2] [20]. More information on the ionisation and recombination effects can be found in Thomas Dusterwald's work on Riometers [12].

Plasma is a fourth state of matter [21] that can be defined as: "a quasi-neutral ionised gas that exhibits a collective behaviour". It is "quasi-neutral" because even though it consists of ions and free electrons; the collective, viewed at a distance, is neutrally charged. Solar energy forces cause electrons to move away from the nuclei but electrostatic forces counter that motion. The result is that electrons undergo oscillatory motion. This oscillation is defined as the plasma electron frequency, ω_{pe} [22]. It is given by

$$\omega_{pe} = \sqrt{\frac{n_e e^2}{m_e \epsilon_0}} \quad [\text{rad/s}] \tag{2.1}$$

where m_e is the electron mass, ϵ_0 is the permittivity of free space and n_e is the number density of electrons per cm^3. For the ionosphere it approximates to

$$f_{pe} = \frac{\omega_{pe}}{2\pi} \approx 9.98\sqrt{N_e} \quad [\text{Hz}] \tag{2.2}$$

where N_e is the number density of electrons per m^3

Figure 2.1 shows the plasma electron frequency as "plasma critical frequency". Radio waves are reflected if the frequency of the incident wave is less than that of the plasma electron frequency [20].

2.1.2 Plasma Convection

The geospace is a region around the earth that is directly effected by the solar wind from the sun [23]. It includes the earth's magnetic field which protects it from solar wind. The sun pushes fully ionised hydrogen/helium plasma towards the earth, and with it comes the interplanetary magnetic field (IMF). The earth's magnetic field acts as a barrier against the IMF and the interaction between the magnetic fields have complex effects on the geospace. Phenomena such as magnetic reconnection, plasma waves, ionospheric convection and aurora are all products of this interaction.

The electric field lines across the magnetosphere are perpendicular to the magnetic field and point from dawn to dusk. The $\vec{E} \times \vec{B}$ field results in a Lorenz force driven convection within the ionosphere [23]. The convection creates two or more global scale convection regions/cells, the placement and shape of which depends on the direction and state of the solar wind. The idealised two-cell model for the North pole is shown on the left in Figure 2.2. These and other ionospheric effects occur over 100s to 1000s of kilometres and are highly variable in both time and space. In order to observe these effects; it is therefore

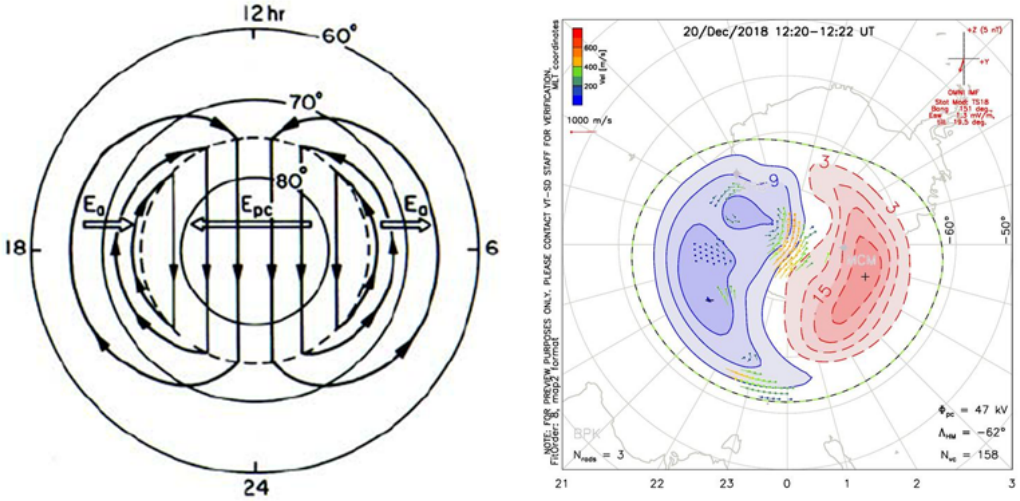

Figure 2.2: Left: Ideal Plasma Convection Cell for the North Pole [24]. Right: An example of plasma convection readings taken from the SuperDARN for the South Pole [6].

critical to have a measurement system that can meet the time and space requirements. As is discussed further in Section 3, a radar can meet these requirements under certain circumstances.

2.1.3 HF Ionospheric Propagation

The propagation of radio waves through the ionosphere is not well understood due instability and inconstancy of the ionosphere's structure. The SuperDARN website makes use of the IRI model [25] of plasma density to plot predicted ray paths for their radars. The IRI is a semi-empirical model that makes use of historical data to extrapolate unmeasured properties of the ionosphere. It is commonly used but its reliability (especially in the Antarctic regions) has been questioned. An example of a ray trace for SANAE is shown in Appendix A. A simplified illustration is given in Figure 2.4.

Ionospheric irregularities of interest appear parallel to the magnetic field lines [23]. The reflections from these irregularities undergo Bragg-scatter interference. Bragg-scatter is illustrated in Figure 2.3. Given an incident angle θ, the reflection from each surface will constructively interfere if

$$n\lambda = 2d\sin\theta \tag{2.3}$$

where $n \,\epsilon\, \mathbb{N}$, λ is the wavelength of the incident wave, and d is the distance between the reflective ionospheric irregularities.

The conditions are however greatly simplified by the reflective properties of the ionosphere for coherent backscatter. The requirement is that the incident wave needs to be

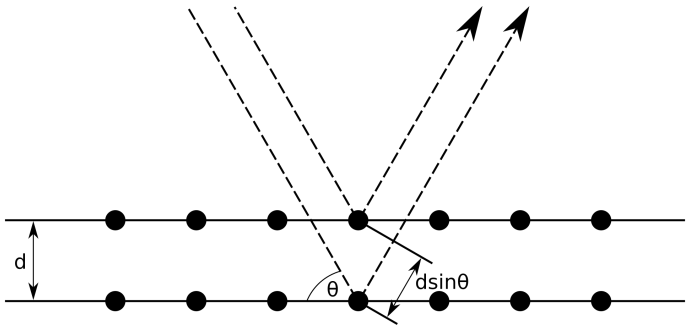

Figure 2.3: Bragg scatter of point targets.

orthogonal to the magnetic field lines (along which the irregularities occur) to maximise backscatter [2]. At low latitudes, radars typically make use of ionospheric refraction to bend the incident wave to meet the orthogonality condition at higher altitudes. UHF and VHF waves undergo little to no refraction and therefore are not well suited (see Figure 1.2). In order to receive backscatter from the high-altitude regions of the ionosphere, HF is very well suited [26].

The angle at which the HF waves enter the ionosphere will effect the altitude that the wave reaches before becoming orthogonal with the magnetic field lines. Figure 2.4 illustrates a simplified overview of the effects that the elevation angle has on propagation.

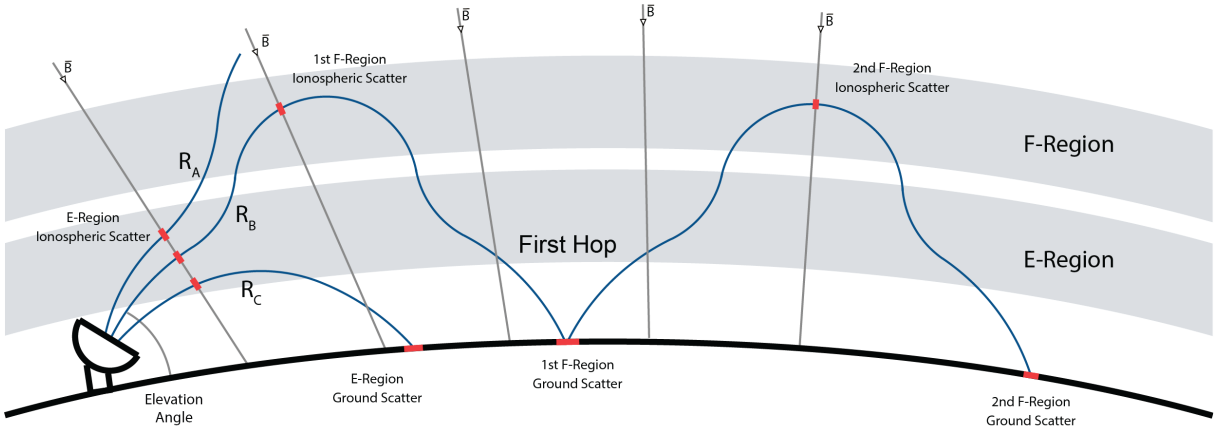

Figure 2.4: Ideal two-hop HF propagation ray paths for the SuperDARN.

The electron density variation in the ionosphere and the large vertical beamwidth of the radar causes multiple ray paths. When HF transmitters are used as communication systems, they can make use of up to two hops to transfer information thousands of kilometres over the horizon and back to the earth's surface [1]. The radar can make use of ionospheric

backscatter from the second hop, although it was found that this only forms a very small fraction of the total backscatter [26]. The ground area illuminated by the radar is called the "footprint" of the radar, and is one of the main sources of backscatter.

The work done by Tighe Barris (see Section 1.1.3) allows the radar to determine at what angle certain types of backscatter is received. This can be used to cancel backscatter that is not of interest to the operator, or provide valuable information on the state of the ionosphere. In contrast, Thomas Dustewald's work on a riometer mode for the SuperDARN relies only on the natural radiation from the ionosphere directly above the receiver.

The irregularities, providing the backscatter, drift with the background plasma motion. Therefore Doppler processing is used distinguish between ionospheric convection and ground scatter in the radar readings. The processed Doppler readings can then be used to generate plasma convection maps, shown on right of Figure 2.2. Studying these maps have greatly aided researchers in gaining a better understanding of the earth's geospace and the effects of the solar wind [5].

Given all the above conditions to maximise backscatter; it is important for the radar to be frequency agile in order to adapt to ionospheric changes, and keep with scientific research requirements.

2.1.4 Additional SuperDARN Data Products

The data obtained from the SuperDARN has provided useful results for research other than plasma convection [27]. Some of those research areas are mentioned here.

Travelling Ionospheric Disturbances (TIDs): In the case of TIDs, the reflections from the ground are studied. The non-uniformity of the ionosphere can focus the rays on specific ground locations producing dominant ground scatter at specific ranges.

Polar Mesospheric Summer Echoes (PMSEs): The Antarctic mesopause is located at 90km to 110km altitudes. The mesopause's sensitivity to climate change has made it a very important research interest in recent years. Backscatter from the mesopause was first identified from the Syowa east radar in 2002.

Ultra Low Frequency (ULF) Pulsations: Another product of studying the ground scatter from the SuperDARN, comes from studying the ULF Doppler from the ground scatter. This low frequency Doppler component originates from the ionosphere's vertical motion. It is well monitored by magnetometers but properties such as the morphology, phase and group velocity can only be determined by the SuperDARN.

2.2 SuperDARN Radar Operation

Each SuperDARN radar is a monostatic pulsed radar that makes use of antenna arrays to produce narrow, electronically steerable beams. In the case of the TIGER-3 (discussed in Section 2.2.1), a 16 element Twin-Terminated Folded Dipole antenna array (shown in Figure 2.5 and Figure 2.6) is used to provide a field of view of $\sim 90°$ in azimuth and $\sim 50°$ in elevation, with each beam providing $\sim 3°$ in azimuth (at 14 MHz). It provides range in excess of 5000 km with 110 range gates. The effective radiated power in the main lobe direction is $\sim$ 10 kW at 12 MHz (2.4 kW x 16 transceivers). Typically, the radar operates in fixed frequency mode but can transmit at 8-17 MHz to accommodate changing ionospheric conditions [10] [23] [13].

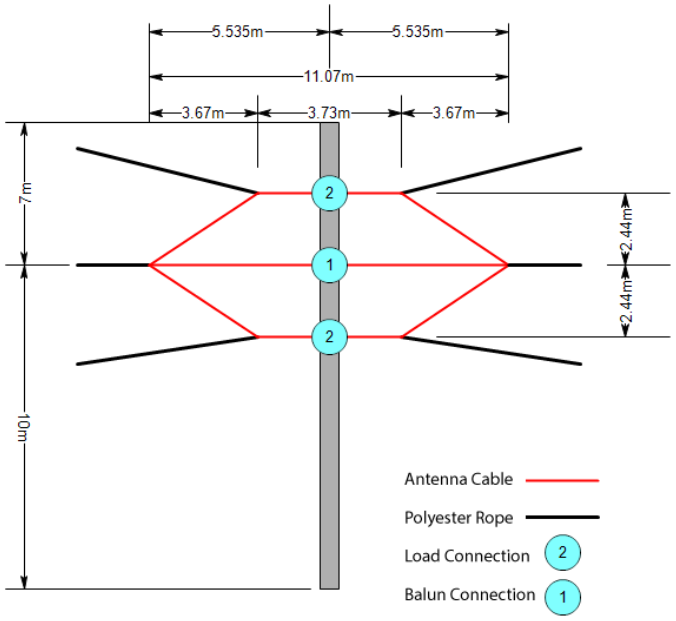

Figure 2.5: Illustration of a SuperDARN TTFD antenna element [*Provided by SANSA*]

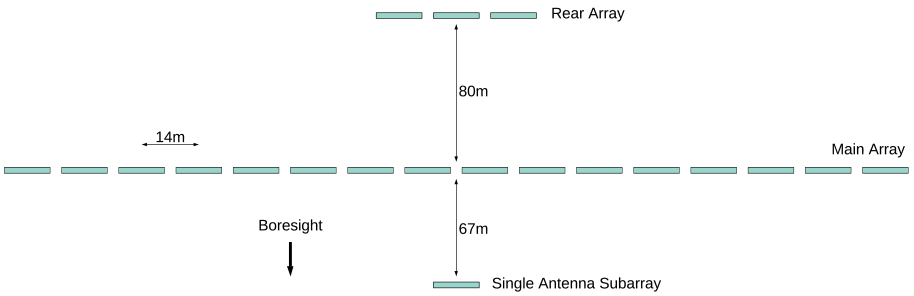

Figure 2.6: Antenna array formation [9].

2.2.1 TIGER Digital Radars

The evolution of the digital version of the TIGER radar was first proposed by J. Whittington et al. in 2002 [28] and was in development to around 2015; when support and development of the radar stalled after the 3^{rd} generation TIGER-3 radar was implemented. Much has changed about the radar architecture, and much of the radar's present hardware has become obsolete.

A digital version of the existing TIGER radar was proposed in a time when digital signal processing (DSP) systems, making use of application specific integrated circuits (ASICs), was becoming the norm for technologies like mobile phones. The initial fabrication cost of ASICs is far too high to be used for low volume production such as for SuperDARN radars. An alternative to ASICs is field programmable gate arrays (FPGAs) which, due to their reconfigurable nature, have more use cases and are therefore more economical to fabricate and buy for low volume applications [28].

The aim of the TIGER project was to improve on the conventional analogue design in the following ways: [15] [28]:

- Using a digital signal synthesiser results in greater frequency output control and higher signal quality.
- Phase adjustments for beamsteering can be controlled and calibrated for each transceiver during run-time.
- Sidelobes are reduced by using a digital windowing function on the transmitted signal.
- Digital control panels can keep track of the state of the system.
- High order digital filters are adjustable, reliable and robust.

The TIGER-3 makes use of a Xilinx Virtex-5 FPGA, which is discussed further in the next section.

2.2.2 An Introduction to Virtex-5 FPGAs

In short, an FPGA device is an integrated circuit containing a large volume of generic components that can be interconnected through switching networks. The primary logic components are logic blocks (LBs) which can be used to perform programmable logic functions based on a set of inputs [29].

The basic architecture of the modern SRAM-based FPGA consists of a matrix of LBs and switching matrices (SMs) [29]. I/O pads connect the FPGAs internal blocks to external devices. The simplified architecture is shown in Figure 2.7. In the case of the Virtex-5;

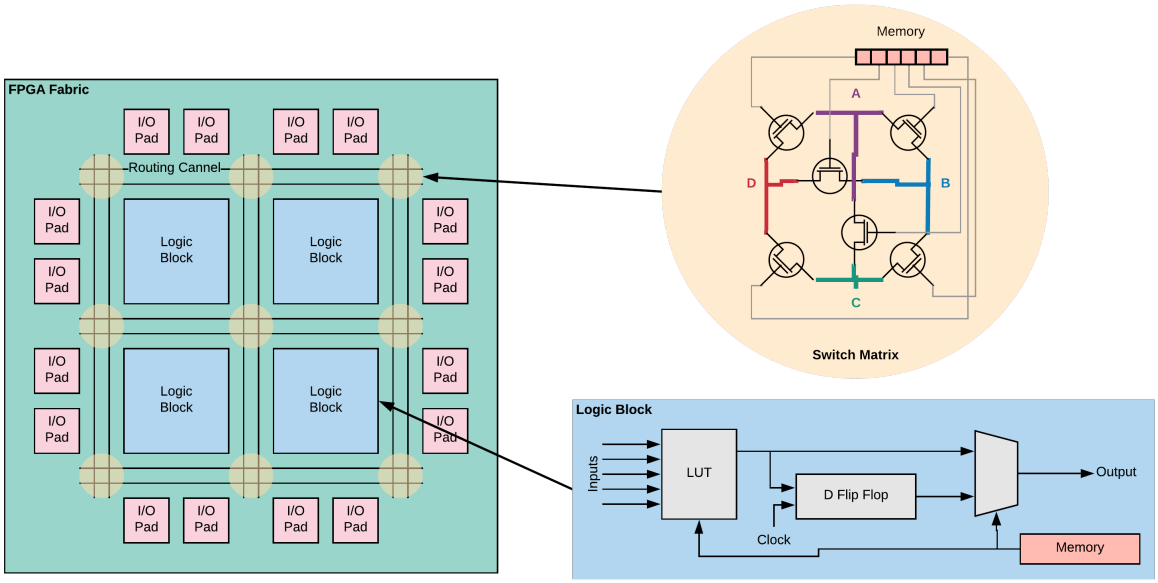

Figure 2.7: Simplified model of a generic FPGA architecture.

LBs contain two 'slices' which each contain four 6-input look-up tables (LUTs) alongside other slice elements. Figure 2.7 shows a functional illustration of a single LUT slice as an LB. This slice element can be used to perform logic operations as combinational or clocked (sequential) logic.

LBs and I/O pads are connected through routing channels and SMs. The SMs allow for each channel to be routed on a specific path. I/O pads allow for the routed signals to interact with external devices through buffers, supporting a range of logic standards.

The term 'primitive' is used to describe the vendor-specific components of an FPGA. They may include basic programmable logic components such as LUTs and flip-flops. Other primitives may include task-specific components such as digital counters, Ethernet interfaces, high speed DSP tiles and clock management tiles (CMTs) [30].

For clocking purposes, the Virtex-5 is divided into regions. Each region allows for clocking signals to be connected through global clock buffers to global clocking lines (GCLs). The GCLs ensure that clocking signals can be transferred between regions optimally, minimising clock skew and capacitive effects.

The reconfiguration of an FPGA is done by loading a bitstream onto the fabric which fills the SRAM (labelled as "memory" in Figure 2.7) and fully defines the operation of the FPGA [29]. Reconfiguration of FPGAs are commonly done through a hardware description language (HDL). The two major HDLs used are; Verilog (Verification & Logic HDL)

and VHDL (Very High Speed Integrated Circuits HDL). VHDL is based on *Ada* which is an object orientated programming language. Verilog closer resembles C programming. HDL synthesises into high-level code and optimally maps the hardware according to constraints from the manufacturers. A bitstream file is generated after synthesis to load to the fabric [31].

Integration

Modern FPGA families include higher level fixed silicon functionality built into the fabric. In the late 1990s it became common for manufacturers to include System-on-chip (SoC) devices as part of the FPGA fabric to provide CPU processing to designs.

Embedded hardware is often referred to as "hard-cores" and form part of the FPGA primitives. In addition to CPUs, it may include DSP slices, Ethernet hardware interfaces, etc. Hard-cores provide a semi-fixed operational state but offer high speeds and low space requirements, similar to that of silicon based ICs.

Soft-cores are similar to functions in C-like programming. They are HDL modules that make use of basic primitives and logic resources to perform specific tasks, based on a set of inputs and parameters. Soft-cores provide design flexibility and modification, but are limited by the amount of logic resources and primitives required for synthesis. A set of soft-cores are made available by manufacturers to perform functional tasks, such as creating communication interfaces or performing complex mathematical functions [32].

2.2.3 TIGER-3 Platform Architecture

The TIGER-3 transceiver was developed by La Trobe University and the University of Newcastle and Adelaide [33]. It makes use of a Virtex-5 FPGA on a breakout board containing peripherals, used for radar operations and FPGA reconfiguration. Each transceiver unit of the radar makes use of an FPGA breakout board and a RF board which connects to an element in the antenna array.

Much of the design was changed for the current implementation at SANAE. A system-wide block diagram of the system, as first planned in 2010 [34], is shown in Figure 2.8. The diagram illustrates the hardware chain from the transceiver to the main computer where the summed receiver data are converted into SuperDARN data.

The system was designed to have interfaces between the transceiver boxes and a main computer. The main computer communicates with a separate "server box", which contains its own TIGER-3 board running Linux OS on a SoC CPU. The server box communicates to the transceivers through switches; transmitting timing and control data and receiving radar data. It combines the received data and transfers the data to the main computer through TCP-IP Ethernet and Firewire [34].

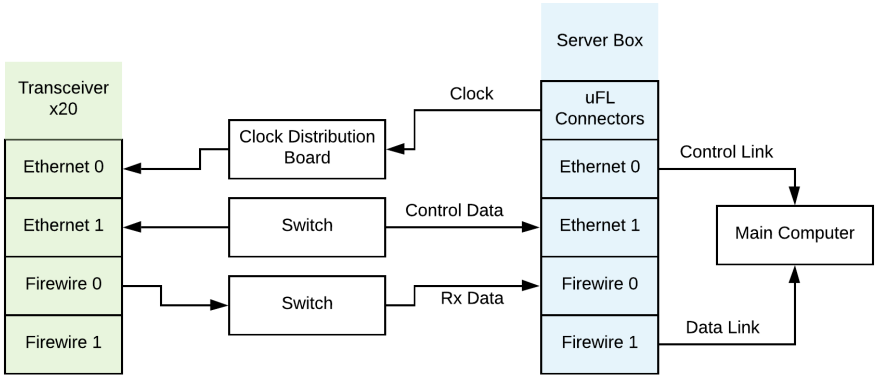

Figure 2.8: Dated TIGER-3 system architecture.

The system architecture underwent multiple changes as development continued. Firewire was a fast ageing technology that quickly became obsolete. Additionally, hardware limitations rendered the second Ethernet port unusable. The result is that the system implementation changed through the years and the currently implemented at SANAE is substantially different. The last documented implementation was that of SANAE IV, shown in Figure 2.9. The current system makes use of a dedicated timing box with an

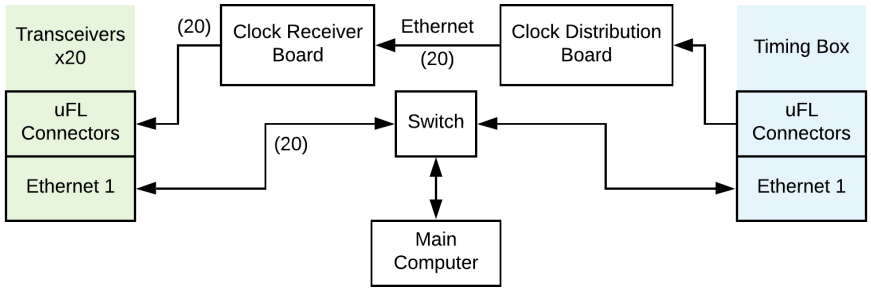

Figure 2.9: Current TIGER-3 system architecture.

FPGA breakout board identical to the transceiver boards, but housing an oven controlled crystal oscillator instead of the crystal oscillator used in the transceivers.

The timing signal from the timing box is distributed with equal delay to each transceiver to ensure that minimal phase error exists between the transceiver boxes. The main computer performs the summing of the receiver data (that was formerly done by the server box), and communicates with the timing box to initiate timing commands sent to the transceiver boxes.

The functions that the TIGER-3 system currently offers are therefore the configuration of the individual transceivers by the main computer, even clock distribution between receivers, the triggering of pulse sequences from the main computer and the reception of the base banded receiver signals.

2.2.4 SANSA TIGER-3 Development Platform

For this project SANSA provided a development platform that includes the Virtex-5, the TIGER-3 breakout board and the corresponding RF board, and the necessary physical interfaces to perform as a complete radar transceiver. The transceiver is shown in Figure 2.10. The system is powered through a fan-cooled AC-DC power supply sourced from

Figure 2.10: SANSA's TIGER-3 based transceiver box.

240 VAC mains. The RF board contains all the periphery used for the SuperDARN on receive. On transmit; some high power related peripherals are left unconnected. The peripherals are further discussed in Section 4.

2.2.5 TIGER Mark-4

The 4th implementation of TIGER was discussed in a presentation at the 2015 annual SuperDARN workshop [15]. TIGER-3 was only just implemented and the concept of the fourth version was only theorised. TIGER-3 largely completed the move to a completely digital SuperDARN, and TIGER-4 aimed at building on that with new features. A "smart radar" was proposed - the concept of a self adapting radar that would alter performance parameters based on real time data analysis.

2.3 Chapter Summary

Firstly, this chapter detailed the physical properties of the ionosphere with respect to plasma convection and RF interference. The ionosphere will reflect RF signals given that the frequency of the incident wave is below that of the critical plasma frequency. In order to measure plasma convection, relatively large irregularities in the F-region of the ionosphere need to be measured. The complex nature of the ionosphere is unpredictable and not well understood, but it has been shown that the reflected waves will only propagate back to the radar when the incident wave is perpendicular to the earth's magnetic field lines. Furthermore, reflection relies on Bragg-scatter, which means that (for maximum return) the frequency of the incident wave has to be double that (according to Bragg's law) of the spatial frequency of the irregularities. In addition to the above limitations in frequency, it was shown that only a certain range of frequencies will undergo refraction in the lower altitude ionosphere. This is important as the refraction of the incident wave allows for the satisfaction of the perpendicular-wave condition mentioned above. It was therefore shown that HF (3-30 MHz) is ideal for the measurement of plasma convection.

The second part of this section provides an overview of the physical design of the system. It briefly covers the antenna formation, corresponding beam parameters, power output, and operational frequencies of the system. An argument is made for the use of digital transceivers in stead of analogue transceivers. Digital transceivers provide a host of advantages, including; system adaptability, accuracy and reliability. The digital transceivers are based on FPGAs to provide high speed time-accurate outputs. The TIGER-3 platform is the third generation TIGER in a series of developments which aimed to eventually complete the shift to a fully digital SuperDARN - much has changed in terms of hardware and software since its inception. An overview of the FPGA architecture was provided, before the past and present transceiver system designs were detailed. The designs show the data and clock paths shared between the transceivers and the main processing computer. Finally details were provided regarding the transceiver box supplied by SANSA to support further development of the system. This box closely resembles that of a single transceiver used at SANAE at the time of writing.

3 System Architecture Description

This chapter concerns the development of supporting models and background theory for the implementation stages discussed in Section 4. It is divided into three sub-sections that detail: (1) The development of a theoretical radar model and complex IQ signal representation. (2) The supporting structures for the Ethernet communication link. (3) The primary FPGA HDL design considerations. The subsections specifically provide supporting formulae and models according to the hardware on the TIGER-3.

3.1 Radar Model

Directly sampling a radar signal is useful for design simplification and frequency versatility. In order to make effective use of the sampled signal, a method is required to transfer, process and store the information. Often a directly sampled signal will contain far more information than what is required for a band-limited signal. Consider a signal sampled at 8-bits per sample at a rate of 125 MHz; the total data rate is 1 Gb/s. In this section, it is detailed how the Nyquist Criterion defines the maximum frequency that can be resolved unambiguously after sampling. Furthermore, it shows how the signal can be manipulated to minimise the data rate while retaining the frequency band of interest. This simplifies the transmission and storage stages of the system and is critical to a digital radar's operation.

3.1.1 Discrete Analytic and IQ Signal Representation

The Nyquist sampling criterion defines the relationship between the maximum frequency unambiguously retained after sampling, and the rate at which the signal is sampled. Here, it is given for a single channel (real) signal, as

$$f_{N_{\Re}} = \frac{f_s}{2} \tag{3.1}$$

where the sampling frequency is given by f_s, and $f_{N_{\Re}}$ defines the maximum unambiguous frequency. All frequency components above the Nyquist frequency limit will be aliased through spectral wrapping - appear as lower frequency components in the sampled signal. Given a sampled signal ($x[n]$) with the frequency representation ($X[\Omega]$), where $\mathcal{DF}\{x[n]\} = X[\Omega]$ denotes the Discrete Fourier transform of the signal $x[n]$;

$$X[\Omega_s] = X[\Omega_s + k\Omega_{Nyquist}] \quad \text{for} \quad k \,\epsilon\, \mathbb{Z} \tag{3.2}$$

where Ω_s denotes the discrete frequency samples.

A signal is therefore typically filtered by analogue electronics before being sampled; ensuring that the high frequency components, which would be aliased after sampling, are sufficiently suppressed prior to sampling.

Radar signals are often processed as baseband IQ data which makes use of the Hilbert Transform to produce an analytic signal, which is basebanded through demodulation to in-phase and quadrature (IQ) data. The process is illustrated in Figure 3.1 and is further detailed mathematically below.

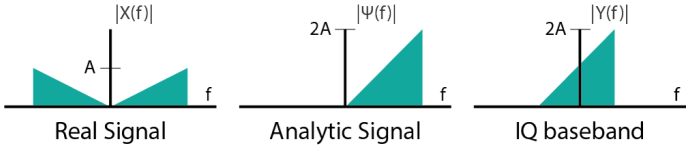

Figure 3.1: Illustration of the spectral content of a signal at specific processing stages for IQ down conversion

A real continuous-time signal ($x(t)$) is fully described by its spectral representation ($X(f)$) for $f \geq 0$, because of the spectral symmetry; $X(-f) = X^*(f)$. The analytic signal ($\Psi(f)$) is a mathematical signal representation of a real signal that only makes use of the positive spectrum [35]. It is given by

$$\Psi(f) = \begin{cases} 2X(f) & \text{for} \quad f > 0 \\ X(f) & \text{for} \quad f = 0 \\ 0 & \text{for} \quad f < 0 \end{cases} \tag{3.3}$$

In order to write $\psi(f)$ in a format that may be inverse transformed, a second spectral signal $Y(f)$ is defined such that $\Psi(f) = X(f) + Y(f)$ and thus

$$Y(f) = \Psi(f) - X(f) = \begin{cases} X(f) & \text{for} \quad f > 0 \\ -X(f) & \text{for} \quad f < 0 \end{cases} \tag{3.4}$$

which can also be represented in complex exponential notation as

$$Y(f) = \begin{cases} je^{-j\pi/2}X(f) & \text{for} \quad f > 0 \\ je^{+j\pi/2}X(f) & \text{for} \quad f < 0 \end{cases} \tag{3.5}$$

A function $\hat{X}(f)$ is defined for $Y(f) = j\hat{X}(f)$, and is therefore given by

$$\hat{X}(f) = \begin{cases} e^{-j\pi/2}X(f) & \text{for} \quad f > 0 \\ e^{+j\pi/2}X(f) & \text{for} \quad f < 0 \end{cases} \tag{3.6}$$

Making use of the above derivations, a time-signal representation can be derived as

$$\psi(t) = \mathcal{F}^{-1}\{\Psi(f)\} = \mathcal{F}^{-1}\left\{X(f) + j\hat{X}(f)\right\} = x(t) + j\hat{x}(t) \tag{3.7}$$

where both $x(t)$ and $\hat{x}(t)$ are real functions (since $x(t)$ has conjugate symmetry). $\hat{x}(t)$ is known as the Hilbert Transform of $x(t)$ and can be found by phase shifting $x(t)$ by $-90°$.

IQ signals are basebanded analytic signals - where the analytic signal is demodulated (further discussed below) to be centred about DC. Due to the non-symmetry of the IQ signal spectrum; the information is processed in real (I channel) and imaginary (Q channel) components. The spectral width is halved in the process and therefore the Nyquist criterion (Equation 3.1) is relaxed to

$$f_{N_{\mathbb{C}}} = f_s. \tag{3.8}$$

Shifting the spectral content of a signal can prove critical for limiting the spectrum and decreasing the required sampling rate of a signal.

The frequency shifting of a signal $x_1(t)$ can be achieved by multiplication with a complex exponential signal $x_2(t)$, given by

$$x_2(t) = e^{j2\pi f_c t} = \cos(2\pi f_c t) + j\sin(2\pi f_c t) \tag{3.9}$$

where f_c defines the frequency of the tone. The Fourier transform of the time-multiplication of two signals is the convolution of their transforms. The operation is given by

$$\begin{aligned} Y(f) &= \mathcal{F}\{x_1(t)x_2(t)\} && (3.10)\\ &= X_1(f) \circledast X_2(f) \\ &= X_1(f) \circledast \delta(f - f_c) \\ &= X_1(f - f_c) && (3.11) \end{aligned}$$

The spectral content of $X_1(f)$ is shifted by f_c. Note that $x_1(t)$ is a complex signal and therefore $y(t)$ is also complex.

In practice, digital frequency shifting is done by the element-wise multiplication of sampled signals by samples from a digitally synthesised signal. Aliasing from the Nyquist criterion will result in a circular shift of the spectrum. Consider a sampled signal $x[k]$ with the DFT pair given by $\mathcal{DF}\{x[k]\} = X[\Omega_s]$. Furthermore, consider Equation 3.2 which shows that the spectral domain of $X[\Omega_s]$ repeats at $\Omega_{Nyquist}$ (represented here by N - the number of samples in the transform). The periodic extension of $X[\Omega_s]$ is given by [36]

$$\tilde{X}[\Omega_s] = X[\Omega_s \bmod N] = X[(\,(\Omega_s)\,)_N] \tag{3.12}$$

where $n \bmod N$ and $(\,(n)\,)_N$ are taken to mean n modulo N, which has the value of the remainder after n is divided by N. Therefore, the discrete equivalent to Equation 3.10 is given by

$$Y[\Omega_s] = \mathcal{DF}\{x_1[n]x_2[n]\} = \frac{1}{N}\sum_{\Omega_s=0}^{N-1} X_1[\Omega_s]X_2[(\,(k-\Omega_s)\,)_N)]. \tag{3.13}$$

The operation is shown graphically in Figure 3.2, by making use of the fast Fourier

transform (FFT) to approximate the discrete Fourier transform (DFT).

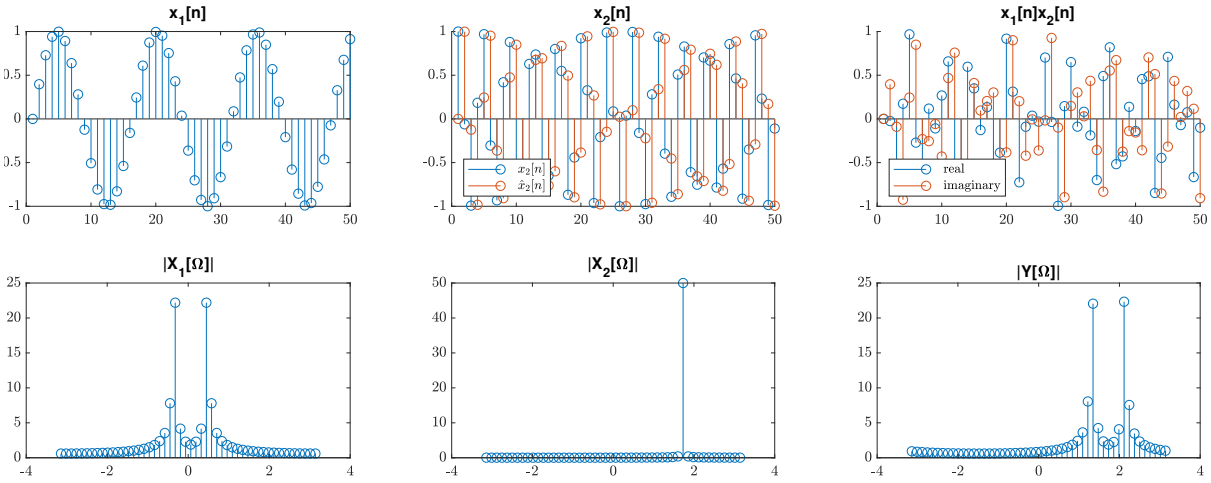

Figure 3.2: Multiplication of X_1 by a tone signal (X_2) to produce Y.

3.1.2 Basebanding and Decimation

In Figure 3.2 a signal (x_1) was translated up in frequency. On receive, digital radars typically down convert to baseband, filter the signal, and down-sample. The process is illustrated in Figure 3.3.

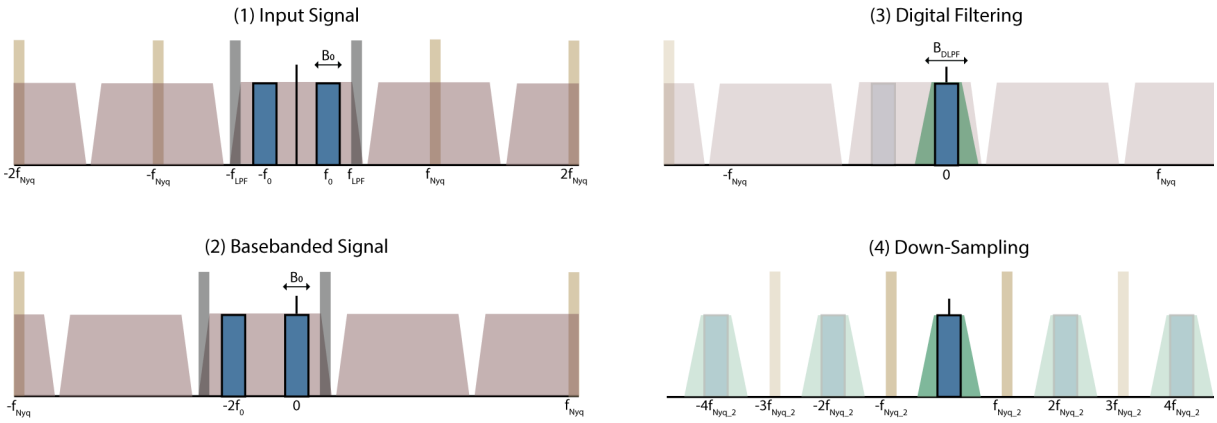

Figure 3.3: Baseband demodulation steps.

Each step is discussed here:

1. The input signal is illustrated by the shaded spectral range, limited by a hardware low-pass filter (LPF). The signal band of interest is highlighted in dark blue. At the initial sampling rate, the Nyquist frequency f_{Nyq} is sufficient to prevent aliasing after the hardware LPF.

2. The input signal is then basebanded by an analytical demodulating signal at a frequency of $-f_0$.

3. Digital low pass filtering removes the image on the left and all other spectral content that underwent circular convolution in the basebanding process. Only the green shaded region contains spectral content after the filtering.

4. The band limited signal may then be down sampled without the risk of aliasing. A new Nyquist rate is set at f_{Nyq_2}. The result is a low sample rate I-Q representation of the band of interest.

3.1.3 The Radar I-Q Model

For a typical radar mode of operation, a signal is transmitted from an antenna and echos are received from N scatterers. Each individual scatterer (n) produces an echo at τ_n that is identical to the transmitted signal (v_{TX}), but attenuated by a_n and delayed in time. The transmit-receive relationship is given by

$$v_{RX}(t) = \sum_{n=1}^{N} a_n v_{TX}(t - \tau_n) \tag{3.14}$$

for a discrete set of scatterers. τ_n is given by

$$\tau_n = \frac{2R_n}{c} \tag{3.15}$$

where R_n is the one way range to the scatterer and c is the propagation speed of the wave. The attenuation is related by

$$a_n \propto \sqrt{\frac{G^2 \sigma_n \lambda^2}{(4\pi)^3 R_n^4}} \tag{3.16}$$

where G is the gain introduced by the antenna (given that the same antenna is used for transmit and receive), σ is the radar cross section of the scatterer and λ is the transmitted wavelength [35].

The Radar Receiver

The translation of the received signal spectrum by f_0 to the left is expressed mathematically as

$$V_{bb}(f) = V_{RX}(f + f_o) \tag{3.17}$$

Following with the steps illustrated in Figure 3.3; a low pass filter is then applied to extract the basebanded signal

$$[\, V_{bb}(f) \,]_{LPF} = V_{RX}(f + f_o) H_{LPF}(f) \tag{3.18}$$

where H_{LPF} is the transfer function of a low pass filter. In the time domain this operation is given by

$$\begin{aligned}
[\, v_{bb}(t)\,]_{LPF} &= v_{RX}(t)e^{-j2\pi f_0 t} \circledast h_{LPF}(t) \\
&= v_{RX}(t)[\cos(2\pi f_o t) - j\sin(2\pi f_o t)] \circledast h_{LPF}(t) \\
&= [v_{RX}(t)\cos(2\pi f_o t)] \circledast h_{LPF}(t) - j[v_{RX}(t)\sin(2\pi f_o t)] \circledast h_{LPF}(t) \\
&= I(t) + jQ(t)
\end{aligned} \tag{3.19}$$

Within a digital radar system, the received signal is typically sampled by one or more ADCs and basebanded by a demodulation signal that is generated digitally. The digital signal synthesis component (used to create the digital demodulation tone) is known as a numerically controlled oscillator (NCO). The demodulation signal is phase shifted (Hilbert-Transform), which is used as an analytic demodulation tone for basebanding.

Transmission

Frequency translation techniques are also useful for the transmission of signals in radar systems. A digital radar can make use of analogue signal multiplication to shift a digitally produced baseband signal to a transmit frequency. The produced (in IQ format) can be translated up in frequency by multiplication with a (complex exponential) sinusoid. However, a real signal is required for transmission. Making use of the real component of the complex signal will result in the spectrum of the complex signal being conjugated and mirrored about the vertical axis. Mathematically the real signal output is given by

$$\begin{aligned}
v_{RF}(t) &= \Re\left\{p(t)e^{j2\pi f_o t}\right\} \\
&= \Re\left\{(I_p(t) + jQ_p(t))\left(\cos(2\pi f_o t) + j\sin(2\pi f_o t)\right)\right\} \\
&= I_p(t)\cos(2\pi f_o t) - Q_p(t)\sin(2\pi f_o t)
\end{aligned} \tag{3.20}$$

where $v_{RF}(t)$ is the transmitted signal, $p(t)$ is the complex baseband digital signal, multiplied by $e^{j2\pi f_o t}$.

Analogue frequency shifting and demodulation is not essential for a radar that can digitally produce or sample a signal at the transmit/receive frequency. This is further discussed in Section 3.1.4.

3.1.4 TIGER-3 Digital Radar Transceiver Model

It should be clear from Section 3.1.3 that I-Q data signal representation does not add or remove signal information. Working with real signals is acceptable given that the sampling criteria are met. Typically, analogue frequency shifting allows for IQ analogue-to-digital conversion (real and complex channels), which means that the total bandwidth in each

channel is halved. The Nyquist criterion is therefore relaxed at the cost of requiring twice the sampling hardware.

Further reason to make use of baseband IQ signals, is that digital filtering is simplified at baseband for digital circuits. The reason for this is that low pass filters are more efficiently implemented digitally than band-pass filters.

For the Tiger-3 SuperDARN radar, the hardware supports the rate required for direct sampling on receive and transmit. However, the data product for the SuperDARN is in IQ format, and therefore this project will make use of IQ data. The radar model designed for this project is shown in Figure 3.4. On transmit, no frequency shifting is required

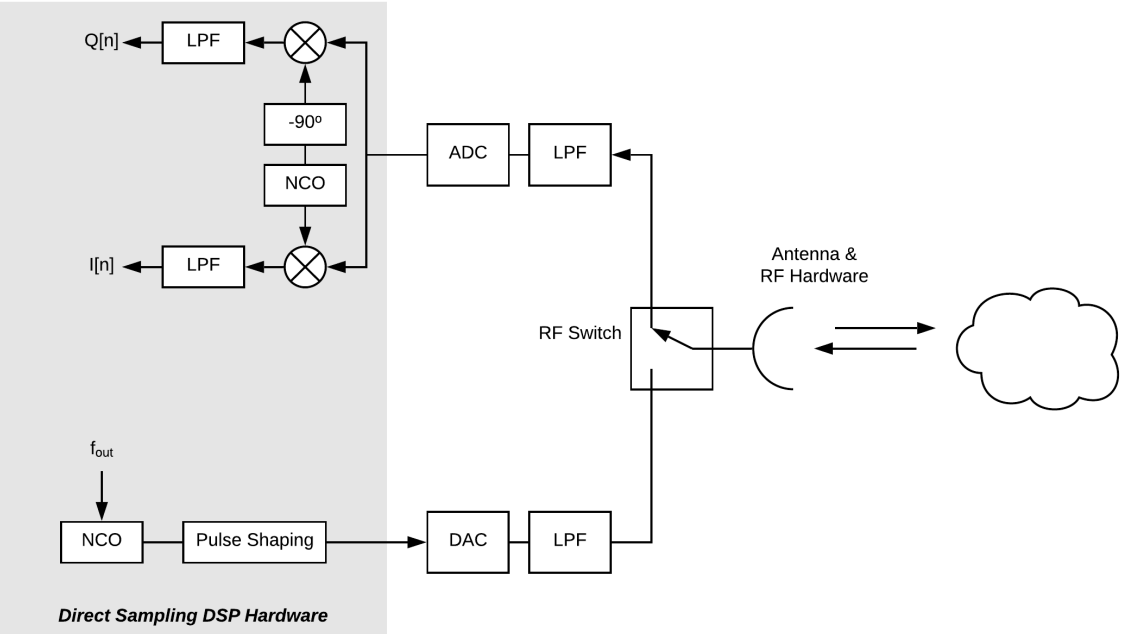

Figure 3.4: Typical digital radar transceiver model with baseband demodulation.

and the only task is to ensure that the transmit signal is correctly phased and shaped according to the mode of operation.

3.1.5 Moving Targets

The radar model has been derived for stationary targets, and the received signal is given in Equation 3.14. However, the SuperDARN is primarily used to measure Doppler shifts in the echos received from moving ionospheric irregularities. This section will expand on the stationary target model to accommodate moving targets [35]. Consider the received

signal for a signal point scatterer, given by

$$v_{RX}(t) = a\, v_{TX}(t - t_d) \tag{3.21}$$

where t_d is the **variable** two-way delay introduced by the moving targets during the time frame that the pulse makes contact with the target. $R(t)$ denotes the target range as a function of time and is given by

$$R(t) = R_0 + vt \tag{3.22}$$

where R_0 represents the range of the target at the time (t_{strike}) that the pulse first makes contact with the target. This occurs when $R(t) = ct$, and t_{strike} is given by

$$t_{strike} = \frac{R_0}{c - v} \tag{3.23}$$

The pulse is compressed or expanded as the time delay (t_d) changes with $R(t)$, during the time that the pulse interacts with the target. The variable delay (t_d) is therefore given by

$$t_d = \frac{2R(t - t_d)}{c - v} = \frac{2(R_0 + v \times (t - t_d))}{c - v} \tag{3.24}$$

which, when solving for t_d, results in

$$t_d = \frac{2(R_0 + vt)}{c + v} = \frac{2R(t)}{c + v} \tag{3.25}$$

Substitution into Equation 3.21 produces

$$\begin{aligned} v_{RX}(t) &= a\, v_{TX}\left(t - \frac{2(R_0 + vt)}{c + v}\right) \\ &= a\, v_{TX}(t\left[\frac{c - v}{c + v}\right] - \frac{2R_0}{c + v}) \\ &= a\, v_{TX}(\frac{c - v}{c + v}\left[t - \frac{2R_0}{c - v}\right]) \\ &= a\, v_{TX}(\alpha[t - \tau]) \end{aligned} \tag{3.26}$$

where the time delay (τ) is given by

$$\tau = \frac{2R_0}{c - v} \tag{3.27}$$

and, since $v/c \ll 1$, the stretching factor (α) estimates to

$$\alpha = \frac{c - v}{c + v} = \frac{1 - v/c}{1 + v/c} \approx 1 - \frac{2v}{c}. \tag{3.28}$$

Similarly,

$$\frac{1}{\alpha} \approx 1 + \frac{2v}{c} \tag{3.29}$$

The energy in the signal theoretically should remain constant for a perfect scatterer. For the conservation of energy; the return signal amplitude will decrease for a positive (*away*) radial velocity and increase for a negative velocity, to accommodate for the time-compression or expansion. The received energy (E_{RX}) is given by

$$\begin{aligned} R_{RX} &= \int_{t=-\infty}^{\infty} |v_{RX}(t)|^2 \mathrm{dt} \\ &= \int_{t=-\infty}^{\infty} a^2 |v_{TX}(\alpha[t-\tau])|^2 \mathrm{dt} \\ &= \frac{a^2}{\alpha} \int_{t=-\infty}^{\infty} a^2 |v_{TX}(t')|^2 \mathrm{dt}' \\ &= \frac{a^2 E_{TX}}{\alpha} \end{aligned} \tag{3.30}$$

The energy in the signal from a stationary target is calculated by changing the received signal in Equation 3.30 to

$$v_{RX}(t) = a_0 v_{TX}(t - 2R/c) \tag{3.31}$$

where a_0 is range dependent. Therefore

$$E_{RX} = a_0^2 E_{TX} \tag{3.32}$$

and since $E_{TX} = E_{RX}$ for energy conservation;

$$a = a_0 \sqrt{\alpha} \tag{3.33}$$

The Fourier Transform of the received pulse is given by

$$\begin{aligned} V_{RX}(f) &= \mathcal{F}\left\{a_0\sqrt{\alpha}\, v_{TX}(\alpha[t-\tau])\right\} \\ &= a_0\sqrt{\alpha}\, \mathcal{F}\left\{v_{TX}(\alpha[t-\tau])\right\} \\ &= \frac{a_0}{\sqrt{\alpha}}\, V_{TX}\left(\frac{f}{\alpha}\right) e^{-j2\pi\tau f} \end{aligned} \tag{3.34}$$

Considering Equation 3.29, and since $v \ll c$,

$$V_{TX}\left(\frac{f}{\alpha}\right) \approx V_{TX}\left(f + \frac{2v}{c} f\right) \tag{3.35}$$

The Doppler shift (f_D) of the receive signal characterises the radial velocity of a target and is given by [35]

$$f_D = -\frac{2v}{c} f \tag{3.36}$$

3.1.6 Doppler Processing

Section 3.1.5 detailed the presence of a Doppler shift in the received signal from a moving target. It is critical for the SuperDARN to be able to measure the Doppler spectrum of the ionosphere (see Section 2.1.2). In practice, radar Doppler processing makes use of multiple echos to better resolve the Doppler frequency of the signal [1]. Consider the Fourier pair for the frequency representation of a square pulse time-signal. It is given by

$$\mathcal{F}\left\{rect\left(\frac{t}{\tau}\right)\right\} = \tau \text{sinc}\left(\frac{\omega\tau}{2}\right) \tag{3.37}$$

where $\text{sinc}(x) = \sin(x)/x$ represents the sinc function. A radar transmitter may make use of a periodic pulses to measure both Doppler and range from the received signal. Practically, only a finite set of pulses can be processed to obtain Doppler information (five pulses are considered in the example given in Figure 3.5). Consider the frequency convolution that occurs with time multiplication; the limited sequence of modulated pulses can be represented as the multiplication of a high frequency continuous signal, a train of small pulses and one large pulse. The resulting spectrum is shown in Figure 3.5.

The Rayleigh bandwidth of a pulse (given by $\delta d \propto \frac{1}{T_d}$) defines the bandwidth occupied by the pulse's spectral representation. It can be used to roughly define Doppler resolution in this case. It shows that Doppler resolution is improved by increasing the number of pulses processed [1].

Just as range ambiguity depends on the pulse repetition frequency (PRF) of the pulse sequence, so does the Doppler ambiguity. The Doppler spectrum repeats at the PRF [1]. The choice of PRF therefore introduces a trade-off between Doppler-range ambiguity. It is for this reason that typical pulsed radars are not well suited for the SuperDARN: The SuperDARN typically measures backscatter from 100-4500 km in range and the velocity of the plasma can exceed 2 km/s [37]. If the corresponding Doppler readings of 186 Hz is required then the PRF needs to be set to at least 377 Hz, which then limits the maximum unambiguous two-way range to 400 km.

To solve the range-Doppler limitations, the SuperDARN makes use of a pulse sequence with staggered pulses resulting in unique PRFs (detailed in Section 3.1.8). The signals are processed using correlation functions, detailed in Section 3.1.7.

3.1.7 The Auto-Correlation Function

To determine the presence of an echo in a noisy received signal, a correlation function is used to maximise SNR at time-delays where the received signal is well correlated to the transmitted signal. The majority of SuperDARN literature refers to the function as an *auto-correlation function*. Strictly this is the incorrect term to use for the function, as auto-correlation refers to the cross-correlation of a signal with itself. In the case of the

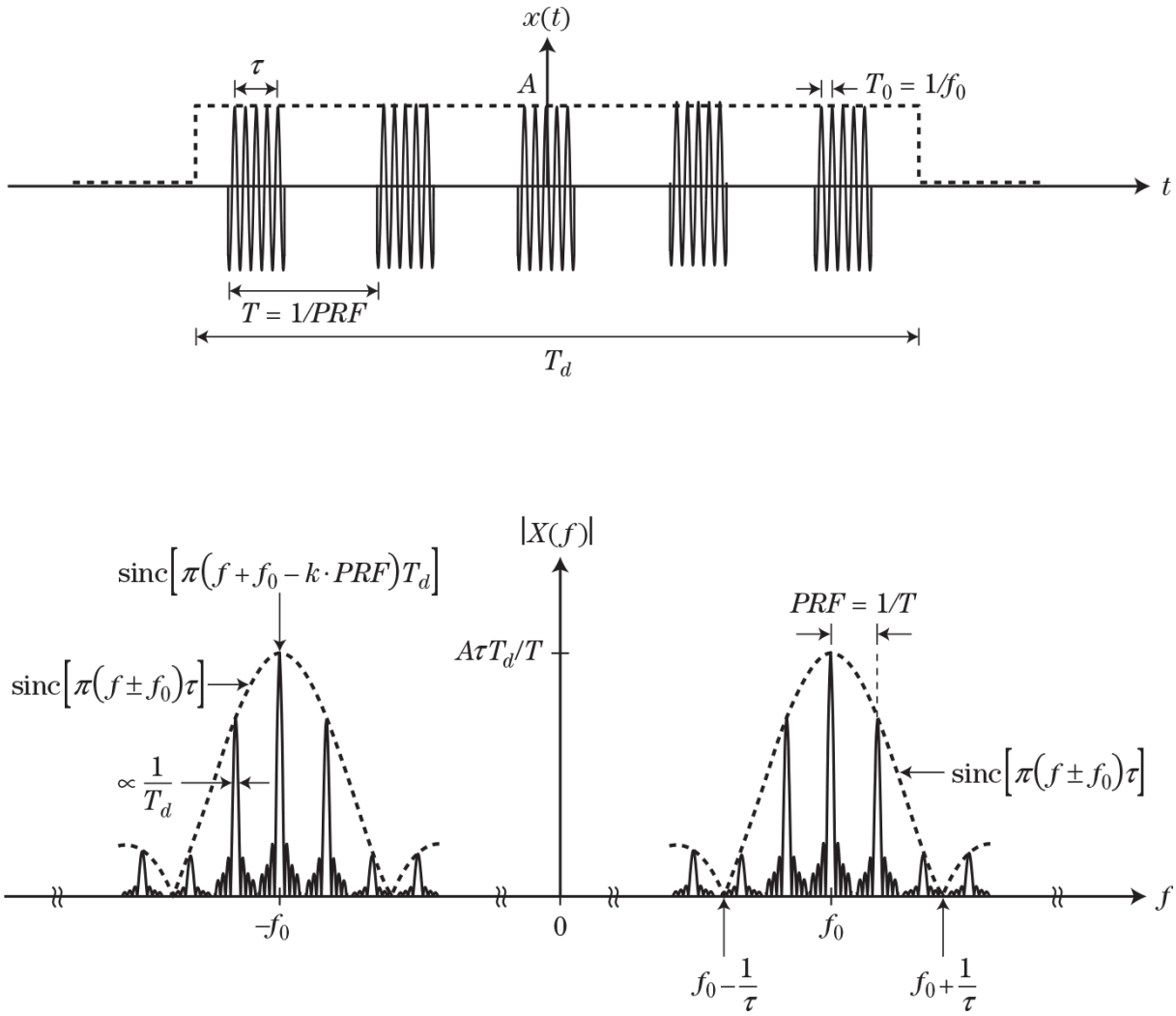

Figure 3.5: Doppler spectrum for a truncated pulse sequence [1].

SuperDARN the received signal is cross-correlated with the transmitted signal. In order to stay consistent with supporting literature the terms are used interchangeably. In the case of the SuperDARN the correlation is performed for a sequence of pulses, as discussed in Section 3.1.8. The cross-correlation ($R_{xy}(\tau)$) of two signals ($x(t)$ and $y(t)$) is given by [38]

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x^*(t)y(t+\tau)\mathrm{dt} \tag{3.38}$$

The auto-correlation function (ACF) defines the cross-correlation of a signal with itself. The ACF of an energy signal is shown to be Fourier pairs with the signal's energy spectral Density (ESD) [38]. The ACF of an energy signal $x(t)$ is given by

$$R_x(\tau) \equiv \int_{-\infty}^{\infty} x^*(t)x(t+\tau)\ \mathrm{dt} \tag{3.39}$$

The Fourier transform produces

$$\begin{aligned}\mathcal{F}\{R_x(\tau)\} &= \int_{-\infty}^{\infty} R_x(\tau)e^{-j\omega\tau}\,\mathrm{d}\tau \\ &= \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} x^*(t)x(t+\tau)\,\mathrm{dt})\right\}e^{-j\omega\tau}\,\mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} x^*(t)\left\{\int_{-\infty}^{\infty} x(t+\tau)e^{-j\omega[t+\tau]}\,\mathrm{d}\tau)\right\}e^{-j\omega\tau}\,\mathrm{dt} \\ &= \int_{-\infty}^{\infty} x^*(t)X(\omega)e^{-j\omega\tau}\,\mathrm{dt} \\ &= X^*(\omega)X(\omega) = |X(\omega)|^2 \end{aligned} \tag{3.40}$$

where $|X(\omega)|^2$ is the energy spectral density of $x(t)$. The ESD is therefore the squared magnitude of the Fourier transform of the $x(t)$, and can be used to measure a Doppler shift in the spectrum. The SuperDARN makes use of the ACF function to integrate multiple received pulses. Echos measured by the SuperDARN are received with inconsistent phase offsets and therefore time-signal pulse integration can not be used. The ESD is independent of phase and provides an effective method of integration. More details about the phase inconsistencies are detailed in Section 3.1.8.

3.1.8 SuperDARN Pulse Sequence

The range-Doppler limitations were detailed in Section 3.1.6. The SuperDARN solves this problem by making use of a pulse sequence. Although not directly relating to the outcomes of this project; this section provides a detailed description of the pulse sequence as it forms a critical part of the SuperDARN's operation. The scope of this study accommodates research that may support future development. Future work on the transmit chain of the transceiver will rely on the theory detailed here.

The SuperDARN pulse sequence is chosen to ensure that the time differences (*lags*) between any two pulses in the sequence are unique. The SuperDARN at SANAE makes use of the katscan-8 pulse sequence [37] which consists of a total of 8 pulses. The transmission sequence is shown, as measured from the current system, in Appendix B. Pulses are transmitted according to a lag pattern and the receiver is enabled between the individual pulses. Figure 3.6 illustrates a 3 pulse sequence with lags given as τ and where d_0 denotes the range of interest.

Note that d_0 is illustrated at a range that results in the reflected pulses returning after the transmission pulses have all been transmitted. Realistically, the range gates under test will be close enough for t_1 to appear well before $t_0 + 3\tau$. The grey blocks show scattering at unwanted ranges that overlap with the echos from the range of interest - marked in red. The grey reflection are known as cross-range interference (CRI), which is unavoidable for any pulse sequence.

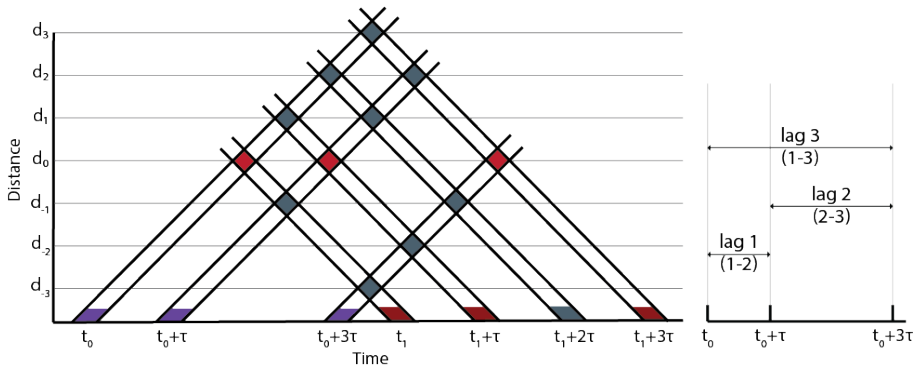

Figure 3.6: Left: 3-pulse sequence range-time plot. Right: Pulse sequence lags.

In Figure 3.7, a two pulse sequence is shown. The pulses ($P11$ and $P12$) are separated

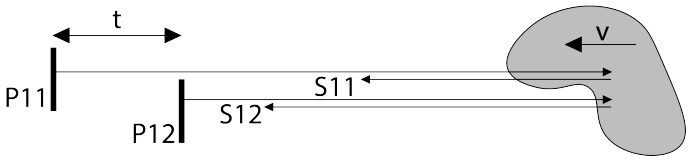

Figure 3.7: Two-pulse, single sequence return.

by time, t. Now consider two pulse sequences, second sequence having pulses $P21$ and $P22$. The return pulses ($S11$, $S12$, $S21$, $S22$) of two consecutive pulse sequences can be mathematically represented by

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} A_1 e^{i\phi_1} & A_1 e^{i(\phi_1+\omega\tau)} \\ A_2 e^{i\phi_2} & A_2 e^{i(\phi_2+\omega\tau)} \end{bmatrix} \tag{3.41}$$

where ϕ_n indicated the phase of the returned signal, and A_n the amplitude. Note that $\phi_1 \neq \phi_2$, but that $\omega\tau$ is assumed to remain constant between the two sequences. $\omega\tau$ directly relates to the velocity of the plasma being measured. Therefore, simply integrating pulse sequences would result in the destructive summation of the random phase components (ϕ_n).

The ACF is given by $R_{(a,b)_c}$, where a and b denote the signals being correlated in reference to pulse sequence c. Hence

$$\begin{bmatrix} R_{(1,1)_1} & R_{(1,2)_1} \\ R_{(1,1)_2} & R_{(1,2)_2} \end{bmatrix} = \begin{bmatrix} S_{11}S_{11}^* & S_{12}S_{11}^* \\ S_{21}S_{21}^* & S_{22}S_{21}^* \end{bmatrix} = \begin{bmatrix} A_1^2 & A_1^2 e^{i\omega\tau} \\ A_2^2 & A_2^2 e^{i\omega\tau} \end{bmatrix} \tag{3.42}$$

which allows for integration, after the removal of the random phase factor.

Now, consider the effects of range aliasing. Given two areas of backscatter, such that

S_{11} from the first area arrives at the same time as S_{12} from the second area (areas spaced $ct/2$ apart). If B_n (amplitude) and ϑ_n (phase) denotes the parameters of the reflection from the secondary scatterer,

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} A_1 e^{i\phi_1} & A_1 e^{i(\phi_1+\omega\tau)} + B_1 e^{i\vartheta_1} \\ A_2 e^{i\phi_2} & A_2 e^{i(\phi_2+\omega\tau)} + B_2 e^{i\vartheta_2} \end{bmatrix} \tag{3.43}$$

and therefore

$$\begin{bmatrix} R_{(1,1)_1} & R_{(1,2)_1} \\ R_{(1,1)_2} & R_{(1,2)_2} \end{bmatrix} = \begin{bmatrix} A_1^2 & A_1^2 e^{i\omega\tau} + A_1 B_1 e^{i(\vartheta_1-\phi_1)} \\ A_2^2 & A_2^2 e^{i\omega\tau} + A_2 B_2 e^{i(\vartheta_2-\phi_2)} \end{bmatrix} \tag{3.44}$$

The random phases of the second term (containing B_n) will statistically tend to cancel out after multiple integration steps.

Furthermore, consider the extension of the current model to 3 evenly spaced pulses. The reflections are then given by

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} Ae^{i\phi} \\ Ae^{i(\phi+\omega_1\tau)} + Be^{i\vartheta} \\ Ae^{i(\phi+2\omega_1\tau)} + Be^{i(\vartheta+\omega_2\tau)} + Ce^{i\xi} \end{bmatrix} \tag{3.45}$$

where R_1 can be calculated by $R_1 = S_2 S_1^*$ and $R_1 = S_3 S_2^*$. Note that range ambiguity for the sequence is unavoidable. Finally, extend the model to a unique-lag 3 pulse sequence (shown in Figure 3.6) where

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} = \begin{bmatrix} S_3 S_3^* \\ S_2 S_2^* \\ S_2 S_0^* \\ S_3 S_0^* \end{bmatrix} = \begin{bmatrix} A^2 + \cdots \\ A^2 e^{i\omega_1\tau} + \cdots \\ A^2 e^{i2\omega_1\tau} + \cdots \\ A^2 e^{i3\omega_1\tau} + \cdots \end{bmatrix} \tag{3.46}$$

with the random phase terms not shown. R_n therefore provides the discrete lags for the ACF function with the random phase terms attenuating after the integration of multiple readings.

It is therefore shown that an evenly spaced pulse sequence produces all ACF lags but that the range aliasing exists for every lag. The unique-lag sequence solves the range aliasing problem at the cost of the following:

- You get missing lags for large pulse sequences (unless there are repeated lags).
- The random phase terms are only effectively removed after multiple integration steps.

As shown in Section 3.1.7, the Fourier transform of the ACF can be used to measure a Doppler shift in the echos. The SuperDARN does, however, not make use of the Fourier transform. A *FITACF* function was developed to estimate the spectral properties directly

from the ACF phase data [39].

The 3-pulse sequence lag pattern illustrated in Figure 3.6 is given by $P_{n=3} = [0, 1, 3]$. The typical katscan parameters are detailed in Table 3.1. The first 24 lags are considered

Table 3.1: SuperDARN pulse sequence parameters.

Number of pulses	8
Pulse Pattern	$P_{n=8} = [0, 14, 22, 24, 27, 31, 42, 43]$
Lag width τ	1.5ms
Pulse width	0.3ms

for the final ACF data product. For the katsacan sequence lags 6 and 23 are missing in the first 24 lags. The ACF function is applied around each lag with reference to a specific range bin. Since the input to the ACF is complex I-Q data the output is also complex.

An example of the 8-pulse ACF output is shown in Figure 3.8. Two lags are missing and a further four are not considered (Bad Lags) due to large CRI scatter corresponding to the range gate under test [37].

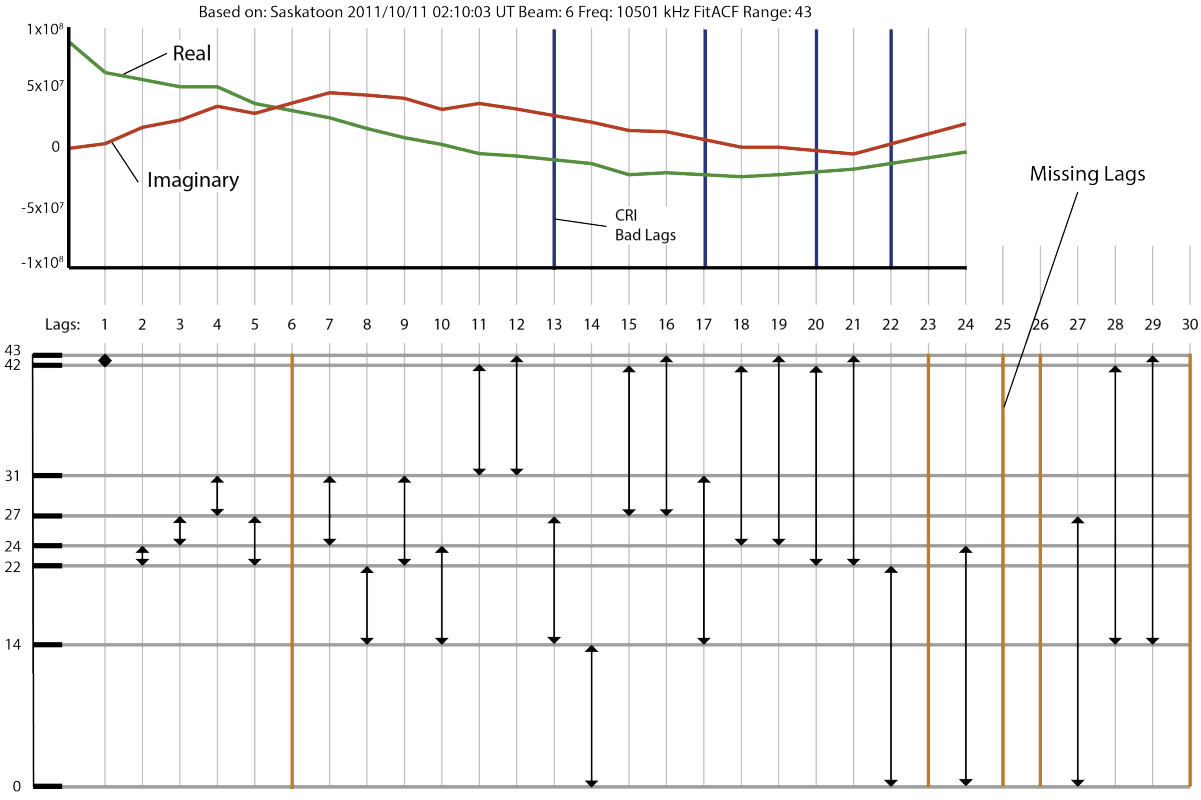

Figure 3.8: ACF function outputs aligned with pulse sequence lags for a set range bin.

The advantage of this approach is the increase of the maximum ambiguous range while

avoiding Doppler ambiguity. Subsequently, effective transmitted power is reduced due the delays between pulses, and the introduction of bad lags for certain rage gates results in longer required integration times to improve SNR [37] [40].

3.1.9 Digital Filtering

Digital filtering is critical to the operations discussed in Section 3.1.2. This section will provide the models and supporting theory for a specific family of digital filters. It has been left for the end of this chapter as it directly concerns the implementations chosen for this project.

Digital filters are often preferred to their analogue counterparts, even for analogue circuits - where analogue signals may be digitised, filtered and converted back to analogue in an existing analogue system [36]. Advantages of using digital filters include; higher order filter capabilities where doing so in analogue would result in large multistage complex circuitry. Digital filters also ensure a linear phase response unlike their analogue counterparts. The greatest advantage, however, is the versatility of digital filters; allowing for reprogramming and simple changes based on the user's needs. Digital filters typically make use of sampled and stored data, where the output sample for a corresponding input sample may depend on samples that were received before or after it. This introduces a latency which could be an application specific disadvantage.

A digital filter is characterised by its difference equation [36]. Finite impulse response (FIR) filters typically make use of a finite set of coefficients. The z-transform can be viewed as a discrete-time equivalent of the Laplace transform. By making use of an inverse z-transform; a difference equation (z-domain transfer function) can be converted to the sampled-time domain, where the coefficients indicate the weighting of past and future input samples for a given output sample. The z-transform of a signal $x[n]$ is defined by

$$\mathcal{Z}\left\{x[n]\right\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = X(z) \tag{3.47}$$

where the discrete-time Fourier transform may be obtained for $z = e^{j\omega}$. The transfer function $H(z)$ is given, for an input $X(z)$ and an output $Y(z)$, as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{q=0}^{M} z^{-q}\beta_q}{\sum_{p=0}^{N} z^{-p}\alpha_p} = \frac{\beta_0 + z^{-1}\beta_1 + z^{-2}\beta_2 + \cdots + z^{-M}\beta_M}{\alpha_0 + z^{-1}\alpha_1 + z^{-2}\alpha_2 + \cdots + z^{-N}\alpha_N} \tag{3.48}$$

where a finite set of β_q and α_p can fully describe a FIR filter.

A sub-class of FIR filters is the cascaded integrator-comb (CIC) filter [41], which strictly performs low-pass filtering - unlike FIR filters which can be used for low, band and high-pass filtering. A CIC filter finds its application in filter decimation and interpolation,

which is a function of its architecture. It only makes use of summation and subtraction, where a typical FIR filter would require multiplication by coefficients. This allows for simpler design architecture for digital implementation. Although the filter is typically implemented in a pipelined form; for simplicity, Figure 3.9 shows the non-pipelined form.

The principles of the CIC filter summarised below is detailed in the core data sheet [41] and is based on work by Eugene B. Hogenauer [42]. The difference function for a CIC decimation filter, is given by

$$H(z) = \left[\sum_{k=0}^{RM-1} z^{-k} \right]^{N} \tag{3.49}$$

which can be described as filtering an input signal by a cascade of N unit-amplitude rectangular windows of length RM. The filter can be implemented as an interpolator or a decimator, where R defines the factor at which the output is up-sampled or decimated.

Both implementations are shown graphically in Figure 3.9, where N is the number of

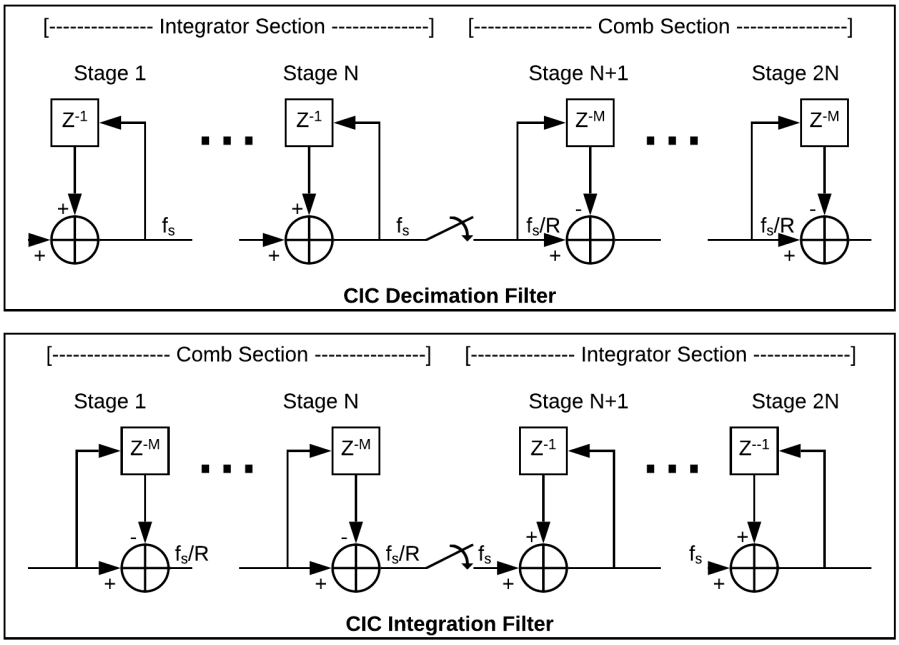

Figure 3.9: CIC filter diagram showing decimation and integration stages

CIC stages and M is the *differential delay* in the comb section stages of the filter. In the case of the decimation filter; the transfer function can be expressed as

$$H(z) = \left(\frac{1 - z^{-RM}}{1 - z^{-1}} \right)^{N} = \frac{(1 - z^{-RM})^{N}}{(1 - z^{-1})^{N}} \tag{3.50}$$

for the interpolation filter (at the lower sampling rate), or as

$$H(z) = \left(\frac{1 - z^{-M}}{1 - z^{-1/R}} \right)^N = \frac{(1 - z^{-M})^N}{(1 - z^{-1/R})^N} \tag{3.51}$$

for the decimation filter (at the higher sampling rate). The denominator and the numerator defines the integrator and the comb section of the filter respectively. Each section containing N stages.

The magnitude frequency response ($z = e^{2j\pi f}$) of the filter is given by

$$|H(z)| = \begin{cases} \left| \frac{\sin(\pi RMf)}{\sin(\pi f)} \right|^N & \text{at the low sampling rate} \\ \left| \frac{\sin(\pi Mf)}{\sin\left(\frac{\pi f}{R}\right)} \right|^N & \text{at the high sampling rate} \end{cases} \tag{3.52}$$

For decimation, it is important to note the effects of the spectral wrapping detailed in Equation 3.13. In this case, filter-suppressed high frequency spectral components are aliased into the output pass-band. For the normalised spectrum, a filter a cut-off frequency given by f_c, and where $\lfloor \, \rfloor$ denotes the floor function; frequency bands in the interval of

$$\frac{k}{RM} \pm f_c, \quad k = 1, 2 \ldots \left\lfloor \frac{R}{2} \right\rfloor. \tag{3.53}$$

alias into the output band. On the left of Figure 3.10, the ideal filter response **before decimation** is shown for a decimation factor of $R = 8$, $N = 3$ stages and $M = 1$. Note

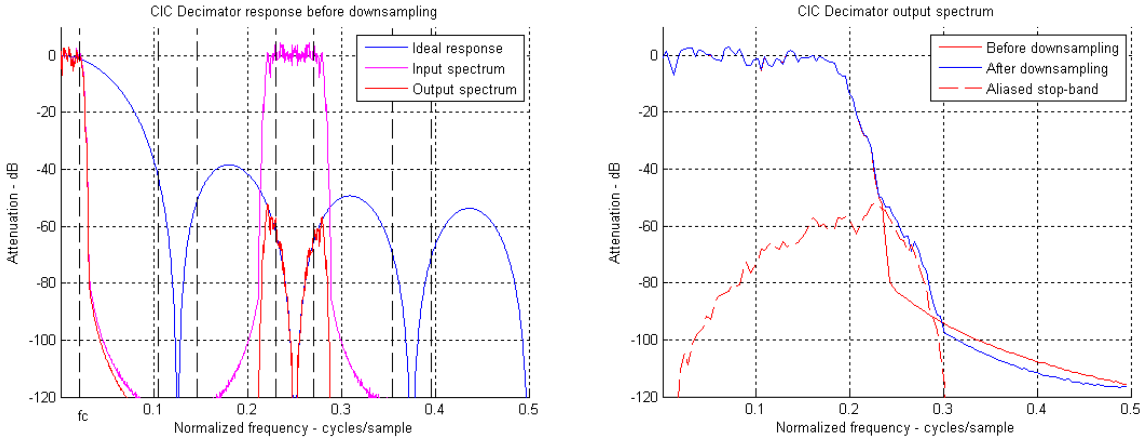

Figure 3.10: Effects of wrapping after CIC filter decimation on the output spectrum [41].

that the input spectrum contains power in the pass band (up to the cut-off frequency, f_c=1/32 cycles/sample), and in the stop-band at approximately 1/4 cycles/sample. The vertical dashed lines illustrate the regions that are aliased according to Equation 3.53.

The plot on the right of Figure 3.10 illustrates the effect that the wrapping from the centre of the plot on the left has on the pass-band. Considering the downsampled output compared to the output before downsampling, indicates a decrease in attenuation for the corresponding stop-band region.

3.2 Communication Model

Ethernet is a computer networking technology that is commonly used for transferring data across local to wide area networks [43]. IEEE standards currently support up to terabit Ethernet (>100 Gb/s). Slower data rate (such as gigabit Ethernet) devices are well established and commercially available to provide small to large area networking solutions to clients. Ethernet is best described as a layered protocol stack. The lowest layer is the physical layer (PHY) which concerns the electrical transmission of bits over a physical medium. Each layer serves a specific task based on the layers below it. Higher layers concern the transmitted data, in which case each higher layer's data set is contained within a section of a lower layer's data.

On a hardware level, the layers may cross over different hardware interfaces. Figure 3.11 illustrates the component-level hardware interfaces that make up the Ethernet communication chain on the TIGER-3. A client typically interfaces with a Media Access Control

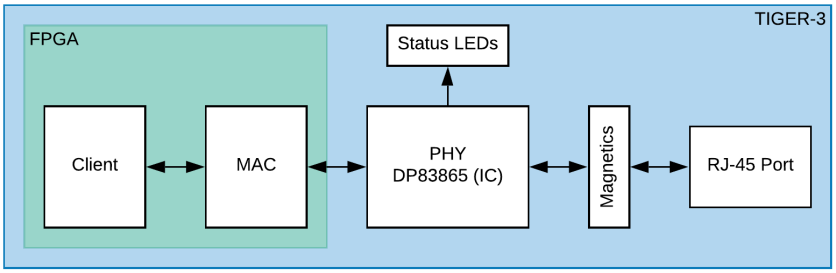

Figure 3.11: Ethernet physical interfaces on the TIGER-3.

(MAC), which is responsible for interfacing with the PHY, which in turn transmits data over a transmission medium. This project makes use of gigabit Ethernet hardware that complies to the Ethernet communication standards set by IEEE802.3 [43].

Data in the higher layers are transferred in frames that can make up a larger stream of data. Each frame makes use of protocol layers which provide information to receiving devices regarding the content of the frame. Figure 3.12 illustrates the complete Ethernet hardware and software interface layers. Above the physical layer, higher layers provide high level information about the data in the frame. This information is provided in the protocol layers - detailed here. The following three generalised layers are used for this

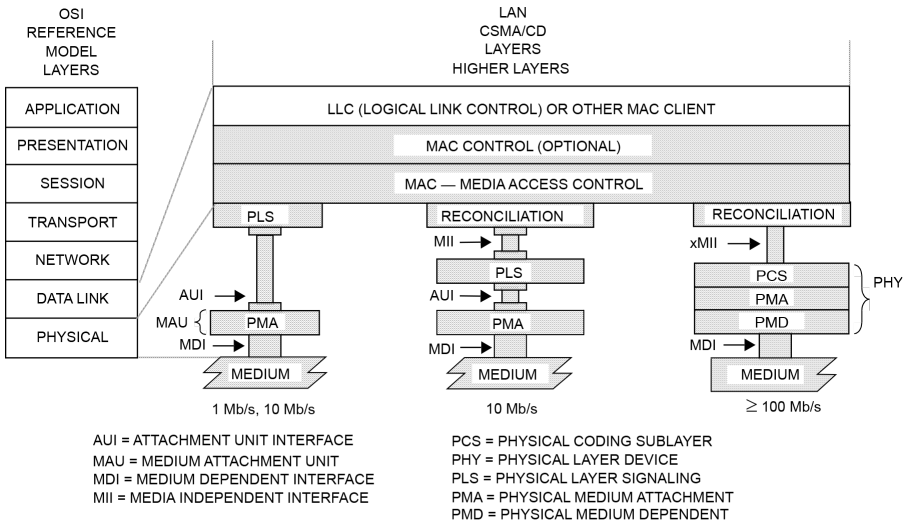

Figure 3.12: Detailed Ethernet layer model according to IEEE802.3 [43].

project: The **Ethernet layer** concerns both the PHY and the data link layer - the latter is the lowest layer that the client interfaces with. In order to provide networking capabilities, the **network and transport layers** follow. Higher layers are typically used for larger networks, and was not required for this project. Each of the generalised layers are discussed below.

3.2.1 Ethernet Layer

The Ethernet layer is the lowest layer of the Ethernet frame and concerns the interface between the MAC, and the PHY. This layer contains all the information transmitted over the transmission media. This implementation makes use of 1000BASE-T half duplex Ethernet standards [43]. It supports theoretical data rates of up to 1Gb/s (*1 gigabit per second*) over twisted-pair transmission media. The IEEE 802.3 provides a standard data frame format which contains all the information that is transmitted to the PHY. The data structure (*with VLAN disabled*) is shown in Figure 3.13. Excluding the client data, the Ethernet header contains 304 bits of information. The **preamble** contains a series of alternating binary for synchronising client timing and is ended by the **start of frame deliminator** to end the alternating sequence. This is immediately followed by the **destination and source MAC addresses**. The **length** field defines the number of octets in the subsequent **MAC client data**. After the data, a **frame check sequence** is provided to ensure the integrity of the frame. On the client's request, the MAC generates all but the **layer 2 Ethernet frame** data - which needs to be provided by the client to the MAC.

Layer	Preamble	Start of frame deliminator	MAC destination	MAC source	Ethertype or length	MAC Client Data	Frame check sequence	Interpacket gap
	7 octets	1 octets	6 octets	6 octets	2 octets	46-1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			64-1518 octets					
Layer 1 Ethernet frame	72-1526 octets							12 octets

Figure 3.13: Ethernet layer data fields.

3.2.2 Network Layer

Within the MAC client data, another protocol standard is used to provide internetworking capabilities. This project makes use of the IPv4 (Internet Protocol version 4), which is commonly used for local area networks (LANs), for which data delivery is prioritised over sequencing and duplicate avoidance (this can be done on a higher layer). The standard is defined in the IETF publication RFC 791 [44], and the packet format is shown in Figure 3.14. Excluding the payload, the header contains 160 bits of information. The fields

MAC Client Data													
Version	IHL	DSCP	ENC	Total Length	Identification	Flags	Fragment Offset	Time to Live	Protocol	Header Checksum	Source IP Address	Destination IP Address	IPv4 Packet Payload
4 bits	4 bits	6 bits	2 bits	2 octets	2 octets	3 bits	13 bits	1 octet	1 octet	2 octets	4 octets	4 octets	26-1480 octets

Figure 3.14: IPv4 protocol layer fields.

include the following: **Protocal** is used to define the IP protocol being used (in this case IPv4). The **header checksum** ensures the integrity of the header information and is followed by the **IP addresses** for network routing. The **packet payload** may be 0 octets in size, in which case the Ethernet layer will pad the transmitted payload. The remaining header fields are elaborated on in RFC 791 [44].

3.2.3 Transport Layer

The final protocol layer used is the User Datagram Protocal (UDP) which forms part of the IP suite and is defined in RFC 768 [45]. The UDP layer format is shown in Figure 3.15. The UDP header consists of 48 bits. The advantage of using UDP, is that no

IPv4 Packet Payload				
Source Port	Destination Port	Length	Checksum (optional)	UDP Payload
2 octets	2 octets	2 octets	2 octets	20-1474 octets

Figure 3.15: UDP protocol layer fields.

prior communication is required to set up data paths. A checksum (optional) is used to confirm data integrity but no handshaking between source and destination is performed. This optimises data size and rate. Data packets may however be dropped or lost and no retransmission will be triggered. The UDP layer defines the source and destination ports which may be used for data multiplexing in complex data systems.

Other transport protocols provide more complex functionality at the cost of retransmission delays and overhead. TCP-IP is commonly used when handshaking between hosts are required.

3.2.4 Data Structure and Transmission

From the previous sections, a maximum data size of 1.5 kB can be transported on a network layer. Alternatively, by making use of jumbo frames (supported by IPv4), the maximum packet size can be increased to approximately 9 kB - given that the Ethernet adaptor supports it. Jumbo frames allow for an increase in the ratio between overhead from the protocol layers and payload data, increasing the total payload data transmission rate. The existing SuperDARN system makes use of jumbo frames, but it was not implemented in this project as PC hardware limitations did not support it. The complete data structure used is illustrated in Figure 3.16.

Ethernet	IPv4	UDP	Packet Info	Payload							Ethernet
				I_1	Q_1	I_2	Q_2	...	Q_N		
22 octets	20 octets	8 octets	6 octets	1396 octets							16 octets

Figure 3.16: Complete data packet structure.

Consider the preamble, header data and the inter-packet gap shown in Figures 3.13 to 3.15. The data are transmitted at 125 MHz (8 bits) according to the standard. The transmission of header data and the hardware required inter-packet gap, decreases the total end-user payload data rate over multiple frames. Therefore the payload data rate is below the rated 1 Gb/s data rate.

A simple approach was taken for the data layout design (elaborated on in Section 4.5), where the I-Q data are transmitted in 32-bit sets (16-bit in-phase, followed by 16-bit quadrature). At the start of the frame one 48-bit dataset is transmitted containing information on the packet for debugging and data classification. Given the packet size limit (L), the number of I-Q data samples (N), transmitted per packet is given by

$$N = \left\lfloor \frac{L - 34}{4} \right\rfloor \tag{3.54}$$

resulting in a maximum of 366 samples per packet. For debugging purposes, this project made use of 349 samples per packet.

3.3 Virtex-5 Hardware

The FPGA in use for the current TIGER-3 platform is the Virtex-5 XC5VSX50T. The Virtex-5 was first introduces by Xilinx in 2006 and is based on 65 nm processing technology. The SXT models (includes XC5VSX50T) are designed for DSP intensive applications as apposed to logic intensive applications (prioritised for the LXT models). The Virtex-5 user guide [46] and data sheet [47] are critical to understanding the limitations associated with FPGA fabric.

3.3.1 Clocking

Before programming, device properties need to be set for the synthesiser to correctly map to the fabric. One of these properties, *speed grade*, is used to define the switching characteristics of the device. The speed grades for the Virtex-5 XC5VSX50T range from -1 to -3, where -3 provides the fastest switching. The speed grade effects properties such as pin-to-pin delay, maximum clocking frequency of primitives, clock jitter and phase alignment. These properties need to be compared to the operational requirements to ensure that the performance standards are met. It may be important for future timing critical applications but for this project, a speed grade of -1 was sufficient.

CMTs contain clocking related peripherals. In the case of the Virtex-5 XC5VSX50T; there are 6 CMTs, each containing two digital clock managers (DCMs) and one phase-locked loop (PLL). Each CMT has dedicated internal routing to the components within it, but one component within a CMT can be used individually. Typically a DCM-to-PLL or PLL-to-DCM configuration is used. DCMs provide clock management tools such as clock deskew, frequency synthesis and phase shifting, while PLLs provide a wider range synthesisable frequencies and jitter filtering.

3.3.2 FPGA IO

FPGA devices are designed to meet a wide range of design requirements, which requires extensive I/O voltage level control. IO pins on the FPGA can be sourced or routed to a specific range of single-ended, differential or tri-state buffers with different logic standards (specified in the Xilinx Constraints Guide [48]). This project made extensive use of *LVCMOS33* (Low-Voltage CMOS 3.3 V) and *LVDS_25* (Low-Voltage Differential Signal 2.5 V). In a Verilog design, the programmer makes use of a user constraints file (UCF) to provide placement and timing constraints for the synthesis of the project. Although an extensive range of constraints can be set, this project only made use of pin placement, simple timing constraints and logic standard definitions - the rest is left to the synthesizer to optimise.

3.3.3 Basic Verilog Design Elements

The design method used for this project makes extensive use of "always" blocks. In Verilog, and always block depends on a sensitivity list to define when to run its content. The two implementations of an always block are shown, as Verilog code and potentially synthesised logic, in Figure 3.17.

A state-based sensitivity list will synthesise into a combinational logic design; such as a cascade of logic gates and multiplexers, where the outputs are only dependant on the inputs. An edge triggered sensitivity list will synthesise into flip-flop clocked logic, where the outputs only changed based on inputs at clock edge and reset events. Latches are synthesised where outputs are not defined for every condition in an always block - sustaining the output until a corresponding condition triggers a change.

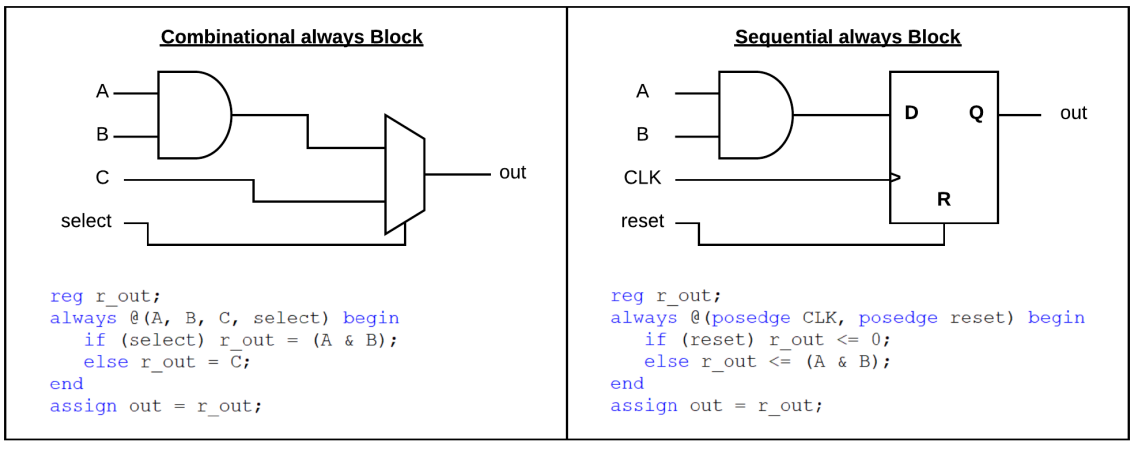

Figure 3.17: Basic *always* blocks used for Verilog design.

3.3.4 Primitive Support

The primitives available for the specific FPGA device used for TIGER-3 are discussed here. The primary primitive that distinguishes the model used in TIGER-3 from the other models in the Virtex-5 family is the XtremeDSP core which makes use of the DSP48E slices. A DSP48E slice supports multiple independent functions, such as multiplication, three-input addition, barrel shifting, wide-bus multiplexing, etc. Furthermore, multiple cascaded DSP48E slices can be used for applications such as digital filtering and other complex arithmetic common to digital signal processing - without extensive use of FPGA fabric space. The XC5VSX50T model contains a total of 288 DSP48E slices. For more details see the manual; *Virtex-5 FPGA XtremeDSP Design Considerations* [49].

Another primitive integral to this project, is the Embedded Tri-Mode Ethernet MAC. The Virtex-5 device family contains two physical Ethernet MACs. The MACs support a host of features, including 10/100/1000 Mb/s full duplex operation. It adheres to the IEEE Std 802.3-2002 specifications [43]. More information can be found in the *Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC User Guide* [50].

Other primitives, such as the SoC, was not used and is therefore not detailed here.

3.4 Chapter Summary

The model design and other design considerations were detailed in this section. First, the radar model was detailed in terms of digital sampling and analytic signal representation. Integral concepts such as the Nyquist criterion and Hilbert transforms are detailed along with illustrations of frequency shifting and down-sampling. The mathematical radar receiver model is then given for I-Q demodulation, followed by the mathematical definition of the auto-correlation function for signal identification.

The radar model was further extended to moving targets, with the derivation for the Doppler frequency given. Doppler processing was further elaborated on in terms of resolution and ambiguity. Integral to Doppler processing, is digital filtering. The implementation of a CIC filter was detailed due to its efficient digital implementation.

The final DSP model discussed in the section, was the pulse sequence used by the SuperDARN. The reason for its use was shown mathematically and visually. It was shown how and why the ACF function is used to extract both Doppler and range information from a received sequence.

The second subsection concerns the communication model used for this project. The interface between the FPGA client, the MAC primitive and the PHY was detailed. The protocols used, includes the Ethernet layer protocol, IPv4 and UDP. Each layer was detailed in terms of header fields and application context was given for relevant fields. Fi-

nally, the data packet structure was given, and the size and speed limitations were defined.

This section concludes with other design considerations concerning the reconfiguration of the FPGA. The clocking-related peripherals were listed along with other design blocks integral to the final implementation.

4 Design and Implementation

This chapter concerns the design and implementation of the listening mode on the TIGER-3 by making use of the models, architectures and signal processing theory detailed in Chapter 3. The sections within the chapter will detail the implementation of signal processing interfaces, followed the Ethernet interfaces. It concludes with details concerning the implementation of the software interface on the receiving end of the Ethernet communication link. See Section 1.7 for the technical requirements for the design. Below, a simplified overview is given to provide further context to the general design process.

4.1 System Design Overview

The purpose of using an FPGA-based platform is to create a processing interface between peripherals. The design task is to collect data from the main ADC peripheral, process the data and transfer it to the Ethernet peripheral. The data should then be received on a personal computer (PC). A block diagram, illustrating the primary data paths through the most relevant peripherals is shown in Figure 4.1. The primary physical interfaces are: The received signal of interest (top-left), the transmission of information to the PC

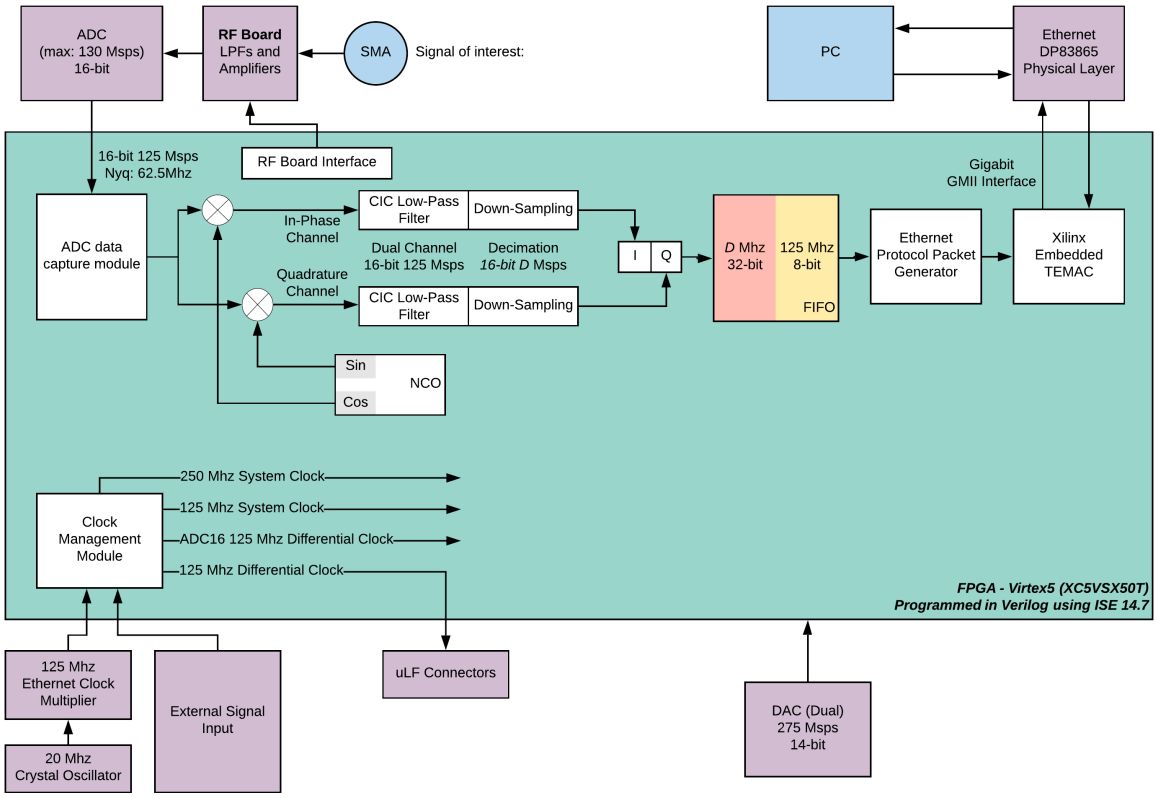

Figure 4.1: TIGER-3 full implementation diagram with related peripherals.

(top-right), and the management of the input clocking signal (bottom-left). The DAC, used for signal transmission is not part of the scope of this project but is included as it is an integral part of the complete transceiver. Further details on the individual process blocks are provided in its respective sub-sections below.

FPGA software design typically involves building interfaces between peripherals and IP-cores, where modules within the design provide inputs and outputs as a functional block. A cascaded modular design approach was taken where top level modules depend on sub-modules and/or IP-cores. The module structure is shown in Figure 4.2. Each top-level

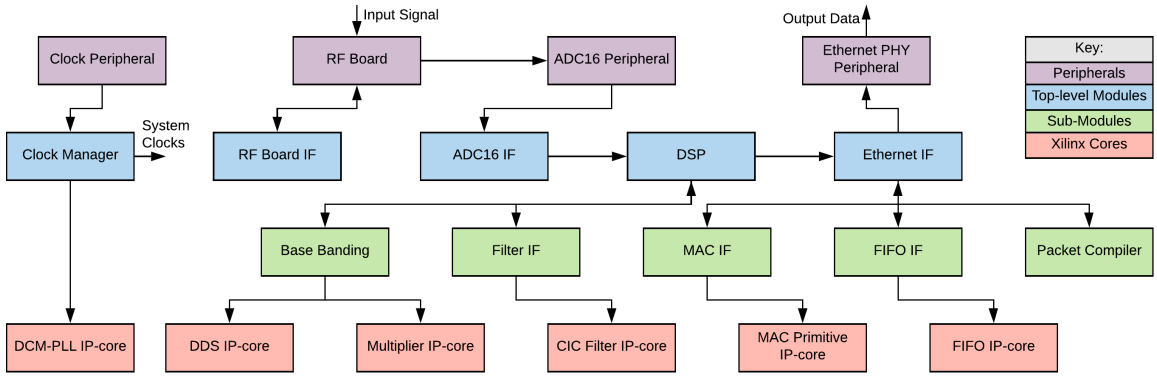

Figure 4.2: Programming design overview.

module provides a data product shared with the other top-level modules. The same applies to sub-modules within each top-level module.

Figure 4.2 clarifies which of the design components existed before this project, and which was designed and implemented for this project. All peripheral components existed as part of the hardware which remained unchanged throughout the project. Top and Sub-level modules were fully designed and implemented for this project. Xilinx cores are existing modules that required configuration and implementation.

The top-level (and subsequent lower level) modules' operation can be adjusted in two ways: Parameters can only be changed before synthesis and are therefore set during operation. Examples include certain Ethernet header content such as IP addresses. Alternatively, module inputs can change module operation during run-time. Module inputs include properties such as synthesis frequency control.

Appendix C and B contains the TIGER-3 transceiver and RF interface board circuit schematics. The schematics were designed by La Trobe University and made available for this project by SANSA. Appendix F contains FPGA HDL design schematics for the designed modules detailed in this chapter. Due to the design complexity of the lower level modules; only modules schematics that are parent modules to sub-modules are shown.

4.2 Clock Management

The external clock is sourced from a 20 MHz crystal oscillator. The 20 MHz clock is passed through a frequency synthesiser and a fanout buffer to produce a 125 MHz LVDS signal to a set of the FPGA input pins. A clock management module was designed and implemented to provide clocking to peripherals modules. A block diagram of the module is shown in Figure 4.3.

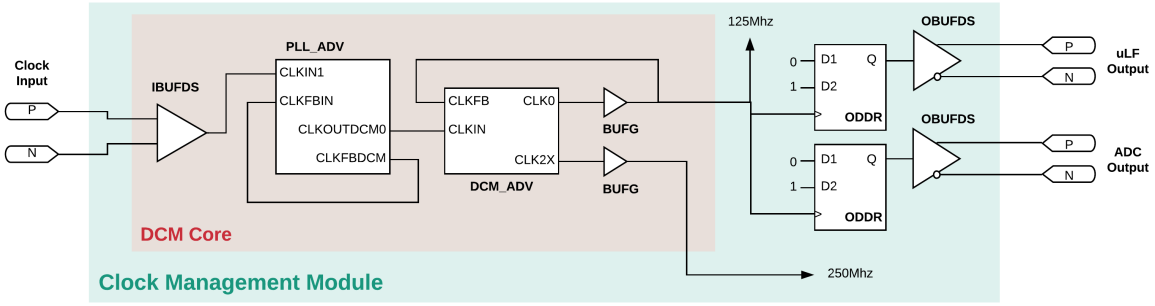

Figure 4.3: Clocking interface design.

The DCM module makes use of the *Xilinx DCM-PLL Advanced* core to generate two clocking signals for the rest of the design. The signals are then passed through BUFG buffers to place the clocks on the FPGA fabric's global clocking network. It also makes use of ODDRs for clock forwarding, and output differential buffers (OBUFDS) to provide stable output clocking to peripherals. Note that the Ethernet interface also makes use of ODDRs, but is instantiated in the Ethernet interface modules - discussed in Section 4.5.2.

4.3 Input Signal Interface

The transceiver makes use of an "RF board" to process the incoming RF signal before passing it to the ADC peripheral on the TIGER-3 through a ribbon cable. The receive side of the RF board is functionally illustrated in Figure 4.4. Furthermore, the signal is passed through additional peripherals on the TIGER-3 before connecting to the ADC. The output of the ADC directly connects to the FPGA pins. The signal path from the SMA input to the FPGA pins is detailed below.

4.3.1 RF Board Receiver

The complete RF board schematic and a block diagram of the board are shown in Appendix G. An input signal is passed from a SMA connector trough a series of switches, amplifiers, attenuators and filters on the RF board, to the TIGER-3 board. For the demonstrator used in this work; the RLC low-pass filter is bypassed. A block diagram illustrating the receiver chain of the RF board (without the RLC filter) is given in Figure

4.4. The first component after the SMA connector, is a single pole double throw voltage

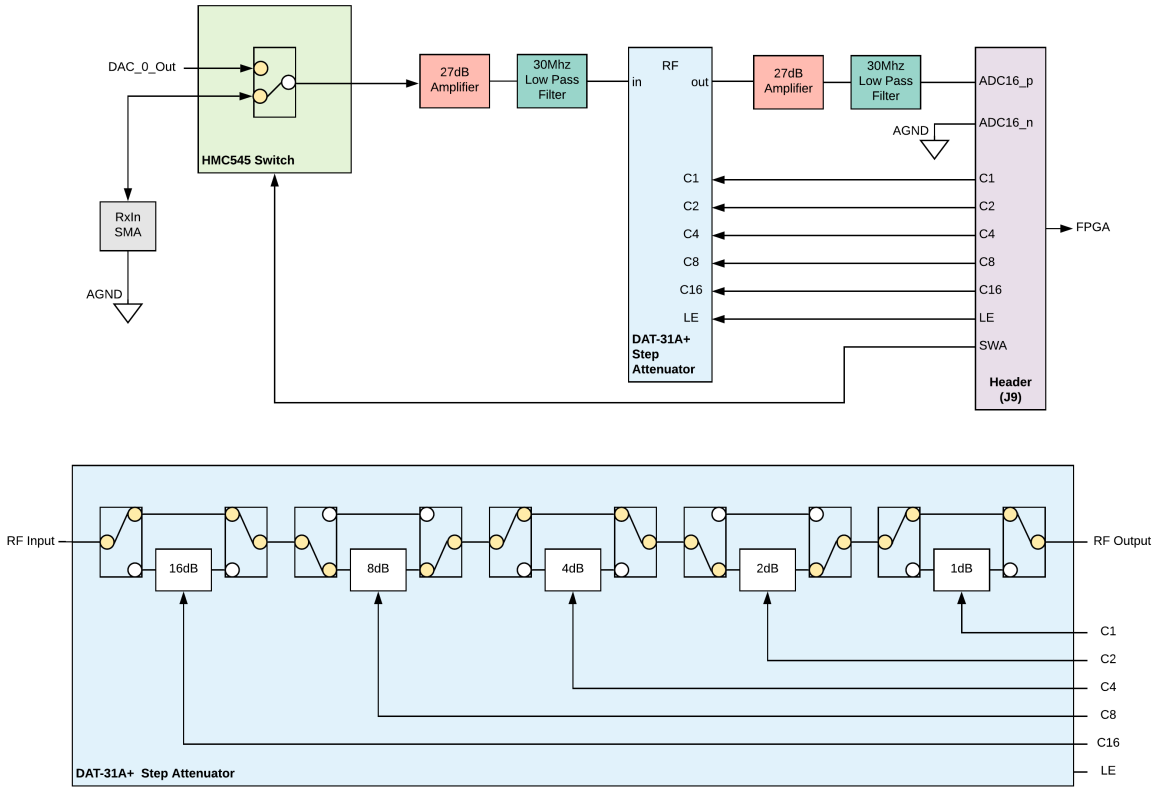

Figure 4.4: RF board peripherals, detailing the digital step attenuator.

controlled switch, which can operate from DC to 3 GHz with a typical insertion loss of 0.25 dB. Based on the bias of *SWA* from the FPGA; the switch allows the receiver chain to be connected to the input SMA port, or to the main DAC output (for receiver chain calibration). *SWA* is **set** to connect the receiver chain to the DAC, or reset to connect it to the input SMA port. A separate output is typically used for transmission and Tx/Rx switching occurs external to the RF board. Further analysis of the DAC-to-receiver chain does not fall within the scope of this project.

The RF switch is followed by an amplifier-filter stage before and after a step attenuator. The attenuator allows the FPGA to control the input signal amplitude, in steps, to ensure that the maximum dynamic range of the ADC is utilised. In the illustration, the attenuator control pins C2 and C8 are set (totalling -10 dB) to provide a final gain of approximately 44 dB. The filters ensure that no high frequency components (above 30 MHz), generated by any of the stages are passed to the ADC. Considering the possible attenuation extremes; the achievable gain ranges from 23 dB to 54 dB (not including the effects of the ADC peripherals and filters).

An RF interface module was written, which routes signals to the correct outputs. Once a change in attenuation is triggered, the module ensures that the latch line to the RF board is kept high for 16 ns (two 125 MHz clock cycles) to meet the IC requirements.

4.3.2 TIGER-3 Receiver

Once the signal is hardware amplified and filtered, it is passed to the TIGER-3 ADC circuitry. The main ADC (used for RF sampling) is a 16-bit 130 Msps ADC. It supports LVDS and CMOS logic standards, and 1's or 2's compliment parallel digital outputs. The ADC modes of operation are elaborated on in the ADC's data sheet [51]. Figure 4.5 shows the three ADC pins associated with the input signal (VCM, AIN+ and AIN-) being driven by an RF transformer coupled pair (T5 and T6). T5 has a centre-tapped

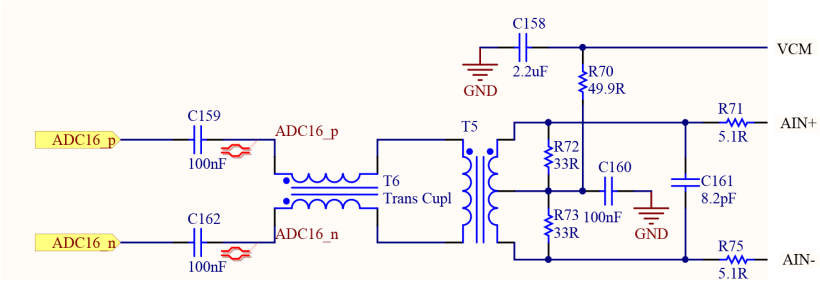

Figure 4.5: ADC input peripheral schematic.

secondary, DC biased by VCM, to set the ADC input DC bias. The disadvantage of using a transformer is the attenuation of low frequencies (with most small RF transformers having poor performance at frequencies below 1 MHz) and centre-tapped transformers often result in large 2nd order harmonics [51].

Although possible to use in single ended CMOS output mode, the ADC hardware is connected on the TIGER-3 to support differential outputs to the FPGA. Important interfacing I/O pins, associated with the RF board and TIGER-3 board, is shown in Table 4.1.

Table 4.1: ADC and related RF board signals.

	Pin Names	**Functional Names**
RF Board	C1, C2, C4, C8, C16	Attenuation Control
	LE	Att. Control Latch Enable
T-3 ADC16	ENC_p, ENC_n	Differential Encode Inputs
	AIN+, AIN-	Differential Analogue Inputs
	LVDS, MODE, RAND, DITH	Mode Select Input Pins
	SHDN	Shutdown Power Input
	CLKOUT_p, CLKOUT_n	Differentail Output Clock
	DA0-DA15 (p and n)	Channel A Differential Data Output

The *ADC16* interface module was designed and implemented for the FPGA to control the input from the main ADC. The module timing diagram is illustrated in Figure 4.6. The

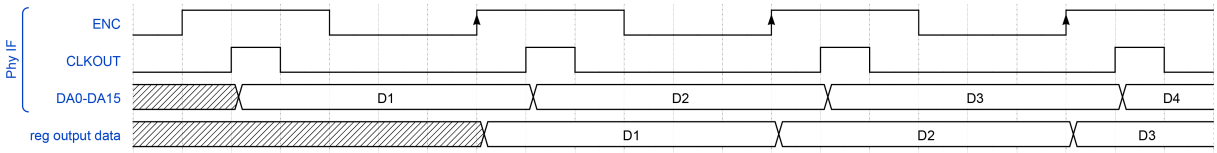

Figure 4.6: ADC interface module timing diagram.

"Differential Encode Inputs" from Table 4.1, indicates the start of the encoding process for the ADC peripheral pins and are driven by the clock lines labelled "ADC Outputs" in Figure 4.3. The "Differential Analogue Input" pins are the analogue inputs to the ADC after being passed through the RF stages discussed above. The "Mode Select Pins" for the ADC are set as pre-synthesis parameters, along with "Shutdown Power".

The clock and data outputs of the ADC are converted into single-ended signals, using the FPGA's IBUFDS primitives, and passed to the ADC16 interface module within the FPGA. In Figure 4.6, The inputs signals are given under "Phy IF", and the output data are registered in the module at the positive edge of the following ENC cycle. This concludes the description of the analogue RF receive module.

4.4 DSP Interface

The next stage in the receiver pipeline is to process the discrete ADC signal to a suitable output format. Figure 4.1 illustrates this process between the ADC capture module and the FIFO. The output is a down-sampled signal with a rate compatible with the Ethernet interface. The data rate should be limited to well below 125MB/s (1Gb/s) for effective transmission. Figure 3.3 illustrates the spectrum for each stage of the DSP chain, discussed below.

4.4.1 NCOs

A numerically controlled oscillator (NCO), also known as a direct digital synthesiser (DDS), is a digital frequency synthesising tool that makes use of a look up tables (LUTs) to generate and store phase samples. The *Xilinx DDS Compiler v4* core [52] was used to generate a NCO that consists of two parts; a *Phase Generator* and a *Sin/Cos Lookup Table*. The simplest form of an NCO makes use of phase truncation, and is shown as a functional block diagram in Figure 4.7.

The phase increment ($\Delta\theta$) is the primary variable input and the clock frequency (f_{clk}) is typically a constant input. The core is synthesised with two primary parameters: the number of bits in the phase accumulator (B_θ), and the number of bits in the LUT (B_Θ).

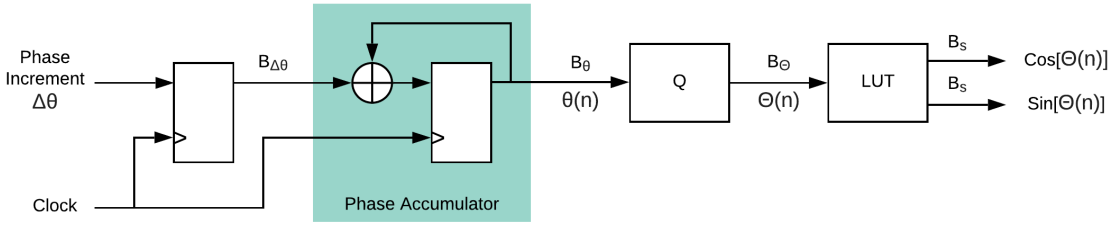

Figure 4.7: Block diagram detailing the internal components of an NCO.

The phase accumulator integrates by $\Delta\theta$ synchronous to the clock. Therefore, an increase in $\Delta\theta$ results in an increase in the rate of accumulation in the phase accumulator.

The accumulated phase $\theta(n)$ is a function of N accumulations. It is passed through a quantiser Q, which simply truncates the bit-string to a lower precision value $\Theta(n)$ with a bit-width of B_Θ. $\Theta(n)$ is used to index the LUT for a conversion from phase-space to time at a bit-width of B_s (which may be decreased as a pre-synthesis parameter but typically $B_s = B_\Theta$).

The LUT will contain the maximum number of unique phase values that can be represented by B_Θ for a single cycle, given by

$$N = 2^{B_\Theta} \tag{4.1}$$

and therefore $\Theta(n)$ is given by

$$\Theta(n) = n\frac{2\pi}{N} \tag{4.2}$$

where n is a sampled time series index. The frequency produced by the NCO is given by

$$f_{out} = \frac{\Delta\theta\ f_{clk}}{2^{B_\theta}} \tag{4.3}$$

and the output frequency resolution is given by

$$\Delta f = \frac{f_{clk}}{2^{B_\theta}} \tag{4.4}$$

Phase and amplitude quantisation errors are observed at the output of the NCO which affect the spectral purity of the output waveform. The errors are minimised by decreasing the quatisation at the cost of an increase in resource usage. The effects of quotation are observed as frequency harmonics, the magnitude of which is primarily dependent on the number of quatisation levels B_Θ (as per Equation 4.1).

Alternatively, the effects of quatisation can be further minimised by enabling additional functionality to the DDS module in the form of *Phase Dithering*, *Taylor Series Correction* [52].

The NCO parameters can be summarised as follows: B_θ influences the frequency precision of the NCO (see equation 4.4). The parameter is limited by a small amount of resource usage in the FPGA - equivalent to a binary counter of equal bit-width. B_Θ influences the occurrence of spurs in the output spectrum. Increasing this parameter will decrease the spurs at the cost of large storage resource usage compared to B_θ.

4.4.2 DSP Module Design

The design and implementation of the DSP interface (see Figure 4.2) is detailed here. It consists of an NCO-multiplier and a CIC decimation filter module discussed in Section 3.1.9. This fully implements the down-conversion of any ADC sampled signal by an adjustable demodulating frequency, and applies a low-pass filter with an adjustable pass band, with the latter directly determining to the output sampling rate.

The *Base-Banding* sub-module makes use of an NCO and multiplier IP core to demodulate a signal to baseband. The multiplier core was configured to receive a signed 16 bit (ADC) signal and a signed 9 bit (NCO) signal. The output is truncated to 24 bits. The timing diagram for the module is shown in Figure 4.8.

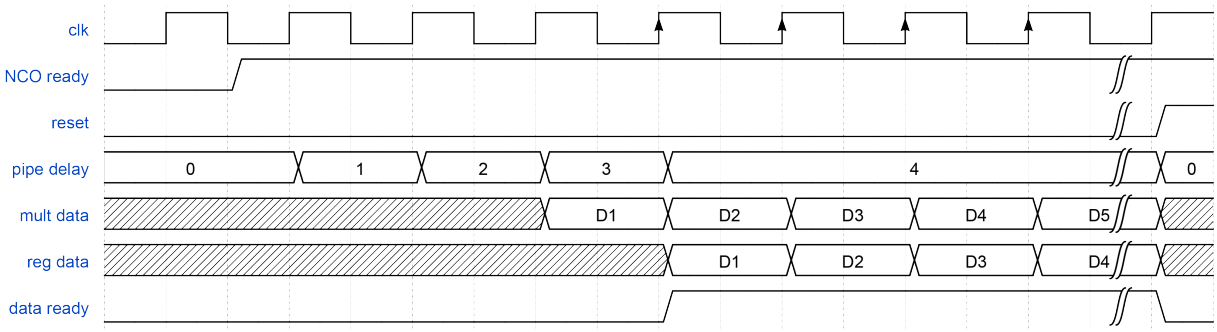

Figure 4.8: NCO interface module timing diagram.

The *mult data* output is the multiplier output. A set latency is introduced by the NCO and multiplier cores. The NCO core asserts *NCO ready* when the output is valid, and a counter with output *pipe delay* is used to indicate the multiplier's registered output validity according to the latency defined by the multiplier core-generator documentation [53].

The *Filter IF* sub-module makes use of the CIC filter core, which can be set to dual channel operation to avoid the additional resource space that would result from using a separate instantiation for each channel.

The timing diagram for the complete DSP module is given in Figure 4.9. The signals

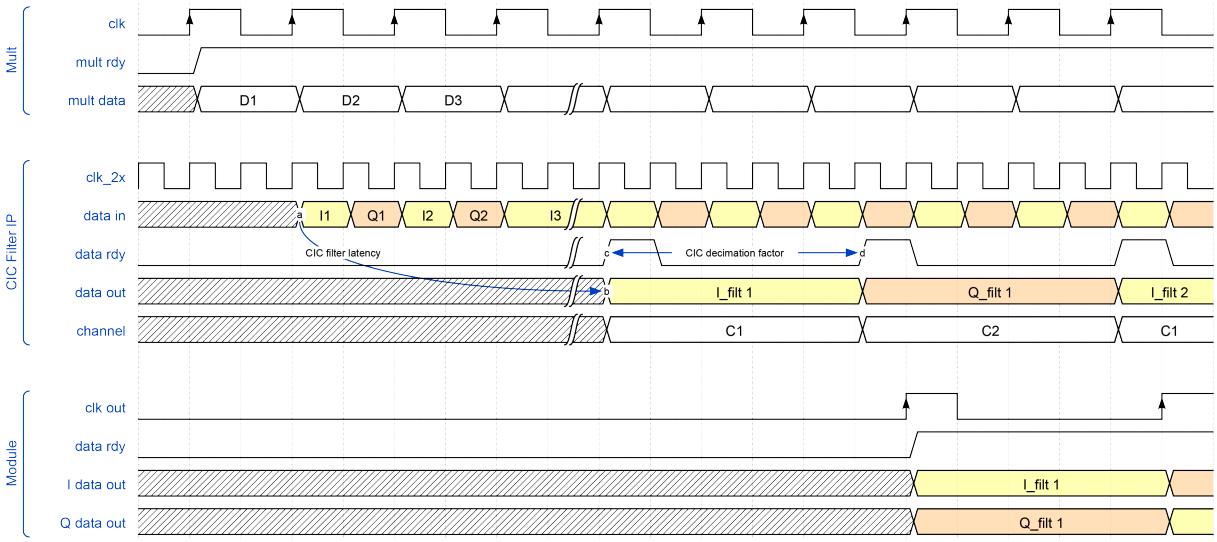

Figure 4.9: Combined multiplier and CIC filter interface module timing diagram.

in the timing diagram are discussed here. For the CIC core to be able to process N channels at an input rate of 125 MHz; it has to be driven by a clock with a frequency of $N \times 125$ MHz, to process each channel input serially. Two multipliers are used to process the I and Q channels independently at the CIC input. The I and Q channel outputs are illustrated as one data bus in Figure 4.9, given by *D1*, *D2*, *D3*, etc.

Synchronous to the 250 MHz clock (*clk_2x*); *D1* is divided into *I1* and *Q1* and presented serially to the input lines of the filter. Care was taken to ensure the first data set presented to the CIC core is the in-phase component, irrespective of when the core is enabled.

The output is similarly produced on a single channel after a latency, and *CIC - data rdy* is set after the each output channel is registered. The CIC core supports up to 16 channels and indicates the channel being output on *CIC - channel*.

By making use of the *mult rdy* bit (set by the multiplier sub-module) and the output indicator bits from the CIC core; the latency introduced by both sub-modules can be accounted for. The *mult rdy* bit is used to control the reset of the CIC core, ensuring that the CIC core does not process the data before the multiplication data are valid. The above design considerations ensure that *clk out* and *Module - data rdy* indicates the output availability of valid filtered data.

To aid in the debugging process; a design decision was made for the module to separate the output channels into parallel output lines *Module - I data out* and *Module - Q data out*.

4.5 Ethernet Interface

Interfacing with the Ethernet peripherals imposes design considerations regarding data rates and format. The interface operates a standard data rate and width depending on the Ethernet speed and physical interface. Gigabit Ethernet makes use of a 125 MHz 8-bit data rate structure, which was used for this project. This means that an absolute maximum rate rate of 1Gb/s is achievable - although this is only a theoretical maximum (discussed in Section 3.2).

4.5.1 FIFOs and Clock Domain Separation

The rate at which the data are presented to the Ethernet interface is dependant on the CIC filter characteristics from Section 4.4.2, which is variable at operation-time (filter bandwidth is a user parameter). This introduces a divide between a certain and uncertain clock domain. Furthermore, the data bit-widths of the two domains do not correspond. This is common in FPGA design and first-in-first-out (FIFO) cores are commonly used to manage situations such as these. They are input-output-clocked storage elements that allow for data to pass through in order, such that the first data set that enters the FIFO is the first to leave. In special cases this can be done regardless of input/output data width and clocking rates.

The FIFO IP core generator allows for a variety of different FIFO implementations. The implementations and the relevant features of each is shown in Table 4.2. Independent clock domains and data widths are required, as mentioned above, and therefore only the *Block RAM* implementation meets all of the requirements.

Table 4.2: Compatible FIFO implementations and features.

Rd/Wr Clock Domains	**Memory Type**	**Supported Features**
Common Clock	Block RAM	FWFT
	Distributed RAM	FWFT
	Shift Register	
	Built-in FIFO	FWFT, BFIFO
Independent Clocks	Block RAM	FWFT, BFIFO, AD
	Distributed RAM	FWFT
	Built-in FIFO	FWFT, BFIFO
FWFT: First-Word Fall-Through BFIFO: Makes use of built-in FIFO primitives AD: Asymmetric input-output data widths		

The FIFO core provides a host of settings and features, which are not all discussed here. It allows the user to pre-define a data width and depth. During operation-time, the core can provide information on the amount of stored data from an input and output size perspective. *First-word fall-through* allows for the input data to be presented to the output immediately when clocked, as apposed to being presented after the first output

clock edge. First-word fall-through was implemented in this design to decrease latency. The FIFO clock diagram is shown as part of Figure 4.14.

4.5.2 EMAC Primitive and Physical Layer

The Virtex-5 contains a pair of Ethernet Media Access Controllers (EMACs) which can be configured individually to 10/100/1000 Mb/s full duplex, or 10/100 Mb/s half duplex operation. Unlike its predecessor (Virtex-4) the Virtex-5 houses stand alone MAC blocks, greatly improving on its functionality. For more supported features and information, see the embedded TEMAC user guide [50].

The EMAC communicates to the PHY through a media-independent interface (MII), variations of which include; gigabit MII (GMII), reduced GMII (RGMII) and serial GMII (SGMII). The TIGER-3 is connected such that it supports GMII - used for this work. More information on the MAC and higher Ethernet layers are provided in Section 3.2. The interface signals from the MAC are listed in Table 4.3. Further information on the standard can be found in IEEE 802.3-2000 [43].

Table 4.3: GMII interface related signals.

Signal Name	**Description**	**Direction**
Transmitter signals		
GTXCLK	Clock signal for 1 Gb/s TX signals (125 MHz)	MAC to PHY
TXCLK	Clock signal for 10/100 Mb/s signals	MAC to PHY
TXD[7..0]	Data to be transmitted	MAC to PHY
TXEN	Transmitter enable	MAC to PHY
TXER	Transmitter error (used to corrupt a packet)	MAC to PHY
Receiver Signals		
RXCLK	Received clock signal	PHY to MAC
RXD[7..0]	Received data	PHY to MAC
RXDV	Signifies data received are valid	PHY to MAC
RXER	Signifies data received have errors	PHY to MAC
COL	Collision detect (half-duplex connections only)	PHY to MAC
CRS	Carrier sense (half-duplex connections only)	PHY to MAC
Management Signals		
MDC	Management interface clock	MAC to PHY
MDIO	Management interface I/O	Bidirectional

The EMAC's primary task is to interface between the client and the PHY (Figure 3.11). It also provides further functionality through other interfaces, shown in Figure 4.10. The host interface provides access to the control registers of the EMAC (set as pre-synthesis parameters for this project) and the optional management data IO (MDIO) interface provides access the management registers of the PHY (not used for this project). Finally, a statistics interface also provides information on the running of the EMAC. The complete EMAC signal interface is shown in Appendix E.

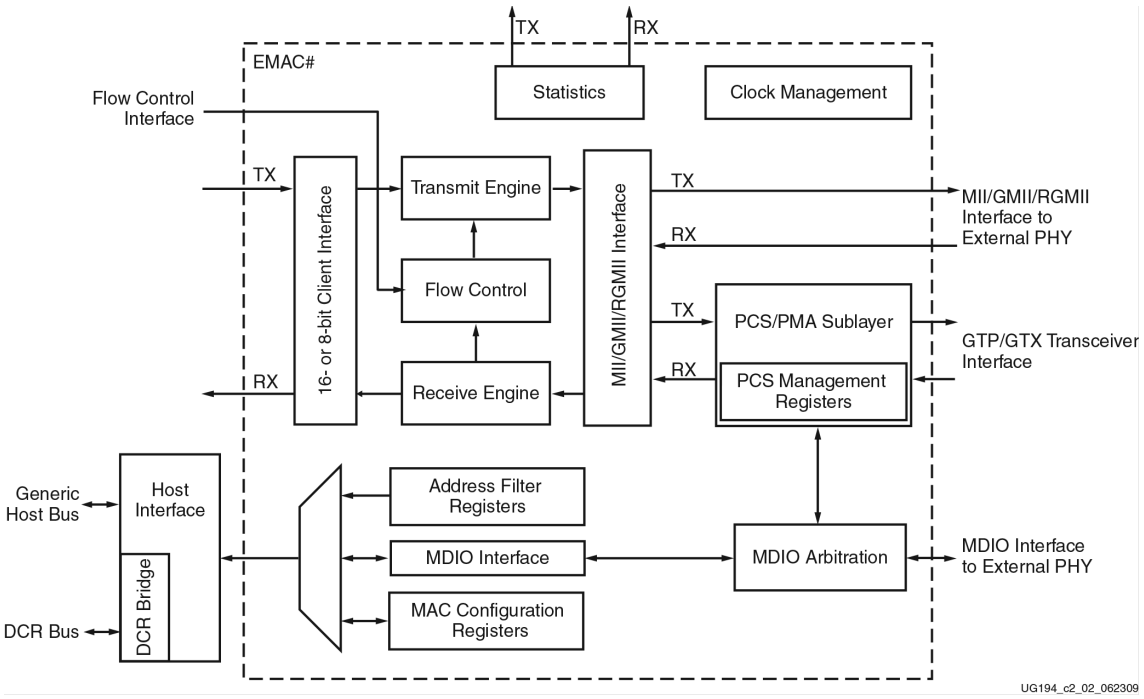

Figure 4.10: MAC interface block diagram [50].

It is important that the MAC is implemented correctly to allow for the signals to interface through the IO buffers, to the PHY synchronously. Multiple clocking schemes are available for different modes of operation. The standard clocking scheme used for gigabit operation is shown in Figure 4.11.

The *GTX_CLK* signal is sourced from the clock management module and is therefore already in the global clock net. It is used to clock the *TX client logic* block and the output of the *GMII_TX* interface. *GMII_RX_CLK* is used to clock the receive side logic and incoming data.

An ODDR is used for clock forwarding of the *GMII_TX_CLK* signal. In this case the ODDR inputs are inverted to invert the output clocking signal. The delay introduced by the ODDR is similar to that of a typical register instantiating, and therefore all other GMII lines are registered to compensate for the output delay of the ODDR clock. This configuration ensures that the positive clock edge occurs at the centre of the *GMII_TXD* data output. No further attention was given to the receiver side as it was not within the scope of this work.

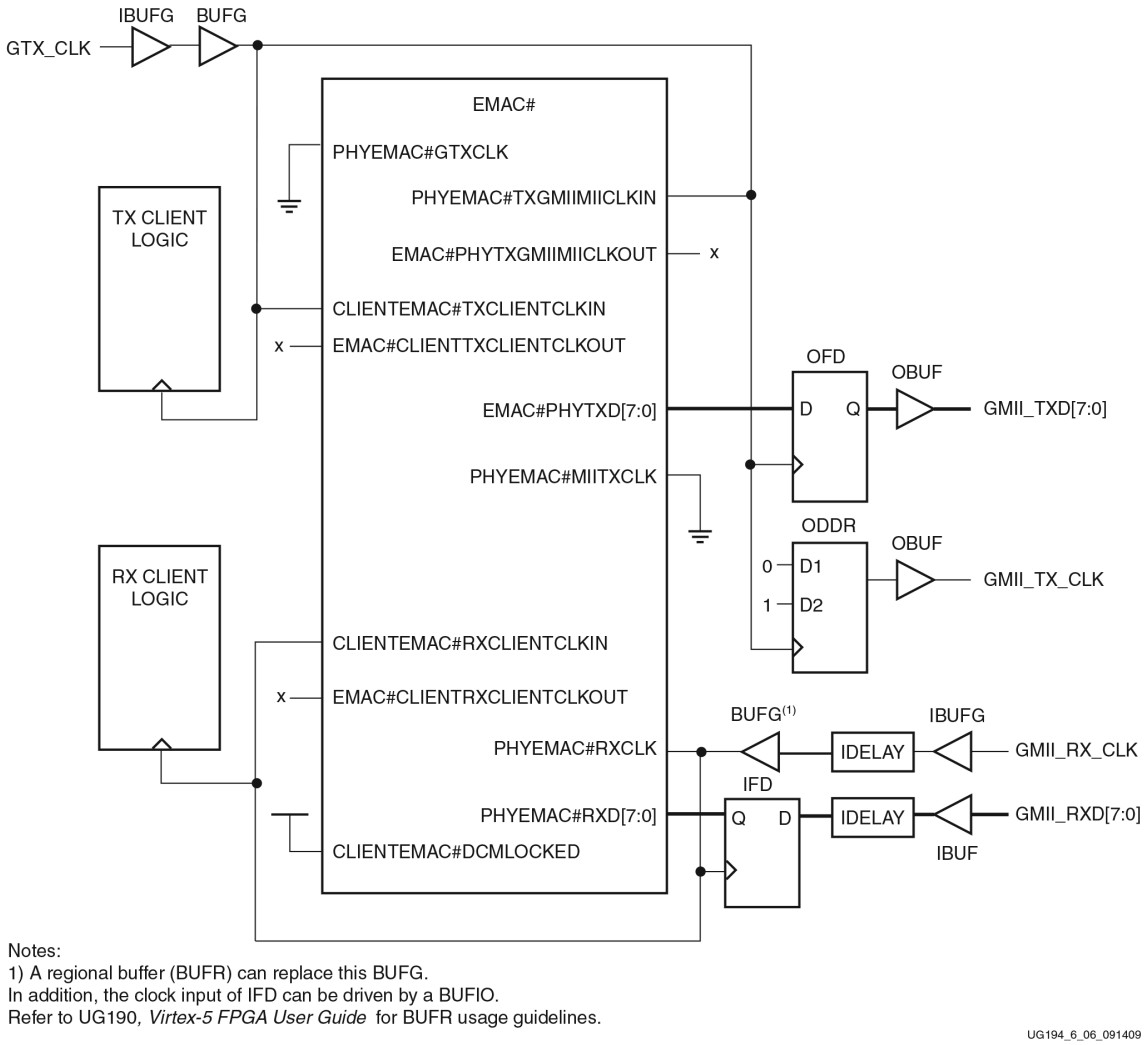

Figure 4.11: EMAC implementation interface for 1 Gb/s operation.

4.5.3 Packet Generation

The tasks performed by the MAC were discussed in Section 3.2, where it was mentioned that the MAC generates the Ethernet frames. However, the majority of the header information needs to be prepared in the *Client Logic Interface* from Figure 4.11. This includes all header information for UDP, IPv4 and layer 2 of the Ethernet frame (see Figure 3.13).

The designed method makes use of a FIFO to collect input data to the point where a certain data storage threshold is met. At this point in time, the MAC is initiated and the 48-bit packet information header (Section 3.2.4) is registered. Once the MAC is ready to receiver data; the packet generator provides 8-bit header and packet information data to the MAC. Once all the header and packet information data have been provided to the MAC; the data from the FIFO is passed to the MAC until a set amount has been

transferred. The process repeats once the FIFO's input threshold is met again. The FIFO ensures that input data continue to be captured while header information is prepared and transferred by the client. The process is further discussed in Section 4.5.4.

The workings of the packet generator are complicated by changes that need to be made to certain header fields before transmission. For example; the packet size may change based on the user's needs, in which case the length parameters in the headers need to change. This in turn also changes the header checksum values. These changes need to be registered before the data can be processed.

A condensed description of the IPv4 checksum field is given below, with the aid of a block diagram in Figure 4.12. The checksum is calculated by summing every 16-bit field

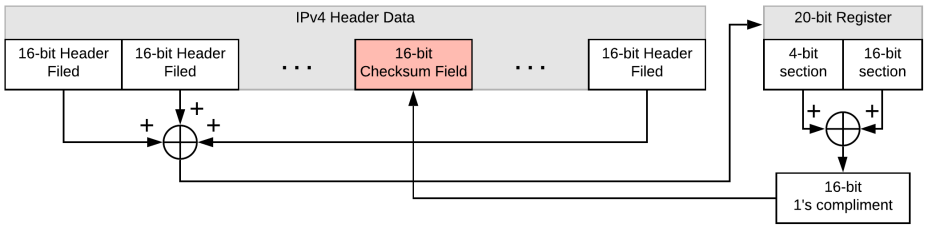

Figure 4.12: Diagram detailing checksum calculation.

in the IPv4 header except for the checksum field itself. There are nine 16-bit fields in the header, resulting in a maximum summed length of 20 bits. As illustrated, the 20-bit summed output is summed again as a 4-bit and 16-bit part. The result is at most 17 bits wide. In the unlikely event that the result is 17 bits wide, the summation process is performed again until the result is 16 bits or less. The 1's compliment of the 16-bit result is assigned to the IPv4 checksum field [44].

It is therefore the task of the sub-module to produce a 336-bit register containing all the header information for a packet, together with the 48-bit packet information, before transmission. Once the sub-module is enabled, it provides the data to the MAC in order. After providing all the data, the sub-module must once again go into a reset state while the FIFO collects data for the next packet. The module has an extensive list of pre-synthesis parameters that need to be set to define the static header details according to the protocols discussed in Section 3.2. These parameters can change at run-time with a few changes to the module.

4.5.4 Interface Design

The Ethernet module brings all that was discussed in the above section together. The module ensured that the correct block is providing information to the MAC at the correct

time and that the remaining data-providing blocks remain disabled. This is a time-sensitive operation that makes use of a counter as a pointer for the module. A functional block diagram, illustrating the operation of the module, is shown in Figure 4.13. The timing diagram of the module is given in Figure 4.14. With the aid of the aforementioned figures, a detailed description of the module operation follows.

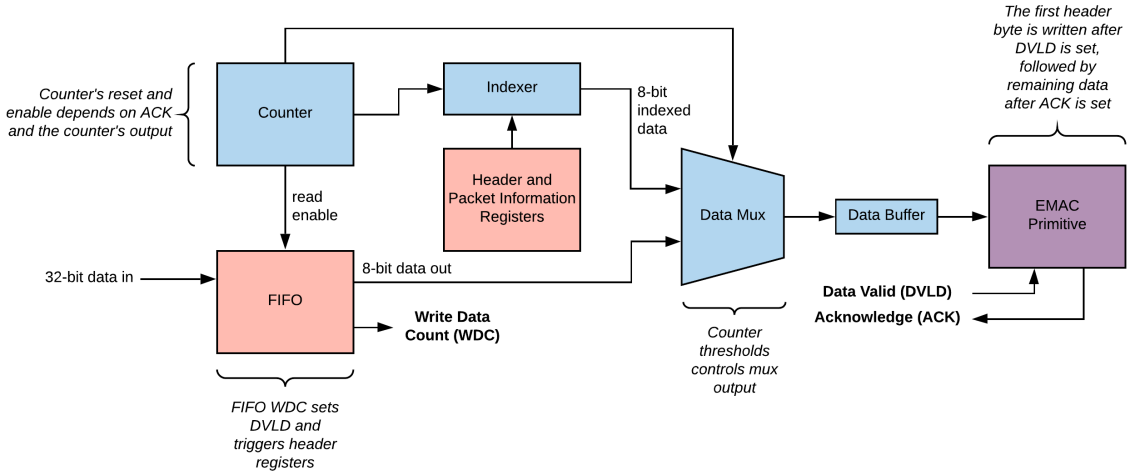

Figure 4.13: Block diagram illustrating the architecture of the Ethernet interface module, with respect to data flow.

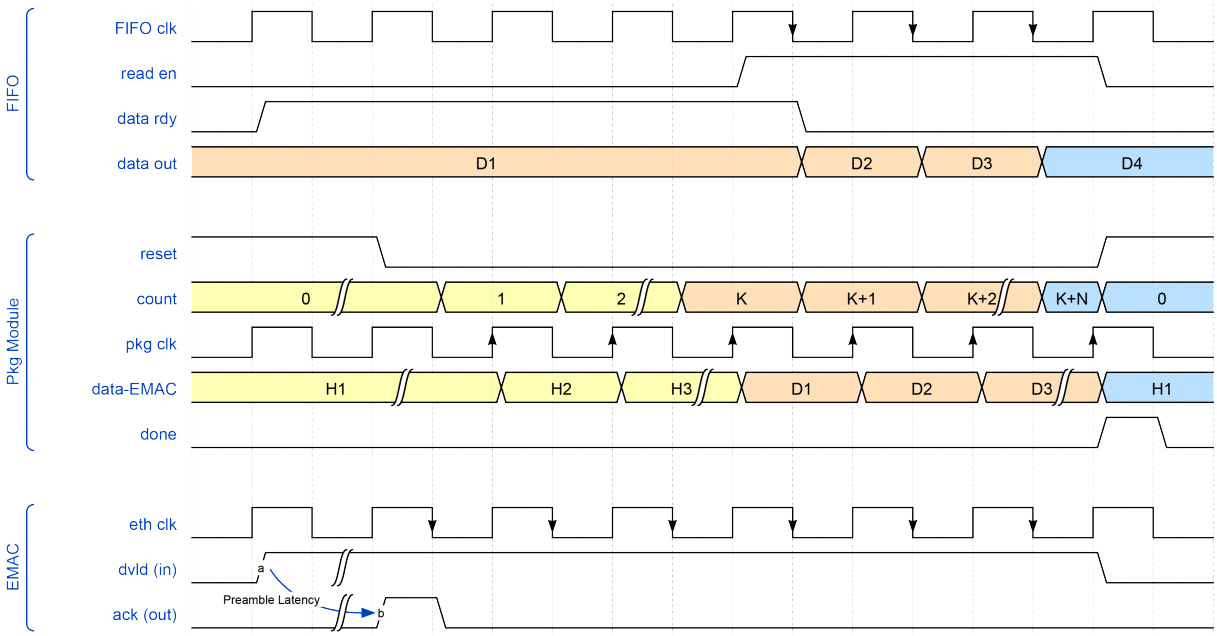

Figure 4.14: Combined Ethernet interface module timing diagram.

The Client-MAC interface relies on two signals; DVLD (Client to MAC) and ACK (MAC

to Client). After DVLD is set by the client, the first byte is written to the MAC on the next positive edge of the shared 125 MHz clock. The following edges are ignored until ACK is set by the MAC to indicate that the preamble and first byte has been transmitted. The remaining bytes must be made available at the following positive clock edges.

The module is designed to always have the first byte (*data-EMAC - H1*) at the output before transmission begins. Once the FIFO's data count is triggered (*FIFO - data rdy*); DLVD is set while the counter and all other operation remains reset. At the positive edge of ACK, the counter is enabled and synchronous operation begins. The synchronous operation is illustrated in Figure 4.15.

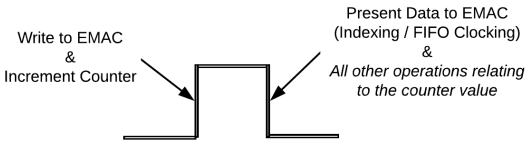

Figure 4.15: Ethernet interface clocking scheme.

It is important that the incrementing of the counter and the operations based on the counter are separated by at least half a clock cycle. At first the counter values index the header register (*data-prim - H*) to provide 8-bit data to the MUX at positive clock edges. When the counter value indicates the end of the header data, the "packet information register" is indexed in a similar way (not shown in the timing diagram).

When the counter indicates the completion of all header and packet information (*count - K*); it triggers a change in the MUX path, the FIFO's read functionality is enabled (*FIFO - read en*), and the remaining data are transmitted (*data-prim - D*). Note that *FIFO - data rdy* will reset once data are transferred out of it.

Once the counter indicates the end of the packet; a signal (*Pkg Module - done*) resets DVLD and all other functionality (including clearing the counter and placing the first byte at the output). The module remains in a reset state until the FIFO indicates that the next packet may be sent, through *FIFO - data rdy.*

4.6 PC Receive Interface

Multiple software packages are used for debugging and processing the received data. The primary debugging tool was Wireshark, used to directly obtain data from the computer's MAC. Python 3.7 was used to capture the data packets and store the information in a usable format. Finally MATLAB 2019 was used to process the stored data for the required visual representation.

Wireshark proved integral to the capturing of data. A PC typically has a network interface controller (NIC) that contains the circuitry required for a computer to communicate over a physical layer such as Ethernet. The NIC performs screening, queuing and sorting of the received packets. If a packet is malformed or contains errors, the NIC will discard the packet.

Wireshark enables promiscuous mode on the NIC which allows for all packet information on a specific interface to be captured regardless of format. This feature is key in debugging the Ethernet interface of the FPGA. Wireshark provides the majority of the protocol layer data along with the payload data.

A python script was written for the capturing of data, making use of the *socket* library for low-level network interfacing. The socket is set up in UDP mode to capture data from a specific physical MAC address (belonging to the FPGA). A buffer size is defined, and the buffer is filled with unformatted binary data and the source address. The source address is discarded and the data are unpacked in the correct format by making use of the *struct* library. Finally, the I, Q and packet information data are separated and stored as arrays in a *CSV* file. The formatting is such that three rows are generated in the CSV file for every received packet. The first containing 6 numbers, formatted as 8-bit unsigned values - corresponding to the packet information data. The second and third row contains 349 samples each formatted as signed 16-bit data for I and Q respectively.

A MATLAB function was written to read the CSV file and combine the multi-row formatted data into a single complex array. MATLAB makes use of the packet information data to notify the user if a packet was missing in the CSV file. The time data are then processed according to the user's requirements.

4.7 Chapter Summary

This chapter has provided the reader with a simplified overview of the listening-mode design, including the primary components synthesised within the FPGA. The basic task is to capture data from the ADC sampling at 125 MHz, and perform base-band demodulation, filtering and the down-sampling of the signal. The signal is then buffered in a FIFO until such a time that enough information is stored for a packet to be transferred to the computer. In that case, the packet is formatted according to protocol. The information is passed to the Ethernet MAC and in turn the MAC output is communicated to the Ethernet PHY for transmission.

To perform the above operations; multiple modules were written, each performing a specific task or creating a specific interface. The modules were summarised in a top-down block diagram, showing the module/sub-module structure of the system, and which modules depend on data from others. The clock management module was discussed first. It is

responsible for generating clocking signals as required by the rest of the design. Additional information was provided on the peripheral interface with a 20 MHz crystal oscillator.

Next, the input signal interface was detailed. The path from the antenna, through the RF board, to the TIGER-3 was shown, along with the physical components on the TIGER-3 that relate to the ADC IC. The primary ADC pins and modes of operation was given. Finally, a timing diagram was provided to show how the output data of the module relate to the input signals.

After the sampling of the input signal, the DSP interface is tasked with processing the data. A description was provided for the design implementation of the NCO core and important design parameters were derived. The NCO core provides a reference signal for the demodulation and the output is generated synchronous to the ADC output for sample multiplication. The output from the multiplication process is transferred to the CIC filter (the CIC filter theory was provided in Section 3.1.9). Timing diagrams were used to detail the flow of data and relevant signals in the module.

The buffering and transmission of data through the Ethernet interface were detailed next. The design options regarding the buffer implementation was listed, followed by the design steps taken to implement the Ethernet interface. The primary interface lines were detailed with special attention given to the clocking schemes used for the MAC. A sub-module is tasked with generating packet headers once enough of the data are available. The header data are transferred from the packet generator to the MAC after which a multiplexing design enables the FIFO and the packet data are transferred to the MAC. Once again, a timing diagram was provided to illustrate the data paths.

Finally, a summary of the PC-side software was given. Multiple software packages were used to perform debugging, data capture and processing of the Ethernet packets.

5 Implementation Results

This chapter details the results obtained from the final implementation of the listening mode on the TIGER-3. Where applicable, results are accompanied by simulations to provide an ideal reference for the discussion to follow. The discussion of the results are detailed in Chapter 6. This chapter is organised according to the functional outcomes of the listening mode: starting with clocking and signal synthesis, followed by signal sampling and processing. Finally, results are given concerning the transmission of data through the Ethernet interface.

5.1 Clocking

The operation of the FPGA fundamentally depends on the input clocking signal. Clock jitter originated from multiple sources, including thermal noise from the crystal and the internal workings of a PLL circuit [54]. The result is a deviation in the timing between the falling and rising edges of the signal. Signals are converted between analogue and digital domains through ADCs and DACs - which depend on input clocking. Jitter can therefore cause inaccuracies in the sampling frequency. Furthermore, excessive jitter can corrupt communication links by causing bit errors.

The first set of results will therefore detail the accuracy of the input clocking signal and the clocking signal generated by the clock management module. Clock forwarding was used to produce a clock output through the uLF connectors on the TIGER-3. Figure C.4 in Appendix C illustrated the circuit schematic counting the measured ports (P10-P9 and P13-P14). Figure 5.1 shows the signals at both differential pairs.

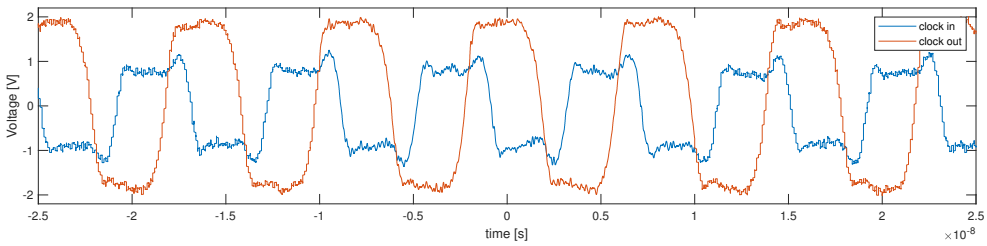

Figure 5.1: Clocking signals at the input and output of the FPGA.

The oscilloscope (Agilent MSO9104A) used to take the measurements in Figure 5.1 record multiple clock cycles to provide statistics for the clock edges. Table 5.1 details the clock properties measured. Each result in the table was obtained for a set of 3000 measurements. It details the maximum, minimum, average and standard deviation of the duty cycle and period for both the input and output signal.

Table 5.1: Duty cycle and period properties of the input and output FPGA clocking signals.

	Duty Cycle		Period	
	Input	Output	Input	Output
Mean	50.3 %	49.7 %	8.0002073 ns	8.01993 ns
Min	47.7 %	48.8 %	7.91599 ns	8.0005420 ns
Max	53.0 %	50.7 %	8.08547 ns	7.93948 ns
Std Deviation	674 m%	267 m%	21.4522 ps	17.0399 ps

5.2 NCO

In order to test the spectral purity of the NCO, the module's outputs are directly transferred to the Ethernet client module for transmission to the computer and further analysis. A block diagram, illustrating the configuration of the FPGA for the results obtained below, is shown in Figure 5.2. The "Packet Counter" fills the packet information field of the Ethernet packet to ensure that there are no missing packets in the output. All of the results in this section are obtained without the loss of packet information.

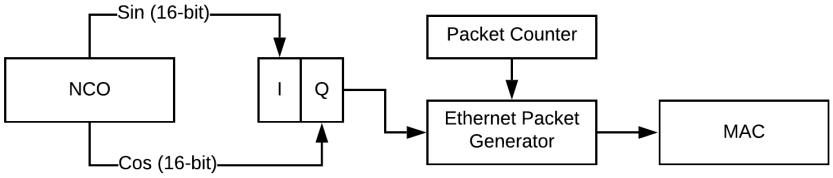

Figure 5.2: Diagram showing the design implementation to capture NCO results.

The NCO was set to produce a tone frequency at 2.5 MHz. The complete parameters set for this trial are given in Table 5.2. The output rate was limited to a frequency significantly lower than the 125 MHz required for the final implementation of the listening mode. This ensures that the maximum packet transfer rates of the Ethernet module is not exceeded when the data are transferred without decimation. Furthermore, decreasing the output rate ensures that packets are not lost due to the computer's processing capabilities.

Table 5.2: List of parameters used to capture and display NCO results.

Total Received Samples	1745
Samples per Packet	349
Sample Output Rate	7.8125 MHz
NCO Output Frequency	2.5 MHz
NCO Phase Width	32 bits
NCO Output Width	9 bits
FFT Zero-Padding	10000

The results for the parameters given above are shown in Figure 5.3. The first plot shows the spectral representation of the NCO output, and the second shows the output from a

simulation. The simulation was done by making use of quantised lookup tables to best mimic the operation of the NCO. The data are processed as an analytic signal, making use of the *sin* and *cos* outputs of the NCO in the form $\cos(\Theta(n)) + j\sin(\Theta(n))$ as it will be used for the base-banding process.

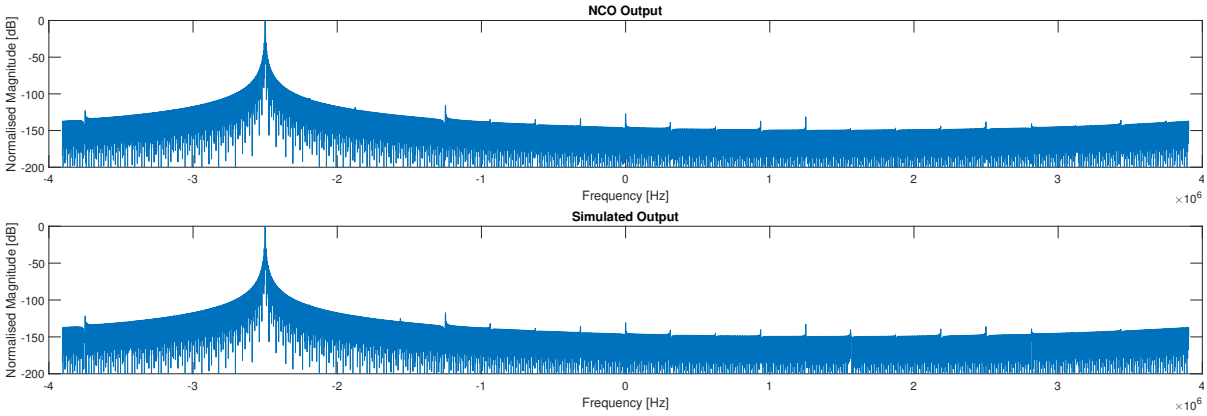

Figure 5.3: Top: Recorded NCO spectral outputs. Bottom: Simulated spectral results for corresponding parameters.

The current implementation of the NCO supports one primary input parameter, which is the phase increment. The NCO core also supports inputs for phase control and phase modulation which were not used for this work. Multiple results were recorded to ensure that the NCO parameters are reliably controllable at run-time. To illustrate user control, the phase increment was assigned to the dip switches on the TIGER-3, and the results for four discrete phase increments are shown in Appendix I (the remaining parameters were kept constant as given in Table 5.2).

The LUT size and phase increment width parameters define the performance of the NCO, and are set after synthesis. The phase increment width directly relates to frequency resolution (Equation 4.4) and simulations are not shown. However, the LUT size relates to the output signal purity and has a significant effect on FPGA fabric usage. It is therefore important to characterise the effects that the parameter has on the output signal to choose an optimal value. Figure 5.4 shows the effects that different LUT sizes have on the output spectrum of a tone.

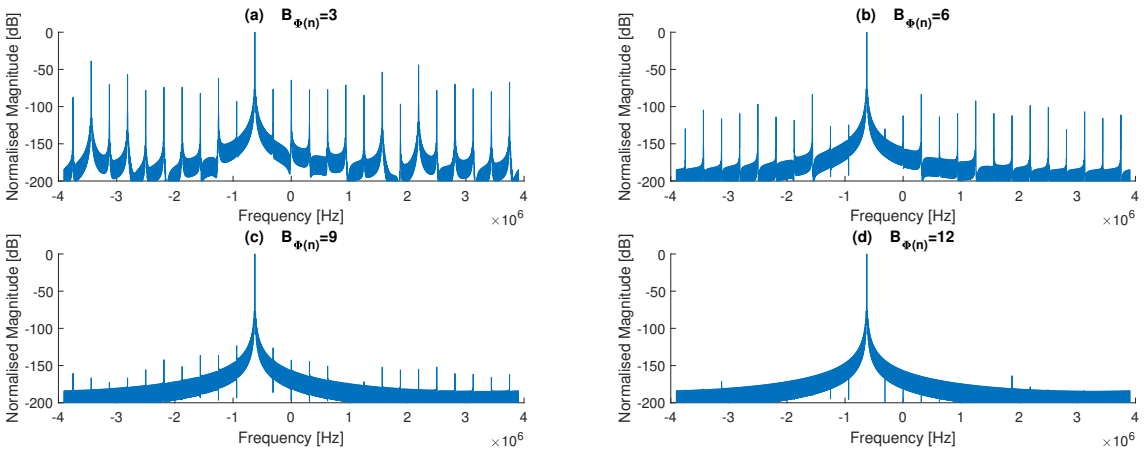

Figure 5.4: Simulated spectral results showing the effect that the LUT size ($B_{\Theta(n)}$ given in bit-length) of the NCO has on the synthesised signal.

5.3 RF Interface and ADC Readings

This section of results illustrates the gain control and filtering of the analogue signal from the input SMA port. It furthermore details results obtained for the sampling of the signal to the digital domain.

5.3.1 RF Board Gain Analysis

The fully detailed characterisation of the RF board was left for future work. The gain parameters of the RF board are controlled by an interface written for this project - the related results are shown here.

The gain of the RF board was measured for a set of discrete points by feeding a signal of known frequency (from 5 kHz and 75 kHz) and magnitude from a signal generator (Agilent 33250A) to the SMA input (see Appendix G for the schematic diagrams). The output amplitude was measured at the connection between the RF board and the TIGER-3. The results are given in Figure 5.5. Note that the measurements were taken before the transformer inputs to the ADC IC, and that in this case the step attenuator control was set to 0 dB attenuation. The input signal was set to 150 mV peak-to-peak. The plot was normalised from a peak gain of 54 dB.

Following the passive gain analysis; the attenuation control parameters were tested. The input signal was kept at 150 mV at a frequency of 12.57 MHz. The attenuator was set to provide 0 dB of attenuation, until a push-button trigger resets the C8 and C4 pins to add 12 dB of attenuation. The resultant waveform, as measured from an oscilloscope, is shown in Figure 5.6.

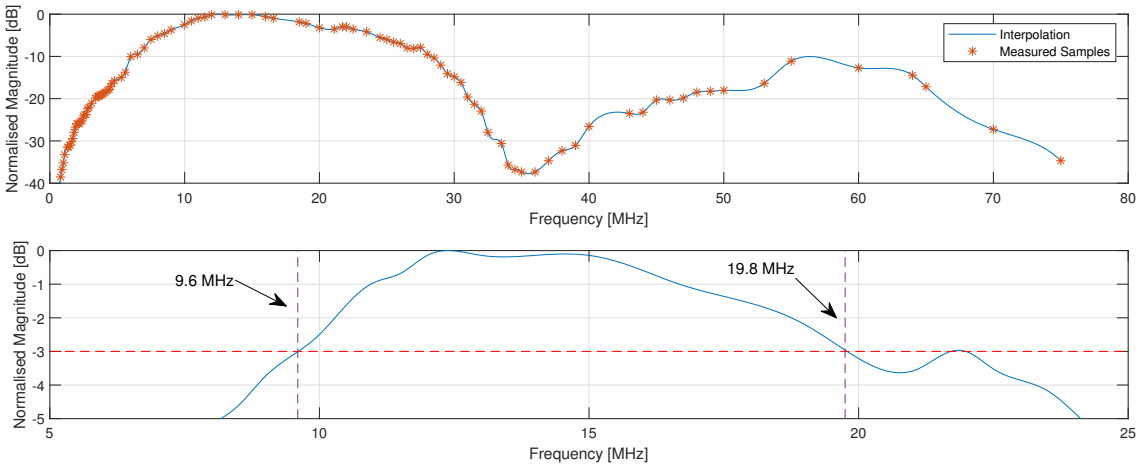

Figure 5.5: Top: Full RF board frequency magnitude response excluding primary input filter. Bottom: Corresponding pass-band.

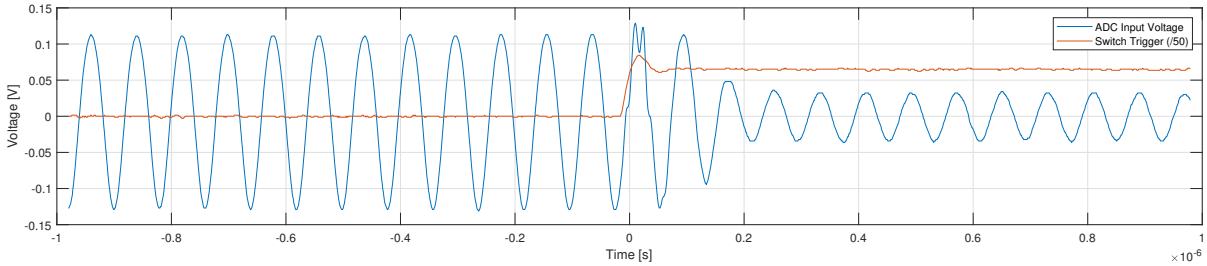

Figure 5.6: ADC input signal for a 12 dB attenuation step.

Finally, a similar test was performed by triggering the SMA/DAC switch (with no DAC output). The results for the switching is shown in Figure 5.7. For both tests; the input trigger was also recorded to allow for analysis of the switching time of the peripherals under test. The "Switch Trigger" signal voltage level was decreased by a factor of 50 in both cases. This concludes the results that exclusively relate to the RF board. By making use of the RF board, the next set of results depended on the ADC.

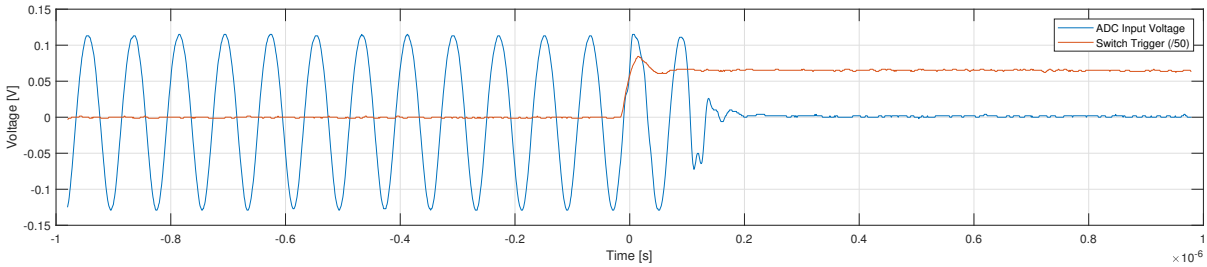

Figure 5.7: ADC input signal for switching from receive to transmit.

5.3.2 ADC Noise

The ADC module was tested in a similar way to the NCO. The output was directly connected to the Ethernet module and therefore the sample rate had to be decreased significantly. The ADC was clocked by the standard 125 MHz differential signal, but the Ethernet module was clocked at 1/16 of the ADC output rate - resulting in a sample rate of 7.8125 MHz. The ADC output was passed to both of the channels of the Ethernet input, and where required; an input signal was supplied by the signal generator.

The first set of results was obtained to record the errors associated with the ADC. To measure the ADC's noise; the input lines of the ADC (before the transformers) were short circuited and 1745 samples were recorded. The ideal output was expected to be zero but quantisation and callibration errors typically affect the ADC readings. Figure 5.8 shows the time and frequency domain data (FFT zero padded by 10000 samples). Figure 5.9 shows histogram of the output data. Note that the ADC outputs were received as two's

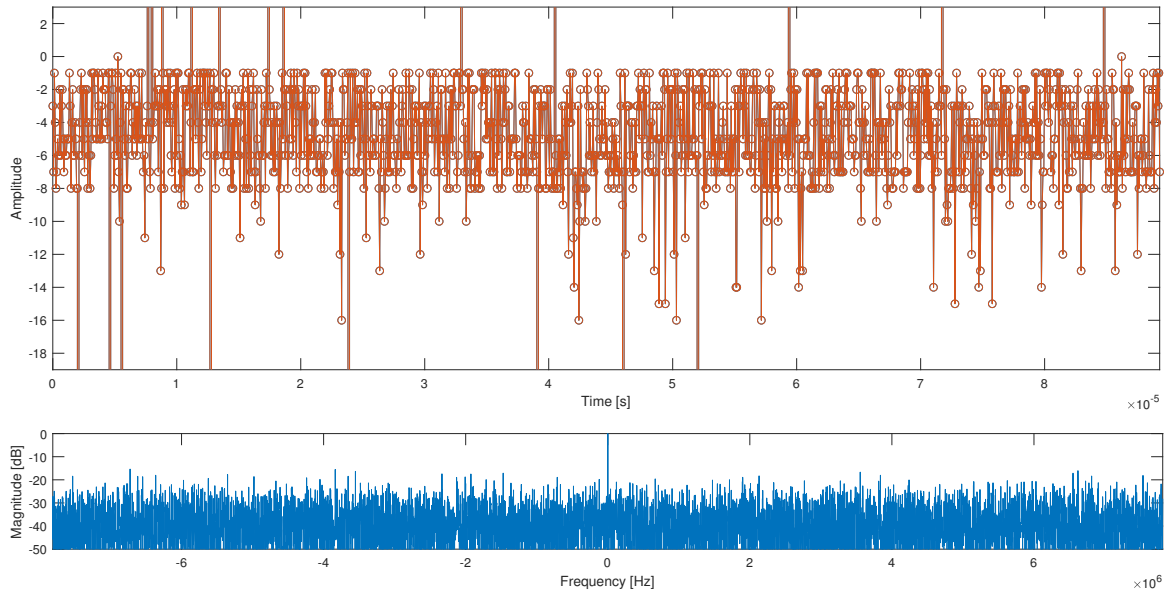

Figure 5.8: Top: Time domain output of the ADC with the ADC input pins shorted. Bottom: Corresponding spectral domain representation.

compliment binary data and therefore 16-bits may represent integer value from -32768 to 32767. The horizontal axis of the histogram in Figure 5.9 was truncated to show the majority of the recorded samples, ranging between -8 to -1. The highest amplitude values were sparsely located and recorded near the maximum binary values. These high amplitude irregularities only exist at high Ethernet throughput rates. Appendix H contains the complete histogram.

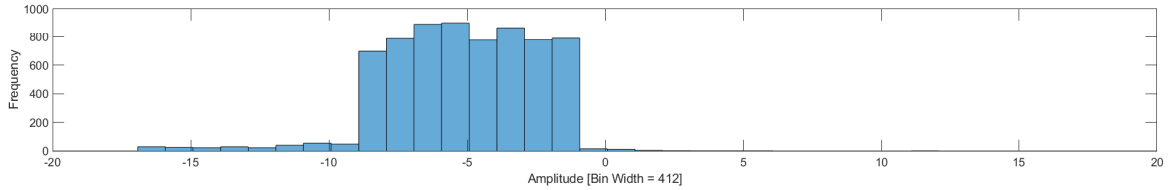

Figure 5.9: Histogram of the ADC output with the ADC input pins shorted

From the results, it is important to note that the sparsely located high amplitude irregularities only appear at high throughput rates. No errors were recorded at a data rate (and equivalent instantaneous bandwidth) of 3.9 Mhz, which is well above the requirement for most HF systems. The FFT results were similar for both data rates, as the irregularities were infrequent enough not to affect the FFT. Results for the higher data rate were used for this project as it provides a larger instantaneous bandwidth for noise and leakage analysis. The results are discussed in further detail in Section 6.

5.3.3 RF System Noise

The spectral components introduced by the ADC was localised and analysed above. The following results were obtained to analyse the RF leakage and noise associated with the RF board and the power supply of the system.

A signal generator was used to drive a signal into the SMA connector on the RF board (at 50 Ω driving impedance). The signal was sampled by the ADC on the TIGER-3 and directly passed to the Ethernet module for transmission. The signal generator output was disabled to attempt to measure the noise introduced by the system. Due to technical faults an amplifier was bypassed on the RF board, and a maximum gain of 27 dB was realisable and used. The results for the test are shown in Figure 5.10. The spectrum shows high power components at approximately 375 kHz. The histogram of the sampled data is presented in Appendix H.

Note again that the same information is transmitted through both channels and is shown separately in the plot as *Channel 1* and *Channel 2*. From the results, it was noted that the data points do not always correlate for both channels - especially in the case of the high amplitude irregularities. These results support the statement that the errors are due to Ethernet related bit errors that were only present with high throughput rates.

5.3.4 ADC Input Signal Sampling

The system parameters were kept constant from the previous set of results, except that an input signal was introduced at a peak-to-peak voltage of 150 mV with 27 dB gain from the RF board and. The gain results in an ADC input at the ADC's full dynamic range of 3.3 V peak-to-peak. Sampled data was recorded by setting the signal generator

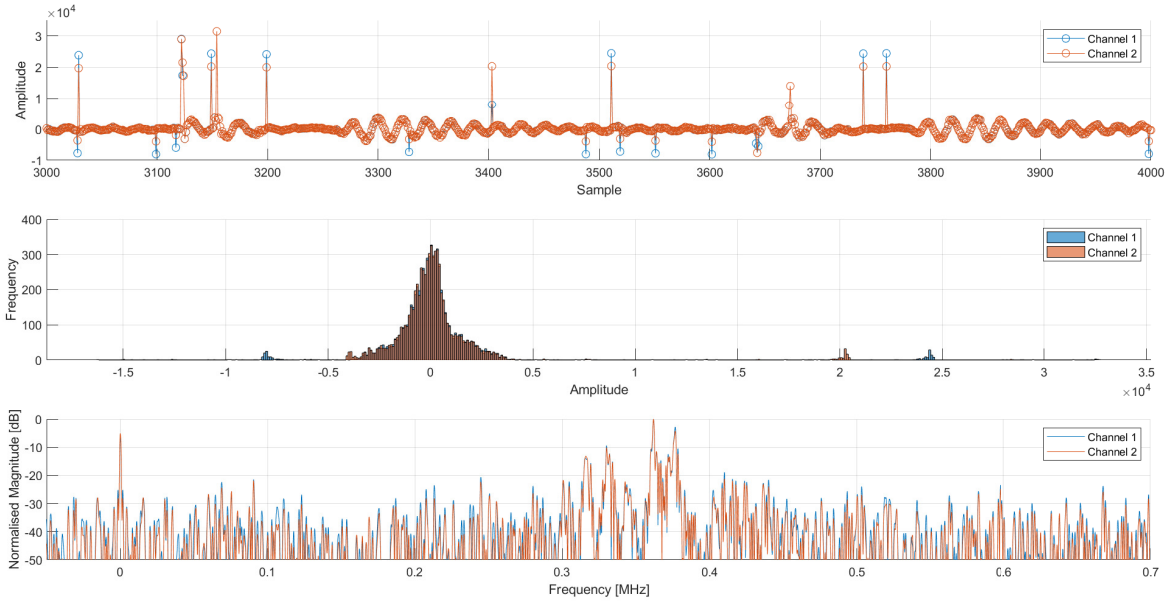

Figure 5.10: Top: Time domain output of the ADC with input pins connected to the RF board. Bottom: Corresponding spectral domain representation.

to four discrete frequencies. The results consisted of 1745 samples, for each frequency, transformed by an FFT with 10000 samples zero-padding. The resultant spectrum for *channel 1* are shown in a combined plot in Figure 5.11.

Excluding the spectral components identified in Figure 5.10 (which are consistently observed in the results in Figure 5.11), spurious spectral power components are observed alongside the expected measured frequencies. The spurious components are observed at less than -60 dB of the peak output for all measurements.

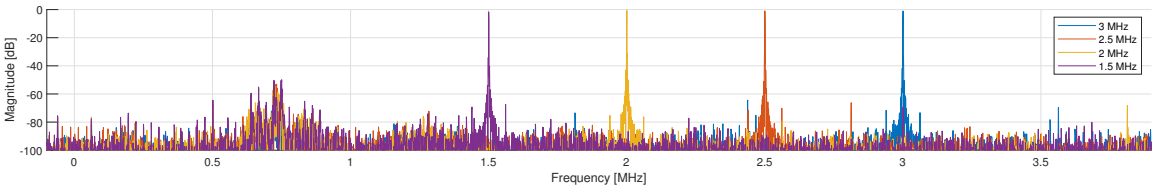

Figure 5.11: ADC sampling spectra from readings for a set of input frequencies. Each signal is normalised to its maximum

5.4 Base Banding

To demonstrate part of the base banding process (signal multiplication) implemented within the FPGA; two NCO signals were generated and input to the demodulation mod-

ule (without filtering). The first NCO (*NCO-1*) was set to produce a tone of 2.1 MHz (representing the ADC input), and the second (*NCO-2*) had a variable phase increment (to demonstrate multiplication frequency control). The signals are multiplied and the output is directly passed to the Ethernet module for transmission. A block diagram illustrating the implementation is shown in Figure 5.12. The output for three demodulating frequencies are shown in Figure 5.13. The black arrows (corresponding to each plot's legend) illustrates the demodulation frequency, given by $e^{-j2\pi f_{NCO-2}}$. The spectrum is expected to shift and be centred about $f_{\text{NCO-2}}$ after demodulation for each plot.

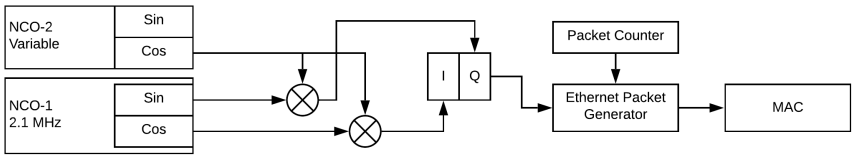

Figure 5.12: Block diagram showing the design implementation used to capture results for the baseband process.

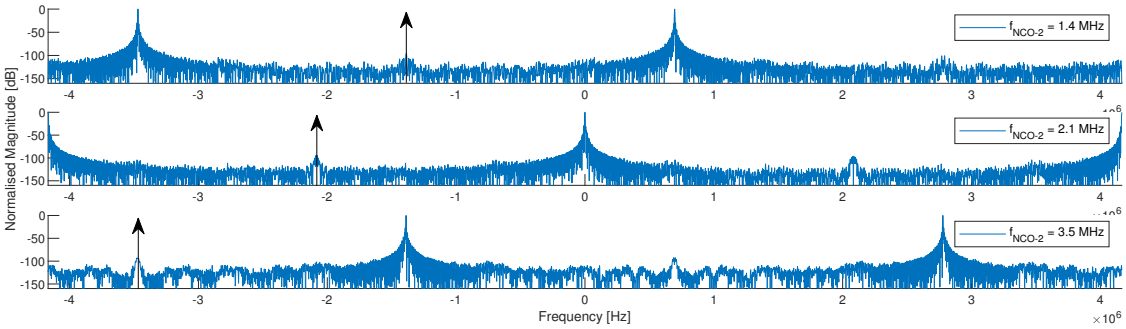

Figure 5.13: Spectral results for the demodulation of a tone by three discrete frequencies.

5.5 Filtering

The final step in the listening mode DSP chain is the filtering and decimation of the signal. It is followed by the transmission of the information to the Ethernet modules in the final implementation. Here, the sampling frequency is set to 125 MHz (as planned for the final design). The decimation decreases the output sample rate for compatibility with the maximum Ethernet rate. For illustration purposes, a low decimation rate of 16 was initially used for the filter. This corresponds to an output sample rate of 7.8125 MHz.

5.5.1 Wide-Band Filtering

To best illustrate the filter's output; Figure 5.14 contains plots of a simulated result (pre and post-decimation) and the recorded output from the designed system. The param-

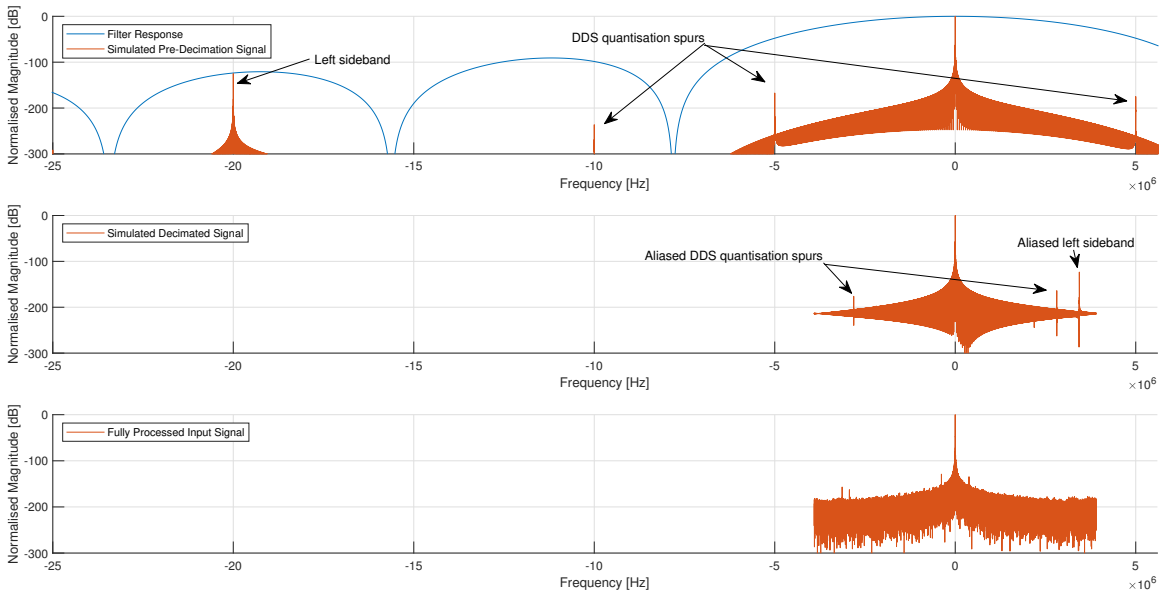

Figure 5.14: Top: Simulated results showing the expected pre-decimation spectral output of the filter. Middle: Simulated results after decimation. Bottom: Recorded results for an input tone measured at the output of the filter.

eters for this trial were as follows: 60 packets were recorded, totalling 20 940 samples, processed with zero-padding for spectral clarity. The input signal was at 10 MHz with the demodulating frequency also set to 10 MHz. The NCO had a phase width of $B_\theta = 32$, and a output width of $B_\Theta = 9$. The CIC filter had a decimation rate of $R = 16$, with $N = 3$ stages and a differential delay of $M = 1$.

Figure 5.15 illustrated the spectral aliasing that occurs during the decimation process as per Equation 3.53. Every aliased band is shown individually. The sum of these bands equates to the simulated result given as the *Simulated Decimated Signal* from Figure 5.14.

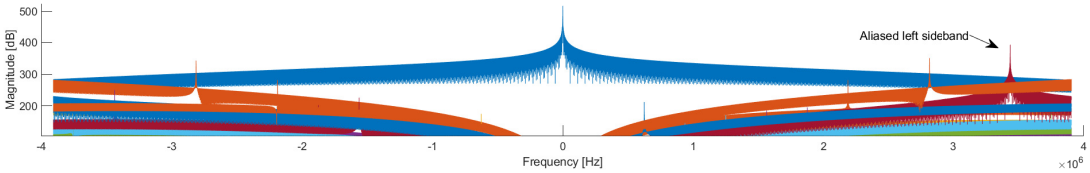

Figure 5.15: Simulation showing the filter's aliased spectral bands individually.

To better understand the source of spurious components in the output signal; the base-banding module's demodulation frequency was varied to produce multiple sets of results that together reproduce a larger spectral range than what can be realised from a single

reading.

A constant input frequency of 11.5 MHz was provided from the signal generator. A total decimated bandwidth of 7.8125 MHz was captured for each reading with a -3 dB pass-band that is 1.35 MHz wide. A 4 MHz bandwidth is shown from each reading as higher frequencies are attenuated by the filter and not required for the results. The demodulation frequencies (f_{dds}) range from 8 MHz to 13 MHz and are spaced 1 MHz apart. The results are given in Figure 5.16.

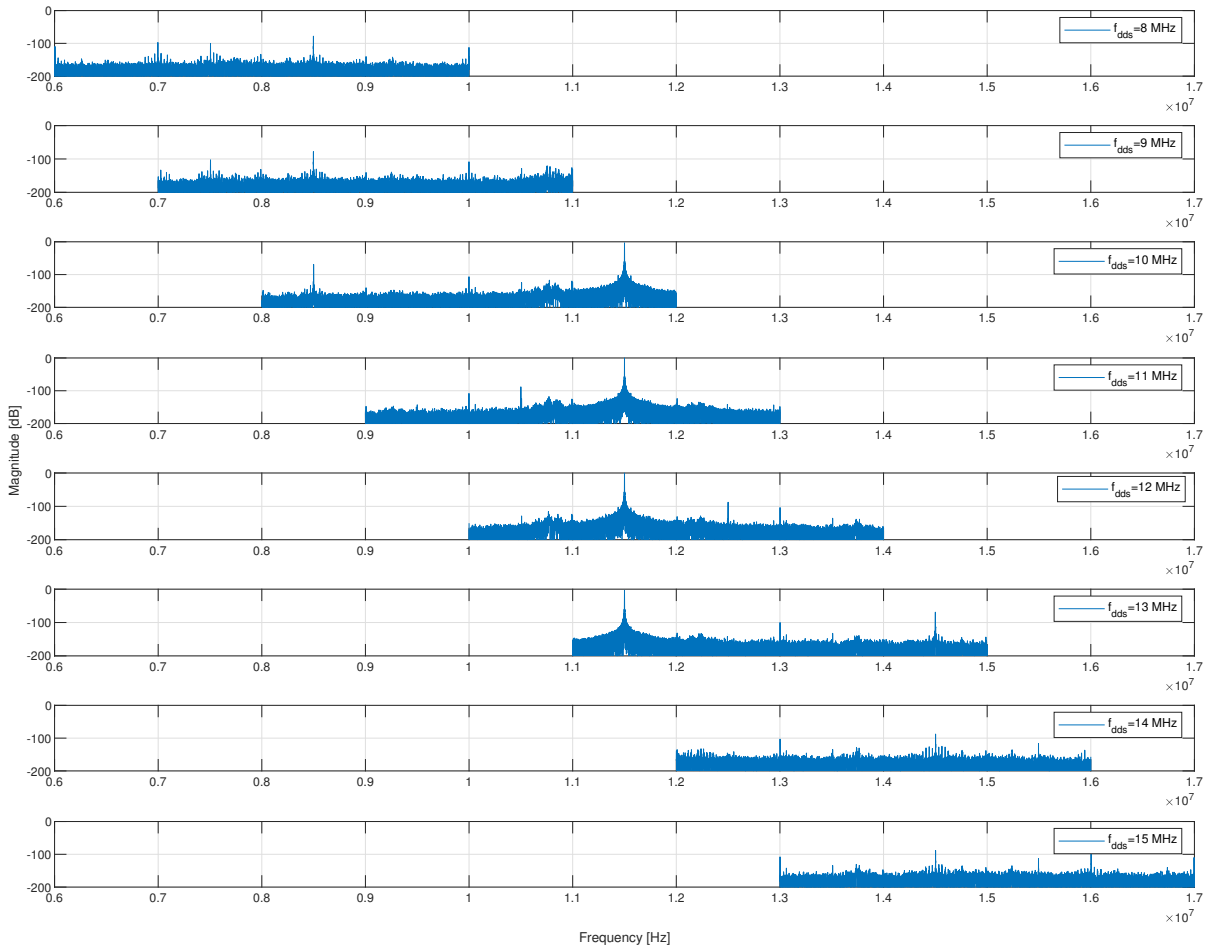

Figure 5.16: Demodulation sweep of an input signal at 11.5 MHz.

Spurious components were recorded in the results at a maximum of -78 dB of the input frequency. It was observed that even though the majority of spurs appear consistently in all of the readings, a spur appeared at 11.5 Mhz for $f_{dds} = 12$ MHz which was not present in other readings. These "moving" spurs appear to be the translated and filter attenuated left sideband, aliased into the pass band after CIC filter decimation. The location of the sideband after decimation depends on f_{dds}. The processing of the left sideband is

simulated in Figures 5.14 and 5.15.

A second set of readings are shown for smaller demodulation steps. The smaller steps show the presence and movement of the aliased left sideband. The maximum left sideband magnitude was recorded for f_{dds} = 12 MHz at -140 dB of the peak output magnitude. The results are shown in Figure 5.17.

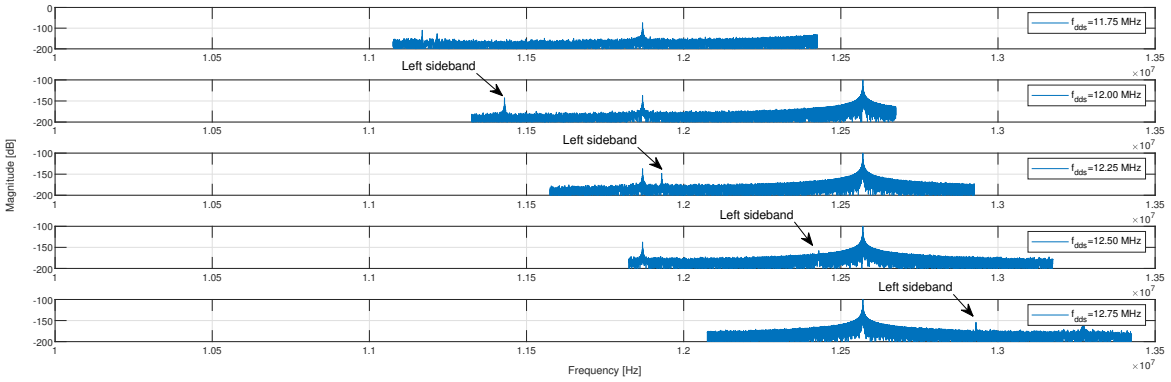

Figure 5.17: Demodulation sweep of an input signal at 12.75 MHz, showing the aliased left sideband.

5.5.2 Narrow-Band Filtering

The SuperDARN operates by measuring Doppler shifts close to 100 Hz. In this case the decimation of the filter can be substantially increased, resulting in a narrower pass-band of frequencies around the demodulating frequency.

For the next set of results; the decimation factor was set to 8000. The filter supports a maximum decimation factor of 8192. The resultant low pass filter has a -3 dB cut-off frequency of approximately 2.7 kHz, The filter's magnitude response is shown in Figure 5.18.

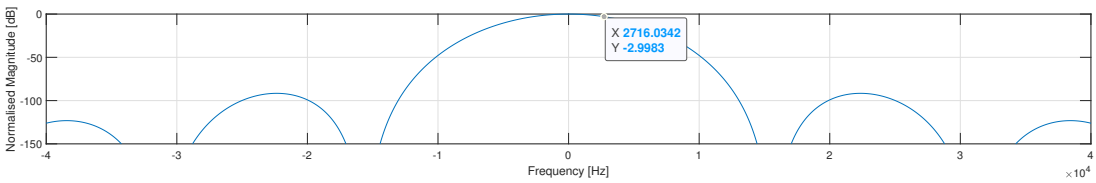

Figure 5.18: CIC filter magnitude response with the decimation factor set to 8000.

The demodulation frequency was set to 10 MHz. Readings were taken for a range of input frequencies, and a near-DC output frequency of 0.5568 Hz was recorded for an input tone of 10 000 146 Hz. The offset of 146 Hz was expected as simple over-the-counter crystal oscillators (such as the on-board crystal used by the TIGER-3) have low tolerance values

that depend highly on temperature. The timing of the crystal may therefore not correspond well with tat of the signal generator, creating an offset in the results.

Decreasing the input frequency to 10 000 136 Hz (corresponding to a Doppler shift of 10 Hz) produced the spectral results shown in Figure 5.19. The input voltage was set to a sine wave, 150 mV peak-to-peak with the RF board gain set to 27 dB. The time-domain signal is shown in Figure 5.20.

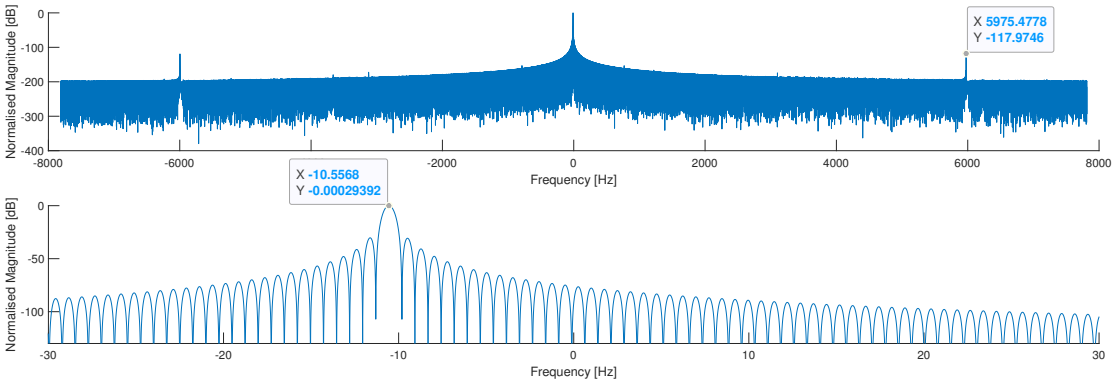

Figure 5.19: Spectrum of the fully processed ADC input tone with a 10 Hz Doppler shift and a filter decimation factor of 8000.

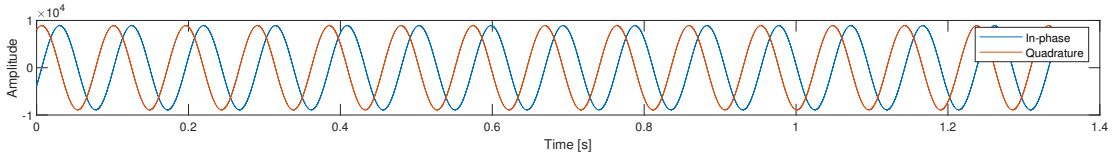

Figure 5.20: Time-domain signal of the fully processed ADC input tone with a 10 Hz Doppler shift and a filter decimation factor of 8000.

A total of 60 packets were collected for 20 940 samples. The NCO and filter parameters remained constant from the previous results apart from the increase in the decimation factor of the filter. One dominant spur was recorded within the full filter output spectrum, at approximately 6 kHz, with an amplitude at -118 dB of the output peak. For a total continuous wave recording time of approximately 1.4 seconds; a -3 dB Doppler resolution of 0.223 Hz was achieved. The spectral peak of the output was located at -10.5568 Hz.

5.6 Wide-Band Signal Sampling

The input signal was set to repeatedly sweep a total bandwidth of 500 kHz over 4.5 ms centred at 12 MHz. A decimation rate of $R = 16$ was used to collect 24 430 samples (3.1 ms of recording). Readings were repeatedly taken until a data set contained the majority

of the sweep data was captured. The data was truncated to 21 801 samples totalling 2.7 ms in time and containing a sweep of 300 kHz. The base-band output is shown in Figure 5.21.

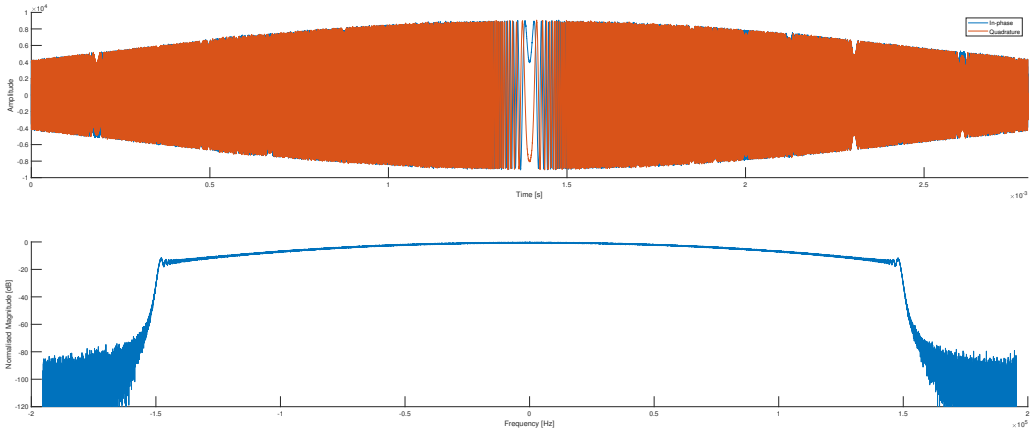

Figure 5.21: Top: sampled wide-band signal. Bottom: Corresponding spectral plot.

5.7 Data Transmission

The results of the Ethernet's performance is not so easily represented graphically. Its operation is either functional or not, and at the end, it operated functionally. The packet integrity in terms of data rate is limited by a bottleneck on the PC receive side - the results of which are given later in this Section.

Initial results produced many errors, the most notable of which was that the FPGA programmed packet manager module (designed for this project) lost a single sample at the end of every packet. The resultant waveforms looked correct but the spectral domain revealed periodic components indicating a periodic error in the compiled time-data. The resultant waveform for an NCO output (expected to be similar to the results in Figure 5.3) is shown in Figure 5.22. After close inspection, the time-domain signal revealed a

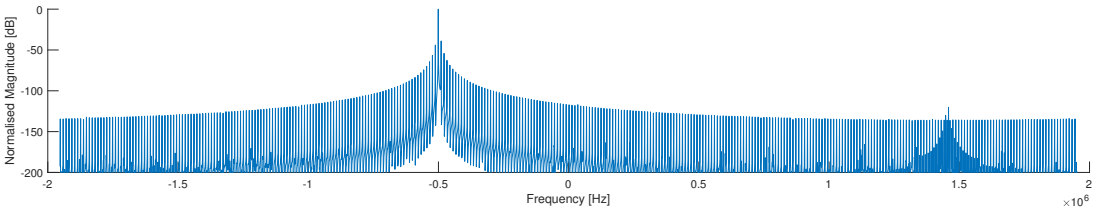

Figure 5.22: Spectral results for an NCO output with one missing sample per packet.

missing sample at the end of every packet. The fault was attributed to the Ethernet

implementation which enabled the FIFO's read functionality for an additional cycle after indicating the completion of the packet to the MAC. The programming error was fixed and the Ethernet functioned as expected.

5.7.1 Data Transmission Rate and Integrity

As formerly mentioned in Section 5.7, a bottleneck was detected on the PC receive side. The Python program tasked with capturing and stitching together packets periodically loses packets based on the number of packets set to be captured and the packet rate. The introduction of the packet information header (to indicate the packet number) aided the PC interface in determining when and how many packets were missed, and parameters would need to be adjusted for new readings. The missing packets were attributed to limitations in the processing capabilities of the PC, as an increase in active processes on the PC would drastically increase the rate at which packets were lost.

The packet rate and the number of received packets were set to a range of values, and the packets were recorded. The packet information header was used to determine the total number of packets transmitted to successfully capture the given number of packets. Table 5.3 details the percentage of packets successfully received compared to the total number of transmitted packages. The frequency given in the table corresponds to the rate at which data (32-bits) were presented to the Ethernet module. The decimation required for an input data rate of 125 MHz is given in brackets.

For example, if the output data frequency was set to 7.8125 MHz and the Python script was set to capture and store 100 packets. This total data rate would be given by 7.8125 MHz $\times$ 32-bits $=$ 250 Mb/s. The packet information header would have indicated a total of 185 transmitted packets, equating to 35% of the total packets being captured.

Note that the performance varied significantly between readings with the same parameters at different times. This could be attributed to the processing capabilities of the PC at the time. Each results in Table 5.3 was calculated by averaging a total of five readings.

Table 5.3: Percentage of packets successfully received at a given output data rate, after a set number of packets are successfully collected.

	Packets Collected			
	20	**50**	**100**	**300**
Frequency (Decimation)	**Percentage Received**			
15.625 MHz (8x)	100%	59%	20%	14%
7.8125 MHz (16x)	100%	100%	35%	23%
3.906 MHz (32x)	100%	100%	68%	45%
1.953125 MHz (64x)	100%	100%	100%	99%

5.8 Resource Usage

A summary of the FPGA resource usage is given by ISE after synthesis. The summary for the final design usage is given in Table 5.4. The highlighted rows indicate the usage of a certain class of resources, and is followed by sub-classes where applicable. The table indicates the number of elements utilised and the number of elements available for the specific device.

5.9 Chapter Summary

The results shown in this section were obtained to confirm the functionality of each design component. The clock peripherals were tested - as the system's performance is largely based on their functionality. These tests were followed by additional tests concerning the NCO, its parameters, and the effect that those parameters have on the output.

The next set of results concerned the analogue processing and sampling of the input signal. The gain of the RF board was characterised and attenuation switching was demonstrated. Sampling results were obtained to aid in characterising the system's noise, and to demonstrate the effective sampling of signals by the ADC.

Furthermore, the demodulation module was tested by providing two known signals to the input of the module and transferring the output to the computer for analysis.

Results were then obtained for the complete processing chain. A signal was sampled by the ADC, base-banded, filtered and decimated for a range of input frequencies (including a frequency sweep) to illustrate the functionality of the complete system. Simulations verified that results were similar to expectations, and were used to identify artefacts in the output signal.

Additionally, the Ethernet's functionality was characterised. It was shown how errors in the interface programming were identified (and corrected) for early results. The performance of the Ethernet module was measured based on packet rate and data loss, detailed in Table 5.3.

In conclusion, the design's resource usage for the FPGA device was detailed.

Table 5.4: Summary of the FPGA's resource usage for this work.

Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	767	32,640	2%
Number used as Flip Flops	767		
Number of Slice LUTs	666	32,640	2%
Number used as logic	538	32,640	1%
Number using O6 output only	386		
Number using O5 output only	98		
Number using O5 and O6	54		
Number used as Memory	69	12,480	1%
Number used as Dual Port RAM	32		
Number using O5 output only	16		
Number using O5 and O6	16		
Number used as Shift Register	37		
Number using O6 output only	37		
Number used as exclusive route-thru	59		
Number of route-thrus	160		
Number using O6 output only	156		
Number using O5 output only	3		
Number using O5 and O6	1		
Number of occupied Slices	388	8,160	4%
Number of LUT Flip Flop pairs used	1,017		
Number with an unused Flip Flop	250	1,017	24%
Number with an unused LUT	351	1,017	34%
Number of fully used LUT-FF pairs	416	1,017	40%
Number of unique control sets	55		
Number of slice register sites lost to control set restrictions	104	32,640	1%
Number of bonded IOBs	99	480	20%
Number of LOCed IOBs	99	99	100%
IOB Flip Flops	28		
IOB Master Pads	2		
IOB Slave Pads	2		
Number of BlockRAM/FIFO	3	132	2%
Number using BlockRAM only	3		
Number of 36k BlockRAM used	2		
Number of 18k BlockRAM used	1		
Total Memory used (KB)	90	4,752	1%
Number of BUFG/BUFGCTRLs	5	32	15%
Number used as BUFGs	5		
Number of DCM_ADVs	1	12	8%
Number of DSP48Es	6	288	2%
Number of PLL_ADVs	1	6	16%
Number of TEMACs	1	2	50%
Average Fanout of Non-Clock Nets	3.04		

6 Discussion of Results

This section concerns the discussion of the results in the previous section. It follows a similar order to the results section.

6.1 Clocking

Initial clock trials did not provide encouraging results. The interface was incorrectly initiated without the use of clock forwarding. Signals on the clocking net were connected through general signal routes to IO buffers, resulting in large amounts of jitter. Until clock forwarding was implemented, peripherals such as the ADC performed poorly. Making use of differential input buffers further increased performance (initially single ended clocking sourced from one of the two differential clock lines was used). Finally, making use of CMTs provided duty cycle stabilisation and synthesis of alternate frequencies.

The performance of the clocking module was characterised by results given in Figure 5.1 and Table 5.1. The clock at the input of the FPGA was compared to the the output and therefore the results include the performance of the input/output buffer configuration, CMT, and clock forwarding. The results show a stable output clock. The results shown an improvement in the maximum deviation from a perfect 50% duty cycle, from 3% to 1.2%. These parameters are well within the requirements of the Ethernet PHY and will result in negligible errors in digital/analogue conversions. The results show the successful completion of the clock's technical requirements.

6.2 RF Board and Gain Control

The results in Figures 5.5, 5.6 and 5.7 demonstrate the passive and active gain control and switching provided by the RF board. The passive gain response suggests that the true pass band for the RF board is not as wide as some of the receive chain components may suggest. A pass-band gain of 54 dB was recorded, which correlated well with the expected output from two 27 dB amplifiers with the step attenuator set to 0 dB. The magnitude response profile suggested optimal operation from approximately 12.5 MHz to 15 MHz.

Even though the low-pass filters used are set at 30 MHz, high signal attenuation was recorded for signals above 20 MHz. The reason for the attenuation could not be accounted for, considering that all passive components in the RF board chain are rated to work from DC to 30 MHz. No further attention was given to the matter, as extensive analysis of the RF hardware is not part of the scope of this work and the spectral range of interest is within the pass-band. The transformers at the input of the ADC act as short circuits at low frequencies, and therefore, attenuation was also measured for signals below 10 MHz.

Attenuation control was also demonstrated with a switching time of approximately 0.2

μs recorded at the output. The component is rated for a typical switch delay of 1 μs. Furthermore, SMA/DAC switching was demonstrated with a switching time just under 0.2 μs.

6.3 Input Signal Sampling and Processing

The trials performed to obtain results for the signal processing modules illustrated the robust design of the modules. The inputs and outputs of each module was chosen to allow for integration into other designs, which was evident when modules were configured in alternate ways to provide the results shown in this work.

6.3.1 NCO

Based on a series of simulations (some of which are shown in Figure 5.4), module parameters for the NCO were chosen. B_Θ was shown to directly influence the presence of spurious noise in the NCO output. The spurious noise from the ADC (Figure 5.11) and the CIC filter capabilities (Figure 5.14) was considered when choosing on a parameter value. Seeing as resource usage was not a concern for the current implementation; $B_\Theta = 9$ was chosen to produce a signal with spurs at no more than -130 dB of the peak output. This means that the NCO spurs appear at similar gain levels to the expected spurious noise in the system (according to Figure 5.14).

Considering that the SuperDARN records Doppler readings up to 100 Hz; the frequency resolution of the NCO had to be high enough to account for the required spectral sensitivity. An NCO frequency resolution of 0.0291 Hz was chosen to be sufficient - corresponding to a phase increment bit-width of $B_\theta = 32$. Since the NCO is used for demodulation, a small offset in frequency could be accounted for in post-processing, given that the base-banded signal bandwidth falls within the pass-band of the digital filter. The resource usage associated with B_θ is low, allowing for the luxury of high precision design.

6.3.2 Signal Sampling

With an output data width of 16-bits and a sampling rate of 125 MHz; the ADC produces data at a rate of 2 Gb/s, which is twice the absolute limit of the Ethernet interface. This justifies subsequent signal processing. The results in Figure 5.11 shows a spurious free range as high as -60 dB of the peak output for a sampling frequency of 7.8125 MHz. It is however important to note from Figure 5.5 that the signals recorded are strongly attenuated by the ADC interface at these frequencies. The results show (although at low sampling frequencies) that the ADC interface module is functional.

Spurious spectral components from the sampling/processing interfaces (such as the aliased left sideband), may interfere with the received spectrum and may interfere with sensitive readings. The results given in Figure 5.14 were taken to illustrate a large bandwidth

recording. In this case a decimation rate of $R = 16$ produces a -3 dB the pass-band approximately 2.72 MHz wide, with a total received bandwidth of 7.8125 MHz. These results are further discussed in terms of filtering in Section 6.3.4 and in terms of noise in Section 6.3.5. Here, it is important to note that an input tone was effectively measured for an ADC sample rate of 125 MHz.

6.3.3 Base Banding

There is not much to add concerning the results obtained from the demodulation. The module performed as expected and the output correlated well with the simulated results. The results in Figure 5.13 show the effects of wrapping as the data in the final two plots were demodulated by a high enough frequency for left sideband to wrap around to the positive spectral domain. The filter is responsible for removing this sideband.

6.3.4 Filtering and Decimation

This section of the discussion relates to the results obtained for Figure 5.14. The results illustrate the functionality of the CIC filter and are accompanied with simulations of the expected results. It illustrates the filter's response to the input spectrum for a single input frequency. As discussed in Section 3.1.9, the CIC filter has a *sinc* function shaped magnitude response (see Equation 3.52).

Equation 3.53 shows that, in this case, a 2.72 MHz spectral range centred at every filter magnitude response null, is wrapped into the decimated output spectral band. Note that the decimated spectrum is 7.8125 MHz wide, and therefore any spectral components outside of the range of -3.9 MHz and +3.9 MHz will undergo wrapping. The first filter response side lobe peaks at -90 dB of the maximum. However, the highest gain provided to components outside of the decimated bandwidth is located at ± 3.9 MHz. The filter provides 27 dB of attenuation at these frequencies, which wrap to the outer extremes of the output spectrum. Any spectral components higher than ± 3.9 MHz will undergo increased filter attenuation as per the simulated magnitude response. This is shown as individual wrapped bands, simulated in Figure 5.15.

For the final set of filter results (Figure 5.19), the decimation factor was substantially increased to a value that would produce an efficient data output rate, and yet allow for a 100 Hz Doppler shift to be recorded. The demodulation centre frequency and input signal frequency were both set to 10 MHz. An output Doppler shift of 146 Hz was measured, suggesting a minor inconsistency between the clocking frequency of the FPGA and the output frequency of the signal generator. The 146 Hz of error can be reduced by calibration in pre or post-processing. The results showed how an increase of 10 Hz was effectively measured by the device, which corresponding well with the requirements of the SuperDARN. As per the theory in Section 3.1.6, the Doppler resolution is increased by increasing the number of measured samples.

6.3.5 System Noise and Interference

Three approaches were taken to analyse the noise from the complete system. The noise was measured by analysing the readings taken from the ADC. The first approach was to measure the noise generated by the ADC peripheral. The second was to attempt to measure the noise or spectral leakage originating from the other electrical components in the system. Finally the output spectrum is analysed for other sources of interference, such as noise originating from the signal processing.

The initial results (ADC, power supply and RF board noise) were analysed by making use of ADC recordings in the case where no signal is fed into the system.

ADC Noise

The result in Figure 5.8 characterises the performance of the ADC by measuring the output when the input lines shorted. The high amplitude artefacts have been attributed to the transmission process (see the end of Section 5.3.3),and are not further discussed here.

The high frequency histogram values (shown in Figure 5.9) indicate that without the DC offset, the measured output fluctuates by the three least significant bits. The noise generated by the ADC was observed to be approximately 72 dB below the maximum ADC level, corresponding to a peak-to-peak ADC input voltage 0.8 mV. A DC offset of -4.5 was measured corresponding to an ADC input voltage of -1 mV. The distribution appears to be normal with the spectral representation suggesting that no RF leakage has affected the results. The noise can therefore be considered Gaussian white noise.

Power Supply and RF Board Interference

The second set of results were taken with the ADC connected to the RF board and no signal provided to the input of the RF board (50 Ω source impedance). The results given in Figure 5.10 show the influence of the RF board and surrounding peripherals (such as the power supply) on the ADC input.

Interference was first identified by making a simple wire loop antenna with an oscilloscope probe and placing it in close proximity to the power supply. A periodic signal was observed on the oscilloscope, which was also found to be present in the second second set of ADC readings which followed. The second set of ADC readings were taken in similar fashion to the ADC readings formerly discussed in this section.

The second set of readings are given in Figure 5.10. The ADC was sampled at 7.8125 MHz, clearly showing interference centred at approximately 725 kHz. This interference was the only high power components recorded. Furthermore, according to the histogram, the interference was approximately centred about zero, with an approximate peak-to-peak

value of 8000. The interference is therefore at approximately 42 dB below the maximum ADC level.

These readings were taken with an RF board gain of 27 dB. For an input signal to occupy a similar ADC dynamic range to that of the recorded noise; the input signal will need to have an amplitude of

$$V_{in} = \frac{ADC_{noise}}{ADC_{maximum}} \times \frac{V_{ADC_{max}}}{G_{RF}} = \frac{8000}{2^{16-1}-1} \times 3.3/10^{\frac{27}{20}} = 36 \text{ mV} \tag{6.1}$$

where ADC_{noise} is the peak-to-peak value of the noise, $ADC_{maximum}$ is the maximum peak-to-peak value of the ADC with a corresponding voltage of $V_{ADC_{max}}$, and G_{RF} is the gain provided by the RF board.

Note that at this stage it is unknown what effects extra gain from the RF board would have on the measured noise. The gain of 27 dB, used here, is half of the typical gain of the system (one amplifier was bypassed due to technical faults). If the extra gain had no effect on the noise amplitude (highly unlikely), and a total gain of 54 dB was realised; the above result would suggest that an input peak-to-peak voltage of 1.6 mV would appear at the same amplitude as the interference.

It is furthermore noted that that the jumper lead used to bypass the passive filter at the input of the RF board may be responsible for capturing the majority of the RF interference from the power supply. The jumper acts as a loop antenna, which is very effective in capturing noise. The interference added by the power supply can only be analysed effectively for the transceiver once the input passive filter is completely implemented (not the case for the current system) and the second faulty amplifier is replaced and not bypassed.

Recorded Output Interference

The final analysis concerns the interference in the filtered results. This discussion largely relates to the results obtained in Figures 5.16 and 5.17.

The results obtained for Figure 5.14 revealed noise spurs in the output spectrum that did not correlate well with the simulation. An attempt was made to create a realistic simulation model that would perfectly model the real system, this was not fully realised for the complete system. The final output in Figure 5.14 contains two noise spurs on either side of the output tone, -125 dB lower than the output tone's magnitude. Other noise spurs were observed at approximately -3.5 MHz. It is difficult to determine whether these spurs originated from the sampling process, or from signal processing.

For the results shown in Figures 5.16 and 5.17, the base-banding frequency was adjusted for multiple readings to visualise a larger spectrum. The individual overlapping readings

provide additional information on the source of the spurious noise in the system. There are four primary sources of interference: input noise as sampled by the ADC, RF leakage from the RF board, noise originating from the quantisation of the NCO LUT, and the filtered left sideband wrapped into the pass-band during decimation.

The results shown in Figure 5.16, show multiple spurs that originated from the sampling process (as they are present for multiple demodulating frequencies). However, consider the peak at approximately 12.5 MHz (in the subplot for $f_{dds} = 12$ MHz) which is not present in the adjacent readings. This suggests that it originates from the DSP process.

A single non-symmetric spur such as the one discussed above is likely to be due to the filter suppressed left sideband. A second set of filter sweep readings were taken for a smaller filter step size - shown in Figure 5.17. The input tone for these readings were set to 12.75 MHz. The readings clearly show the wrapped image move across the decimated bandwidth as the demodulating frequency is changed. The spur moves by the same spectral increment for each change in the demodulating frequency, confirming its origin as the left sideband. The magnitude of the left sideband is greater when it is further from the baseband centre frequency, supporting the simulated results and supporting the theory given for Figure 5.15.

6.4 Wide-Band Input Signals

This result is shown simply to illustrate that the system is capable of receiving large bandwidth signals effectively. Theoretically, at a decimation rate of $R = 16$, the system supports bandwidths of up to 7.8125 MHz. In this case a sweep frequency input was recorded, centred at 12 MHz with a bandwidth of 300 kHz.

6.5 Ethernet Data Transmission

To effectively obtain signal information from the FPGA, Ethernet communication proved to be integral. The results show that the Ethernet module was functional and could transmit packets according to the protocols used. It was however noted from the ADC results that bit errors exist for transmit frequencies of 7.8125 MHz (250 Mb/s) and more. It is however critical to note that transferring data at this rate was only necessary for high instantaneous bandwidth readings, which is not the case for typical SuperDARN operation or most applications of HF receivers. The SuperDARN measures Doppler shifts of up to 100 Hz which requires far lower data rates. The CIC filter is capable of decimating the input signal to an approximate minimum data rate of 15 kHz (488 kb/s). No bit errors were measured at transmit rates of 3.9 MHz or less and therefore the Ethernet interface is deemed to be successfully implemented.

Performance related results were however difficult to obtain due to a bottle neck in on

the PC receiver-side of the communication link. The processing power of the computer limited the number of packets that could be recorded without dropped packets, for a given data rate. Results were obtained specific to the hardware that was in use at the time. Table 5.3 showed that data can be transferred at 15.525 MHz, corresponding to 496.8 Mb/s. In this case the PC's processing power limited the number of coherent packets that could be recorded without loss to approximately 20 packets. The data capturing capabilities of the PC varied significantly with the CPU usage of the PC.

6.6 FPGA Resource Usage

The resources used for the current implementation show that there is a large amount of resource space available for further development. The results are encouraging, as the SuperDARN allows for many features to be added to the current implementation.

It is however worth noting that the I/O buffer usage was not well managed in the final design implementation. *3.3 V CMOS* logic standards were used as often as possible, which resulted in a shortage of the corresponding buffers for certain FPGA banks. The solution was to change the logic standard of individual pins (in the case that the peripheral supported other logic levels) to *2.5 V CMOS*, for the design to synthesise.

Only 2% of the DSP48E slices are currently utilised for the CIC filter and for multiplication. The availability of DSP slices could support extensive DSP features in future applications.

7 Conclusions and Recommendations

The project is aimed at detailing a complete implementation of a listening mode for the TIGER-3 based transceiver box, provided by SANSA. The design task as whole was met with individual technical requirements adressed in the results given. The individual components as well as the system-wide implementation were detailed. Preceding the details of the design process, this report details the researched literature to support the design task implementation, and to support future development.

The listening mode allows for the control of the input gain followed by sampling, processing and transmission of the data to a personal computer. The input to output interfaces consist of the RF board and the TIGER-3. The FPGA modules written for this mode were designed to be as versatile as possible to support further development for current and alternate modes of operation.

This section details the conclusions drawn from the discussions in the previous section and provides recommendations for future work.

7.1 The Design Process

The system design was largely based on current SuperDARN operation, which requires base-banded IQ output data for an adjustable frequency range of up to 30 MHz. Through the design process the plan of development changed, eventually taking the form of the block diagram shown in Figure 4.1. Interfacing with the relevant peripherals proved a challenge on a hardware and software level. Modules were written to provide interfaces with peripherals and to perform specific tasks within the FPGA. Results and recommendations for these interfaces and tasks are discussed here.

7.1.1 Clock Management

In its current form, the clock manager simply generates the relevant clocking signals as required by the rest of the system. It currently generates two 125 MHz ODDR differential output, a 125 MHz clocking signal, and a 250 MHz PLL-synthesised clocking signal. The clock management module is shown in Figure 4.3. The technical requirements therefore were wholly met, with relevant clocking signals generated, improved and forwarded according to the needs of other modules.

Critical for future development involving phased array steering, is the functionality of clock synchronisation. It is important that all transceivers within an array share a common clock to minimise phase/timing errors. Each transceiver has two rotary switches (on the TIGER-3 board) which are used to provide each arrayed transceiver with a unique identity. If the clocks on each transceiver is in phase, further functionality within the NCO module can allow for phase offsets based on the transceiver box's identity.

Currently, a separate timing box provides a global clocking signal to all of the transceivers through the transceiver uLF interface. This clocking signal can be used to synchronise the transceiver on-board clock through feedback within the *PLL_ADV* component in Figure 4.3. Alternatively, the SuperDARN currently makes use of the DCM to switch between the internal and external clock as the system input. The fourth switch on the transceiver DIP switch is connected to specific FPGA pin that supports clock switching. It is thus used for the current SuperDARN implementation.

7.1.2 RF Board Interface

The RF signal captured by the antenna is variable in nature. The signal power is dependent on the backscatter properties of the ionosphere, which is in turn dependent on time of day and the state of the solar weather. Furthermore, interference signals from external sources may be captured by the antennas and transferred to the transceiver. The HF board provides filtering and gain control to compensate for the unpredictable power of the backscatter signal and the presence of non-HF interference signals.

An FPGA interface for the control of the external RF board was developed and its performance was confirmed to be reliable and accurate. The low-pass filtering of the incoming signal is a passive feature of the RF board, and proved effective in removing unwanted high frequency components. It is however worth noting that the the primary hardware filter was not present in the current hardware. Two active features are provided for the receiver chain: ADC/DAC switching from the SMA port and variable step attenuation. Both features were tested and performed well, with switching times of approximately 0.2 μs.

Attenuation control is a critical component of the receiver chain, as it is tasked with ensuring that the full dynamic range of the ADC is utilised. With variable attenuation, the total gain is adjustable in 1dB steps from 23dB to 51dB as per Figure 4.4.

The technical requirements for this project were therefore met by demonstrating gain control and switching. Future work could implement automatic gain control for the receiver. The ADC readings can be analysed during run-time and the gain can be changed based on the maximum amplitude. The gain would then be communicated to the personal computer through the Ethernet interface along with the receive signal data.

7.1.3 Digital Signal Processing

The results obtained for each DSP design element were compared to simulations, taking all digital operations into account. The results show the functionality of each DSP module individually and as a complete DSP chain. Furthermore, the results show that each module's operation can be adjusted by pre-synthesis parameters or variable input lines.

Below, conclusions are drawn and recommendations made for each DSP module in terms of design.

The NCO module was implemented in its most basic form, and performance correlated well with simulations. The module supports variable demodulation frequencies. Given the driving clock frequency of 125 MHz, the required tuning range of 8 MHz to 17 MHz is well within the capabilities of the NCO. Improvements could be considered in the form of phase dithering and Taylor series implementations of the core. The core's parameters can be changed to improve resource consumption if necessary. Additionally, more attention can be given to the input/output data widths of the NCOs, multipliers and the CIC filter, to further optimise resource usage.

The filter design offered a larger range of implementation options. The first decision that was made in the design process was to use dual channel operation for the I and Q channels. The advantage of dual channel operation is a decrease in resource usage, as apposed to generating a separate core for each channel. The disadvantage is that the module complexity was increased in terms of clocking and data handling.

It was shown that the CIC filter provides a resource efficient digital low-pass filter, as apposed to conventional FIR filters. The filter's magnitude response is however limited to that of a *sinc* waveform (see Figure 5.18 and Equation 3.52).

To meet the technical requirements, the filter must be adjustable in bandwidth and maximise the suppression of unwanted noise components. These goals were achieved in full. All the narrow band results show that noise levels are suppressed to well below -75 dB of the input signal. The maximum noise amplitude was recorded in Figure 5.16. A bottleneck in PC Ethernet receive stage limited the testing of pass bands wider that 300 kHz, which was shown to be achievable in Figure 5.21.

In order to create a more constant pass-band with a higher roll-off, a compensation filter can be implemented. More details on the compensation filter can be found in the CIC filter data sheet [41]. The testing of the compensation filter in PC post-processing or implemented on the FPGA, is left for future work. Furthermore, attention can be given to the choice of parameters to attempt to introduce filter response nulls at strategic positions in the spectrum. Care must be given to the aliased left sideband when considering the placement of filter nulls.

7.1.4 Ethernet Communications

The implementation of the Ethernet core unexpectedly proved the biggest challenge of the design process. Exact clocking schemes (see Figure 4.11) were difficult to debug and writing a module to generate the packets in the correct format proved a challenge to design efficiently. The storage of data in the FIFO is integral to dividing clock and data-width

domains, and proved very effective in doing so. The packet generator module allows for 48 bits of the Ethernet packet data to be reserved for debugging or other information relating to the packet. Uses may include:

- A packet counter helps the PC to identify missing packets.
- If an automatic gain control module is introduced; the gain can be communicated to the computer.
- In a network, the transceiver ID can be communicated with to server.
- The demodulating frequency can be indicated for the transmitted set of data, etc.

Future work can look to implement the receiver side of the client-MAC interface, which will allow for a host of additional functionality. The PC could communicate parameters such as the demodulating frequency, the decimation factor and the gain control parameters, eliminating the need to use external switch and push-button periphery - as in the current implementation.

If the bottleneck at the PC side can improved, the speed limitations of the communication link can be optimised for large instant bandwidth operation. Jumbo frames can further aid the data rate of the communication link. Jumbo frames were enabled for the current FPGA MAC implementation, but the packets could not be collected on the PC side for an unknown reason. Debugging the PC networking was out of the scope of the project. The data could not be captured by the Python script unless Wireshark was used to force the NIC into promiscuous mode. Along with developing a PC to FPGA communication link, the development of an improved PC interface would add great value to the project for future applications.

The conclusion is therefore that the technical requirements were not met in its entirety due to PC-side constraints. Although it was shown that the Ethernet protocol was implemented successfully, the maximum data rate could not accurately be confirmed and jumbo frames was never successfully implemented. Strictly, the requirements for the supporting PC-side software were met but the bottleneck could not be solved which resulted in the shortfall of some of the other technical requirements for this project.

7.2 Input Sampling, System Noise and Interference

The ADC interface was designed to collect data from the peripheral at a maximum frequency of 130 MHz (125 MHz for the current implementation) according to the hardware constraints of the ADC. Simulations were used to predict the effects of quantisation on the ADC readings, but could not account for the spurious noise recorded in the results (see Figure 5.11). The system noise and interference was analysed and three sources were identified:

- Noise from the ADC peripheral.
- Sampled interference, from the analogue electronics before digitisation.
- Spurs introduced by the digital signal processing.

All known sources of noise and interference were discussed in Section 6.3.5. The noise generated by the ADC was measured to occupy approximately a peak-to-peak ADC input voltage of 0.8 mV. A corresponding DC offset of approximately 0.2 mV was measured. Gaussian white noise was measured from the ADC and therefore it appears as a "noise floor" in the output spectrum. Considering the magnitude of the interference from the RF board, the noise from the ADC can be neglected. From the results it is concluded that the ADC technical requirements met.

Considering the noise from the RF interface, significant interference was measured with an ADC input peak-to-peak voltage level of 400 mV. An input signal at the SMA connector would undergo 27 dB of gain and therefore the RF interference corresponds to a peak-to-peak SMA input voltage of 17.5 mV. The interference appeared to be band-limited to approximately 100 kHz centred at 350 kHz. It is expected that the interference will be sufficiently attenuated by the digital filter after base-banding.

Finally, the processing noise from the DSP was discussed. The typical sources of DSP noise comes in the form of quatisation and aliasing. The noise introduced by the NCO was due to the LUT size, which is a pre-synthesis parameter that has a substantial effect on the resources used by the FPGA. The parameters chosen for the NCO core were as follows: the phase bit-width was set to $B_\theta = 32$, and the output bit-width was set to $B_\Theta = 9$, which results in an output frequency resolution of 0.029 Hz and a spurious free spectral magnitude range of -130 dB (see Figure5.4). The full NCO implementation only made use of one BRAM (18k) component and no DSP slices. Future implementations should make use of DSP slices by adding Taylor series correction noise shaping to the NCO core, and increasing the output bit-width at the cost of additional BRAM usage.

The base-banding process results the suppressed left sideband to alias into the output. The maximum magnitude of the sideband at the output is dependent on the filter's decimation rate. At high decimation rates, the image is likely to appear in highly attenuated side lobes and is unlikely to be identifiable in the output.

By analysing the full ADC output spectrum (Figures 5.16 and 5.17) it is clear that additional noise is introduced by the ADC or related analogue peripherals. For all measured cases, these noise spurs occur below -75 dB of the peak, are symmetrical and appear relatively far from the sampled tone frequency. This concludes all technical requirements for the system's noise.

7.3 FPGA Resource Usage

A complete list of resource usage and availability is given in Table 5.4. It was noted in Section 6.6 that the resources could have been managed better. The basic requirements were met, which is shown by the system's functionality and the majority of the obtained results. It is however recommended that future work make better use of the available resources.

7.4 Post-Listening Mode

The current implementation of the listening mode sets the basis for substantial future development. Other than improving on current architecture, which was discussed above, there are many additional features that could be implemented to not only bring the system closer to SuperDARN operation, but to open the possibility of a host of secondary applications such as sea state monitoring or over the horizon communication.

The obvious first step would be the development of the RF transmit chain interface. The current NCO module allows for agile frequency control and (although not within the scope of this work) has been configured to generate FMCW chirp signals. Formalising the transmit chain would be a big step towards the realisation of complete radar modes of operation. The development of a new amplifier front-end would allow for high power transmission to an antenna, and allow for field tests. This would prove valuable for confirming the operational limitations of the transceiver. Additionally, the development of the Ethernet FPGA receiver interface would allow for a host of features to be realisable.

Another peripheral feature of the TIGER-3 that was not implemented, is the use of flash memory to store information such as filter coefficients or bitstreams for the FPGA to use. Ethernet can be used to program the FPGA remotely by transferring and storing the bitstream in flash memory. For remote applications, this feature could prove invaluable.

In conclusion, this work set out to develop a base level for future projects by the implementation of a listening mode. This was realised by providing a detailed literature review to aid future development, and by meeting the design tasks set forth by UCT and SANSA.

References

[1] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar*. Scitech, 2010.

[2] R. A. Greenwald *et al.*, "Darn/SuperDARN," *Space Science Reviews*, vol. 71, no. 1-4, pp. 761–796, 1995.

[3] *ADC12J4000 12-Bit, 4-GSPS ADC with Integrated DDC*, Texas Instruments, 2014.

[4] I. Haggstrom, "Incoherent and Coherent Scatter Radars," January 2001.

[5] G. Chisham *et al.*, "A Decade of the Super Dual Auroral Radar Network (SuperDARN): Scientific Achievements, New Techniques and Future Directions," *Surveys in Geophysics*, vol. 28, no. 1, pp. 33–109, 2007.

[6] "VT SuperDARN Home : Virginia Tech SuperDARN," http://vt.superdarn.org/tiki-index.php, accessed: 15 August 2019.

[7] B. G. Fejer and M. Kelley, "Ionospheric Irregularities," *Reviews of Geophysics*, vol. 18, no. 2, pp. 401–454, 1980.

[8] M. Parkinson, J. Whittington, and J. Devlin, "Why do we Need "DigiDARN" - A Global Network of Digital SuperDARN Radars?" January 2006.

[9] E. Custovic *et al.*, "New Antenna Layout for a SuperDARN HF Radar," *Radio Science*, vol. 48, no. 6, pp. 722–728, November 2013.

[10] J. C. Devlin *et al.*, "Buckland Park Radar Overview (2013 SuperDARN Workshop)," 2013.

[11] P. J. Cilliers *et al.*, "Calibration of a SuperDARN Radar Antenna by Means of a Satellite Beacon on a CubeSat," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, July 2013, pp. 1290–1293.

[12] T. Dusterwald, "Implementation of a Low Cost Demonstrator Riometer on a Flexible FPGA Backend: The First Steps in Adding a Riometer Mode to the Digital SuperDARN Radar at SANAE IV," Master's thesis, University of Cape Town, 2018.

[13] T. Barris, "Modelling and Resolving of the Ambiguous Angle of Arrival Measurements of the SANAE IV SuperDARN Radar," Master's thesis, University of Cape Town, 2017.

[14] J. Burger, "High Frequency Surface Wave Radar Demonstrator," Master's thesis, University of Cape Town, 2018.

[15] J. S. Whittington and J. C. Devlin, "Architecture for a "Fully Digital" Implementation of a TIGER Radar (2015 SuperDARN Workshop)," June 2015.

[16] M. Ruohoniemi, "SuperDARN Transitions (2013 SuperDARN Workshop Presentation)," May 2013.

[17] *Virtex-5 FPGA Configuration User Guide*, Xilinx, May 2017.

[18] J. M. Headrick and J. F. Thomason, "Naval Applications of High Frequency Over-the-Horizon Radar," *Naval Engineers Journal*, vol. 108, no. 3, pp. 353–362, 1996.

[19] L. Mark, "The Super Dual Auroral Radar Network (SuperDARN): An Overview of its Development and Science," *Advances In Polar Science*, vol. 24, no. 1, pp. 1–11, July 2014.

[20] M. I. Skolnik, *Radar Handbook*. McGraw-Hill, 2008.

[21] R. J. Goldston and P. H. Rutherford, *Introduction to Plasma Physics*. CRC Press, 2018.

[22] J. D. Huba, "NRL Plasma Formulary," Naval Research Lab Washington DC Plasma Physics Div, Tech. Rep., 2006.

[23] P. Dyson *et al.*, "The TIGER Radar: An Extension of SuperDARN to Sub-Auroral Latitudes," *Workshop on Applications of Radio Science Proceedings*, pp. 9–31, January 2000.

[24] S. Eriksson, "Global Magnetospheric Plasma Convection," 2019.

[25] D. Bilitza, "International Reference Ionosphere 2000," *Radio Science*, vol. 36, no. 2, pp. 261–275, March 2001.

[26] G. Chisham, T. Yeoman, and G. J. Sofko, "Mapping Ionospheric Backscatter Measured by the SuperDARN HF Radars - Part 1: A New Empirical Virtual Height Model," *Annales Geophysicae*, vol. 26, pp. 823–841, May 2008.

[27] J. Stepherson, "Observing Geospace Above Antarctica with SuperDARN (2019 SANSA Space Weather Camp)," 2019.

[28] J. S. Whittington, "Digital TIGER Radar Receiver - an Outline of Possible Architectures," May 2002.

[29] P. Minev and V. Stoianova Kukenska, "The Virtex-5 Routing and Logic Architecture," pp. 14–17, January 2009.

[30] *Vivado Design Suite 7 Series FPGA Libraries Guide*, Xilinx, 2012.

[31] J. Serrano, "Introduction to FPGA Design," 2008.

[32] *Virtex-5 Libraries Guide for HDL Designs*, Xilinx, 2010.

[33] P. L. Dyson *et al.*, "The Tasman International Geospace Environment Radar (TIGER) - Current Development and Future Plans," in *2003 Proceedings of the International Conference on Radar (IEEE Cat. No. 03EX695)*. IEEE, 2003, pp. 282–287.

[34] H. Nguyen *et al.*, "Timing Control and Radar Server for the TIGER 3 Radar (2010 SuperDARN Workshop Poster)," May 2010.

[35] A. Wilkinson, "Master's Course Notes on Radar/Sonar Signal Processing: Fundamentals, University of Cape Town," 2017.

[36] F. Nicolls, "Masters Course Notes on Discrete Signal Processing, University of Cape Town," 2019.

[37] R. A. Greenwald, "Multipulse Sequences and SuperDARN (2018 SuperDARN Workshop Presentation)," May 2018.

[38] A. Wilkinson, "Signals and Systems 2 Course Notes, University of Cape Town," 2017.

[39] K. McWilliams, "SuperDARN Pulse Sequences - Optimization and Testing (2003 SuperDARN Workshop Presentation)," May 2003.

[40] D. T. Farley, "Multiple-Pulse Incoherent-Scatter Correlation Function Measurements," *Radio Science*, vol. 7, no. 6, pp. 661–666, June 1972.

[41] *LogiCORE IP CIC Compiler v2.0*, Xilinx, 2011.

[42] E. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155–162, 1981.

[43] "IEEE Standard for Ethernet," *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)*, pp. 1–4017, March 2016.

[44] J. Postel *et al.*, "RFC 791: Internet Protocol," 1981.

[45] ——, "RFC 768: User Datagram Protocol," 1980.

[46] *Virtex-5 FPGA User Guide*, Xilinx, 2010.

[47] *Virtex-5 FPGA Data Sheet: DC and Switching Characteristics*, Xilinx, 2016.

[48] *Constraints Guide*, Xilinx, 2009.

[49] *Virtex-5 FPGA XtremeDSP Design Considerations*, Xilinx, 2017.

[50] *Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC User Guide*, Xilinx, 2011.

[51] *LTC2208: 16-Bit, 130Msps ADC*, Linear Technology, 2005.

[52] *LogiCORE IP DDS Compiler v4.0*, Xilinx, 2011.

[53] *LogiCORE IP Multiplier v11.2*, Xilinx, 2011.

[54] A. Eric Bogatin. Designcon 2014 - Essentials of Jitter. Teledyne Lecroy. [Online]. Available: http://cdn.teledynelecroy.com/files/whitepapers/designcon2014-essentialsofjittertutorial.ppsx

Appendices

A Propogation

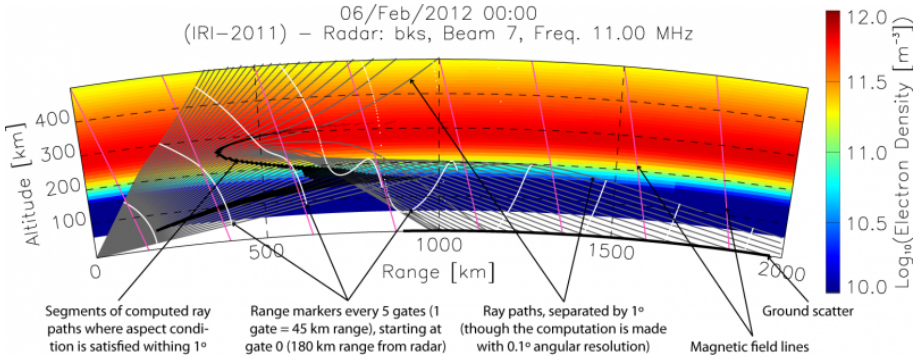

Figure A.1: SuperDARN ray path simulation with labels.

B Transmission Waveforms

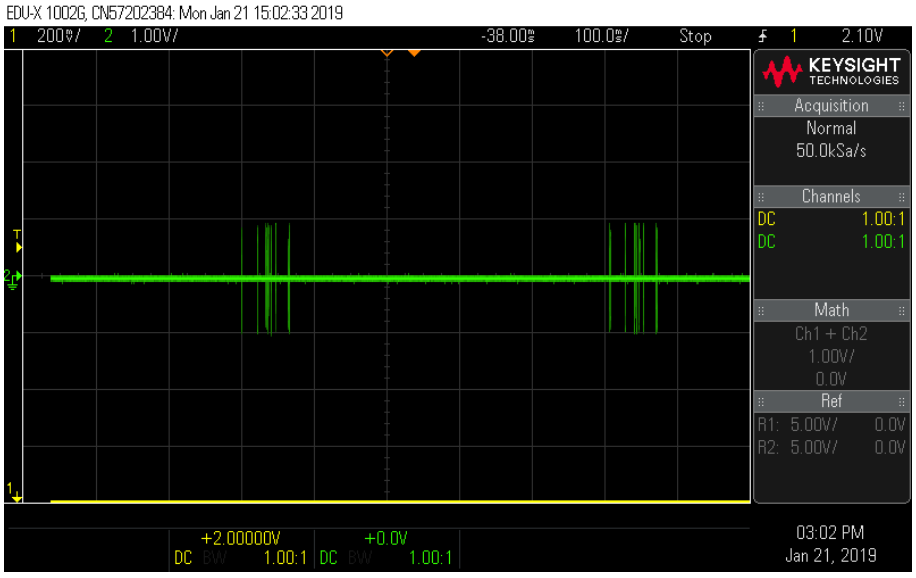

Figure B.1: SuperDARN mode of operation transmitted sequence (two sequences).

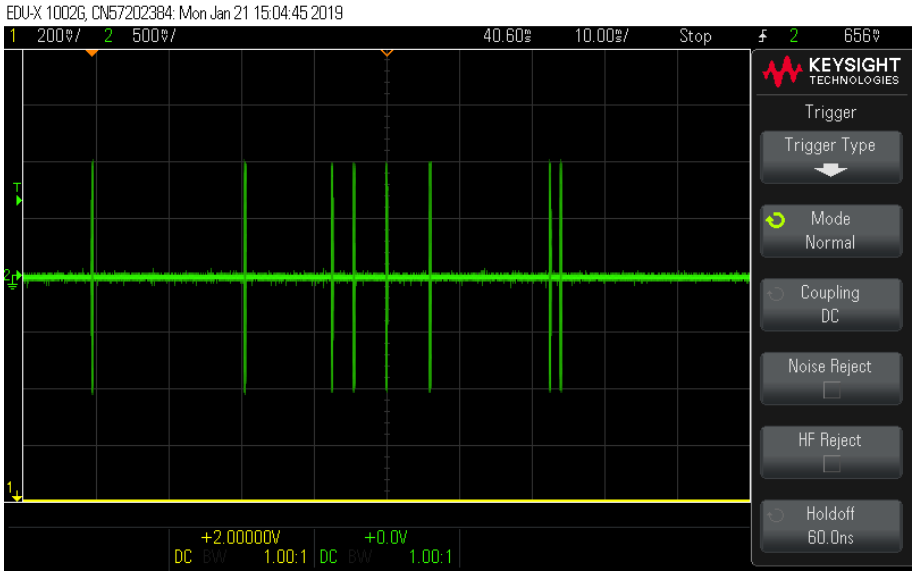

Figure B.2: SuperDARN mode of operation transmitted sequence (single sequences).

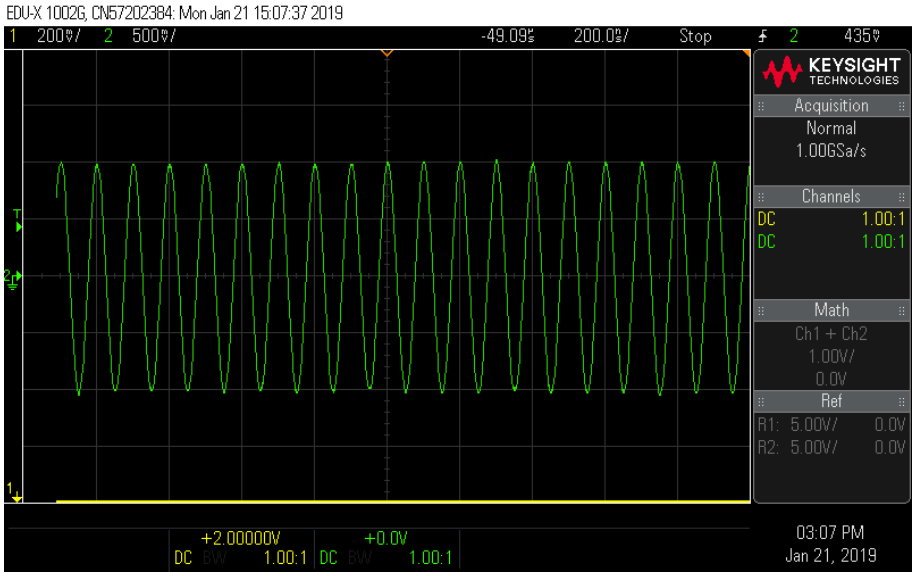

Figure B.3: SuperDARN mode of operation transmitted sequence (sectional pulse).

C TIGER-3 Schematic Diagrams

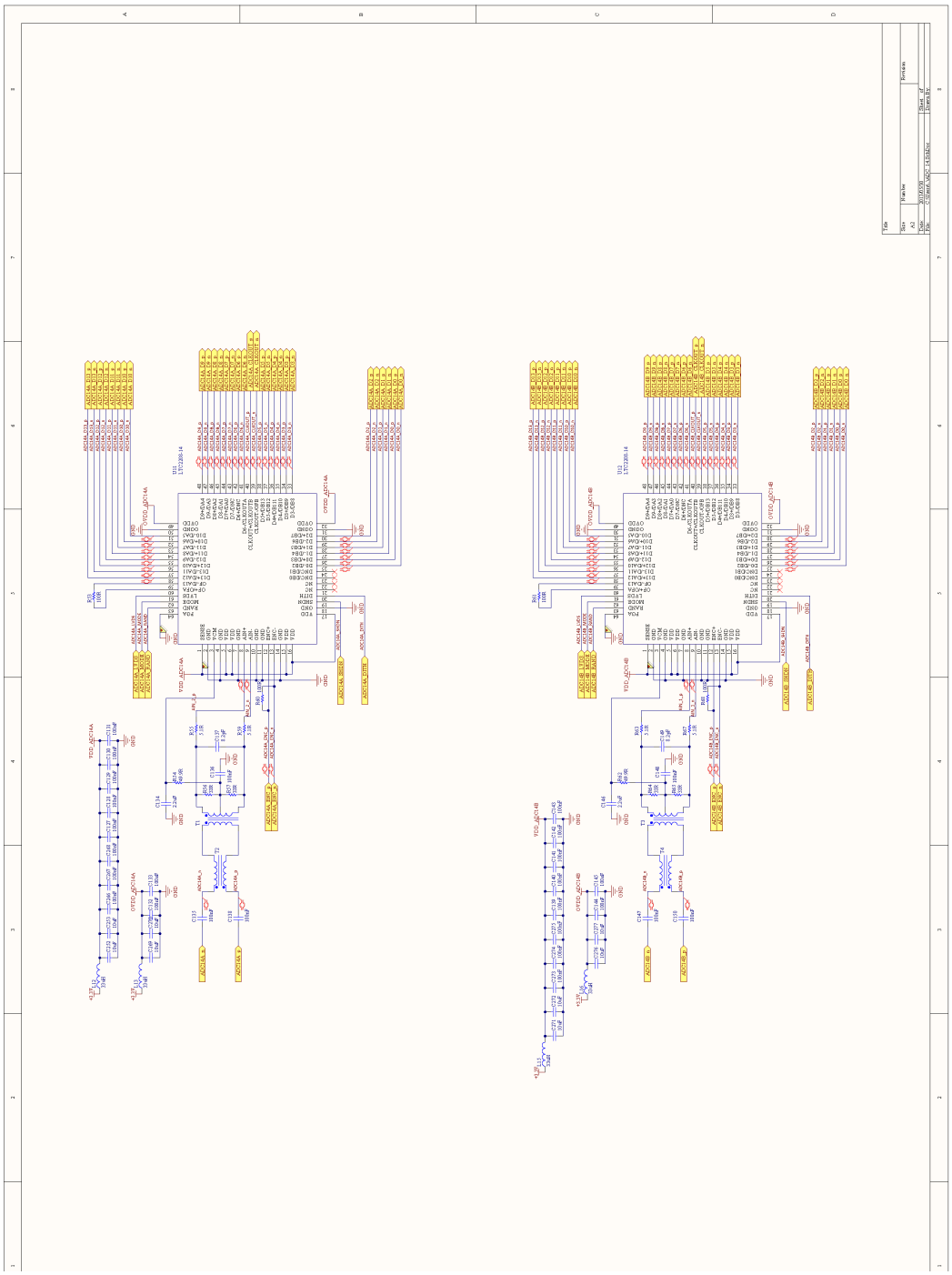

Figure C.1: ADC14A/B schematic.

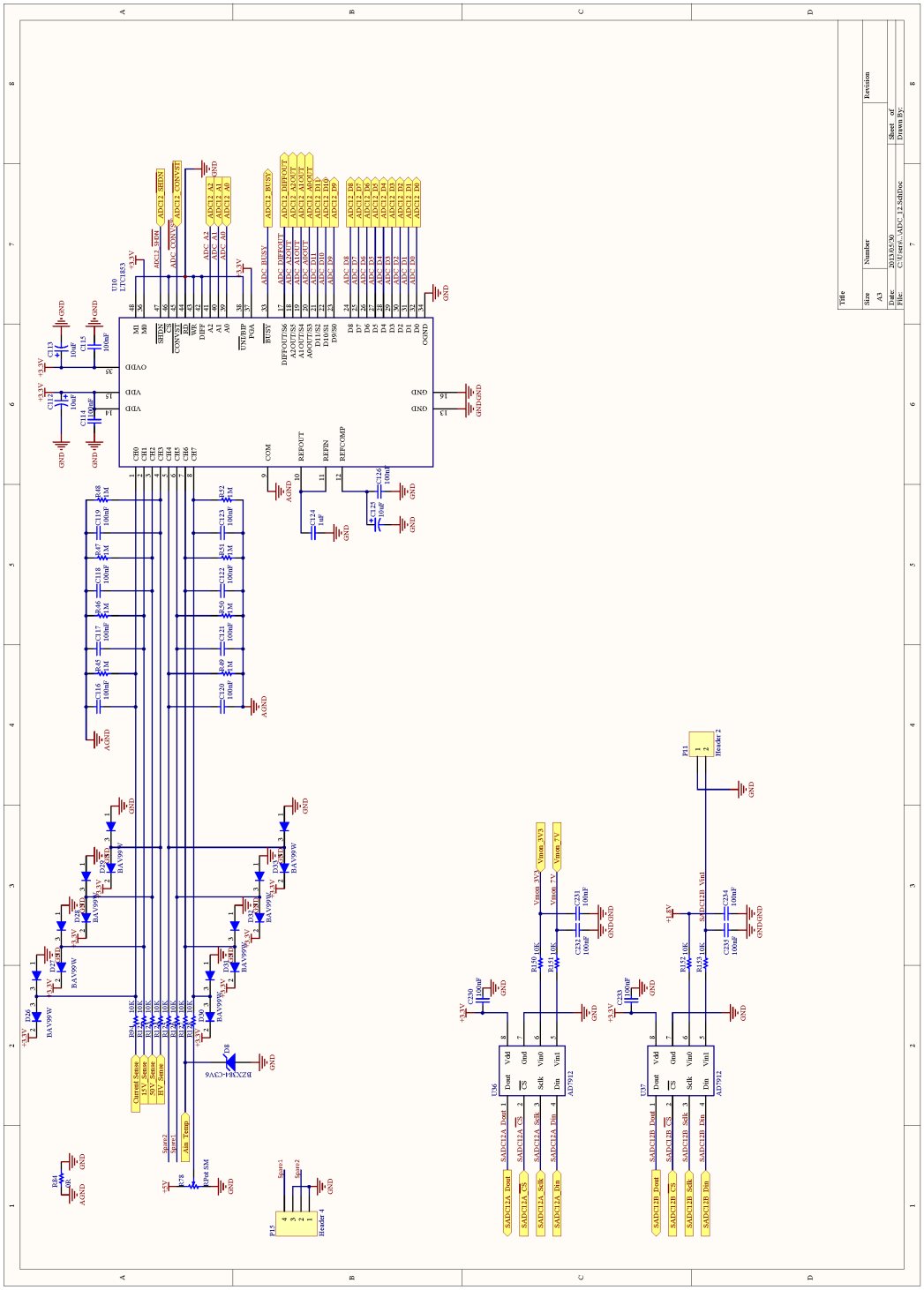

Figure C.2: ADC12 schematic.

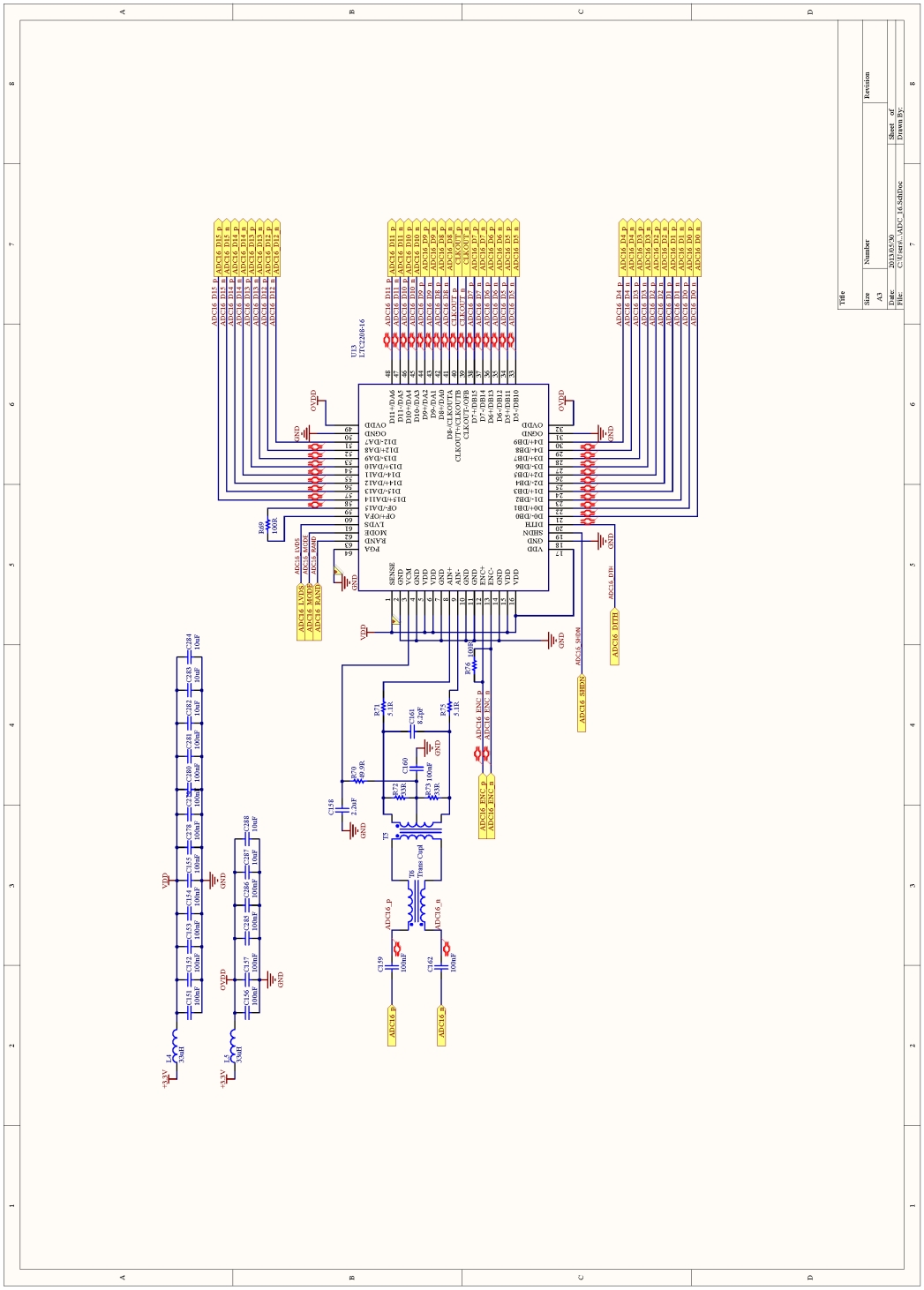

Figure C.3: ADC16 schematic.

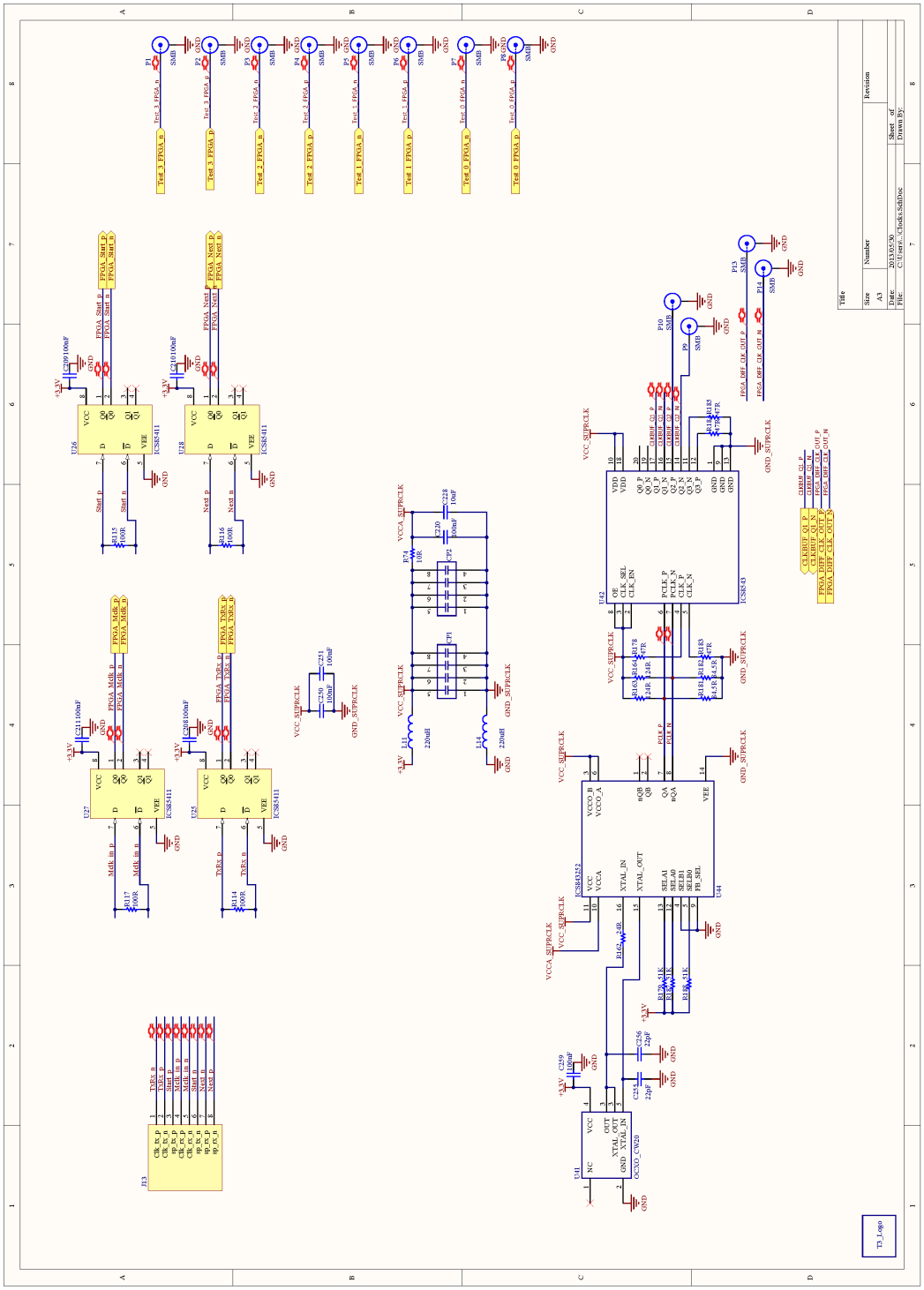

Figure C.4: Clocking schematic.

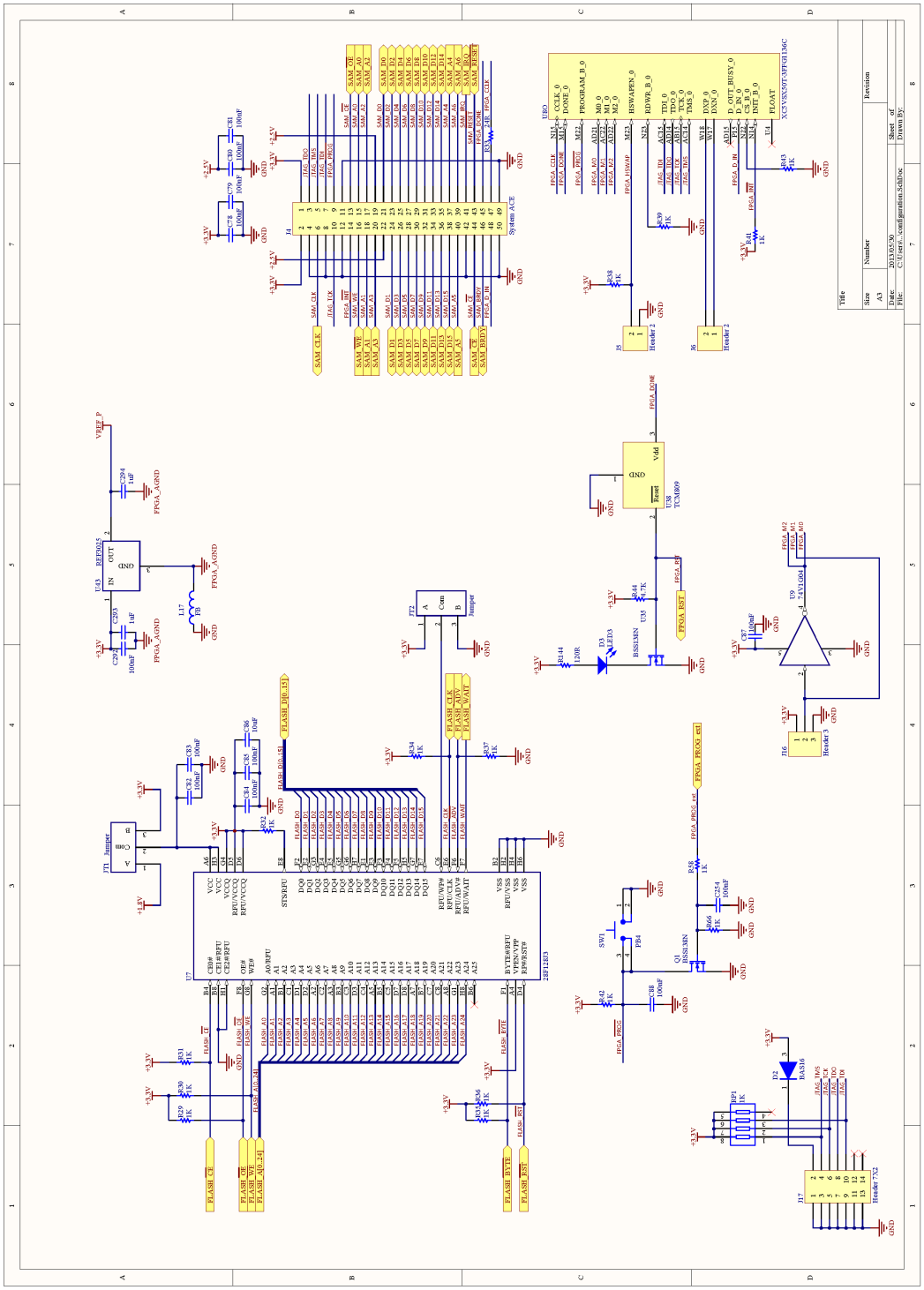

Figure C.5: Flash memory schematic.

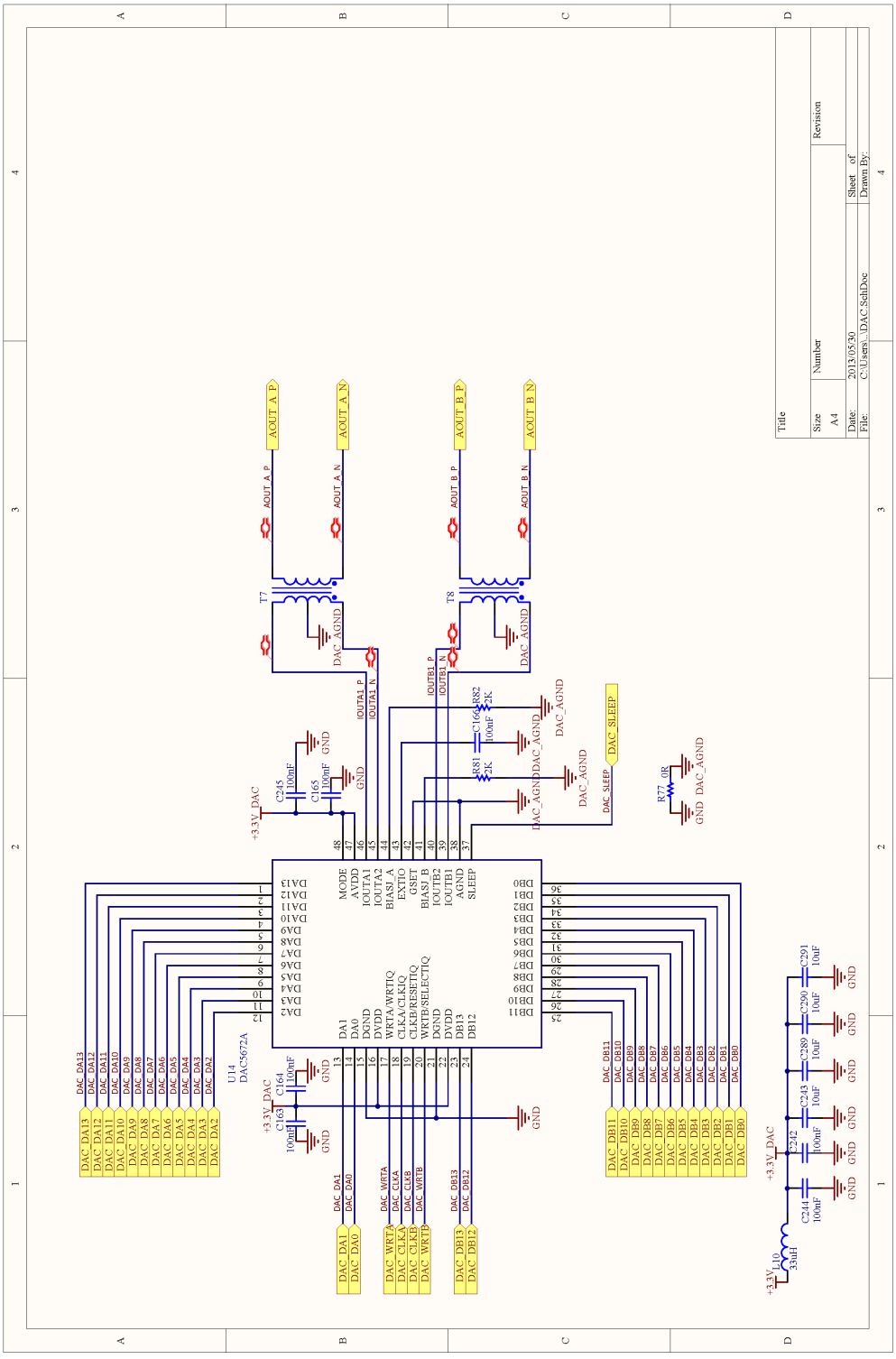

Figure C.6: DAC schematic.

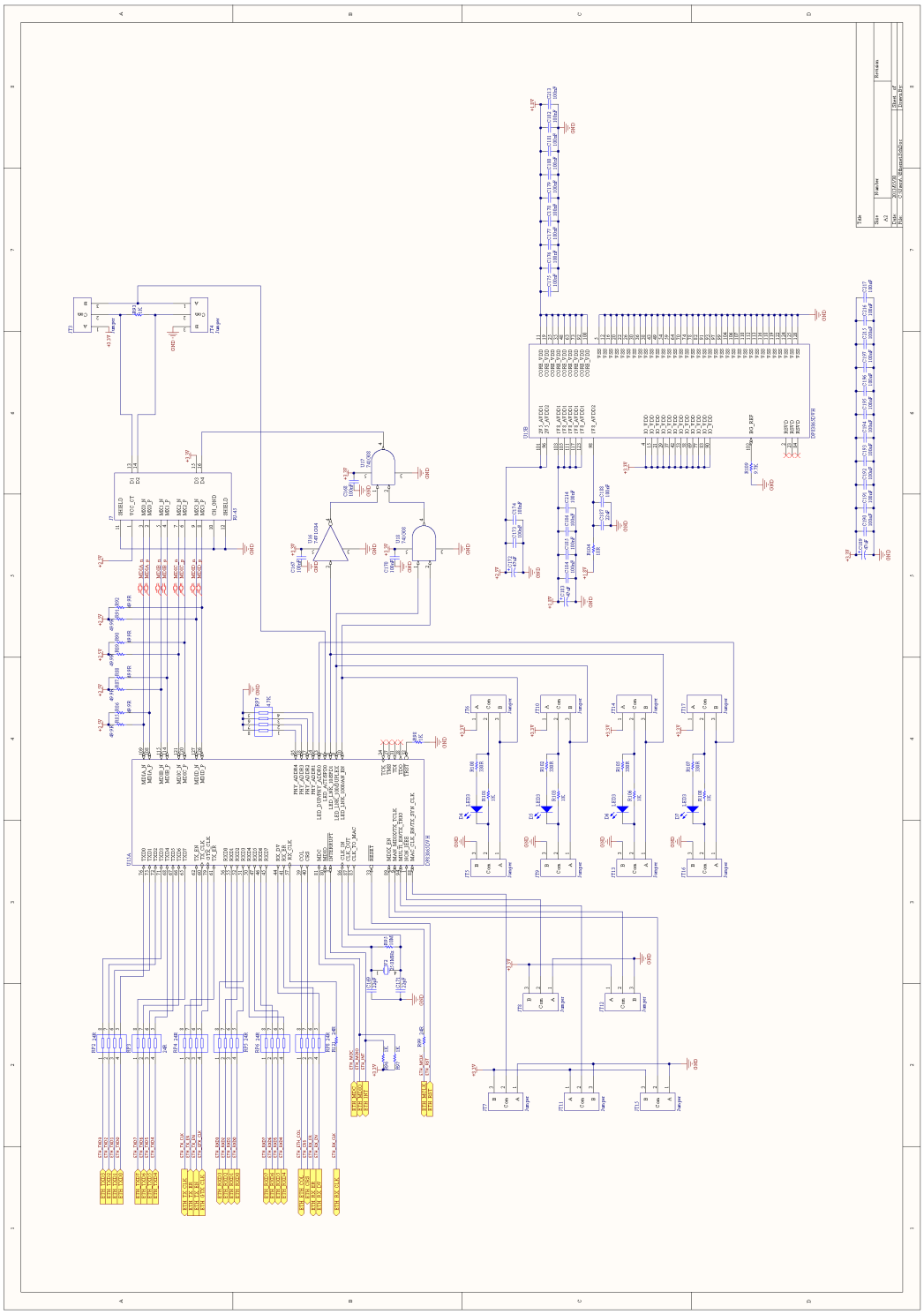

Figure C.7: Ethernet PHY schematic.

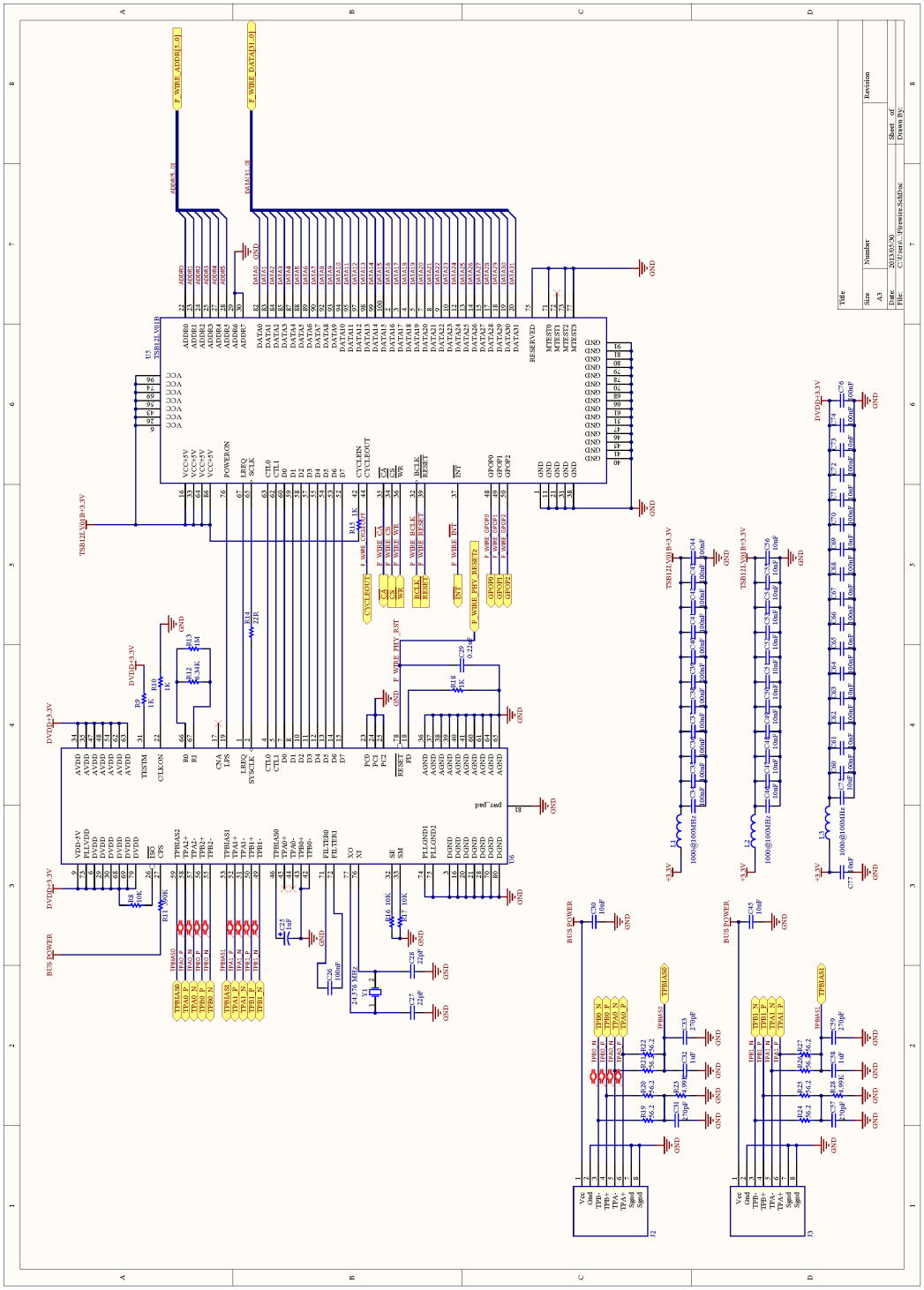

Figure C.8: Firewire schematic.

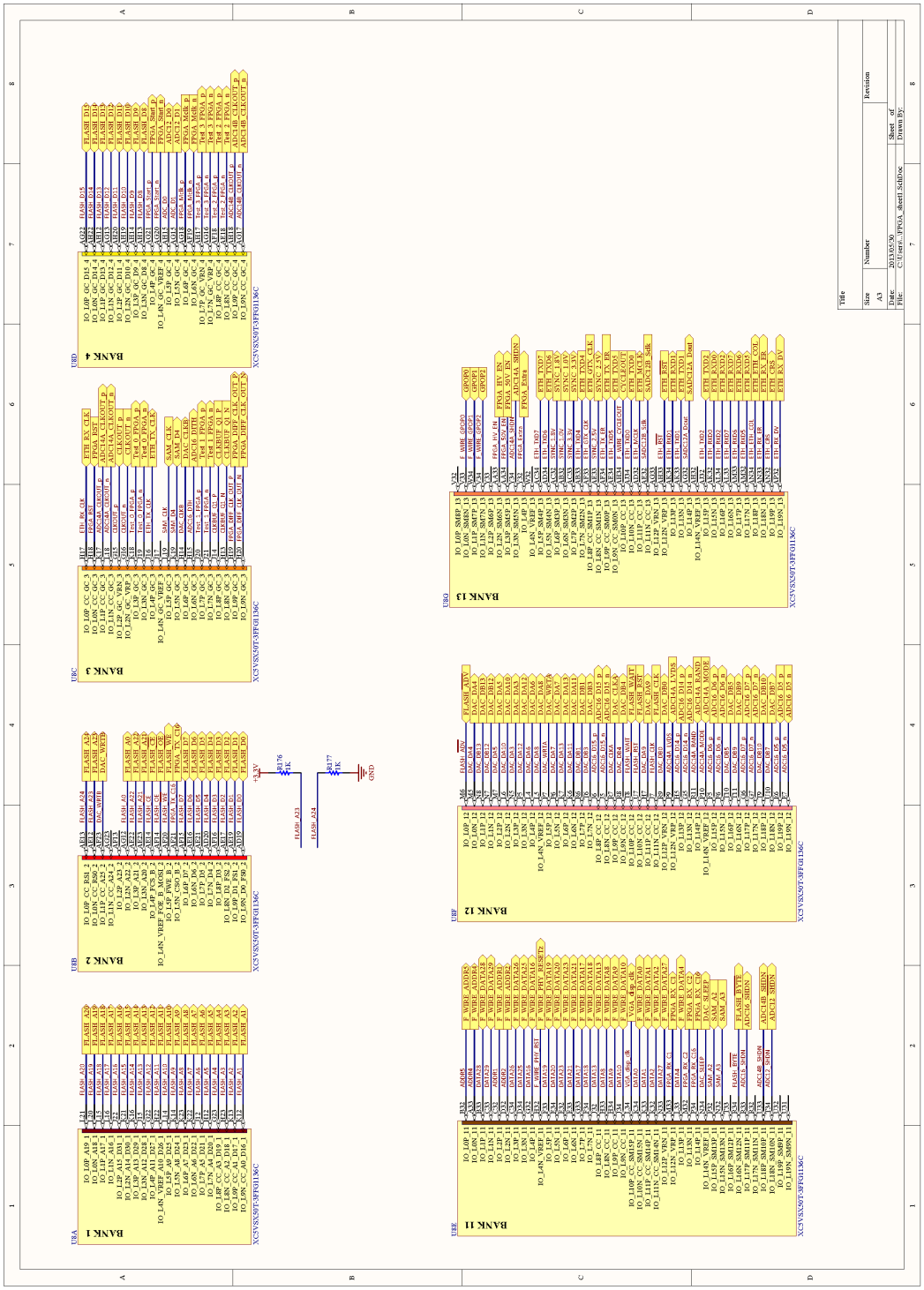

Figure C.9: FPGA banks - part 1.

Figure C.10: FPGA banks - part 2.

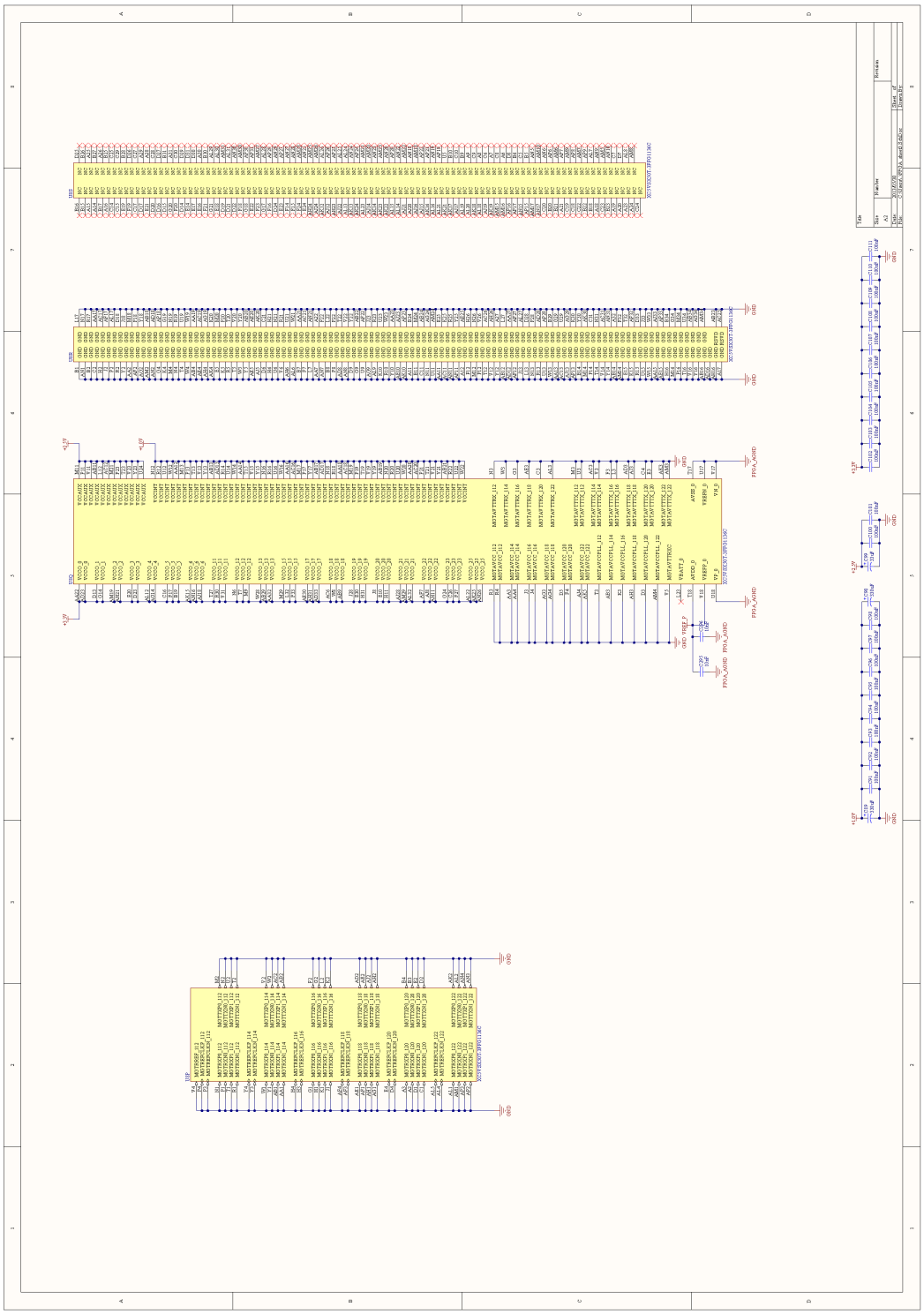

Figure C.11: FPGA banks - part 3.

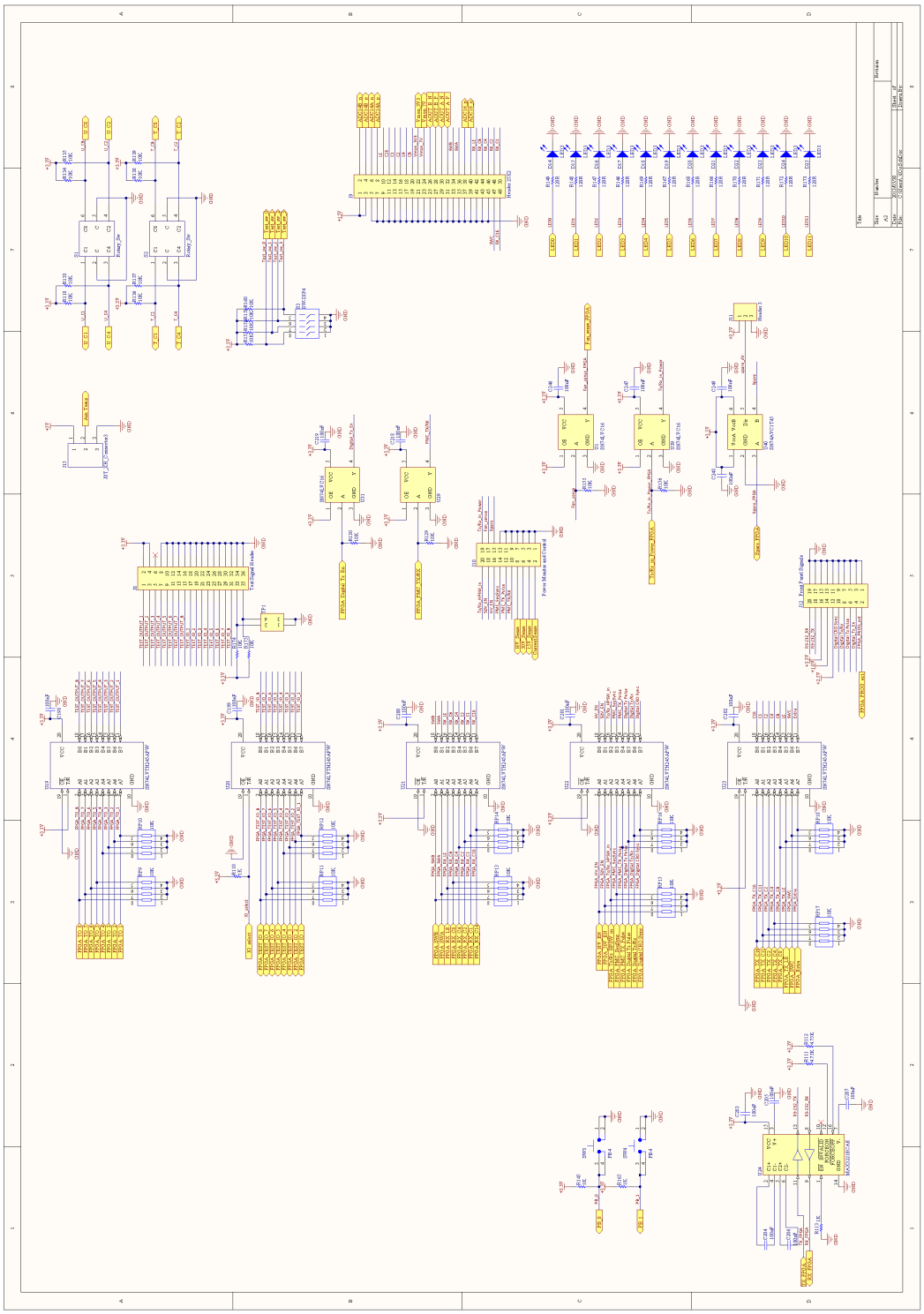

Figure C.12: I/O schematic.

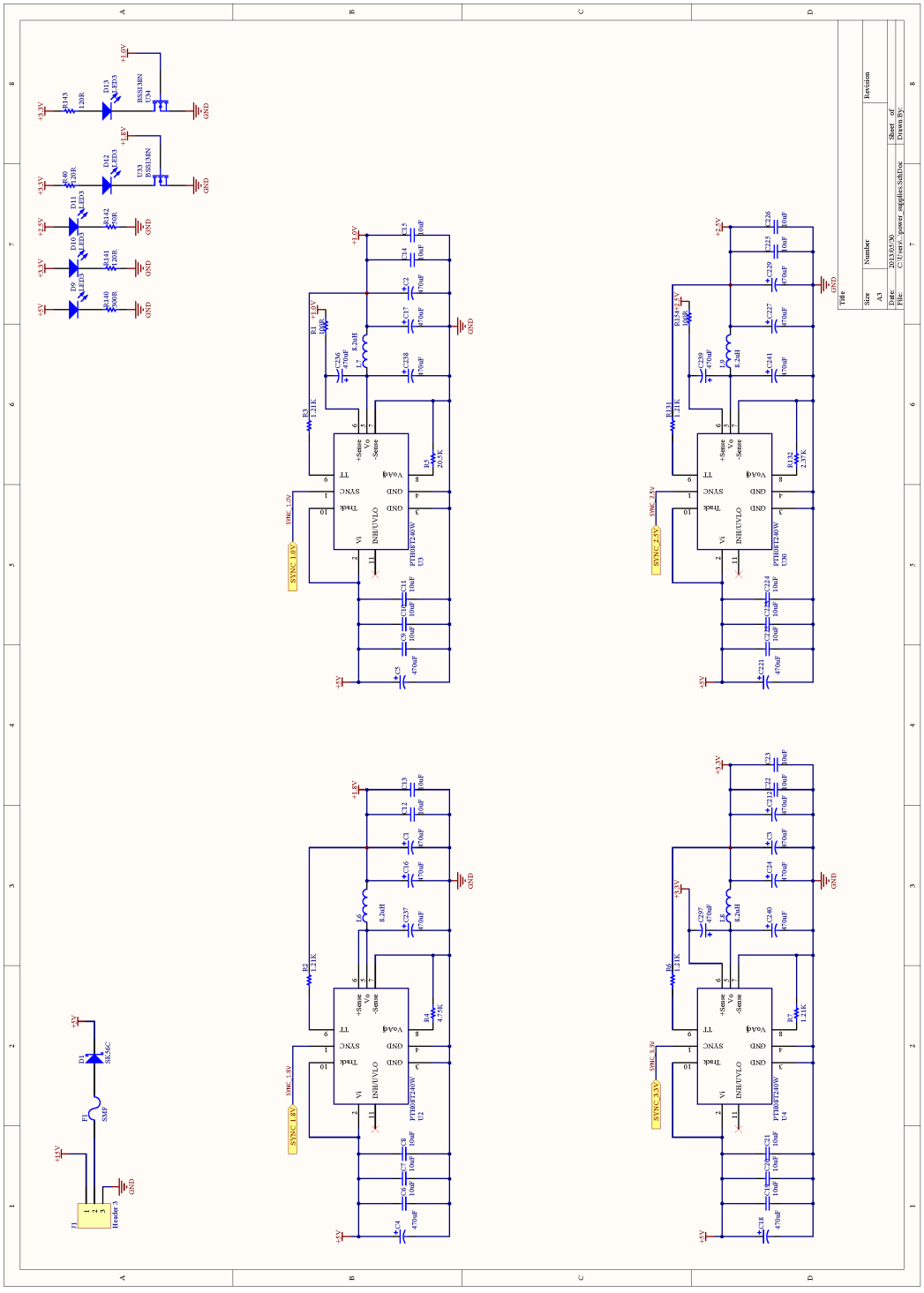

Figure C.13: Power supply schematic.

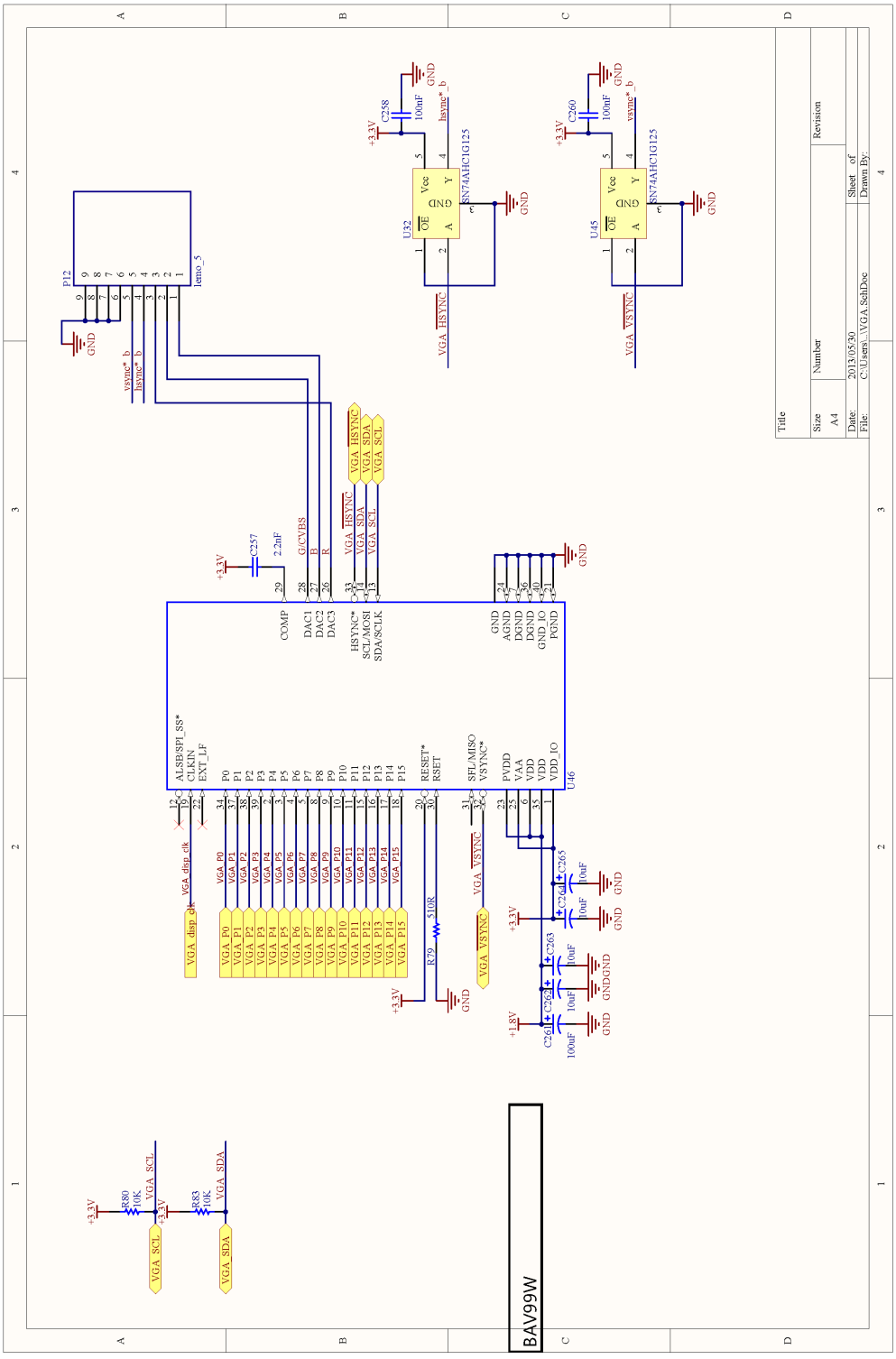

Figure C.14: VGA schematic.

D RF Board Schematic Diagrams

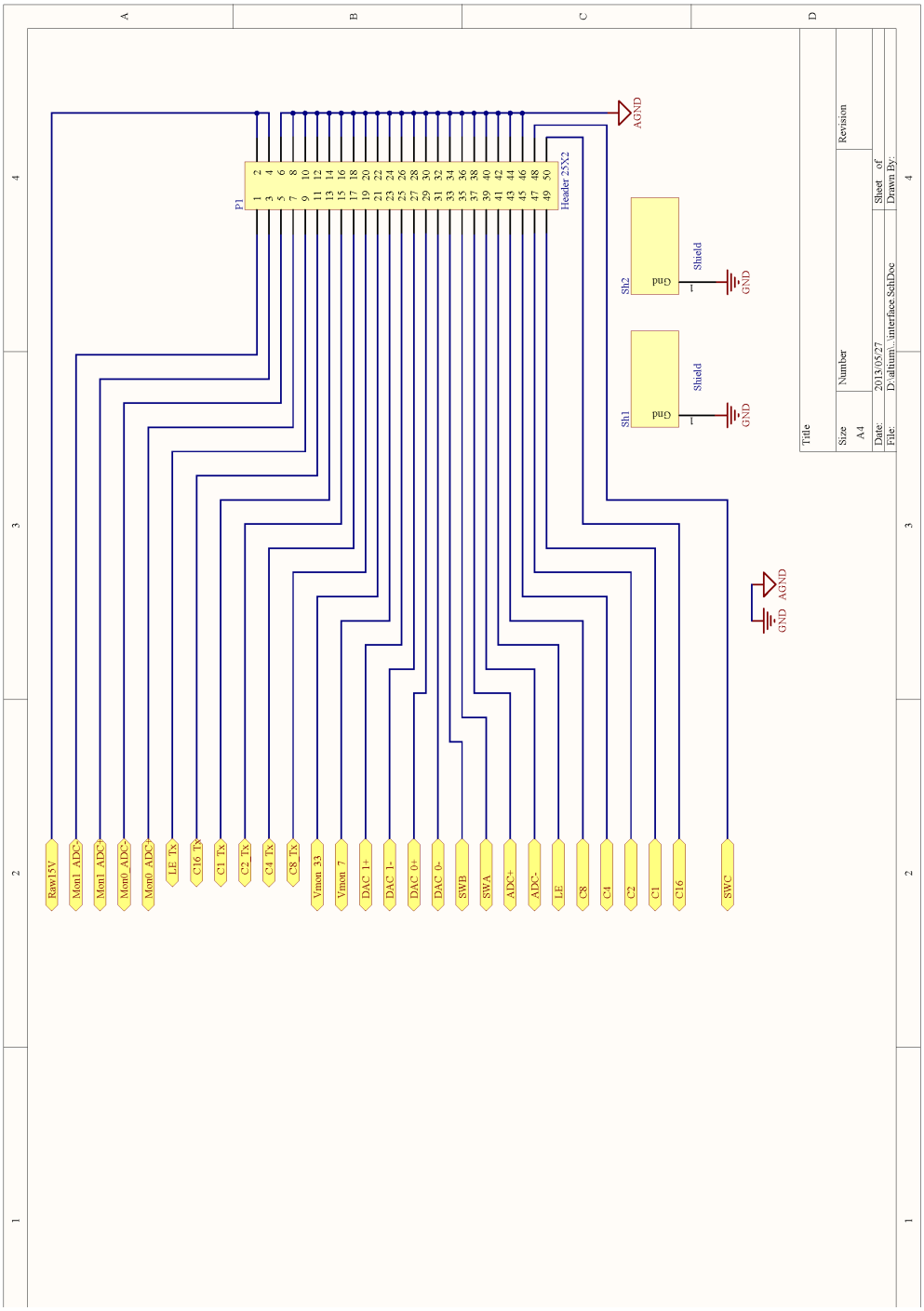

Figure D.1: RF to FPGA header schematic.

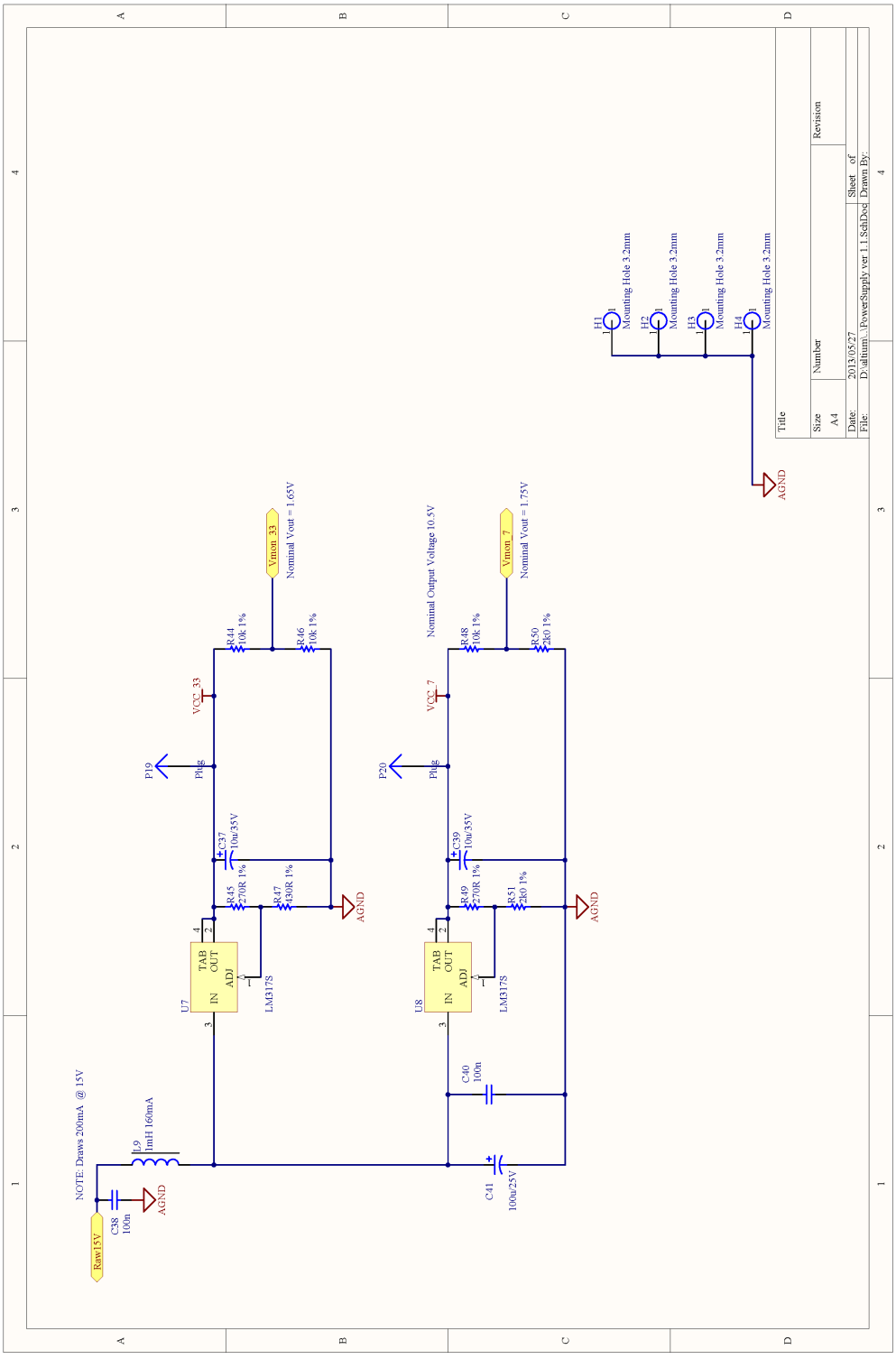

Figure D.2: RF power supply schematic.

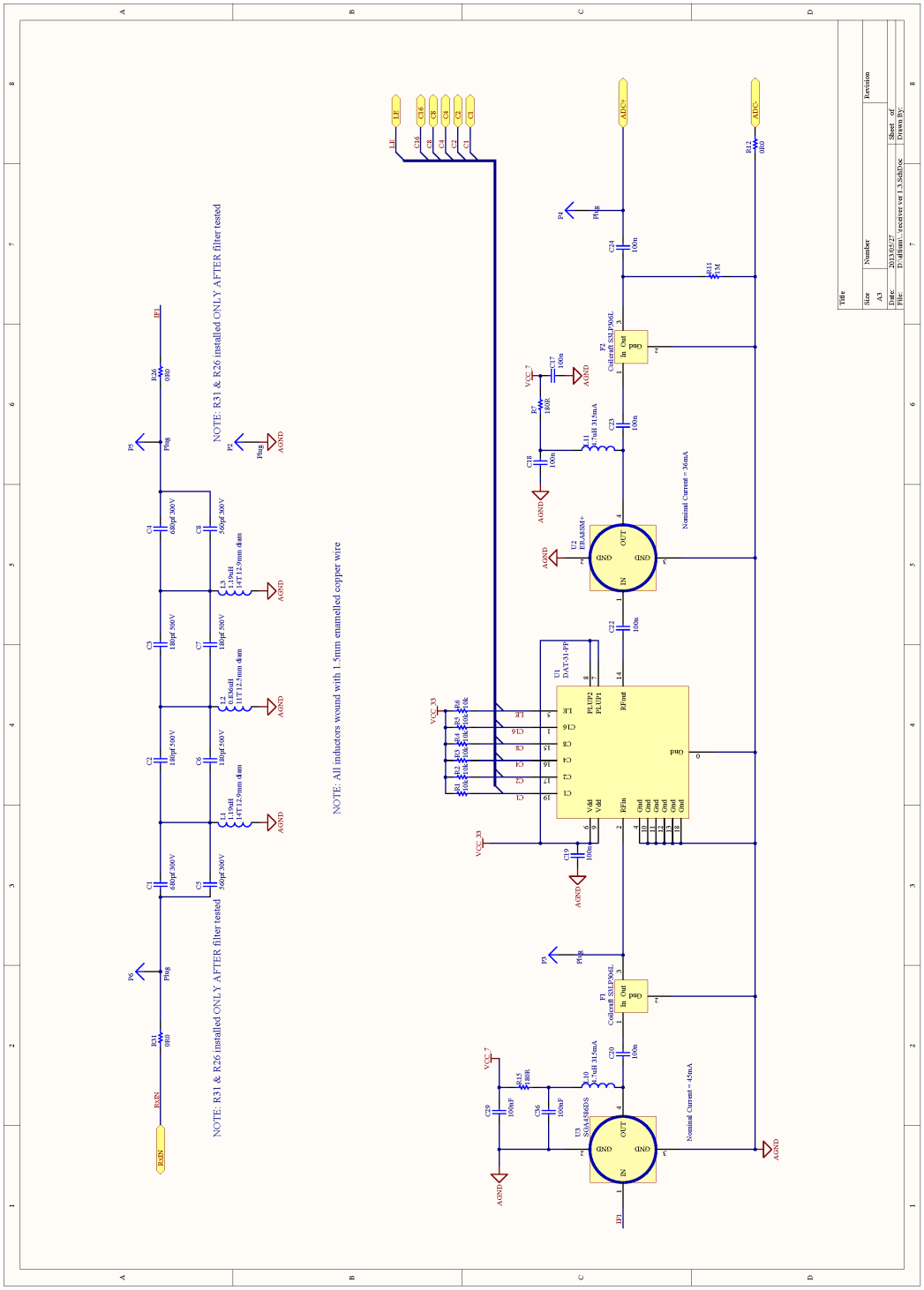

Figure D.3: RF receive chain schematic.

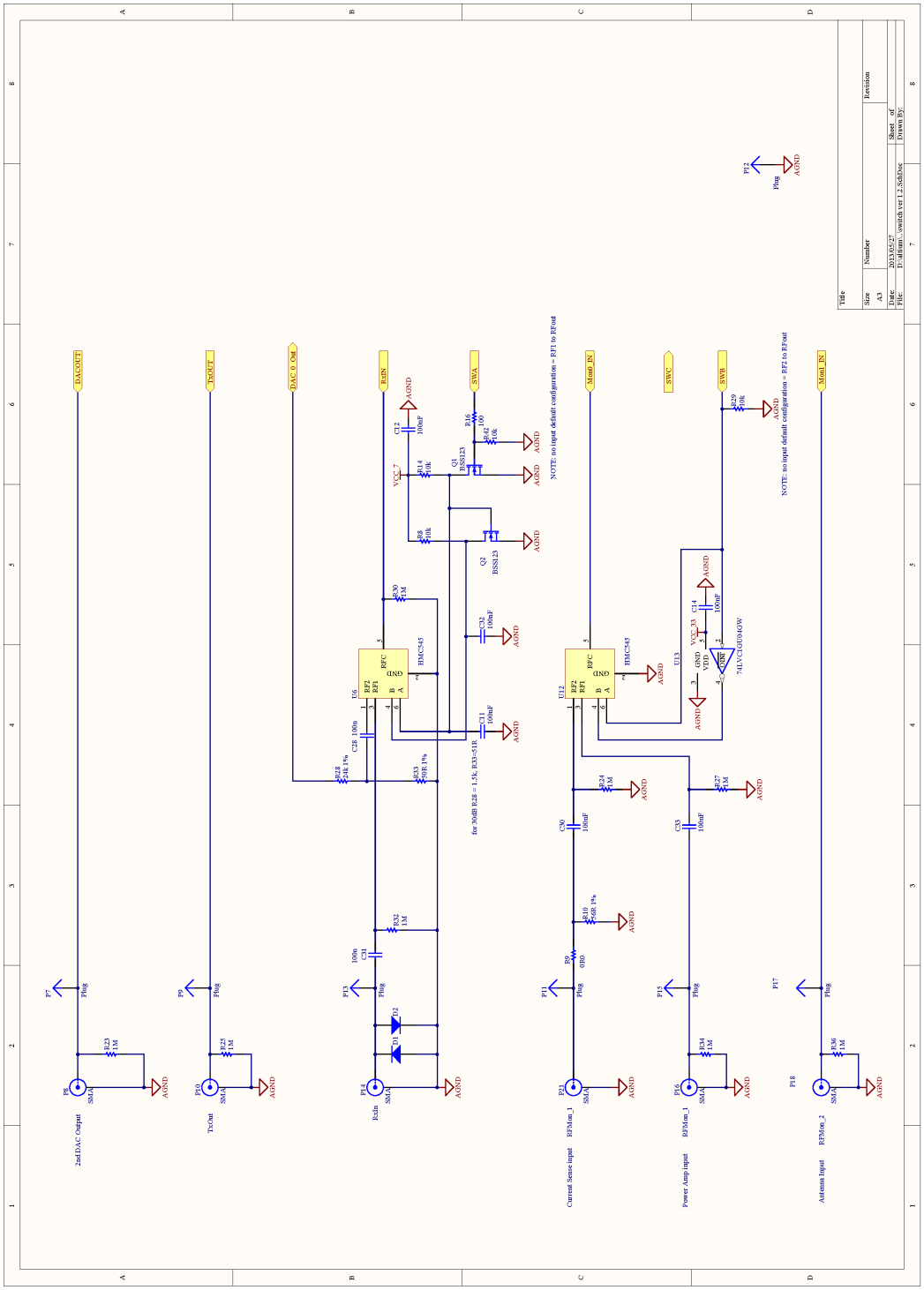

Figure D.4: RF switching schematic.

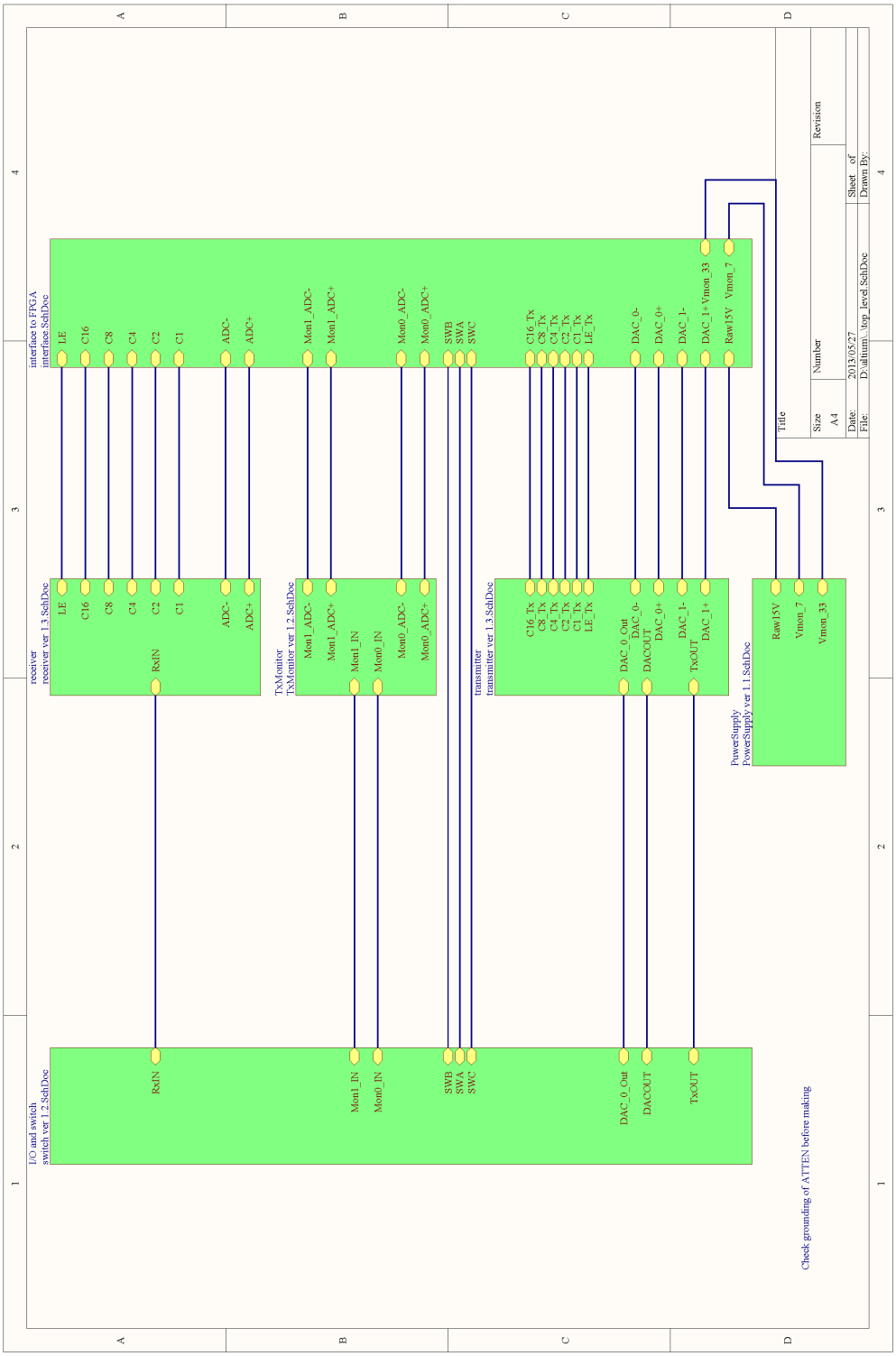

Figure D.5: RF board top-level schematic.

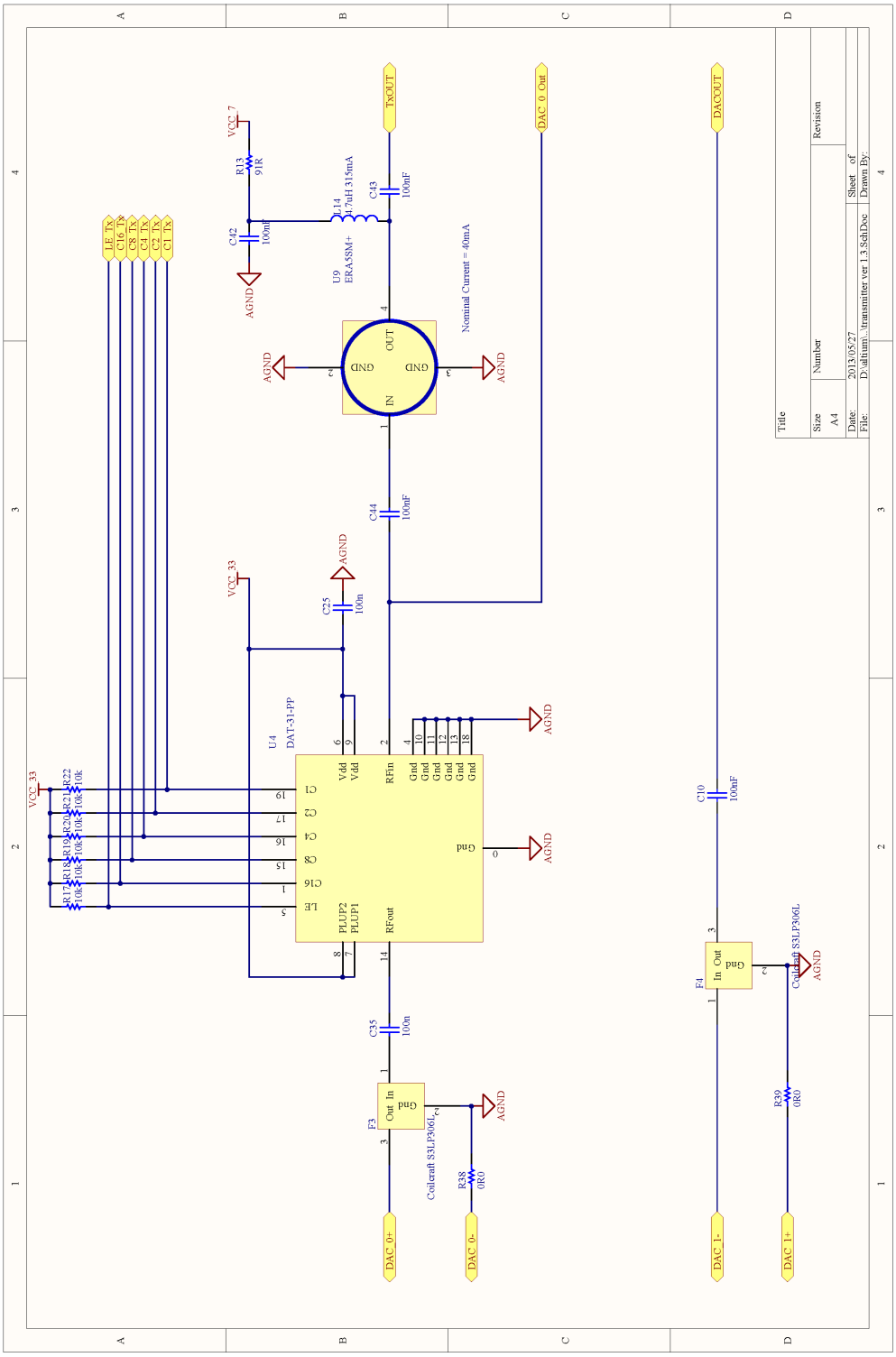

Figure D.6: RF transmitter chain schematic.

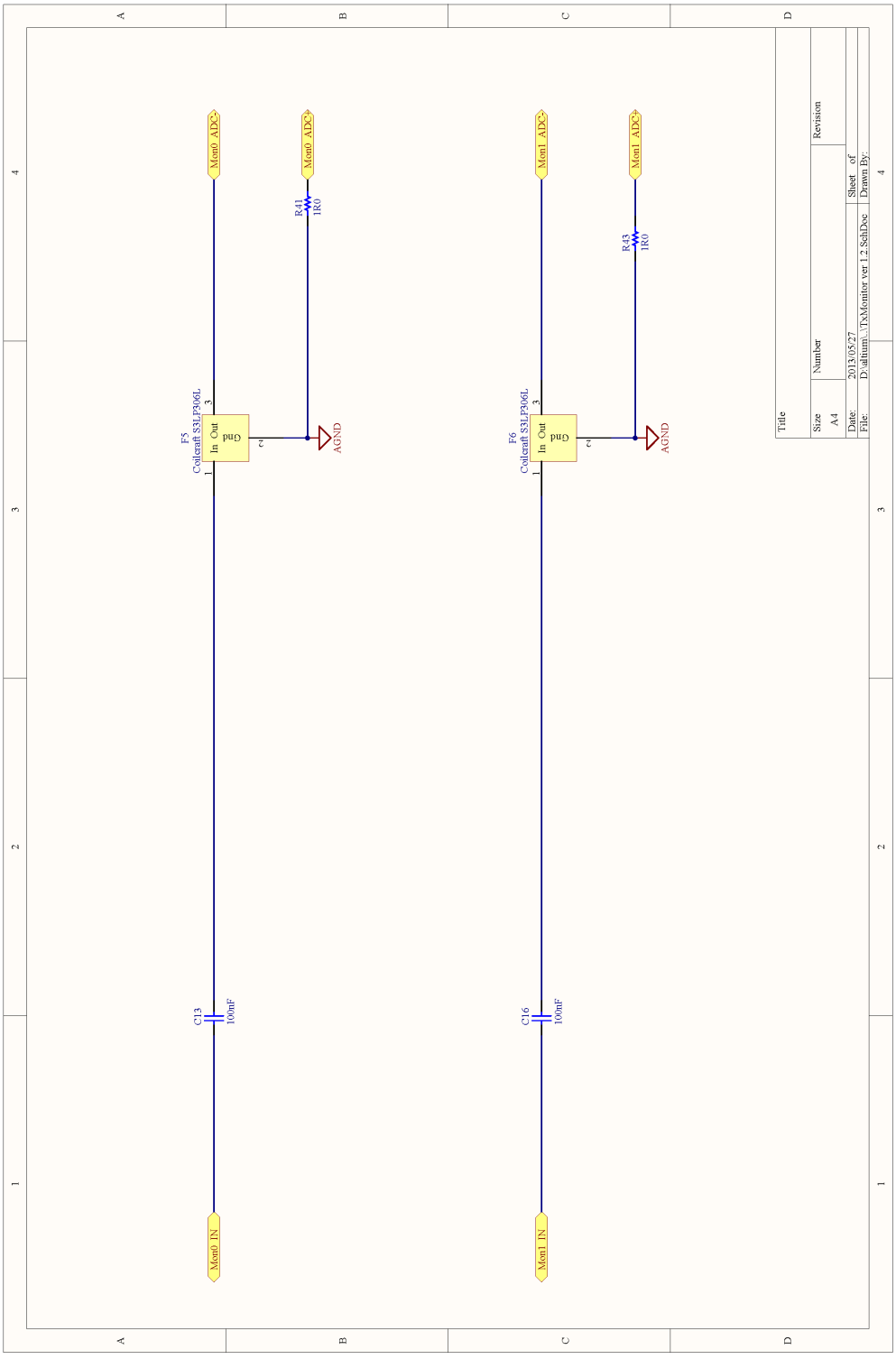

Figure D.7: RF monitor schematic.

E MAC Interface

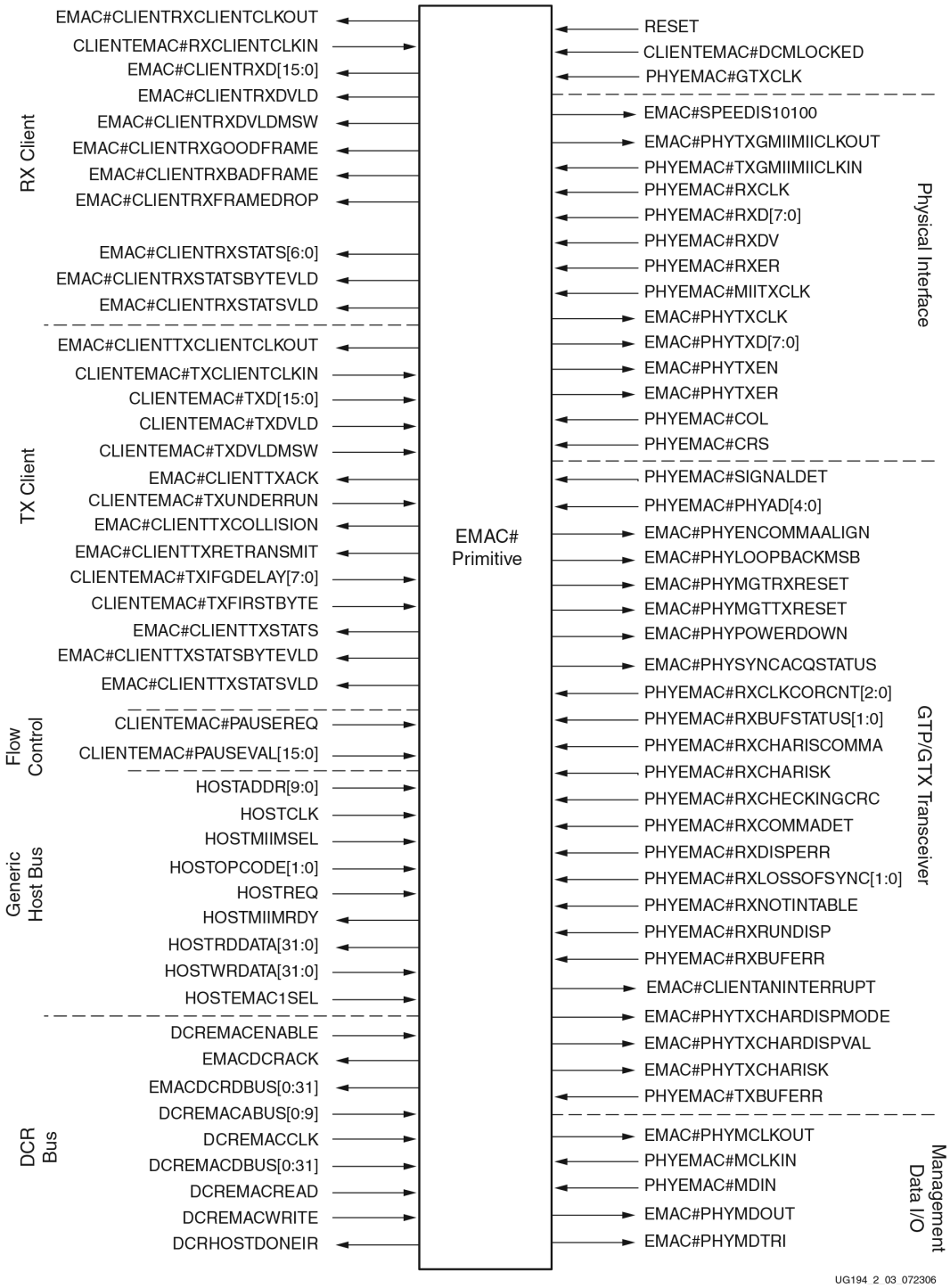

Figure E.1: MAC FPGA core interface pins [50].

F FPGA Module Layout

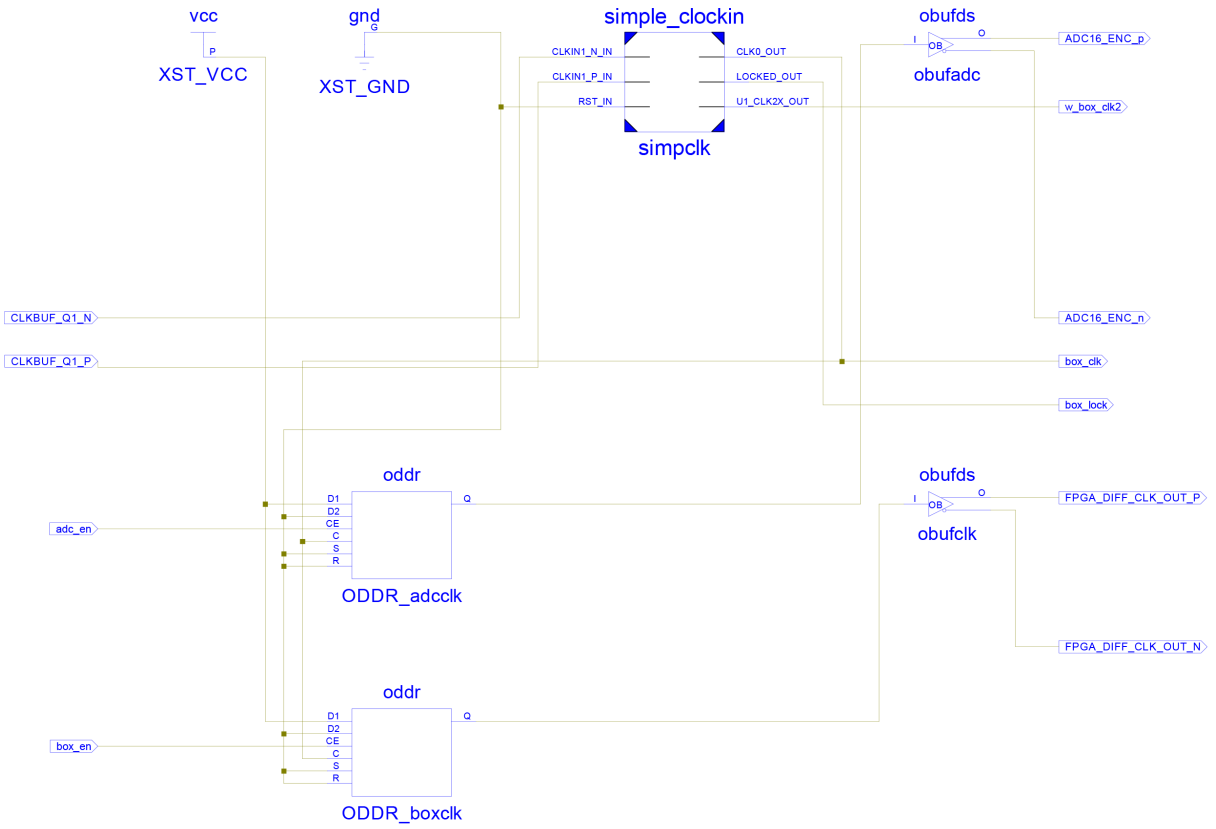

Figure F.1: Clock management module FPGA schematic.

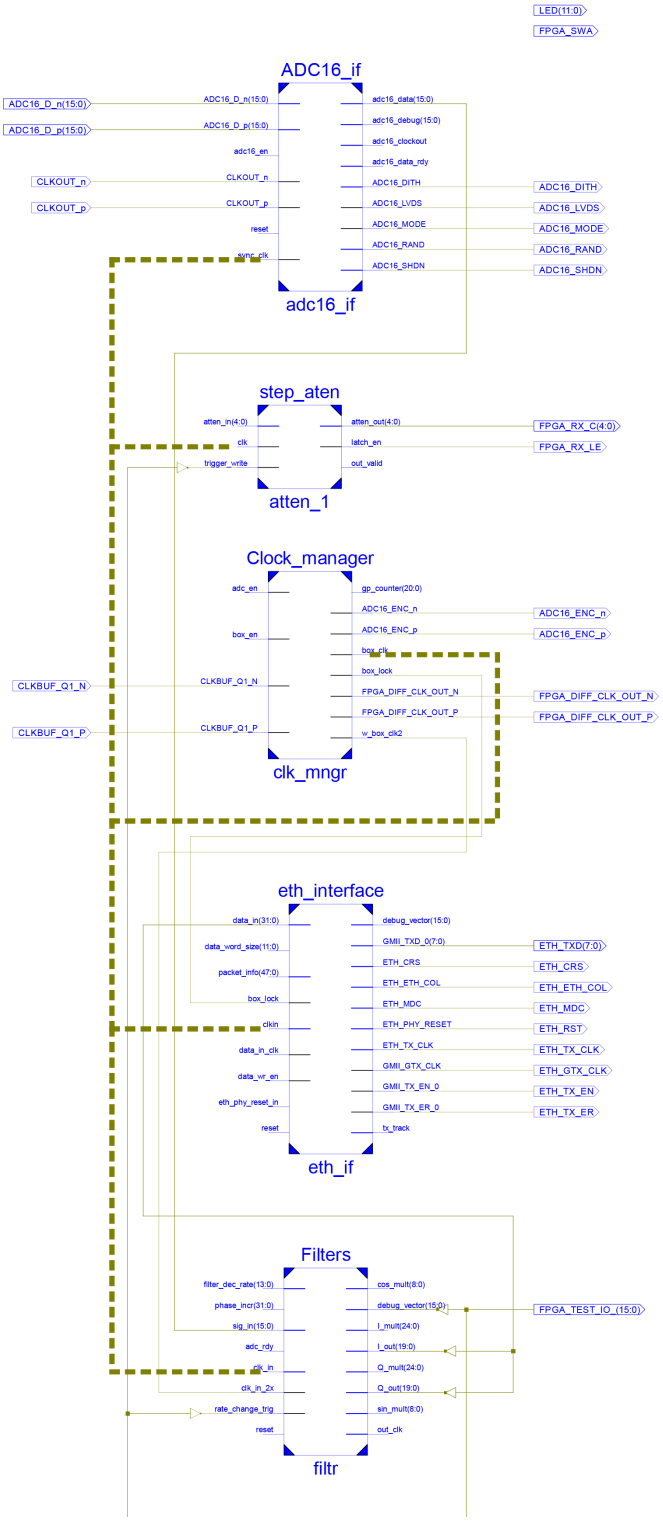

Figure F.2: TIGER-3 main module FPGA schematic.

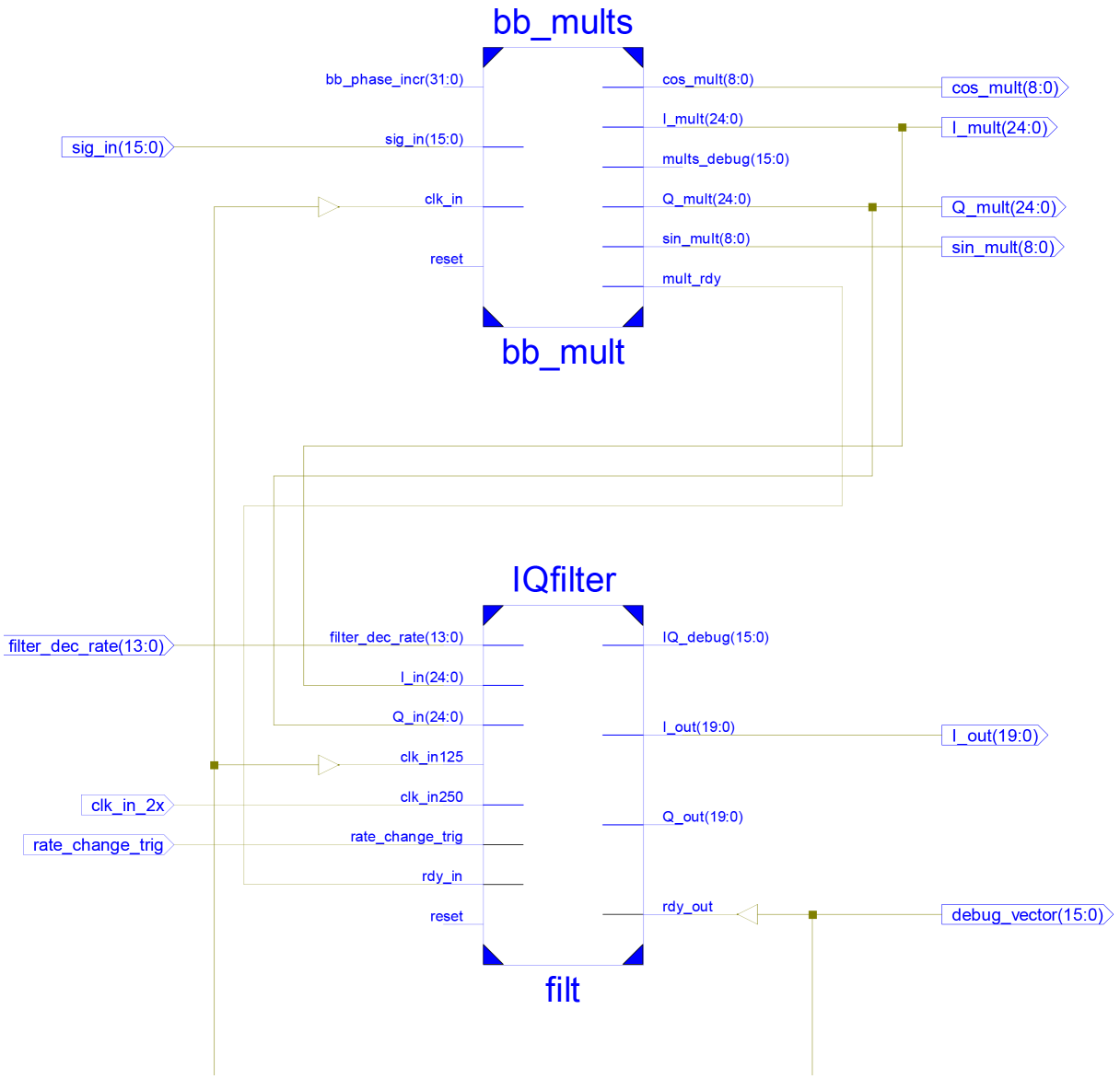

Figure F.3: DSP module FPGA schematic.

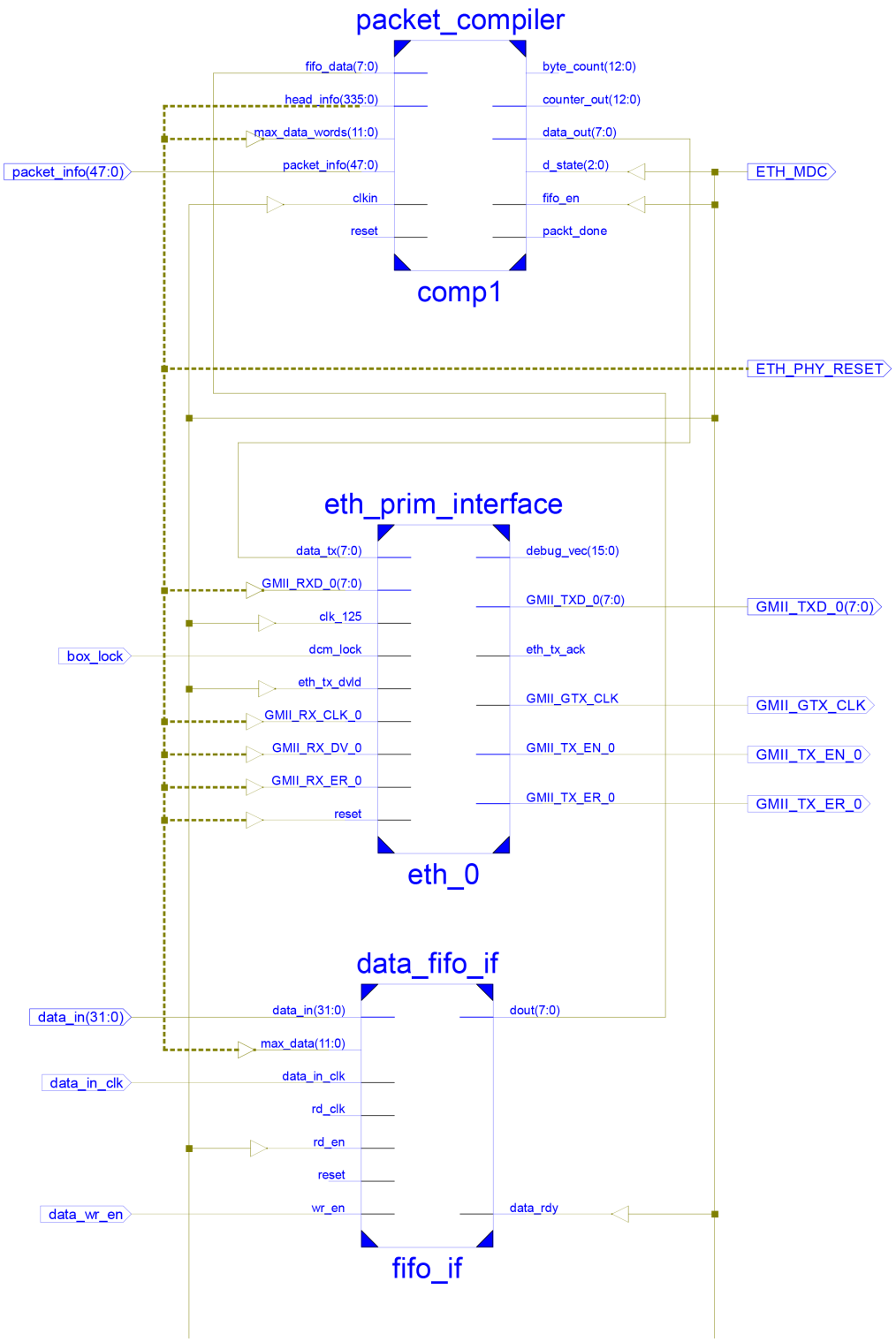

Figure F.4: Ethernet interface module FPGA schematic.

G RF Board Schematic Diagrams

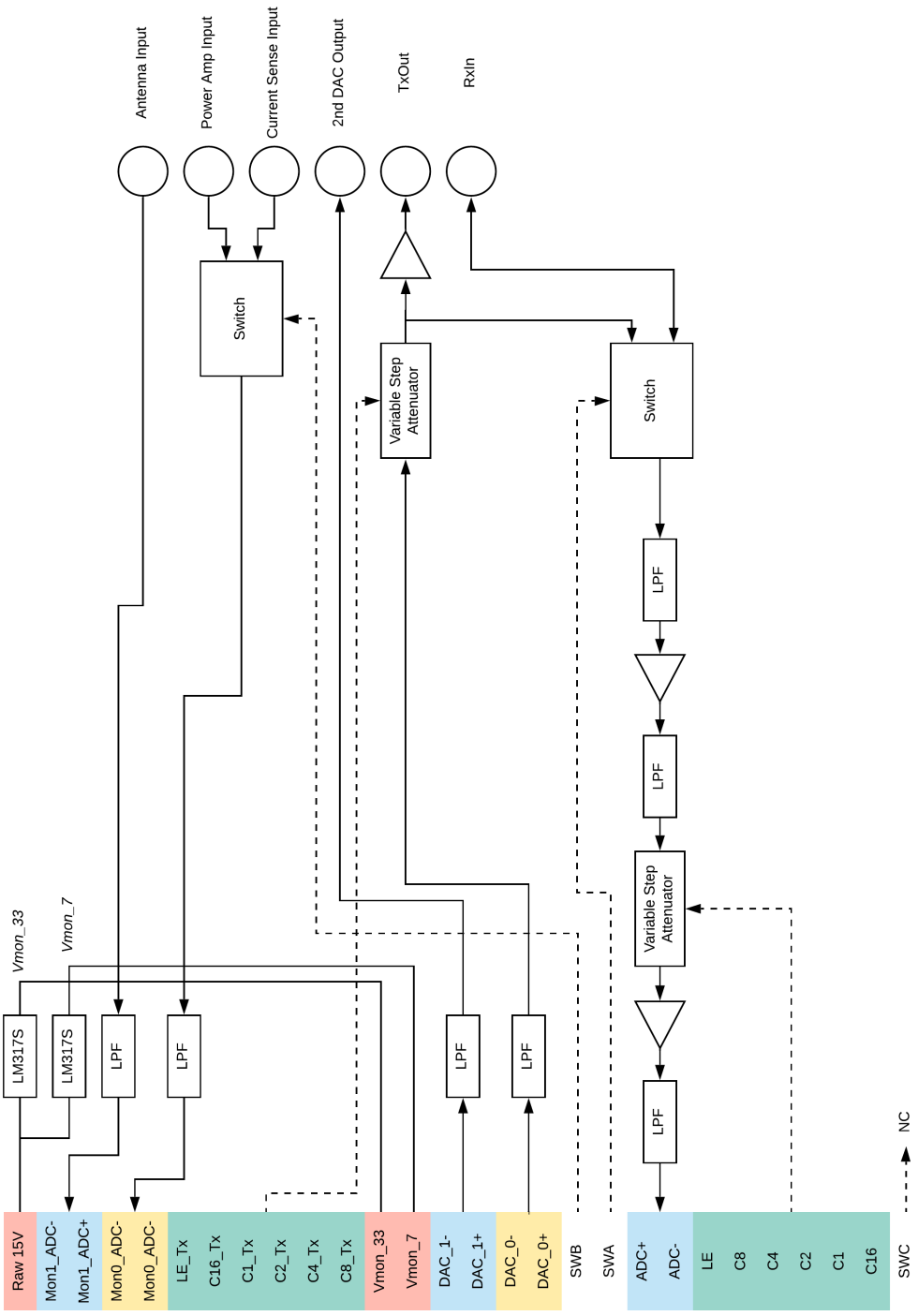

Figure G.1: Complete RF to FPGA block diagram.

H ADC Results

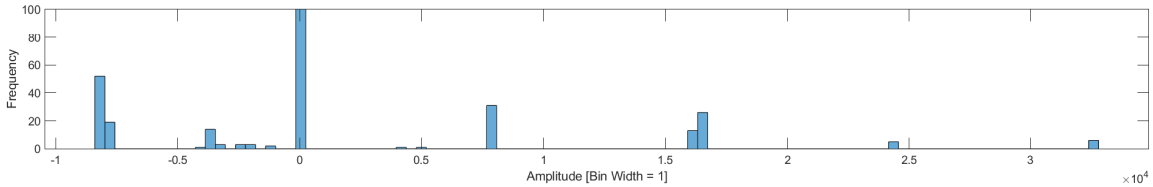

Figure H.1: Complete histogram of the ADC output data with the ADC input short-circuited (bin size of 412). The vertical axis is truncated to an frequency of 100

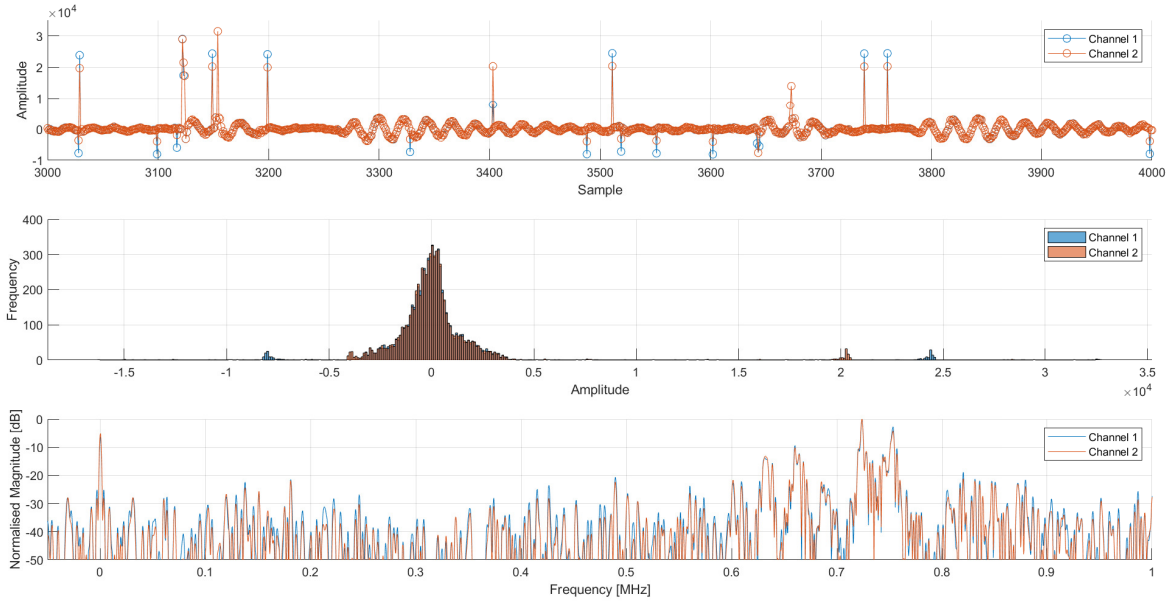

Figure H.2: Top: Time-domain data received from the ADC output, reading through the RF board with the input terminated by a 50 Ohm impedance. Middle: Corresponding histogram of the output data. Bottom: Corresponding truncated spectral data (FFT zero padded by 10000 samples).

I NCO Results

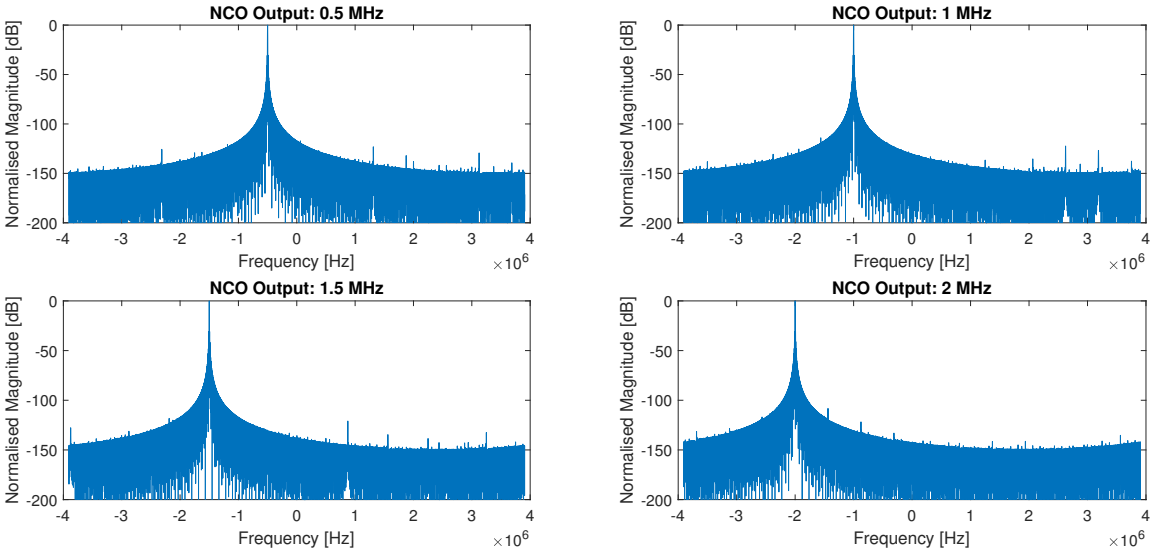

Figure I.1: NCO spectral results for different centre frequencies.