Homodyne Zoom FFT for Electronic Warfare Applications and Implementation in FPGA

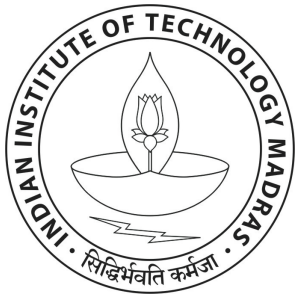
A THESIS

submitted by

R PAVAN KUMAR

for the award of the degree of

MASTER OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.

JUNE 2014

THESIS CERTIFICATE

This is to certify that the thesis titled **Homodyne Zoom FFT for Electronic Warfare Applications and Implementation in FPGA**,submitted by **R Pavan Kumar**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology** , is a bonafide record of the work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Nitinchandrachoodan

Project Guide,

Professor,

Dept. of Electrical Engineering,

IIT-Madras, 600 036

Place: Chennai

Date: 13th June 2014

ACKNOWLEDGEMENTS

I would like to thank Dr. Nitin Chandrachoodan for giving me an opportunity to work under his guidance. I thank for his valuable suggestions and constructive feedback throughout my project work.Its been a great learning experience and a definite improvement in the approach towards a problem under his guidance.

I would like to thank Dr.Umesh for his excellent teaching on Advanced Topics in DSP. This has triggered many ideas to explore some of the challenges in the the field of Electronic Warfare. I thank Dr. Dileep Nair for being my faculty adviser.

My sincere thanks to A.K. Singh, R.V. Hara Prasad, R. Rama Rao and the director of DLRL S.P.Dash for keeping faith in me by sending to pursue M.Tech in IIT under DRDO sponsorship. Also I thnak HRDC members of DLRL and DRDO head quarters for their kind support. My sincere acknowledgements to Gautam, Raju, Deepti,Vijayalakshmi,papaji for their valuable cooperation.

I would like to thank my wife Lakshmi Prasanna for her continuous support and encouragement. The friendly talks with her were helped me a lot to overcome my difficult times. I would like to thank Girish, Senthil, Ajmal, Selvin and all my classmates for being my good friends and classmates for these two years.

This thesis is dedicated to my parents Laxmi Narsimha, my wife and all my family members.

ABSTRACT

The present work is carried out to meet one of the requirements of Electronic Warfare (EW), which is primarily a Defence field.Recently we observe increased interest devoted to Frequency Modulated Continuous Wave (FMCW) Radars, mainly because of their Low Probability of Intercept (LPI) properties. Detecting these Radars in a complex EW environment mixed with high noise, interference and simultaneous signals, requires advanced digital signal processing techniques which can be implemented on FPGA. This thesis presents the concept of *Homodyne Zoom FFT* to Electronic Warfare for the detcetion of FMCW radar signals and its implementation in FPGA.

Modern day FMCW radar has very wide band width (BW) which can switch between few MHz (1 MHz) to hundreds of MHz (500 MHz) having modulation periods from few us (100 us) to several ms (50 ms). To handle this wide band width, sampling frequencies greater than 1 GHz is required, which reduces the frequency resolution achieved for the given number of samples using conventional FFT. In order to detect the smallest frequency variations for the selected sampling frequency a long length FFT can be used. As most of the information computed in a long length FFT is discarded, an efficient scheme is required to be implemented on FPGA with reasonable resources.

A technique is proposed here to implement a long length FFT using two short length FFTs for the application of EW to speed up the design process. The proposed technique works on the principle of Homo dyne mixing and Spectrum Zooming and so, it is called Homodyne Zoom FFT or HDZFFT in short. As the

design involves high data rates care has be taken at every stage to handle the design efficiently. Also simulations are carried out to compare proposed HDZFFT with one of the most popular high resolution spectrum estimation technique, MUSIC.

The design goal is to detect FMCW signals (having modulation patterns triangular, up ramp, down ramp and hopping along with CW) at minimum of 0 dB SNR and implement in FPGA with reasonable area and timing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

EW	Electronic War or Electronic Warfare
ESM	Electronic Support Measure
ECM	Electronic Counter Measure
ELINT	Electronic Intelligence
RADAR	Radio Detection And Ranging
CW	Continuous Wave
FMCW	Frequency Modulated Continuous Wave
ADC	Analog to Digital Converter
FPGA	Field Programmable Gate Array
HDZFFT	Homodyne Zoom FFT
MUSIC	Multiple Signal Classification
LO Freq	Local Oscillator Frequency
PFFT	Primary FFT
CIC	Cascaded Integrator & Comb
DDS	Direct Digital Synthesiser
HDM	Homodyne Mixer
BW	Band Width
LPI	Low Probability of Intercept

CHAPTER 1

Introduction

1.1 Motivation

The modern day EW receivers are required to match with current Radar technologies for the detection of LPI (Low Probability of Intercept) signals with wide instantaneous bandwidth, high sensitivity, wide instantaneous dynamic range, simultaneous signals in Real-Time with accurate parameter measurements. The frequency coverage of the radar generally lies between 0.5GHz to 40 GHz. The input signal of 0.5-40GHz is down converted to an IF of 750-1250MHz (i.e. 500MHz bandwidth) to feed it to the state of the art Digital Receiver. The Digital Receiver hardware consists of an high speed ADC with sampling frequency of 1.35GHz and an FPGA to implement the required algorithm.

To meet the modern day EW receiver requirements of High sensitivity of the order of -90dBm to process FMCW signals it is required to process the signals with minimum of 0 dB SNR. FFT has been a widely applied digital technique to estimate the frequency of a signal. But the intrinsic contradiction between frequency resolution and computational complexity limits its application. If the sampling frequency increases then the FFT length goes up further which need more memory,long processing time. Therefore there is a need to optimise the memory requirements by keeping the design simple.

This is the motivation behind the proposed *Homodyne Zoom FFT* technique to detect FMCW signals with high frequency accuracy at low SNR, in order to reconstruct the waveform for Electronic Attack (EA) purpose.

1.2 Design Objective

The main objective of this work is to develop a digital technique which can detect FMCW signals with high sensitivity (or low SNR) and high frequency accuracy. Also, implementation of this high speed design in FPGA with reasonable resources.

1.3 Basic Design Principles

The two major principles involved in the proposed technique are 1) Homodyne principle and 2) Spectrum Zooming technique. Homodyne principle is used to convert a wide band signal to a narrow band signal by deriving LO frequency from the input signal itself. Then the spectrum zooming technique is applied on this narrow band signal to get finer details of the signal.

1.4 Design Requirements

An FMCW Radar is used as a case study to understand the design requirements in a better way.

PILOT FMCW Radar:

- Center frequency,F_c = 9.375 GHz
- Frequency deviation,Δ_F = 1.7 MHz, 3.4 MHz, 6.8 MHz, 13.75 MHz, 27.5 MHz, 55 MHz.
- ΔF_{min} = 1.7 MHz, ΔF_{max} = 55 MHz
- Sweep time,Δ_T = 1 ms.

Hardware: The modern day Digital EW Receiver hardware consists of an high speed ADC and an FPGA with high density gate capacity. The ADC has the sampling frequency of 1.35 GHz with an instantaneous BW of 500 MHz using Band Pass Sampling (BPS) technique. FPGA (SX series Virtex-6 FPGA) from Xilinx

has maximum clock Frequency of 550 MHz. The proposed algorithm should be implemented on this hardware to meet the real time requirements of EW receiver.

Wide Modulation Band width (ΔF_{max}): The maximum frequency deviation in an FMCW radar can be as large as 500 MHz. To detect this wideband width FM signal, the ADC must have large BW that indirectly pushes the Sampling frequency to higher value. It is known that the Frequency Resolution (F_r) for a given length of data,M is given by equation 1.1.

$$F_r = \frac{F_s}{M} \tag{1.1}$$

where,(F_r) is directly proportional to F_s. Therefore, if ΔF_{max} increases, F_s increases and F_r reduces. So, the proposed technique should be able to handle the largest frequency deviation without effecting frequency resolution.

Narrow Modulation Band width (ΔF_{min}): The smallest frequency deviation determines the length of data, M (equation 1.1) for the given F_s. The F_s is decided based on ΔF_{max} that can occur in an FMCW Radar. But if length of data is increased the area and time consumed to compute FFT in FPGA increases.

Modulation Period: The frequency modulation occurs over a period of time and this pattern is repeated continuously. Typical values for modulation period range from few ms (1 ms) to several ms (50 ms). Slow varying FM requires high frequency resolution where as fast varying FM requires real time processing.

FMCW Pattern: Frequency modulation in an FMCW Radar can have various patterns Ex: Positive Ramp, Negative Ramp, Triangular, Hopping etc. As the EW system does not have any prior information about the incoming signal, detecting these patterns with the above requirements is very important for launching Electronic Attack (EA).

High Sensitivity: The minimum sensitivity requirement for an EW receiver to process LPI FMCW signals is -90 dBm. This requires minimum SNR of 0 dB in a multi signal environment for the given specifications.

1.5 Design Specifications

Let us assume that the Radar signals from 0.5-40 GHz are down converted to an IF of 1000± 250 MHz, which is then applied as input to Digital EW Receiver. Following specifications are derived based on the requirements mentioned above.

- **Sampling Frequency** $F_s = 1350$ MHz The following Band Pass Sampling equation 1.2 is used for the selection of F_s (Band Pass Sampling technique is used here because of the non availability of ADC at Nyquist Rate for the given Specifications)

$$\frac{2 * F_c - BW}{m} \geq F_s \geq \frac{2 * F_c - BW}{m + 1} \tag{1.2}$$

  where F_c is Center Frequency = 1000 MHz, BW is Band Width = 500 MHz, m is any integer say 1 in this case. From this equation F_s can be any value between 1250 MHz to 1500 MHz and it is chosen as 1350 MHz.

- **Lowest frequency deviation** $\Delta F_{min} = 1.7$ MHz
- **Modulation Period** $t_m = 1$ ms
- **Modulation Rate** is given by

$$\frac{F_{min}}{t_m} = \frac{1.7MHz}{1ms} = \frac{170KHz}{100us} \tag{1.3}$$

Frequency deviation of 1.7 MHz in 1 ms duration is equal to 170 KHz in 100 us duaration. If 170 KHz is detected within 100 us then total 1.7 MHz deviation can be easily detected in 1 ms. To achieve this let us consider the following

- Length of data, **M** is 16384 samples
- sampling frequency, F_s is 1350 MHz
- **Frequency Resolution** is

$$F_r = \frac{F_s}{M} = \frac{1350MHz}{16384} = 82KHz \tag{1.4}$$

- **Time Resolution** is

$$T_r = \frac{M}{F_s} = \frac{16384}{1350MHz} \simeq 12us \tag{1.5}$$

So, any frequency variation $\geqslant$ 82 KHz can be detected with a time resolution of 12 us i.e. the minimum duration of the signal required is 12 us in order to

resolve signal with frequency resolution of 82 KHz. If higher frequency resolution is required then the number samples to processed will be more.

1.6 Thesis organisation

The rest of the thesis is organized as described below.

Chapter 2 The basics of the back ground, which is essential to understand rest of the thesis has been discussed in this chapter. Basics of Electronic Warfare, Radar, Digital EW Receiver have been discussed in brief. Also some of the EDA tools were mentioned which makes it easy to understand the simulations presented in the following chapters.

Chapter 3 discusses the detailed explanation on the concept of Homodyne Zoom FFT technique. Concepts of each block present in the design are discussed.

Chapter 4 Implementation details of the design in FPGA are presented

Chapter 5 deals with the simulation results achieved using Matlab and Verilog. Also simulations showing comparisons between HDZFFT and MUSIC are presented

Chapter 6 deals with the Performance Trade off's and performance comparisons in the design

Chapter 7 summarizes the design with concluding remarks and future scope

CHAPTER 2

Background

2.1 EW Scenario

The present work is carried out to meet one of the requirements of Electronic Warfare (EW), which is primarily a Defence field. The EW scenario can be better understood by the following figure 2.1.

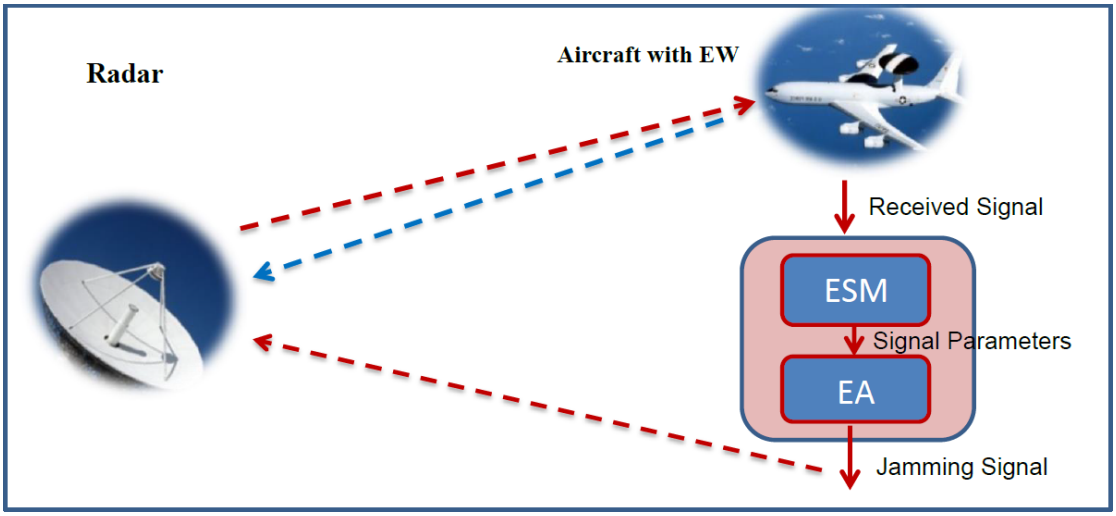


Figure 2.1: Electronic Warfare Scenario

Radar transmits Radio Frequency (RF) signal, if there is a target (in this case the target is Aircraft) then part of the energy will be reflected back which indicates the presence of a target. Radar compares this reflected signal with the original transmitted signal to find out the Range and Velocity of the target. Based on the information obtained by the Radar it then launches its attack towards the target. When there is no target present in the environment the energy will not be reflected, this indicates the absence of a target

Let us assume the Aircraft in the above figure 2.1 is equipped with Electronic Warfare System. Basic building blocks of EW system are Electronic Support Measure (ESM) and Electronic Attack (EA). These two systems are shown in the above bock diagram. The major role of an ESM system is to find some of the important intelligence from the received RF signal and pass this information to ECM system. The ESM system can be used as a defensive system to avoid the threat from the Radar or an offensive system to launch attack towards the Radar from EA system.

2.2 Radar

Radar is an acronym for **Ra**dio **D**etection **A**nd **R**anging. As discussed in the previous section Radar transmits an RF signal and look for its reflected signal if the signal finds a target. The returned echo signal is compared with the transmitted signal to find the the range and velocity of the target.

2.2.1 CW Radar

Continuous Wave (CW) Radar radiates RF signal continuously with frequency f_0. If the target is in motion the received signal is shifted in frequency from the transmitted frequency f_0 by an amount $\pm f_d$ called the Doppler frequency. the sign of f_d indicates, whether the target is approaching (+ sign) or going away (- sign) from the radar. The Doppler shift is zero, in case of stationary target. The advantage of CW Radar is it uses low average power to measure the target velocity. But, its inability to obtain measurement of target range limits its usage.

2.2.2 Pulsed Radar

Pulsed Radar measures the range to target by measuring the elapsed time between sending a pulse of radio energy and receiving a reflection off the target. Equation 2.1 is used for the calculation of target range. Velocity of target is measured by Doppler frequency shift similar to CW radar. Pulsed radar can measure both target range and velocity but requires high peak power to achieve good range resolution.

$$R = \frac{C* \triangle t}{2} \tag{2.1}$$

where R is range to target, C is velocity of light, $\triangle t$ is the elapsed time between transmitted and the received signal.

2.2.3 LPI Radar

Low Probability of Intercept (LPI) radar is defined as a radar that uses a special emitted waveform intended to prevent an EW receiver from intercepting and detecting its emission. The LPI requirement is in response to the increase in capability of modern EW receivers to detect and locate a radar emitter.The main drawback of a pulsed radar is the high peak to average power ratio. The average power is what determines the detection characteristics of the radar. For high average power a short pulse (high range resolution) must have a high peak power. The high peak power transmissions can easily be detected by EW receivers. The duty cycle $duty_{cycle}$ for a pulsed radar relates the average transmitted power P_{avg} to the peak power P_{peak} as

$$duty_{cycle} = \frac{P_{avg}}{P_{peak}} \tag{2.2}$$

The duty cycle can also be calculated as

$$duty_{cycle} = \frac{T_{pw}}{T_{pri}} \tag{2.3}$$

where T_{pw} is the pulse width and T_{pri} is the pulse repetition interval. Typical duty cycles are 0.001 (the peak power is 1000 times the average power). In modulated CW signals, however, the average to peak power ratio is 1 or 100% duty cycle. This allows a considerably lower transmit power to maintain the same detection performance as the pulsed radar. consequently most LPI radars use CW signals. Since CW signals can easily be detected using narrow band receivers, LPI radars use periodically modulated CW signals resulting in large bandwidths and small resolution cells, and are ideally suited for pulse compression.

There are three major pulse compression techniques namely Frequency modula-

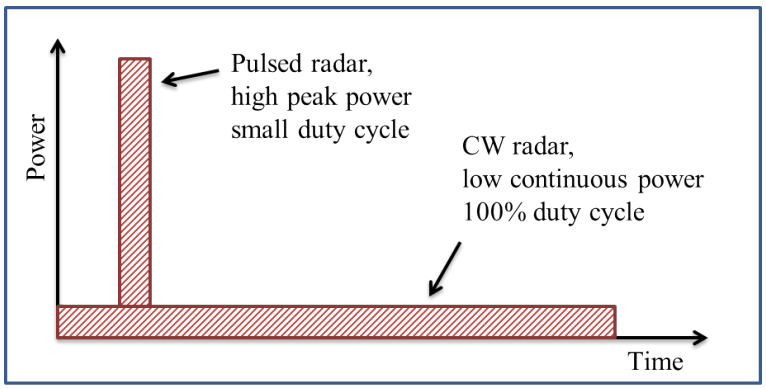


Figure 2.2: Comparison of pulsed Radar and CW radar

tion, phase modulation and combined phase frequency modulation, that provide a wide bad LPI CW transmit waveform. Any change in the radars signature can help confuse an EW receiver and make intercept difficult. The wide band width makes the interception of signal more difficult. For an EW receiver to demodulate the waveform, the particular modulation technique used must be known, which is typically not the case.

2.2.3.1 FMCW Radar

Frequency Modulated CW radar is a class of LPI radar and an effective LPI technique. The timing mark (timing mark is the changing frequency) used in FMCW permits the time of transmission and the time of return to be recognised. The greater the transmitter frequency deviation in a given time interval, the more accurate the measurement of the transit time and greater will be the transmitted spectrum. Due to the very low energy transmitted,the EW receivers interception range is significantly reduced. The frequency modulation spreads the transmitted energy over a large modulation bandwidth ΔF, providing good range resolution that is critical for discriminating targets. The power spectrum of the FMCW signal is nearly rectangular over the modulation bandwidth, so EW interception is difficult. Since the transmit waveform is deterministic, the form of the return signals can be predicted. This gives it the added advantage of being resistant to jamming. Since any signal not matching this waveform can be suppressed. Consequently it is a difficult matter for EW receiver to detect FMCW waveform and measure the parameters accurately enough to match the jammer waveform to the radar waveform.

2.3 Electronic Warfare

The extensive application of Electro Magnetic (EM) spectrum to Radar and Electronic War provided the armed forces with powerful tools; coordination between forces,accurate location of the deployed forces and surveillance of the battle space. The EM spectrum is so extensively exploited, today without its use, the survivability of armed forces is jeopardized.

2.3.1 Introduction

Electronic War (EW) is the science of manipulation and control of EM environment for its own survivability but denies or limits it to the adversary. This information is then exploited to influence adversary's capability to coordinate its activities, to restrict its communication media, to deny the use of radar for weapon launching or guiding. EW enhances the survivability of own forces by denying the use of EM spectrum. EW is classified in many ways based on frequency spectrum, functionality, application etc. Classification based on functionality is considered here for further discussion, other details can be found in the literature. According to functionality an EW system is broadly classified in to three categories (i) Electronic Support (ES), (ii) Electronic Attack (EA) and (iii) Electronic Protection (EP).

Electronic Support (ES): ES also is known by ESM (Electronic Support Measure). ES involves search, intercept, locate, record and analyse radiated EM energy for the purpose of exploiting the radiation information either for formulating EOB (Electronic Order of Battle) or to provide the real time information to EA system.

Electronic Attack (EA): EA also is known by ECM (Electronic Counter Measure). EA involves action taken to prevent or reduce enemy's effective use of EM spectrum. It can be active like Jammers, or it can be passive like Chaff.

Electronic Protection (EP): EP also is known as ECCM (Electronic Counter Counter Measures). EP involves actions taken to ensure friendly use of EM spectrum despite the use of ECM. EP protects own platform against EA by the adversary.

2.3.2 Basic ESM System

Most of the Radars occupy frequency range of 0.5-40 GHz. The ESM system is designed to operate in this frequency band. The modern day ESM system is configured with Wide Band (WB), Narrow Band (NB) and Digital EW Receivers. Basic configuration of the system is shown in figure 2.3 figure. The incoming

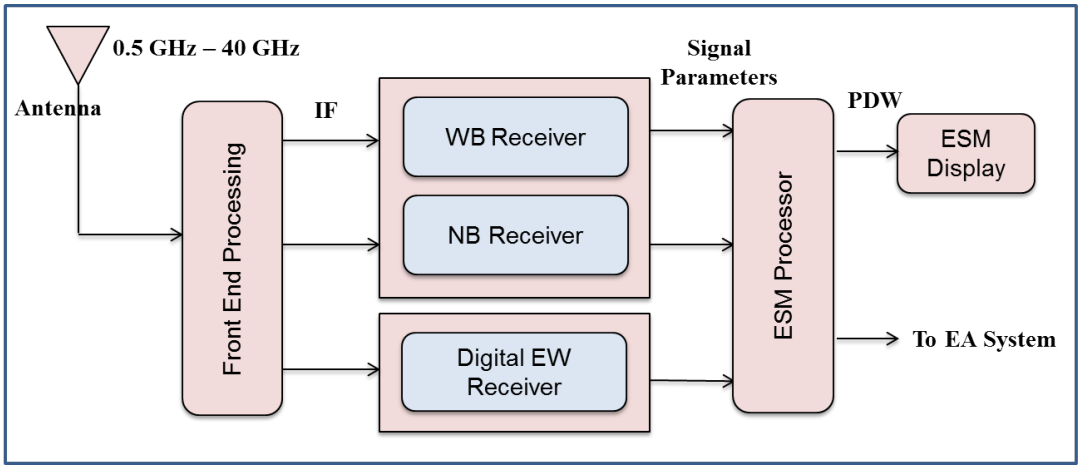


Figure 2.3: Basic Configuration of ESM System

signal incident on antenna is down converted properly to an intermediate frequency (IF) depending on the receiver requirements. WB and NB Receivers use analog techniques where as Digital Receiver uses digital techniques for the detection and processing of received signals. The parameters measured through receivers are further processed to create a Pulse Descriptor Word (PDW) which consists of all the information about the signal. Finally the processed PDWs displayed on ESM display which will give location of the radar with various parameters. Also the same information is passed on to EA system for jamming purpose. ESM system is a passive system and EA system is an active system.

2.3.3 Digital EW Receiver

Digital EW Receiver is a state of the art technology with single board solution for EW receivers. Digital receiver accepts the input of 750 MHz - 1250 MHz and digitizes it using the high speed ADCs and processes the digitized data in FPGAs to extract the radar signal parameters. The hardware consists of an high speed ADC with sampling frequency of 1350 MHz using band pass sampling technique and high density virtex-6 FPGA. The function of the hardware is to estimate the direction of arrival of the radar signal using digital techniques implemented in FPGA. The simplified block diagram of the hardware is shown in figure 2.4.

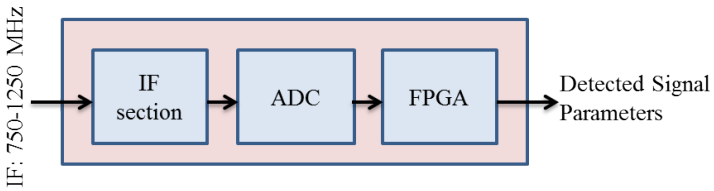


Figure 2.4: Simplified block diagram of Digital EW Receiver

2.4 EDA Tools

The EDA (Electronic Design Automation) tools Mtalb, FDA tool, ISIM, GTK-WAVE and ISE are used for the simulation and synthesis of proposed technique. Matlab is used for simulating the mathematical concepts of the technique. FDA is a filter design and analysis tool used for the implementation of filters. ISIM is used for the simulation of HDL code written for the design in verilog and GTK wave for visualising the simulation results in analog form. Synthesising of the design is done using ISE to estimate the area consumed in FPGA.

CHAPTER 3

Homodyne Zoom FFT

3.1 Concept of Homodyne Zoom FFT

The Homodyne Zoom FFT is a spectrum zooming technique with homodyne principle to suit the requirements of Electronic Warfare. It is a very interesting technique in spectrum analysis, when fine spectral resolution is needed within a small portion of a signal overall frequency range. When the duration of the signal to be analysed is long, this technique is more efficient than the traditional FFT. We could collect many time samples and perform a large size FFT to satisfy our fine spectral resolution requirement. This solution is inefficient because we would be discarding most of our FFT results. The Homodyne Zoom FFT can help us improve computational efficiency to speed up the design process.

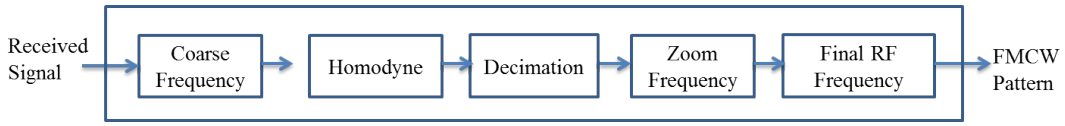


Figure 3.1: Concept of Homodyne Zoom FFT

The two major principles involved in the proposed technique are 1. Homodyne 2. Spectrum Zooming

Homodyne: The first stage of the proposed technique involves converting a wide band signal to a narrow band signal. To perform this operation an incoming signal is mixed with a known complex signal using mixer to shift the frequency spectrum. But, we know in an EW Receiver the incoming signal is unknown and there is no information about the frequency to be used for mixer. So, there is a need to first generate frequency to be used for tuning the mixer from the incoming signal itself.

This is exactly done at this stage using a principle called Homodyne.

Spectrum Zooming: The mixer produces two frequency components at its output, one with low frequency component (near zero frequency) and a high frequency component (twice the incoming signal frequency). The required narrow band width component which is centred around zero frequency is used for zooming purpose. Spectrum analysis within a small portion of frequency spectrum is carried out to extract the fine frequency details. Because of these two principles this technique is given the name ***Homodyne Zoom FFT***. The block diagram shown in figure 3.1 gives an overall view of HDZFFT. The discussion is divided in to four sections (i) Coarse Frequency Estimation, (ii) Homodyne, (iii) Decimation (iv) Zoom Frequency estimation and (v) RF frequency calculation.

3.2 Coarse Frequency Estimation

The first step in Homodyne Zoom FFT is to estimate the incoming signal frequency. As discussed earlier the frequency information in an EW Receiver is unknown. Without this frequency information it is not possible to generate Local Oscillator (LO) frequency which is required for mixer to operate. If LO information is unknown then mixing of two signals can not happen in this case. Even though input samples can be used for mixing purpose, calculation of RF frequency can not be done without estimating LO frequency. So a *Primary FFT* is used to estimate the incoming signal frequency. The block diagram representing coarse frequency estimation is given in figure 3.2

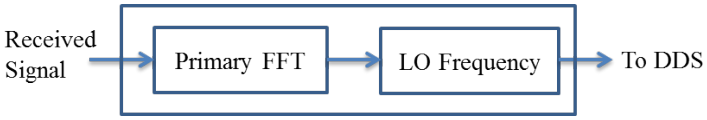


Figure 3.2: Block diagram representing coarse frequency

3.2.1 Primary FFT

We know FFT is an abbreviation for Fast Fourier transform which translates time domain sequence to frequency domain. The frequency domain gives the information about the frequency components present in the input signal. Since the ADC generates real data, a real FFT is used here produces frequency spectrum with mirror image. the mathematical equation for the calculation of FFT is given by following equation 3.1

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\Pi kn}{N}}, k = 0, 1, ...N-1 \tag{3.1}$$

where x(n) is the sampled signal, N is the length of FFT, k is frequency index

The primary FFT (PFFT for convenience) plays a very important role in this design as sensitivity of the system is determined by this module. The main function of PFFT is to find the incoming signal frequency. If the signal power is low which is not adequate for the FFT to detect the signal then the signal frequency can not be estimated. So depending on the sensitivity requirements of the design the number samples required for PFFT computation are chosen. In order to detect FMCW radar signals the sensitivity of around -90 dBm is required for an EW receiver. This is explained using following equation 3.2 which is very popular in EW community.

3.2.1.1 Sensitivity Calculation

$$Sensitivity(S_{min}) = -114 + 10\log\sqrt{2 * B_{IF} * B_v} + NF + SNR_{min} \tag{3.2}$$

where

$$B_V = \frac{F_s}{N} \tag{3.3}$$

S_{min}: the minimum sensitivity required is -90 dBm, Thermal noise is -114 dBm, B_{IF}: IF band width in MHz is 500 MHz, NF:the Noise Figure of the receiver is 8 dB, SNR_{min}: the minimum SNR required is 0 dB, B_v: video band width or resolution band width in MHz calculated in equation 3.4.
let us chose the number of samples for PFFT, N is 512 arbitrarily and calculate B_v by substituting the values of F_s and N in equation 3.3 we get

$$B_v = \frac{1350MHz}{512} = 2.637MHz \tag{3.4}$$

Now substitute equation 3.4 in 3.2 to get

$$Sensitivity(S_{min}) = -114 + 10\log\sqrt{2 * 500 * 2.637} + 8 + 0 = -89dBm \tag{3.5}$$

From the above discussion we can conclude that length of PFFT is selected based on minimum sensitivity requirement. If the length of PFFT is increased the sensitivity is improved but the area and time consumed in FPGA increases.

3.2.2 LO Frequency Estimation

The PFFT gives magnitude information with corresponding frequency indices for the applied signal. This information is used to calculate the Local Oscillator (LO) frequency which is coarse estimation of the frequency because the number of samples here is less. The magnitude of PFFT is converted to absolute magnitude. Here real magnitude is used as the imaginary value is zero for real FFT. By using a search technique peak value is estimated and the corresponding frequency index

at which the maximum value occurs is the required index. The index is multiplied with frequency resolution to calculate the frequency as given in equation 3.6. where $PFFT_{index}$ is the frequency index of Primary FFT

$$LOfrequency = (PFFT_{index}) * \frac{F_s}{N} \tag{3.6}$$

3.3 Homodyne

Homodyne is a process of mixing a signal with it self to shift the frequency spectrum of the input signal to zero frequency. The LO frequency of input signal estimated in the previous section is used to tune Direct Digital Synthesiser (DDS). DDS generates digital samples based on the frequency applied. These DDS samples are then mixed with the input samples to produce the mixer output. This is shown in figure 3.3

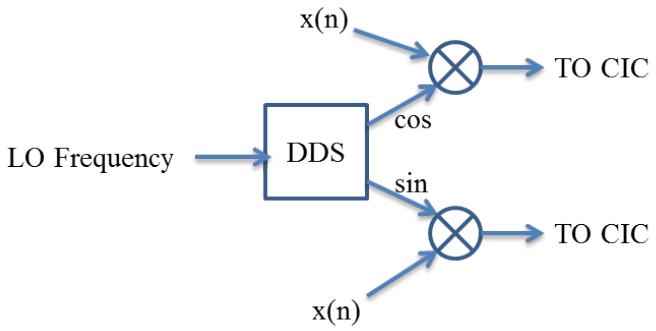


Figure 3.3: Homodyne operation using DDS

3.3.1 Direct Digital Synthesiser

Direct Digital Synthesizer (DDS) generates a sine/cosine waveform at a given frequency directly using digital techniques. The ability to digitally program and reprogram the output waveform with high resolution and accuracy make DDS an extremely attractive solution. A DDS consists of a Phase generator and a

SIN/COS Lookup Table. The basic operation of DDS is explained through following figure 3.4

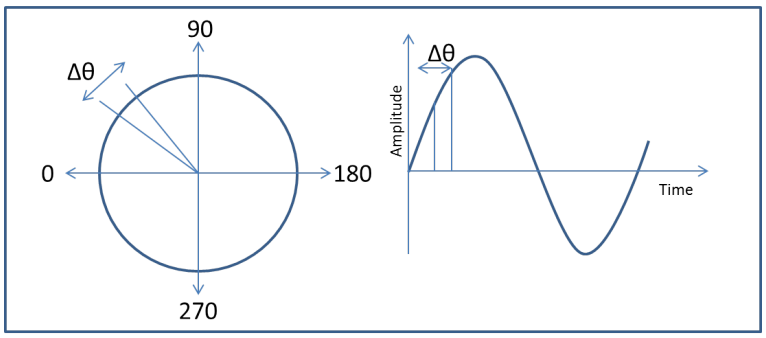


Figure 3.4: operation of DDS using phase diagram

Let us consider the phase advances by an amount $\Delta\Theta$ around a unit circle as shown in figure 3.4. As the phase advances around the circle this corresponds to advances in the waveform, i.e. the greater the number corresponding to the phase increment, the greater the advancement along the waveform. The digital number representing the phase is held in the phase accumulator. The number held here corresponds to the phase and is increased at regular intervals. In this way it can be seen that the phase accumulator is basically a form of counter. When it is clocked, adds a preset number to the one already held. When it fills up, it resets and starts counting from zero again. In other words this corresponds to reaching one complete circle on the phase diagram and restarting again.

Once the phase has been determined it is necessary to convert this into a digital representation of the waveform. This is accomplished using a sine/cosine look up table (LUT) which stores a number corresponding to the voltage required for each value of phase on the waveform. The next stage in the process is to convert the digital numbers coming from the sine/cosine look up table into an analogue voltage. Tuning is accomplished by increasing or decreasing phase increment between different sample points. A larger increment at each update to the phase accumulator will mean that the phase reaches the full cycle value faster and the frequency is correspondingly high. Smaller increments to the phase accumulator value means

that it takes longer to increase the full cycle value and a correspondingly low value of frequency. In this way it is possible to control the frequency. It can also be seen that frequency changes can be made instantly by simply changing the increment value.

3.3.2 Homodyne Mixer

Mixers are used for frequency conversion and are critical components in modern Radio Frequency (RF) systems. A mixer converts RF power at one frequency into power at another frequency to make signal processing easier and also inexpensive. Process of Shifting a signal frequency from one frequency range to another frequency range is called *heterodyne* (hetero means different and dyne means mixing). mixing of signal frequency with local oscillator frequency to get common intermediate frequency (IF) at the output of mixer is called *super heterodyne* mixing.
In an EW environment information about the incoming signal is unknown and generation of LO frequency is not possible with either heterodyne or super heterodyne principles mentioned above. So, a special technique is required to derive LO frequency from the received signal for the purpose of mixing. This is exactly the concept of *Homodyne* and is described below. The process of deriving LO frequency from the received signal and mixing with the same signal is called *Homodyne*.

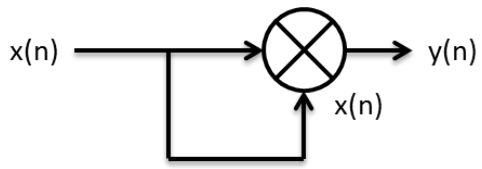


Figure 3.5: Homodyne Mixing for EW Receivers

Let us assume the received signal is x(n) and output of mixer is y(n), see figure

3.5. x(n) is the sampled sequence and is given by equation 3.7

$$x(n) = \cos(2\pi f n T_s) \tag{3.7}$$

y(n) is the product of x(n) with x(n) using homodyne principle and is described in equation 3.8

$$\begin{aligned} y(n) &= x(n) * x(n) \\ &= \cos(2\pi f n T_s) * \cos(2\pi f n T_s) \\ &= \cos(2\pi (f-f) n T_s) * \cos(2\pi (f+f) n T_s) \\ &= \cos(2\pi (0) n T_s) * \cos(2\pi (2*f) n T_s) \end{aligned} \tag{3.8}$$

It is observed in equation 3.8 that the resultant signal contains two frequency components, one is at *f=0* and the other is at *f=2f*. This shows that the us of homodyne principle translates the original frequency to low frequency component (zero frequency) and high frequency component (twice the signal frequency).

3.4 Decimation

Decimation is the process of reducing the sampling rate, this usually implies low pass filtering a signal, then down sampling. A down sampler with a down sampling factor R, where R is a positive ineteger creates an output sequence $y_d[n]$ with a sampling rate that is (1/R)th of that of the input sequence y[n](in this case y[n] is the output of mixer). Down sampling operation is implemented by keeping every Rth sample of the input sequence and removing (R-1) in between samples to generate the output sequence according to the equation 3.9.

$$y_d[n] = y[n\mathbf{R}] \tag{3.9}$$

The frequency response of down sampled sequence is given by relation 3.10. This implies that $Y_d(e^{j\omega})$ is a sum of **R** uniformly shifted and stretched versions of $Y(e^{j\omega})$, scaled by a factor of (1/R). The original shape of $Y(e^{j\omega})$ is lost due to aliasing that takes place due to under sampling. Aliasing due to a factor of **R** sampling is absent if and only if the signal y[n] is band limited to $\pm\Pi/R$. This is the reason an anti alias low pass filter is used prior to down sampling in decimation process.

$$Y_d[e^{j\omega}] = \frac{1}{R}\sum_{k=0}^{R-1} Y(e^{j\frac{(\omega-2\pi k)}{R}}) \tag{3.10}$$

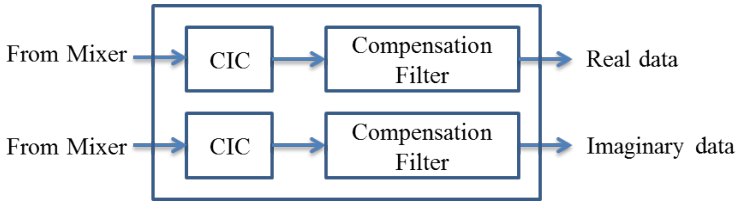


Figure 3.6: block diagram of decimation process

The output of homodyne mixer contains LF and HF components one is near zero frequency and the other is twice the signal frequency. There is a need to extract this zero frequency component from a wide band output. As the required frequency spectrum occupies very narrow band width, filter requires a sharp cut off response at the pass band edge. This makes the order of the filter to be very high as the sampling rate of the filter is the input sampling rate which is also very high (F_s = 1350 MHz in this case). The amount of hardware complexity increases because the number of multipliers and hence the area increases. So, there is a need for an efficient filter design that can keep the number of multipliers minimum and can still operate at high sampling rates. This exactly what a CIC filter does in reality. The decimation process is shown as a block diagram in figure 3.6.

3.4.0.1 CIC Filter

Cascaded Integrator-Comb (CIC) filter is a class of digital liner phase FIR filters used for the purpose of decimation or interpolation to alter the sampling rate. The present discussion is restricted to decimation as we are interested in reducing the sampling rate at the output of the mixer. The structure of CIC filter consists of cascaded ideal integrator stages operating at high sampling rate and an equal number of comb stages operating at the low sampling rate. Together, a single integrator comb pair produces a uniform FIR.

Figure 3.7 shows the basic structure of the CIC decimation filter. The integrator

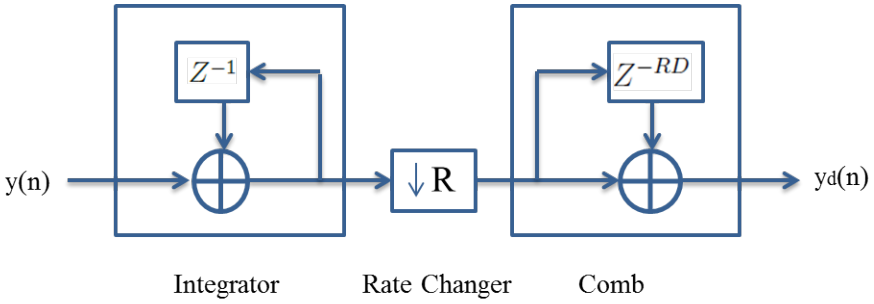


Figure 3.7: Decimation by R using CIC filter

section of CIC filters consists of L ideal digital integrator stages operating at the high sampling rate,F_s. Each stage is implemented as a one-pole filter with a unity feedback coefficient. The system function for a single integrator is given in equation 3.11

$$H_I[Z] = \frac{1}{1 - Z^{-1}} \tag{3.11}$$

The comb section operates at the low sampling rate F_s/R ,where R is the integer rate change factor. Comb section consists of L comb stages with a differential delay of D samples per stage. The differential delay is a filter design parameter used to control the filter's frequency response. In practice, the differential delay is usually held to D= 1 or 2. The system function for a single comb stage referenced

to the high sampling rate is given in equation 3.12

$$H_C[Z] = 1 - Z^{-RD} \tag{3.12}$$

There is a rate change switch between integrator and comb sections. For decimation, the switch subsamples the output of the last integrator stage, reducing the sampling rate from F_s, to F_s/R. It follows from equations 3.11 and 3.12 that the system function for the composite CIC filter referenced to the high sampling rate,F_s is

$$H_{CIC}[Z] = H_I^L[Z] * H_C^L[Z] = \frac{(1 - Z^{-RD})^L}{(1 - Z^{-1})^L} = (\sum_{k=0}^{RD-1} Z^{-k})^L \tag{3.13}$$

Equation 3.13 shows that even though a CIC has integrators in it, which by themselves have an infnite impulse response, a CIC filter is equivalent to L FIR flters, each having a rectangular impulse response. Since all of the coefcients of these FIR flters are unity, and therefore symetric, a CIC flter also has a linear phase response. Frequency response of CIC filter can be obtained by evaluating equation 3.13 at Z=$e^{j2\pi(f/R)}$ where, f is the frequency relative to the low sampling rate F_s/R. The power response of CIC filter can be given by equation 3.14 or 3.15

$$P(f) = \left(\frac{\sin(\pi D f)}{\sin(\frac{\pi f}{R})}\right)^{2L} \tag{3.14}$$

By using the relation sin(x)≈ x for small x and some algebra, we can approximate this function for large R as

$$P(f) = \left(RD\frac{\sin(\pi D f)}{\pi D f}\right)^{2L}, 0 \leq f \leq \frac{1}{D} \tag{3.15}$$

the output spectrum has nulls at multiples of f = 1/D. Thus, the differential delay D can be used as a design parameter to control the placement of nulls. For CIC decimation filters, the region around every Dth null is folded into the passband

causing aliasing errors. The response of CIC filter is shown in figure 3.8 for D=1 and D=2.

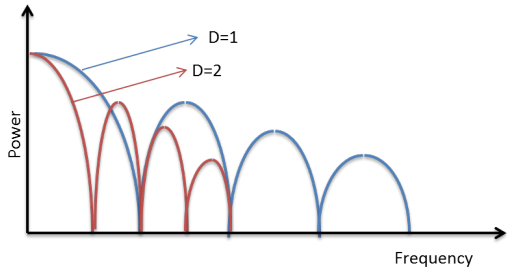


Figure 3.8: CIC filter response for D=1,2

3.4.0.2 Compensation Filter

The essential function of a decimation filter is to decrease the sampling rate and to keep the passband aliasing error within prescribed bounds. Using CIC filters, the amount of pass band aliasing error can be brought within prescribed bounds by increasing the number of stages in the filter. while increasing the number of stages improves the alias rejection, it also increases the passband droop. So, the width of the passband and the frequency characteristics outside the passband are severely limited. These limitations can be overcome by using CIC filters at high sampling rates, and to use conventional filters at the low sampling rate to compensate or shape the frequency response. In this manner, CIC filters are used at high sampling rates where economy is critical, and conventional filters are used at low sampling rates to keep the number of multipliers is low.

An FIR filter is used for the purpose of compensating the frequency response of a CIC filter. An FIR filter of length P with input $x(n)$ and output $y(n)$ is described by the difference equation3.16

$$y(n) = \sum_{k=0}^{P-1} b_k x(n-k) \tag{3.16}$$

where $\{b_k\}$ is the set of filter coefficients. Alternatively , we can express the output sequence as the convolution of the unit sample sequence *h(n)* of the system with the input signal. Thus we have

$$y(n) = \sum_{k=0}^{P-1} h(k)x(n-k) \tag{3.17}$$

where the lower and upper limits on the convolution sum reflects the causality and finite duration characteristics of the filter. The transfer function of FIR filter in Z domain is given in equation 3.18. Frequency response for this filter can be obtained by substituting $e^{j\omega}$ in place of Z in the transfer function.

$$H(Z) = \sum_{n=0}^{P-1} h(n)Z^{-k} \tag{3.18}$$

An FIR filter has linear phase if its unit sample response satisfies the condition

$$h(n) = \pm h(P-1-n), n = 0, 1, ...P-1. \tag{3.19}$$

3.5 Zoom Frequency Estimation

The estimation of zoom frequency involves two stages, one is secondary FFT to perform short length complex FFT on decimated data samples and the other is zoom frequency estimation module to get zoom frequency information. The figure 3.9 shows the blocks present in zoom frequency estimation.

Figure 3.9: blocks present in zoom frequency estimation

3.5.1 Secondary FFT

In the evaluation of FFT it is general practice to calculate the whole transform, spanning the whole frequency range, determined by the sampling rate of the time domain samples. This is done, regardless of whether we are interested only in certain regions of the frequency domain. The idea of computing Zoom frequency is to move the interesting part of the spectrum to a lower frequency band, decimate and zoom in on this predetermined frequency to cut down both computation time and storage requirements.Due to the decimation process the number of calculated secondary FFT input samples will be divided by the zooming factor R. This makes the secondary FFT to operate only on Z samples where Z is given by the following equation **??**. The words secondary FFT, complex FFT and Z point FFT refer to the same in this discussion.

$$Z = \frac{M}{R} \tag{3.20}$$

Where, M is the total length of input samples and R is the decimation factor and Z is the number samples at the input of secondary FFT. A complex FFT is used for the calculation of zoom frequency here, unlike in primary FFT where, we have used real FFT for computing the Lo frequency. Because of the complex FFT, the resulting frequency spectrum has no redundant mirrored frequency in its frequency domain. This is very useful for the estimation of signal frequencies when multiple signals are present at the input. Otherwise, extracting zoom frequencies under multi signal condition is difficult when the frequency spectrum is mirror imaged.

3.5.2 Zoom Frequency Estimation

The output of secondary FFT gives amplitude of each frequency index. This information is used to to estimate maximum amplitude using peak search technique.

The frequency index corresponding to maximum peak is multiplied with frequency resolution, given in equation 4.10 to get frequency. In order to calculate correct zoom frequency the first half of the spectrum is appended at the end of second half of the spectrum. Now,subtracting Z from the peak index gives the accurate zoom frequency. Frequency resolution of Z point complex FFT is given by

$$F_{r(zfft)} = \frac{F_s/R}{M/R} = \frac{F_s}{M} \tag{3.21}$$

3.6 RF Frequency calculation

The calculation of final RF frequency is shown symbolically in figure 3.10.By adding LO frequency (estimated using primary FFT) to the zoom frequency(estimated using secondary FFT) the fine frequency is calculated with high frequency resolution of (F_s/M). This gives the estimation of actual RF frequency incident on the receiver, which is then used to classify FMCW signals based on the pattern in which frequency is varied. In a multi signal environment the signal with maximum amplitude is extracted for classification.

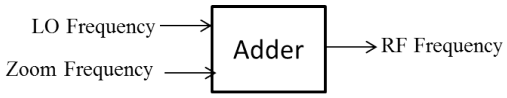


Figure 3.10: RF frequency calculation

CHAPTER 4

Implementation in FPGA

Assumptions: The implementation starts with the following few assumptions and entire discussion is carried out keeping FPGA implementation in view.(i) Input signal frequency range is 750-1250 MHz,(ii) The sampling frequency of ADC is 1350 MHz with ADC resolution of 8 bits, (iii) The sampled data is down converted to 337.5 MHz using FIFO to match maximum clock frequency of FPGA and (iv) The ADC samples are available in FPGA at a rate of 337.5 MHz

4.1 Architecture of Homodyne Zoom FFT

The architecture of HDZFFT for the implementation on FPGA is shown figure 4.1. The implementation details of each of these modules have been explained in the following subsections. It is observed that the real and imaginary components of the signal are processed separately.

4.1.1 Primary FFT

The implementation of Primary FFT or PFFT is done using FFT IP core available in Xilinx FPGA. FFT core implements the Cooley-Tukey FFT algorithm, a computationally efficient method for calculating the Discrete Fourier Transform (DFT). The FFT core performs a transform on real-valued data by setting all imaginary input samples to zero. Due to finite word length effects the noise is more prominent in the lower frequency bins. Therefore, it is recommended that the upper half (N/2+1 to N points) of the output data is used when performing a real-valued FFT.

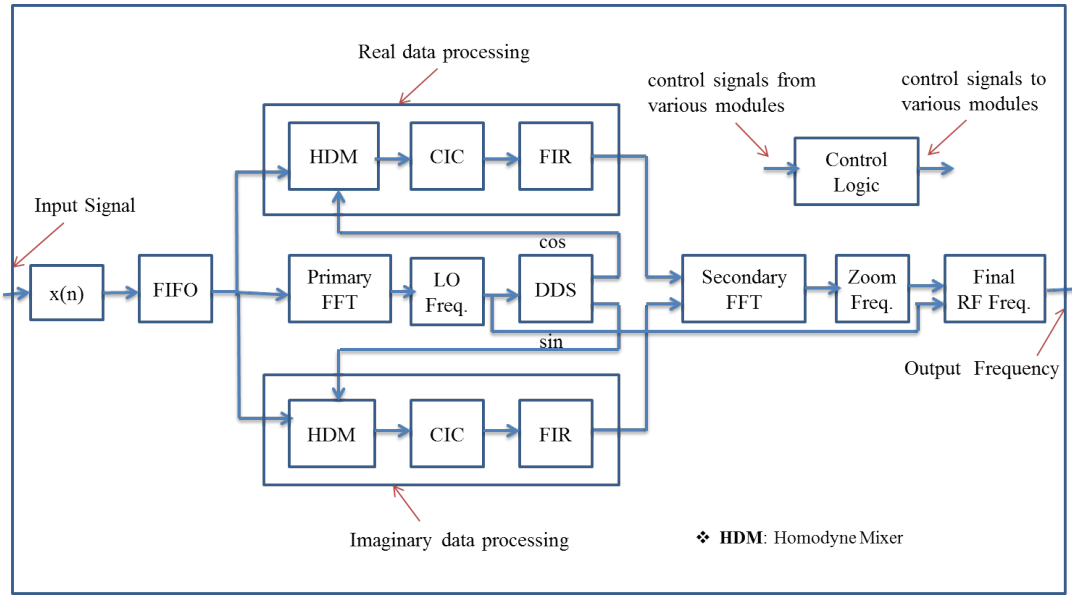


Figure 4.1: Architecture of Homodyne Zoom FFT

The Pipelined, Streaming I/O architecture is implemented, which pipelines several Radix-2 butterfly processing engines to offer continuous data processing. Each processing engine has its own memory banks to store the input and intermediate data. The core has the ability to simultaneously perform transform calculations on the current frame of data,load input data for the next frame of data,and unload the results of the previous frame of data. The data can continuously stream in and, after the calculation latency, can continuously unload the results. This arrangement is shown in figure 4.2. Detailed explanation about the core is available in Xilinx data sheet.

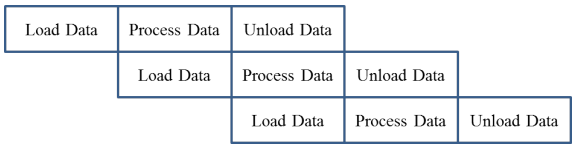


Figure 4.2: pipeline implementation of FFT

The I/O diagram of primary FFT is shown in figure 4.3 to describe input and output details of the module.

clk: Input,clock with given sampling frequency

start: Input, start begins data loading,transform calculation and then data un-

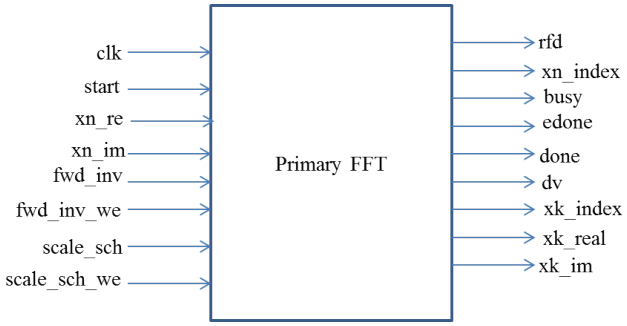


Figure 4.3: I/O details of Primary FFT

loading.

xn_re: Input, 8 bit two's complement real data

xn_im: Input, imaginary component is set to zero to perform real FFT

fwd_inv: Input,set to 1 for forward transform and 0 for inverse transform computation

fwd_inv_we: Input, enable signal for the transform computation

scale_sch: Input, specified with two bits for every pair of Radix-2 stages for scaling output magnitude

scale_sch_we: Input, Enable signal for scaling magnitude

rfd: output, It is High during the load operation

xn_index output, Index of input data

busy: output, It is High during transform computation

edone: Output,Indicates early done by going high one cycle before done

done:Output, Indicates transform completion

dv:output,It is High when valid data is available at the output.

xk_index: Output,Index of output data

xk_re: Output, Real component in two's complement

xk_im: Output, Imaginary component in two's complement

The primary FFT is simulated and synthesised for four different conditions and the results are tabulated in 4.1 and 4.2. Table 4.1 shows the logic utilisation in FPGA is minimum when inbuilt multipliers (xtreme DSP slices or DSP48) are

Table 4.1: Area utilisation in FPGA

Multiplier	Scaling	Slice Reg	Slice LUT	LUT FF	BRAM	DSP48
CLB Logic	unscaled	4844	4323	5263	2	0
CLB Logic	Scaled	4202	3782	4538	2	0
XDSP	unscaled	2574	2345	2874	2	12
XDSP	Scaled	2262	2070	2554	2	12

Table 4.2: Timing utilisation in FPGA

Multiplier	Scaling	Latency(clock cycles)
CLB Logic	unscaled	1641
CLB Logic	Scaled	1646
XDSP	unscaled	1637
XDSP	Scaled	1642

used for multiplication with scaled arithmetic. The area is maximum when CLB logic is used for multiplication instead of xtreme DSP slices without scaling the output magnitude. This analysis helps in choosing suitable FPGA for the given application.

4.1.2 LO Frequency Estimation

The two's complement output from primary FFT is converted to absolute magnitude. only real component is used to compute absolute magnitude, as the imaginary value is zero for real FFT. using peak search technique peak value is estimated and the corresponding frequency index is multiplied with frequency resolution to calculate the coarse frequency. I/O diagram of LO frequency module is given in figure 4.4. *dv_pfft* acts as enable signal for the computation of LO Frequency. *LO_Freq_MHz_out* is used for the purpose of tuning DDS as the DDS core accepts frequency in MHz only. *LO_Freq_KHz_out* is used for calculating final RF frequency in KHz. Area utilisation in FPGA for this module is indicated in table 4.3

clk: Input,clock with given sampling frequency

rst: Input, reset

dv_pfft: Input to LO freq module and output from PFFT module,indicates the

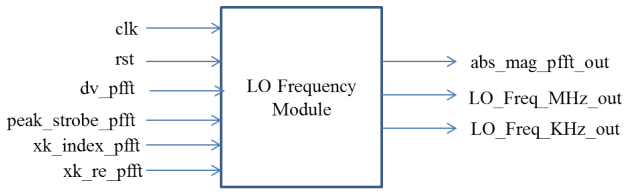


Figure 4.4: I/O details of LO frequency Module

Table 4.3: Area utilisation in FPGA

Module	Slice Reg	Slice LUT	LUT FF	BRAM	DSP48
LO Freq. Module	57	252	252	0	0

validity of PFFT output when this signal is high

peak_strobe_pfft: Input to LO freq module and output from control module, PFFT output is latched when this signal is high

xk_index_pfft: Input to LO freq module and output from PFFT module, Frequency index of PFFT

xk_re_pfft: Input to LO freq module and output from PFFT module, Real output of PFFT

abs_mag_pfft_out: Output of LO freq module, Absolute magnitude of PFFT

LO_Freq_MHz_out: Output of LO freq module, Frequency for tuning DDS

LO_Freq_KHz_out: Output of LO freq module,Frequency for final RF frequency calculation

4.1.3 Direct Digital Synthesiser

DDS (Direct Digital Synthesizer) generates a sine/cosine samples at a given frequency directly using digital techniques. A DDS consists of a Phase generator and a SIN/COS Lookup Table. The basic operation of DDS is explained in earlier sections. So, only implementation details are discussed here.

The DDS core can generate maximum frequency of 400 MHz as per the data sheet. But, the LO frequency measured using PFFT is in the range 750-1250 MHz.

The clock frequency used in DDS is 337.5 MHz which is $F_s/4$. So, if the the LO frequency is used directly to tune DDS at this rate then there is chance of aliasing. To avoid this problem the LO frequency is scaled by 4 to bring down the frequency range from 750-1250 MHz to 187.5-312.5 MHz which will produce identical samples as produced by original F_s. This can be explained using simple equation 4.1. Let us consider a signal x(n) having frequency f and sampling frequency F_s

$$x(n) = \sin(2\pi f n T_s) = \sin(\frac{2\pi f n}{F_s}) = \sin(\frac{2\pi (f/4) n}{(F_s/4}) \tag{4.1}$$

The equation 4.1 shows when the signal frequency (f)and the sampling frequency (F_s) are scaled by same amount the samples generated are identical to the samples generated without scaling. The following steps are involved to generate digital samples corresponding to LO frequency. 1) Interpolation of frequency, 2) Calculation of phase increment ($\Delta\Theta$) using equation 4.2, 3) selection of the number of bits ($B_{\Theta(n)}$) to be used in phase accumulator, 4) Apply phase increment as input and generate sine/cosine samples at the output of DDS.

$$\Delta\Theta = \frac{(LO\_Freq)2^{B_\Theta(n)}}{f_{clk}} \tag{4.2}$$

where $\Delta\Theta$ is phase increment value to produce sine/cosine waveform, LO_Freq is the frequency (in MHz) estimated using PFFT in previous section, $B_\Theta(n)$ is the number of bits used in phase accumulator,f_{clk} is the sampling clock frequency of DDS module. Width of sine and cosine outputs is chosen as 8 bits to produce 48 dB dynamic range. $B_\Theta(n)$ is of 14 bits wide to produce $2^{14} = 16384$ samples to match with input data length. $B_\Theta(n)$ can also be calculated using equation 4.3.

$$B_{\Theta(n)} = \log_2 \frac{f_{clk}}{\Delta f_{dds}} \tag{4.3}$$

Table 4.4: FPGA utilisation for DDS

Module	Slice Reg	Slice LUT	LUT FF	BRAM	Latency
DDS for Sine/Cosine	47	18	62	1	3 cycles

where Δf_{dds} is DDS frequency resolution and is given by equation 4.4

$$\Delta f_{dds} = \frac{f_{clk}}{2^{B\Theta(n)}} = \frac{337.5MHz}{2^{14}} = 20.599KHz \tag{4.4}$$

From the above discussion we can say, the dynamic range of output can be enhanced by increasing the resolution of sine/cosine output. High frequency resolution at the output of DDS output can be achieved by increasing Phase width of accumulator, this increases the depth of LUT. The I/O diagram of DDS is shown in figure 4.5 and described below.

CE: Input,enable signal

Figure 4.5: DDS I/O Diagram

CLK: Input, clock signal

PINC_IN: Input, phase increment calculated based on LO frequency

RDY: Output, indicated the availability of data at the output

SINE: Output, sine wave output in two's complement form

COSINE: Output, cosine wave output in two's complement form

PHASE_OUT: Output, phase output

FPGA utilisation for DDS module is shown in table 4.4. The latency is the time taken to produce the sine/cosine wave at the output once the input is applied.

Table 4.5: FPGA utilisation for mixer

Module	Slice Reg	Slice LUT	LUT FF	BRAM	DSP48E	latency
with Multiplier	0	0	0	0	1	0 cycles
without Multiplier	0	0	72	0	0	0 cycles

4.1.4 Homodyne Mixer

Homodyne mixing is a process of mixing an input signal with it self to produce zero frequency. The mixer operation is implemented using a multiplier in FPGA. The implementation scheme is shown in figure 4.6. As DDS generates sine and cosine signals separately and not a combined complex signal, real and imaginary components are processed separately. The upper multiplier in the figure processes real data and the lower multiplier processes imaginary data. Both of these operations happen in synchronous with F_s. Area and timing utilisation in FPGA are

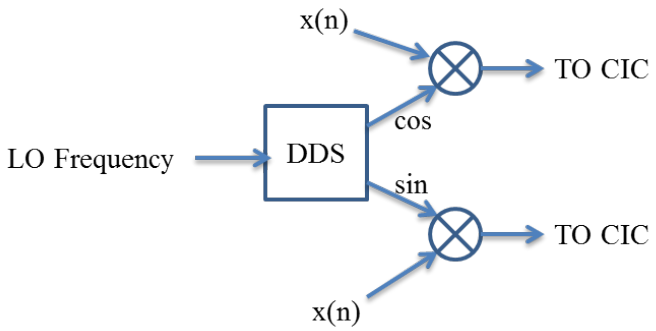


Figure 4.6: Implementation of mixer using multipliers

shown in table 4.5.

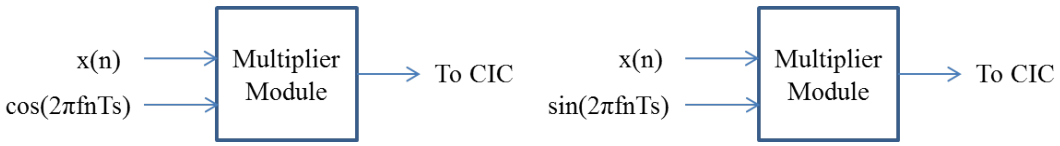


Figure 4.7: I/O diagram of Multiplier core for real and imaginary data samples

4.1.5 CIC Filter

Cascaded Integrator Comb (CIC) filters are used for decimation purpose to implement large sample rate change in hardware design. Implementation of CIC filter structure consists of only adders, subtracters and delay elements. The hardware-efficient implementation of CIC filters makes it suitable for proposed design.

Figure 4.8 shows I/O diagram of CIC filter and brief description of the signals

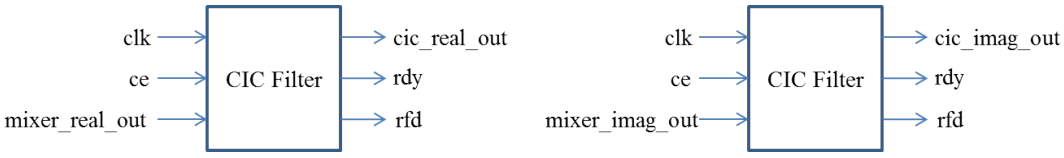


Figure 4.8: I/O diagram of CIC filter for real and imaginary components

given below.

ce: Input,clock enable

clk: Input, clock with given sampling frequency

mixer_real_out: Output, real component at mixer output

mixer_imag_out: Output, imaginary component at mixer output

rdy: Output, indicates that a new filter output sample is available on the output port.

rfd: Output, indicates when the filter can accept a new input sample

cic_real_out: Output, real component at CIC filter output

cic_imag_out: Output, imaginary component at CIC filter output

Xilinx IP core is used for the implementation of CIC filter. To get minimum of 40 dB attenuation in the stop band three stage CIC filter chosen. Block diagram of a 3 stage CIC filter is represented in figure 4.9. Area and timing utilisation for 3 stage CIC filter is shown in table 4.6.

Effect of D: The inegrator stage works at F_s and the comb stage works at (F_s)/R. Frequency response for D=1 and D=2 for R=64 is shown in figures 4.10 and 4.11 where D is the differential delay parameter. It is very clear from the plots that the

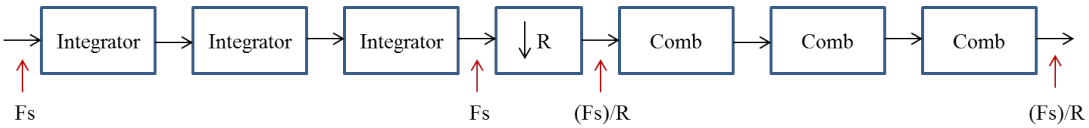


Figure 4.9: Three stage CIC filter

position of Nulls in the magnitude response (transfer function zeros) are present at integer multiples of f =1/(RD). Thus, the differential delay parameter, D, can be used as a design parameter to control the placement of the nulls. Besides the effect on the placement of the response nulls, increasing D also increases the amount of attenuation in side lobes of the magnitude response.

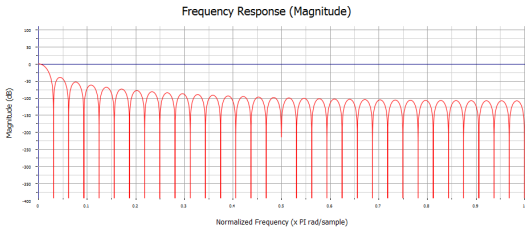


Figure 4.10: CIC response for D=1

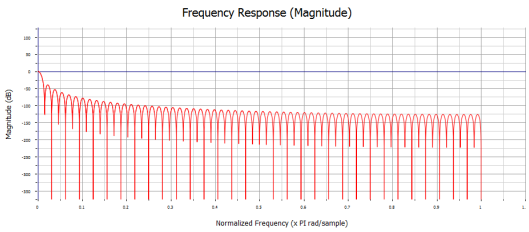


Figure 4.11: CIC response for D=2

Effect of R:The rate change parameter R can also be used to control the frequency response of the CIC filter. Increasing the rate change increases the length of the cascaded unit-amplitude, rectangular window of length R*D. This results in an increase in attenuation and decrease of the width of the response side lobes.
Effect of L: The number of stages parameters, L, can also be used to affect the CIC filter magnitude response. This effect can be understood from the fundamental concept of a cascade of L filtering stages, each with an impulse response of a unit-amplitude, rectangular window. The larger the number of cascaded stages, the more attenuated the magnitude response side lobes become. Increasing L has the effect of increasing the order of the zeros in the frequency response. This, in turn, increases the attenuation at frequencies in the locality of the zero. As the order of the zeros increase, the passband droop also increases, thus narrowing the filter bandwidth. As the increased droop is not acceptable it is corrected using an additional compensating filter after the CIC decimator.

Table 4.6: FPGA utilisation for 3 stage CIC filter

Multiplier	O/P mode	Slice Reg	Slice LUT	LUT FF	DSP48	latency(cycles)
CLB Logic	truncated	314	278	428	0	13
CLB Logic	full precision	446	376	595	0	13
XDSP	truncated	143	89	157	4	15
XDSP	full precision	219	119	233	4	15

Effect of B_{max}: The CIC data path undergoes internal register growth that is a function of all the design parameters: L, D, R in addition to the input sample precision B. The output bit width of a CIC decimator with full precision is given by equation 4.5

$$B_{max} = \lceil L\log_2 RD + B \rceil \tag{4.5}$$

where $\lceil\rceil$ denotes the ceiling operator. The CIC compiler supports both full and limited precision output. For full precision, the CIC decimator implementation uses Bmax bits internally for each of the integrator and differentiator stages. This introduces no quantization error at the output. For limited precision (that is, output bit width less than Bmax), the registers in the integrator and comb stages are sized to limit the quantization noise variance at the output. Consequently, the hardware resources in a CIC decimator implementation can be reduced when using limited precision output at the cost of quantization noise. This ability to trade off resources and quantization noise is important to achieve an optimum implementation.

4.1.6 Compensation Filter

An FIR low pass filter after CIC filter is used to compensate the droop in frequency characteristics at cut off frequency. The decimated output from CIC filter is given as input to FIR filter at F_s/R and xilinx IP core is used for the implementation. A conventional single rate FIR filter of the computes the convolution sum defined

in equation 4.6.

$$y(n) = \sum_{k=0}^{L-1} b_k x(n-k) \tag{4.6}$$

where $\{b_k\}$ is the set of filter coefficients and L is the number of filter coefficients. The actual FPGA realization of this computation can be done using either Multiply and Accumulate (MAC) or Distributed arithmetic. Two MAC architectures are available in the FIR Compiler, one that implements a Systolic filter structure (SMAC) and the other a Transpose filter structure (TMAC). The number of multipliers required to implement a filter is determined by calculating the number of multiplies required to perform the computation and then dividing by the number of clocks available to process each input sample.

Systolic MAC: The architecture is directly supported by the XtremeDSP Slice and results in area-efficient and high performance filter implementations. The structure also extends to exploit coefficient symmetry offering further resource savings.

Transpose MAC:This architecture is also directly supported by the XtremeDSP Slice. This structure offers a low latency implementation,and for some configurations can also offer extra resource savings over the Systolic structure. It does not require an accumulator and can use fewer data memory resources, although it does not exploit coefficient symmetry.

Distributed Arithmetic:An efficient technique for calculation of inner product or multiply and accumulate(MAC). MAC using multipliers are fast but they consume considerable hardware in FPGA. In their most obvious and direct form, DA-based computations are bit-serial in nature. The basic operations required are a sequence of table look-ups, additions, subtractions, and shifts of the input data sequence. All of these functions efficiently map to FPGAs. The architecture of DA accomplishes MAC without an explicit multiplier.

In a conventional MAC based FIR realization, the sample throughput is coupled to the filter length. With a DA architecture,the system sample rate is related to the bit precision of the input data samples. Each bit of an input sample must be indexed and processed in turn before a new output sample is available. For B-bit precision input samples, B clock cycles are required to form a new output sample for a non-symmetrical filter, and B+1 clock cycles are needed for a symmetrical filter. The rate at which data bits are indexed occurs at the bit-clock rate. The bit-clock frequency is greater than the filter sample rate (fs) and is equal to BF_s for a non-symmetrical filter and (B+1)F_s for a symmetrical filter. In a conventional method, required number of MAC operations is implemented using a time-shared or scheduled MAC unit. The filter sample throughput is inversely proportional to the number of filter taps. As the filter length is increased, the system sample rate is proportionately decreased. This is not the case with DA-based architectures. The filter sample rate is decoupled from the filter length. As the filter length is increased in a DA FIR filter, more logic resources are consumed, but throughput is maintained.

Filter Coefficients: The filter coefficients are supplied to the FIR Compiler using a coefficient file with a .coe extension which is generated using FDA (Filter Design and Analysis) tool from Matlab. This is an ASCII text file with a single line header that defines the radix of the number representation used for the coefficient data,followed by the coefficient values themselves.

Ex: radix=coefficient_radix;

coefdata= b(0),b(1),b(2),..b(N-1);

Sampling frequency for FIR core is F_s/R = 1350 MHz/64 = 21.09375 MHz, the stop band frequency is 10.546875 MHz and the pass band frequency is 9 MHz. The stop band attenuation required is minimum of 25 dB. The fixed point arithmetic is used for the generation of coefficients. Coefficient width of 6 bits and input sample width of 16 bits chosen in the FDA tool. Then the coefficients required for FIR LPF is generated and stored as .coe file. These coefficients are then fed to FIR

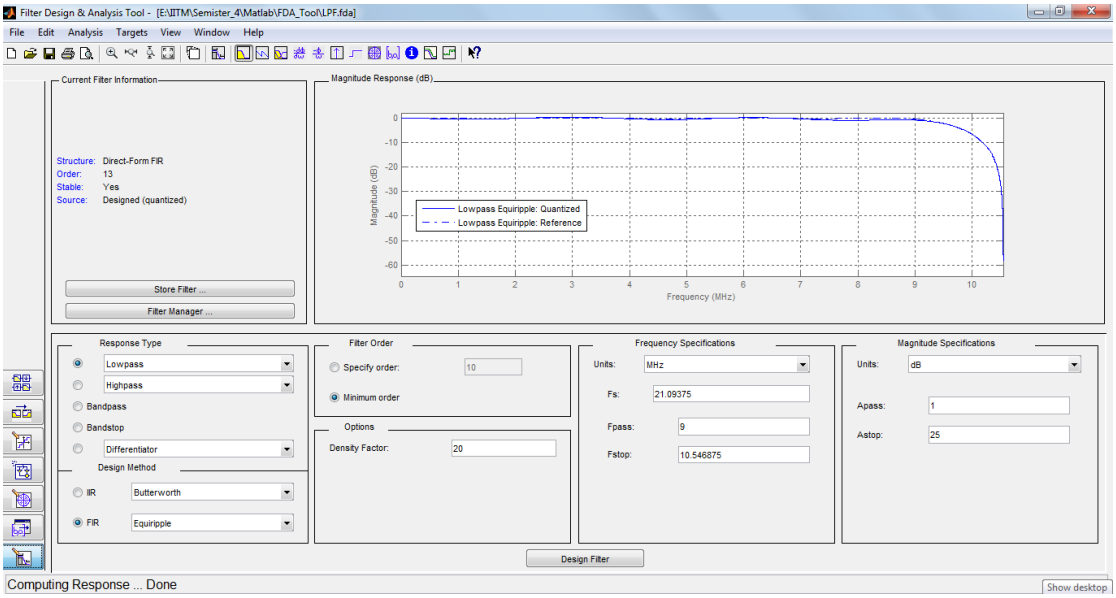


Figure 4.12: FIR filter response using FDA

IP core to implement LPF in FPGA. The area and timing utilisation in FPGA is given in table4.7. For the present design Distributed Arithmetic structure is used to minimise number of multipliers. I/O diagram of FIR LPF using DA is shown in figure 4.13 and description of signals given below.

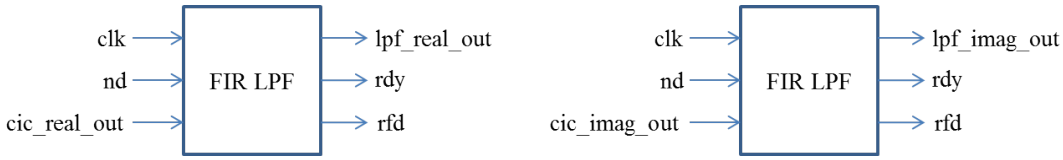


Figure 4.13: I/O diagram of compensating FIR LPF

nd: Input,enable signal for LPF coming from Control Module

clk: Input, clock with given sampling frequency

cic_real_out: Input, real component at cic output

cic_imag_out: Input, imaginary component at cic output

rdy: Output, indicates the LPF output is ready from this point onwards.

rfd: Output, if '1' accepts data and producs output

lpf_real_out: Output, real component at FIR LPF filter output

lpf_imag_out: Output, imaginary component at FIR LPF filter output

Table 4.7: Area utilisation in FPGA

Structure	O/P Width	Slice Reg	Slice LUT	LUT FF	DSP48E	Latency(cycles)
TMAC	23	69	33	70	14	6
SMAC	23	2823	207	284	7	13
DA	26	1158	821	1168	0	10

4.1.7 Secondary FFT

The number of samples after decimation is Z=M/R = 16384/64 = 256. Therefore Z is the number samples at the input of secondary FFT. We have seen that the real and imaginary components are processed separately using identical modules. Apply real and imaginary outputs from FIR LPF to FFT input to perform complex FFT. Because of the complex FFT, the resulting frequency spectrum has no redundant frequency in its frequency spectrum. The mathematical equation for the calculation of Z point secondary FFT is given by following equation 4.7

$$X(k) = \sum_{n=0}^{Z-1} x(n) e^{\frac{-j2\Pi kn}{Z}}, k = 0, 1, ...Z-1 \tag{4.7}$$

where x(n) is the LPF output in complex form,Z is the length of secondary FFT,k is frequency index

Implementation of secondary FFT is carried out using xilinx IP core. For fixed-point inputs, the input data is a vector of Z complex values represented as dual b_x-bit twos-complement numbers, that is, b_x bits for each of the real and imaginary components of the data sample. where b_x is 25 bits for the present design. Similarly, the phase factors b_w is 8 bits wide. All memory is on-chip using either block RAM or distributed RAM. The Z element output vector is represented using b_y bits for each of the real and imaginary components of the output data. where B_y can be equal to b_x if scaled arithmetic is used or higher than b_x for unscaled arithmetic. Input data is presented in natural order and the output data is also in natural order.

The FFT core uses the Radix-4 and Radix-2 decompositions for computing the DFT. The decimation-in-frequency (DIF) method is used for the Pipelined,Streaming I/O architecture. When using Radix-4 decomposition, the N-point FFT consists of log_4Z stages,with each stage containing Z/4 Radix-4 butterflies. Point sizes that are not a power of 4 need an extra Radix-2 stage for combining data. A Z-point FFT using Radix-2 decomposition has log_2Z stages, with each stage containing Z/2 Radix-2 butterflies. For Radix-2, the bit growth can be up to 2 bits. This bit growth can be handled by 1)Performing the calculations with no scaling and carrying all significant integer bits to the end of the computation, 2)Scaling at each stage using a fixed-scaling schedule. The width of the output is

$$Width_{o/p} = inputwidth + \log_2 Z + 1 \tag{4.8}$$

This accommodates the worst case scenario for bit growth. When using scaling, a scaling schedule is used to divide by a factor of 1, 2, 4, or 8 in each stage. As a result of the scaling applied in the FFT implementation, the transform computed is a scaled transform. The scale factor s is defined as

$$s = 2^{\sum_{r=0}^{\log Z-1} b_i} \tag{4.9}$$

where, b_i is scaling applied in ith stage and the scaling results in the final output sequence being modified by the factor 1/s. The I/O diagram of secondary FFT (or complex FFT or Z point FFT) is shown in figure 4.14 and details of each I/O is discussed below.

clk: Input,clock with sampling frequency of F_s/R

start: Input, start begins FFT when lpf output becomes ready

xn_re: Input, two's complement real data from FIR LPF

xn_im: Input, two's complement imaginary data from FIR LPF

fwd_inv: Input,set to 1 for forward transform and 0 for inverse transform computation

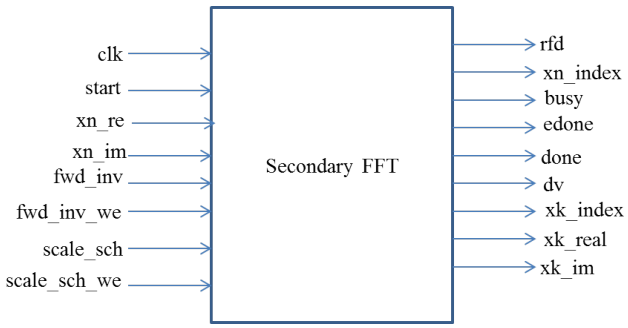


Figure 4.14: I/O diagram of Secondary FFT

Table 4.8: Area utilisation in FPGA

Multiplier	Scaling	Slice Reg	Slice LUT	LUT FF	BRAM	DSP48
CLB Logic	unscaled	5525	4975	5952	2	0
CLB Logic	Scaled	5191	4786	5556	1	0
XDSP	unscaled	3054	2867	3366	2	9
XDSP	Scaled	2932	2719	3243	1	9

fwd_inv_we: Input, enable signal for the transform computation

scale_sch: Input, specified with two bits for every pair of Radix-2 stages for scaling output magnitude

scale_sch_we: Input, Enable signal for scaling magnitude

rfd: output, It is High during the load operation

xn_index output, Index of input data

busy: output, It is High during transform computation

edone: Output,Indicates early done by going high one cycle before done

done:Output, Indicates transform completion

dv:output,It is High when valid data is available at the output.

xk_index: Output,Index of output data

xk_re: Output, Real component in two's complement

xk_im: Output, Imaginary component in two's complement

Area and timing utilisation of secondary FFT in FPGA is given in 4.8 and 4.9

Table 4.9: Timing utilisation in FPGA

Mult	Scaling	Latency(clock cycles)
CLB Logic	unscaled	863
CLB Logic	Scaled	867
XDSP	unscaled	860
XDSP	Scaled	864

4.1.8 Zoom Frequency Estimation

Using peak search technique peak value is estimated and the corresponding frequency index is multiplied with frequency resolution to calculate the zoom frequency. I/O diagram of zoom frequency module is given in figure 4.15. Area utilisation in FPGA for this module is indicated in table 4.10

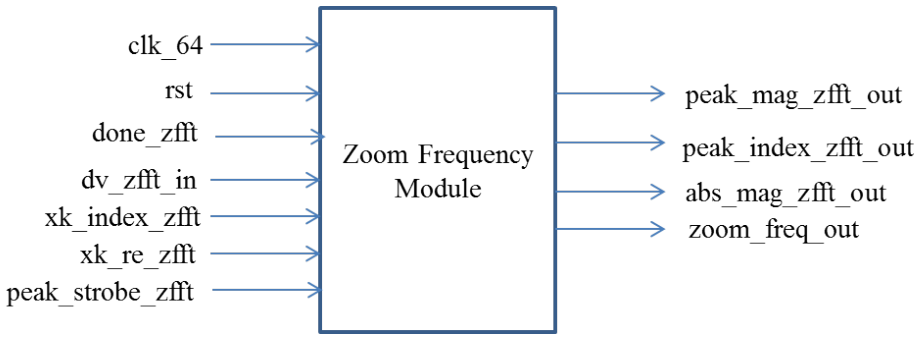


Figure 4.15: I/O details of zoom frequency module

clk_64: Input,clock with sampling frequency of F_s/R

rst: Input, reset

dv_zfft:Input,indicates the validity of secondary FFT output when this signal is high

done_zfft: Input, when it is high, indicates the the output from secondary FFT is available from this point onwards.

peak_strobe_zfft: Input, when this is high, zoom frequency is latched and also used as enable signal for spectrum inversion

xk_index_zfft: Input, frequency index from secondary FFT

xk_re_zfft: Input, real component from secondary FFT output

abs_mag_zfft_out: Output, absolute magnitude of secondary FFT

Table 4.10: Area utilisation in FPGA

Module	Slice Reg	Slice LUT	LUT FF	BRAM	DSP48
Zoom Frequency Module	66	138	151	0	0

peak_index_zfft_out:Output, index of peak value for secondary FFT

zoom_freq_out: Output,estimated zoom frequency

In order to calculate correct zoom frequency the first half of the spectrum is appended at the end of second half of the spectrum. Now,subtracting Z from the peak index gives the accurate zoom frequency. Frequency resolution of Z point complex FFT is given by

$$F_{r(zfft)} = \frac{F_s/R}{M/R} = \frac{F_s}{M} = 82.397KHz \tag{4.10}$$

It is clear from the from equation 4.10 that the frequency resolution achieved by Z point secondary FFT is same as a conventional FFT with M samples.

4.1.9 RF Frequency calculation

The Zoom frequency estimated using secondary FFT is added with coarse frequency estimated using primary FFT to get actual frequency of RF signal. If the LO frequency measurement is higher than the actual frequency the zoom frequency is negative and if the LO frequency is lower than the actual signal frequency then the zoom frequency is positive. So, the sign of the zoom frequency is important while adding it to the LO frequency to calculate accurate signal frequency. An adder/subtracter is used for this purpose and the I/O diagram of this given in figure 4.16. The adder/subtracter block consumes 10 LUTs for the implementation.

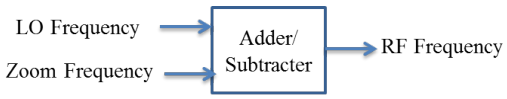


Figure 4.16: Adder/Subtracter to calculate actual RF signal

Table 4.11: FPGA Area Utilisation of Homodyne Zoom FFT

Hardware type	Slice logic	Slice LUT	LUT FF	BRAM	DSP48E
Design with Multipliers	8831	7952	1069	9	23
Design without Multipliers	13572	12182	15188	9	0

4.2 Multiplier vs Multiplier less design

Xilinx FPGAa are available in three forms (i) SX series:These FPGAs are rich in DSP slices which are suitable for DSP applications.(ii) LX series: These FPGAs are suiatble for logic intensive applications with more CLB logic and less in DSP slices.(iii) FX series: These FPGAs are suitable for embedded applications and both CLB logic and DSP slices are limited in these FPGAs. SX series FPGAs are costlier than other two categories. So,to choose proper FPGA it is better to understand area utilisation with and without multipliers.

The hardware utilisation for individual modules have been discussed thoroughly in previous sections. The overall area utilisation of Homodyne Zoom FFT in FPGA with and with out multipliers is shown in table 4.11. It is observed that the design takes least amount logic when multipliers used and consumes maximum logic when the design is multiplier less. This analysis helps in choosing proper FPGA for a given application. As multiplier is the costly element in FPGA it is advisable to use minimum number of multipliers in any FPGA based design, if the latency is manageable.

CHAPTER 5

Simulation Results

5.1 Simulations using Matlab

The concept of Homodyne Zoom FFT is simulated using Matlab for various test conditions. Some of the test results are presented in this chapter. The following parameters were used for the simulations during testing.

- Sampling Technique : Band Pass Sampling
- Sampling Frequency (F_s): 1350 MHz
- I/P signal Frequency Range: 750-1250 MHz
- Signal Type: CW, FMCW with AWGN
- Modulation Period (t_m): 1 ms
- Length of data (M): 16384 samples
- i/p sample width: 8 bits
- Length of Primary FFT (N): 512 samples
- Decimation Factor (R): 64
- Differential Delay parameter (D): 2
- Length of Compensation FIR filter (L): 13
- Length of Secondary FFT (Z): 256

5.1.1 Single Signal with AWGN

The basic performance of Homodyne Zoom FFT is tested by applying a single signal with AWGN at the input and observed the detected signal with frequency variation at the output.The frequency spectrum is plotted for the applied input signal to monitor the frequency components present in the signal. Also frequency variation at the output is plotted against the number of data segments. Each data segment in the plot (on X-axis) is of 12 us duration (M*t_s = 16384*0.74 ns = 12.136 us). So, this gives an idea of frequency pattern over a period of time. Various test conditions for a single signal are presented here. Figures 5.1,5.2,5.3,5.4,5.5 shows the extraction of CW, FMCW with positive ramp (up ramp), negative ramp (down ramp), triangular pattern and hopping patterns respectively in presence of AWGN at an SNR of -7 dB.

5.1.1.1 CW without FM

Input Signal: F_0: 1000 MHz, Δ_F: 0 MHz, SNR: -7 dB

Output Signal: F_0:1000.1 MHz, Error: 0.1 MHz

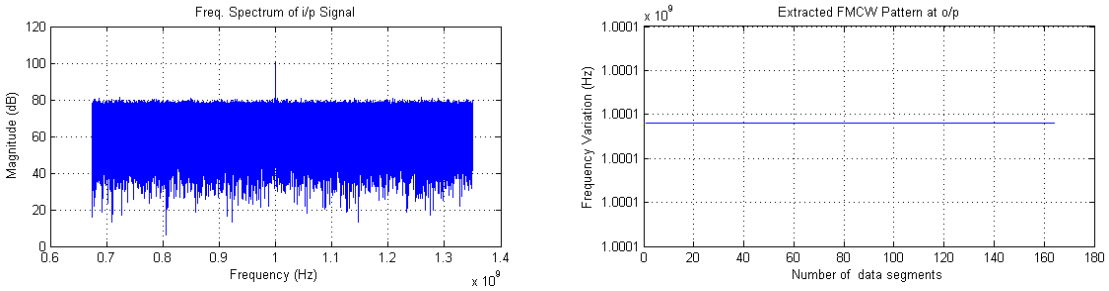


Figure 5.1: CW without modulation

5.1.1.2 FMCW with Positive Ramp

Input Signal: F_c: 1000 MHz, Δ_F: 1 MHz, SNR: -7 dB

Output Signal: Δ_F: 0.988 MHz, Error: 0.0112 MHz

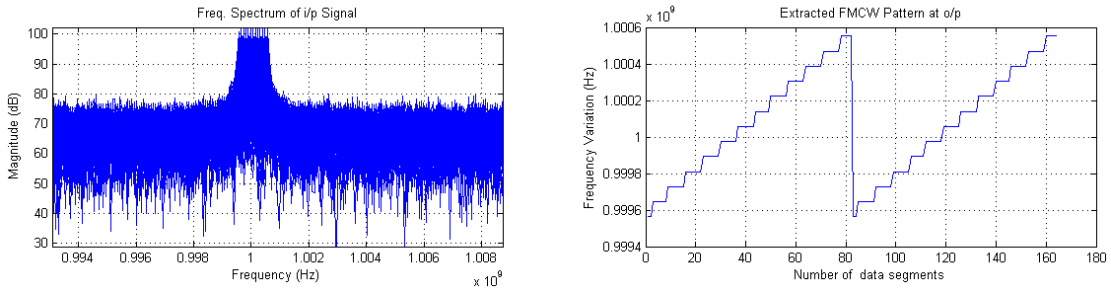


Figure 5.2: FMCW with Positive Ramp Pattern

5.1.1.3 FMCW with Negative Ramp

Input Signal: F_c: 1000 MHz, Δ_F: 10 MHz, SNR: -7 dB

Output Signal: Δ_F: 9.805 MHz, Error: 0.195 MHz

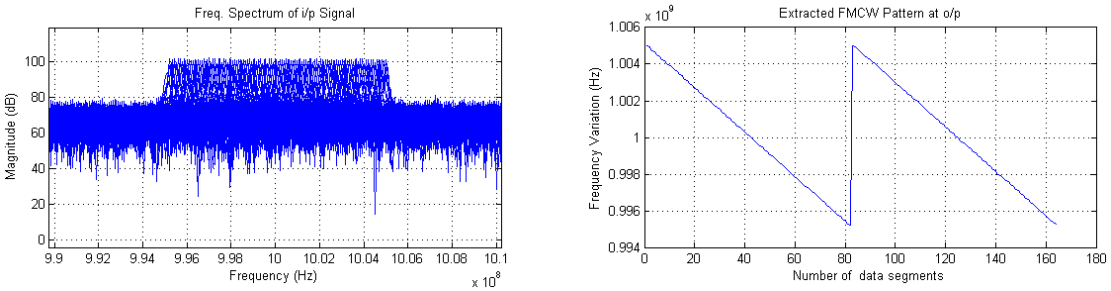


Figure 5.3: FMCW with Negative Ramp Pattern

5.1.1.4 FMCW with Triangular Pattern

Input Signal: F_c: 1000 MHz, Δ_F: 500 MHz, SNR: -7 dB

Output Signal: Δ_F: 498.26 MHz, Error: 1.74 MHz

5.1.1.5 FMCW with Hopping

Input Signal: F_1: 750 MHz,F_2: 1250 MHz, Δ_F: 500 MHz, SNR: -7 dB

Output Signal:Δ_F: 499.99 MHz, Error: 0.012 MHz

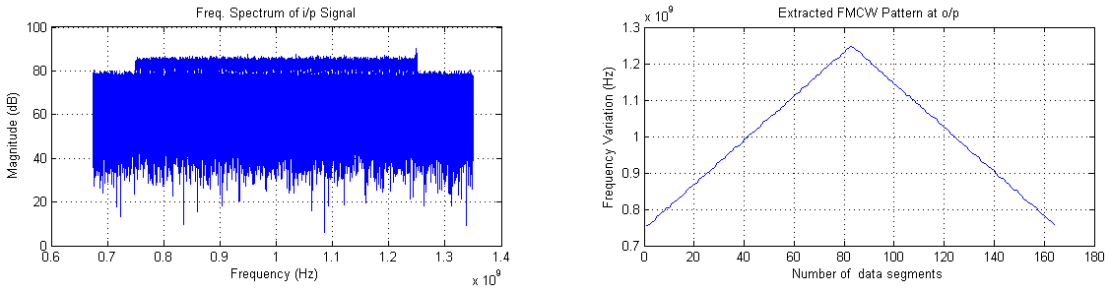


Figure 5.4: FMCW with Triangular Pattern

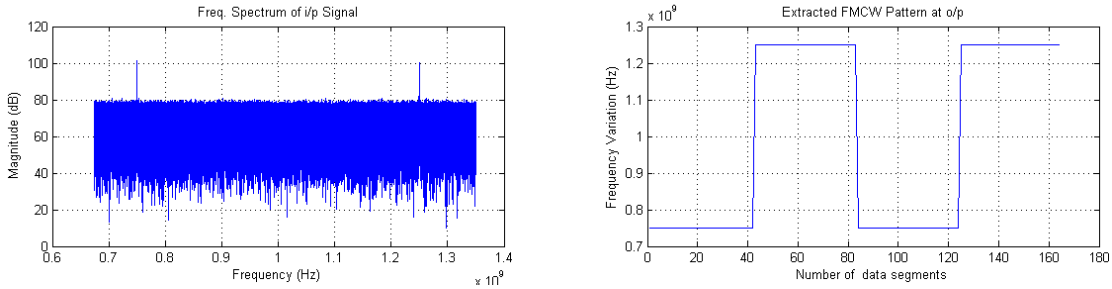


Figure 5.5: FMCW with Hopping between two frequencies

5.1.2 Two Simultaneous Signals with AWGN

The performance of Homodyne Zoom FFT is tested by applying two simultaneous signals with AWGN at the input and observed the detected signal with frequency variation at the output. The signal with maximum amplitude is detected in case of two signals at the input. The SNR requirement for two signal condition is 2 dB higher than single signal case. The signal with higher amplitude must have an amplitude difference of 3 dB atleast compared to the other signal to get faithful frequency variation at the output. Various test conditions for two signals simultaneously are presented here.

5.1.2.1 CW and Triangular FMCW

1) Extraction of CW signal:

Input Signal: CW signal power is 3 dB higher than Triangular FMCW signal

CW:1000 MHz, SNR: -5 dB

Triangular FMCW: F_c: 1000 MHz, Δ_F: 100 MHz, SNR: -5 dB

Output Signal: F_{cw}: 1000.1 MHz, Δ_F: 0 MHz, Error: 0.0579 MHz

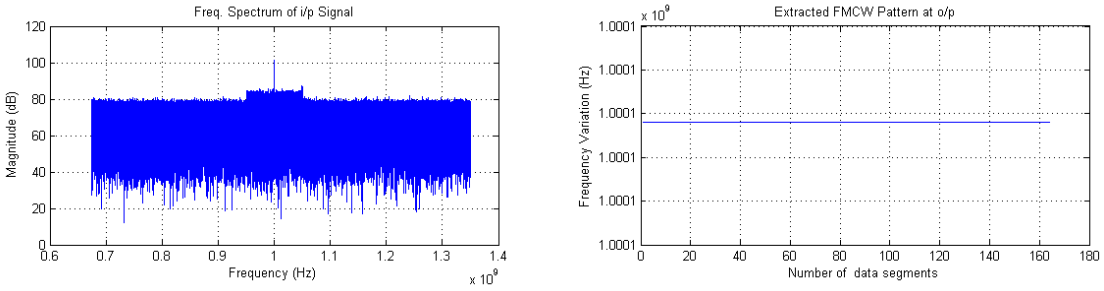


Figure 5.6: Extraction of CW from two simultaneous signals

2) Extraction of Triangular FMCW signal:

Input Signal: Triangular FMCW signal power is 3 dB higher than CW signal

CW:1000 MHz, SNR: -5 dB

Triangular FMCW: F_c: 1000 MHz, Δ_F: 100 MHz, SNR: -5 dB

Output Signal: Δ_F: 98.87 MHz, Error: 1.123 MHz

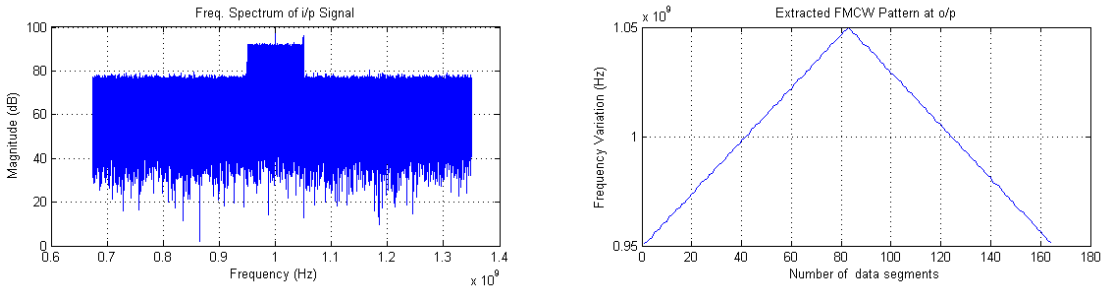


Figure 5.7: Extraction of Triangular FMCW from two simultaneous signals

5.1.2.2 FMCW with Positive & Negative Ramp

1) Extraction of FMCW with positive ramp:

Input Signal: FMCW with positive ramp is 3 dB higher than negative ramp

Positive Ramp: F_c: 1000 MHz, Δ_F: 1 MHz, SNR: -5 dB

Negative Ramp: F_c: 1000 MHz, Δ_F: 10 MHz, SNR: -5 dB

Output Signal: Δ_F: 0.988 MHz, Error: 0.112 MHz

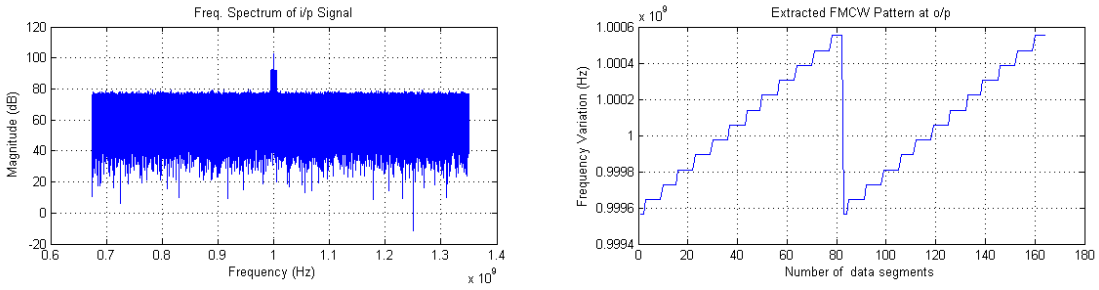


Figure 5.8: Extraction of FMCW with Positive Ramp from two simultaneous signals

2) Extraction of FMCW with negative ramp:

Input Signal: FMCW with negative ramp is 3 dB higher than positive ramp

Positive Ramp: F_c: 1000 MHz, Δ_F: 1 MHz, SNR: -5 dB

Negative Ramp: F_c: 1000 MHz, Δ_F: 10 MHz, SNR: -5 dB

Output Signal: Δ_F: 9.888 MHz, Error: 0.112 MHz

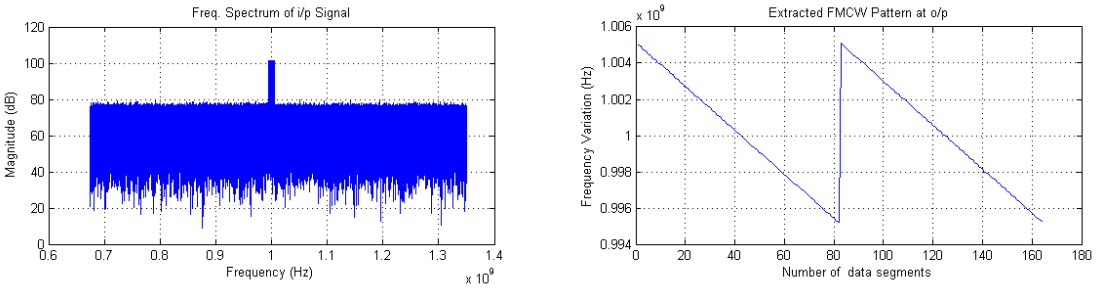


Figure 5.9: Extraction of FMCW with Negative Ramp from two simultaneous signals

5.1.2.3 FMCW with Triangular & Hopping Patterns

1) Extraction of FMCW with hopping:

Input Signal: FMCW with Hopping is 3 dB higher than Triangular FMCW

Triangular FMCW: F_c: 1000 MHz, Δ_F: 200 MHz, SNR: -5 dB

FMCW with Hopping: F_1: 750 MHz, F_2: 1250 MHz, Δ_F: 500 MHz, SNR: -5 dB

Output Signal: Δ_F: 499.99 MHz, Error: 0.011 MHz

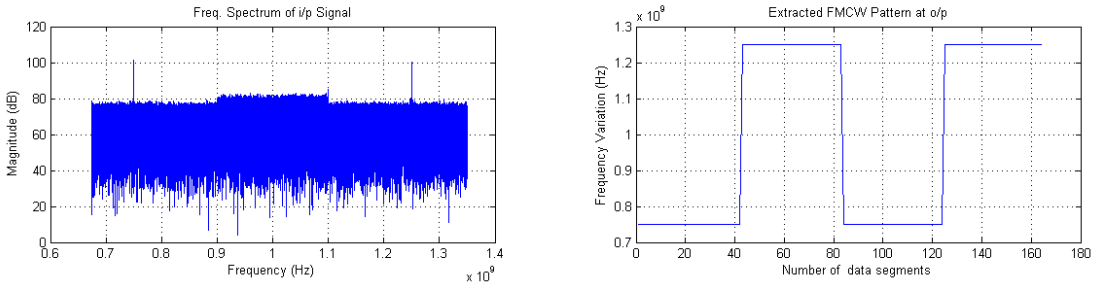


Figure 5.10: Extraction of Hopping pattern from two simultaneous signals

5.1.3 Multi Signal Environment with AWGN

The performance of Homodyne Zoom FFT is tested by applying more than two signals with AWGN at the input and observed the detected signal with frequency variation at the output. The signal with maximum amplitude is detected in a multi signal environment. The SNR requirement for three signal condition is 2 dB higher than two signal case. When five signals applied at the input with AWGN the SNR required is 0 dB which is 3 dB higher than three signal condition and 7 dB higher than single signal case. The signal with higher amplitude must have an amplitude difference of 3 dB atleast compared to the other signals to get faithful frequency variation at the output. some of the test conditions for multi signal environment are presented in this sub section.

5.1.3.1 Three Signals with AWGN

1) Extraction of triangular FMCW:

Input Signal: FMCW with triangular pattern is 3 dB higher than other FMCW signals

FMCW with triangular pattern: F_c: 1000 MHz, Δ_F: 200 MHz, SNR: -3 dB

FMCW with Hopping: F_1:750 MHz, F_2:1250 MHz, Δ_F: 500 MHz, SNR: -3 dB

CW without FM: F_{cw}: 1000 MHz, Δ_F: 0 MHz, SNR: -3 dB

Output Signal: Δ_F: 198.9 MHz, Error: 1.01 MHz

2) Extraction of FMCW with hopping:

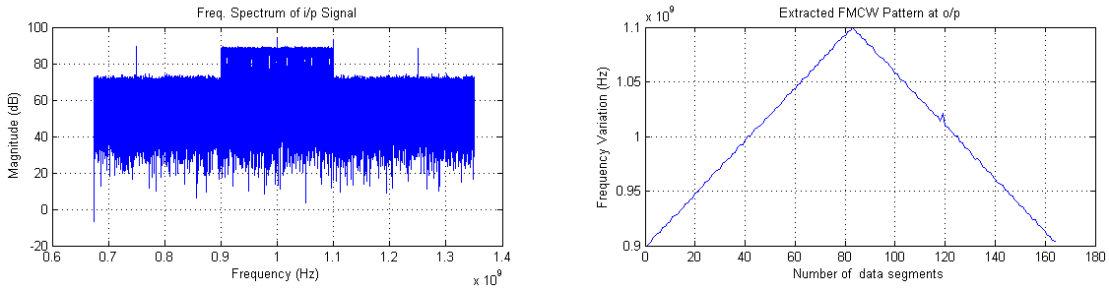


Figure 5.11: Extraction of FMCW with triangular pattern in a multi signal environment

Input Signal: FMCW with hopping is 3 dB higher than other FMCW signals

FMCW with triangular pattern: F_c: 1000 MHz, Δ_F: 200 MHz, SNR: -3 dB

FMCW with Hopping: F_1:750 MHz, F_2:1250 MHz, Δ_F: 500 MHz, SNR: -3 dB

CW without FM: F_{cw}: 1000 MHz, Δ_F: 0 MHz, SNR: -3 dB

Output Signal: Δ_F: 499.99 MHz, Error: 0.01 MHz

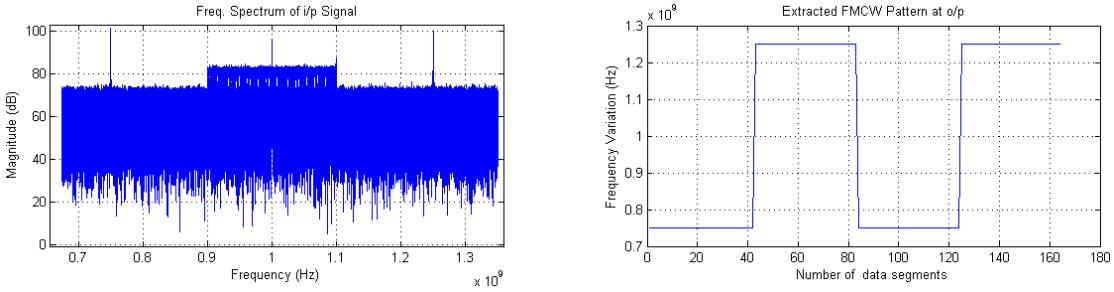


Figure 5.12: Extraction of Hopping pattern in a multi signal environment

3) Extraction of FMCW with negative ramp:

Input Signal: FMCW with negative ramp is 3 dB higher than other FMCW signals

FMCW with negative ramp: F_c: 900 MHz, Δ_F: 50 MHz, SNR: -3 dB

FMCW with positive ramp: F_c: 800 MHz, Δ_F: 10 MHz, SNR: -3 dB

FMCW with triangular pattern: F_c: 1000 MHz, Δ_F: 100 MHz, SNR: -3 dB

Output Signal: Δ_F: 49.686 MHz, Error: 0.314 MHz

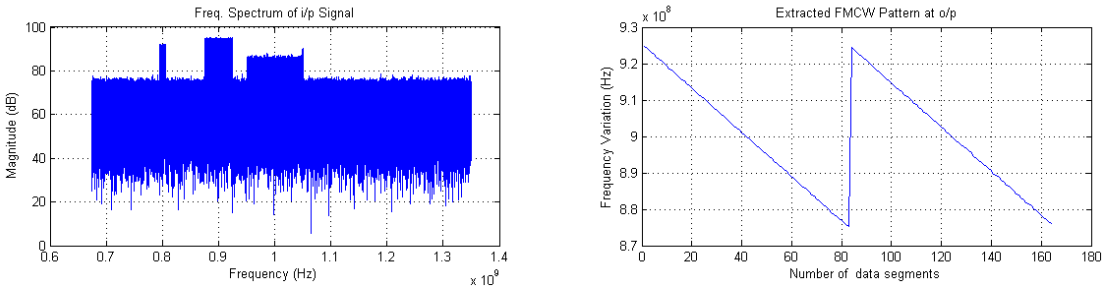


Figure 5.13: Extraction of FMCW with negative ramp in a multi signal environment

5.1.3.2 Five Signals with AWGN

1) Extraction of triangular FMCW:

Input Signal: FMCW with triangular pattern is 3 dB higher than other FMCW signals

CW signal: F_{cw}: 1100 MHz, Δ_F: 0 MHz, SNR: 0 dB

FMCW with hopping: F_1: 1050 MHz, F_1: 1200 MHz, Δ_F: 150 MHz, SNR: 0 dB

FMCW with positive ramp: F_c: 800 MHz, Δ_F: 10 MHz, SNR: 0 dB

FMCW with negative ramp: F_c: 900 MHz, Δ_F: 50 MHz, SNR: 0 dB

FMCW with triangular pattern: F_c: 1000 MHz, Δ_F: 1 MHz, SNR: 0 dB

Output Signal: Δ_F: 0.988 MHz, Error: 0.0112 MHz

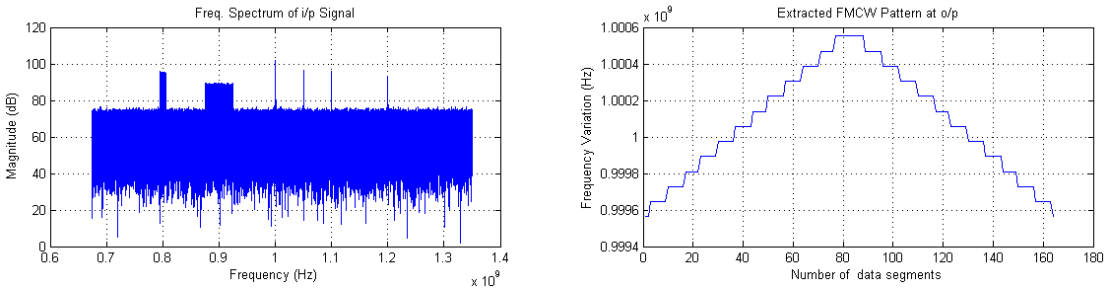


Figure 5.14: Extraction of FMCW with triangular pattern in a multi signal environment

2) Extraction of CW:

Input Signal: CW signal is 3 dB higher than FMCW signals

CW signal: F_{cw}: 1100 MHz, Δ_F: 0 MHz, SNR: 0 dB

FMCW with hopping: F_1: 1050 MHz, F_1: 1200 MHz, Δ_F: 150 MHz, SNR: 0 dB

FMCW with positive ramp: F_c: 800 MHz, Δ_F: 10 MHz, SNR: 0 dB

FMCW with negative ramp: F_c: 900 MHz, Δ_F: 50 MHz, SNR: 0 dB

FMCW with triangular pattern: F_c: 1000 MHz, Δ_F: 1 MHz, SNR: 0 dB

Output Signal: Δ_F: 0.988 MHz, Error: 0.0112 MHz

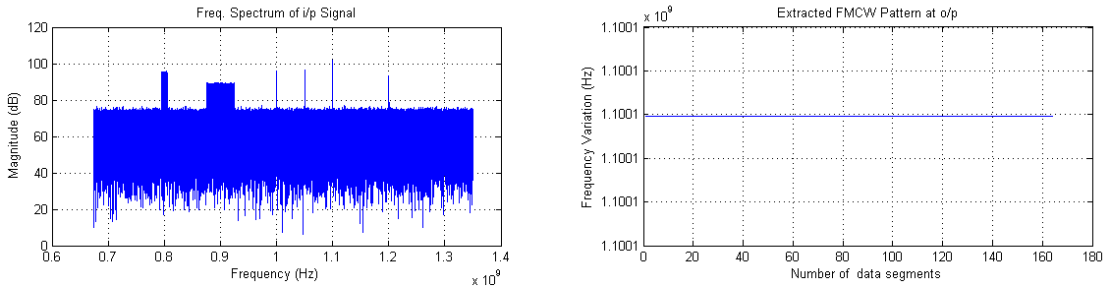


Figure 5.15: Extraction of CW signal in a multi signal environment

5.2 Simulations using Verilog

ISIM and gtkwave are used for the simulation of various modules present in the Homodyne Zoom FFT block diagram. Figure 5.16 shows the LO frequency variation captured using ISIM. Figure 5.17 shows frequency spectrum of primary FFT using gtkwave. Figure 5.18 shows the output of DDS, Homodyne Mixer, CIC filter and FIR LPF using gtkwave. Figure 5.19 shows the frequency variation of an FMCW signal at the output of Homodyne Zoom FFT using ISIM. Figure 5.20 shows the plot of frequency variation of an FMCW signal at the output of Homodyne Zoom FFT using gtkwave.

5.3 Comparisons with MUSIC

The basic performance of MUSIC is tested by applying two CW signals with AWGN at the input. The frequency spectrum is observed at the output by using

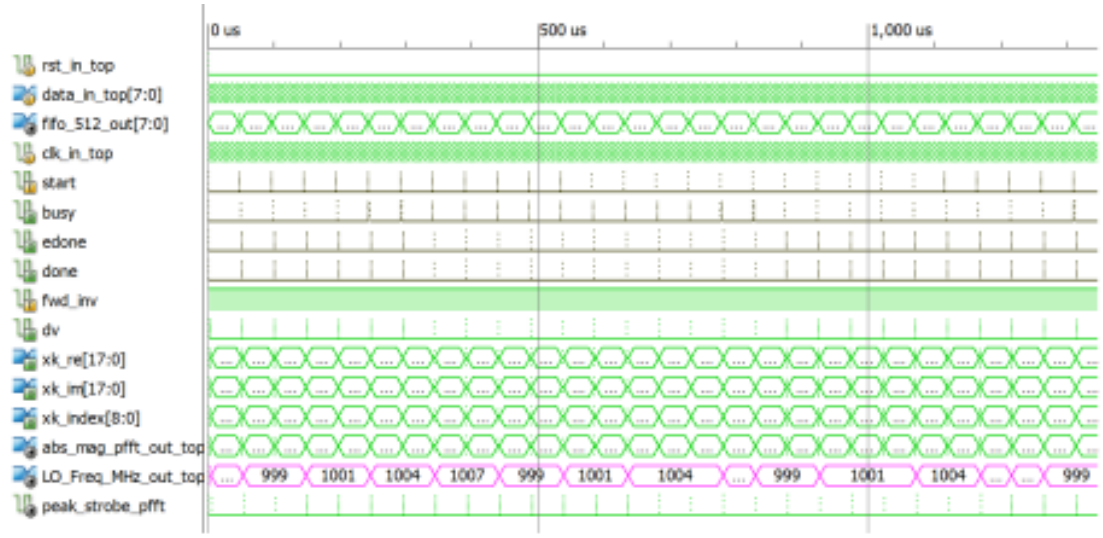


Figure 5.16: Simulation showing LO frequency variation

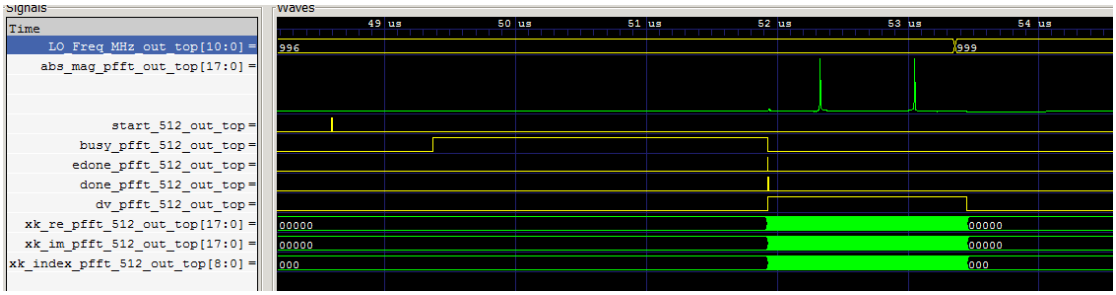


Figure 5.17: Simulation showing frequency spectrum of Primary FFT using gtkwave

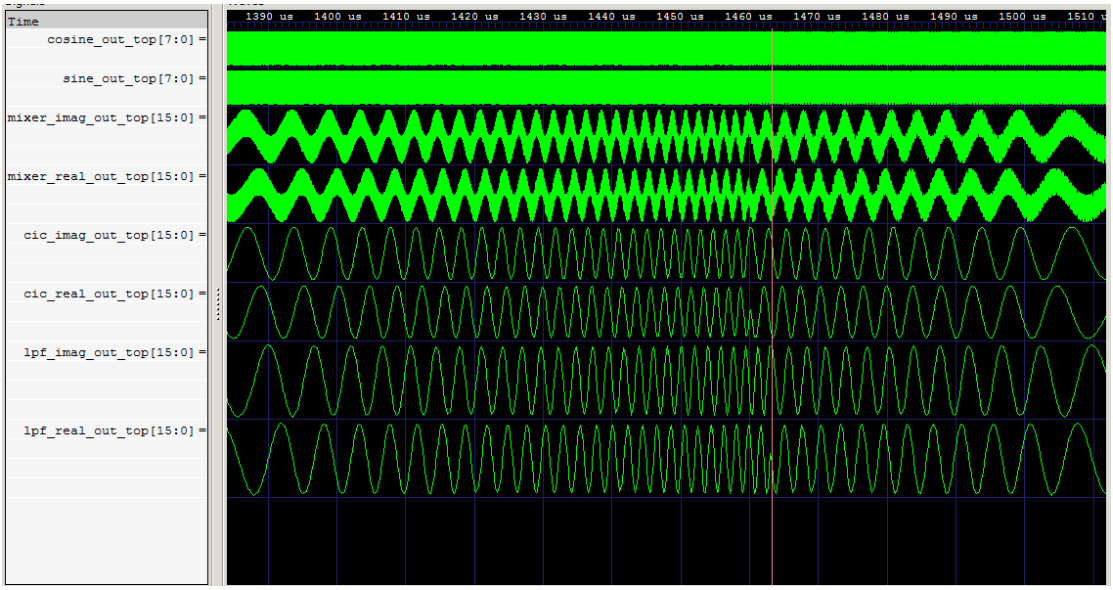


Figure 5.18: Simulation showing DDS,Mixer,CIC,FIR LPF outputs using gtkwave

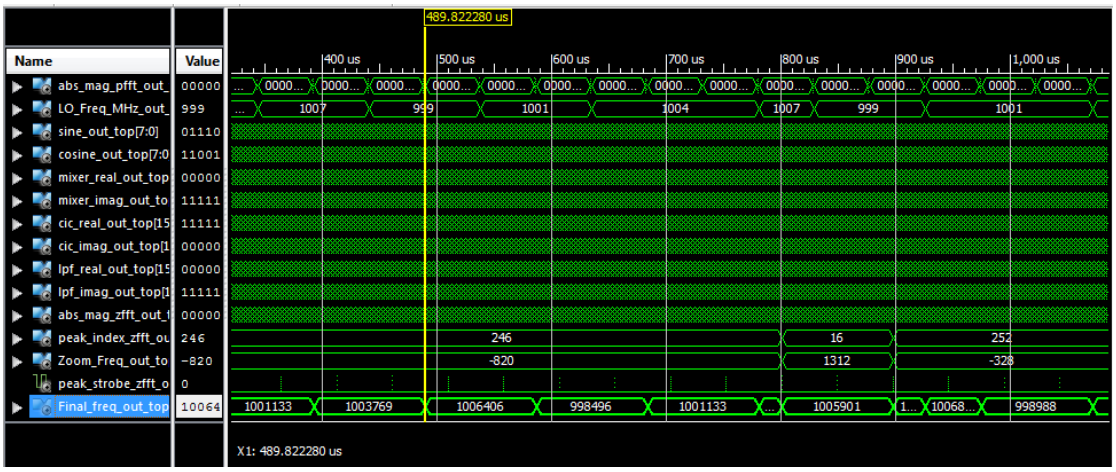


Figure 5.19: Simulation showing output frequency variation using isim

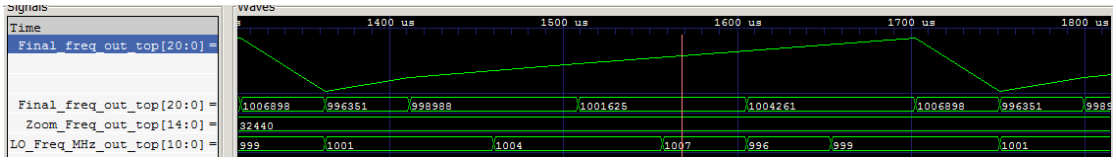


Figure 5.20: Simulation showing output frequency variation using gtkwave

FFT, Homodyne Zoom FFT and MUSIC techniques for the purpose of comparison. Also the estimated frequency by using HDZFFT and MUSIC are compared with the original input signals. A combination of two complex CW signals having a frequency difference of 10 MHz with AWGN is applied at the input. The power levels of the two signals kept same for the sake of simplicity. The length of input data, M is chosen as 16K and 4K for the two test cases simulated. The decimation factor for HDZFFT is 64 and the number of signals, P=2 for MUSIC. The test is carried out by varying SNR and the measured frequencies are compared with input frequencies.

5.3.1 Case1: M=16K,F1=1000,MHz,F2=1010 MHz,R=64,P=2

Simulation results for M= 16K is shown in figures 5.21 to 5.28.

Table 5.1: HDZFFT vs MUSIC for M=16K

SNR(dB)	HDZFFT		MUSIC	
-10	F1=1000.06 MHz	F2=1010.11 MHz	F1=1006.81 MHz	F2 not detected
0	F1=1000.06 MHz	F2=1010.11 MHz	F1=1005.25 MHz	F2 not detected
+10	F1=1000.06 MHz	F2=1010.11 MHz	F1=1005.17 MHz	F2 not detected
+20	F1=1000.06 MHz	F2=1010.11 MHz	F1=1004.51 MHz	F2 not detected
+25	F1=1000.06 MHz	F2=1010.11 MHz	F1=999.481 MHz	F2=1009.2
+30	F1=1000.06 MHz	F2=1010.11 MHz	F1=1000.63 MHz	F2=1010.69
+35	F1=1000.06 MHz	F2=1010.11 MHz	F1=998.93 MHz	F2=1009.86 MHz
+40	F1=1000.06 MHz	F2=1010.11 MHz	F1=1000.06 MHz	F2=1010.03 MHz

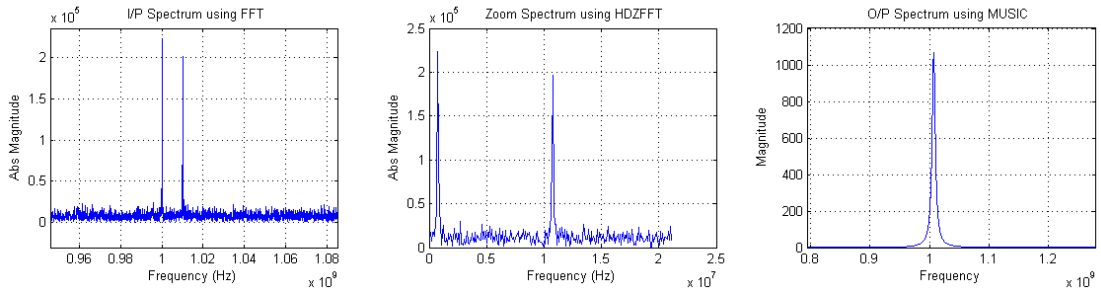


Figure 5.21: I/P, HDZFFT and MUSIC Spectrum for SNR = -10 dB

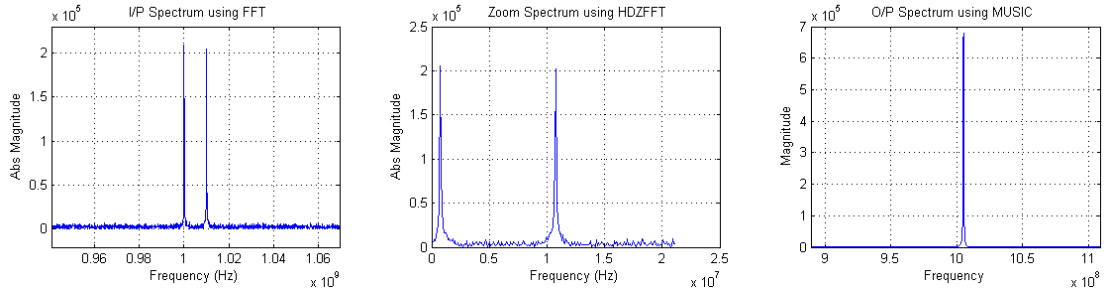


Figure 5.22: I/P, HDZFFT and MUSIC Spectrum for SNR = 0 dB

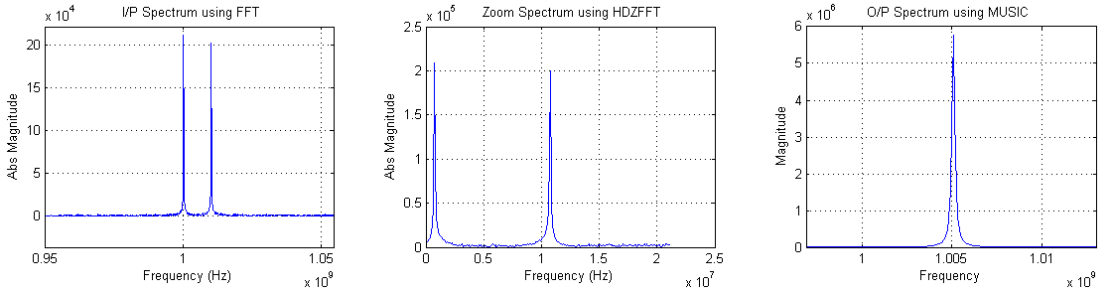


Figure 5.23: I/P, HDZFFT and MUSIC Spectrum for SNR = +10 dB

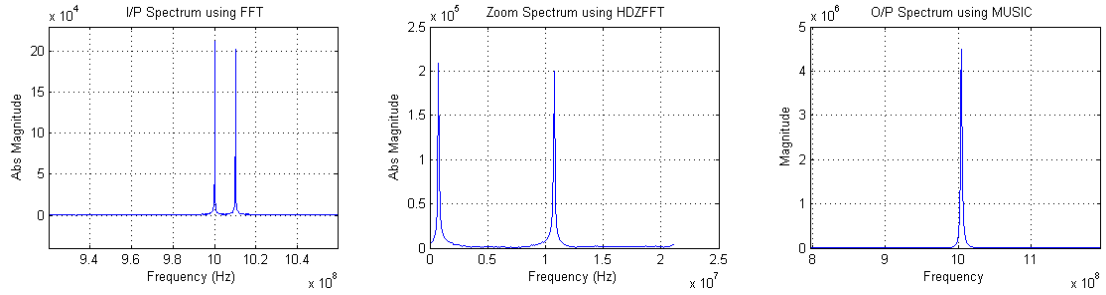


Figure 5.24: I/P, HDZFFT and MUSIC Spectrum for SNR = +20 dB

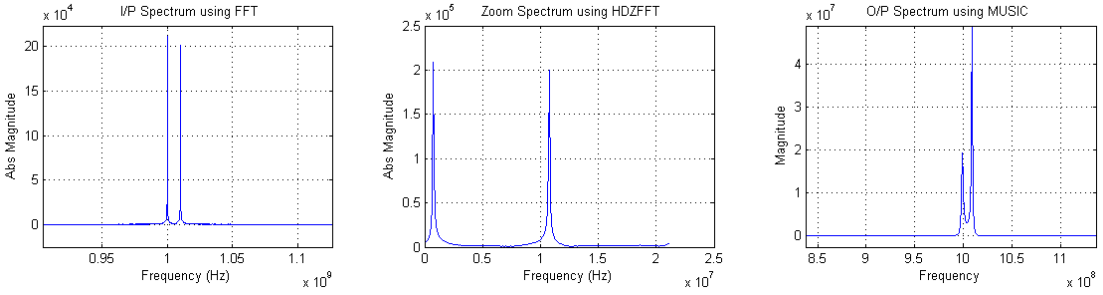


Figure 5.25: I/P, HDZFFT and MUSIC Spectrum for SNR = +25 dB

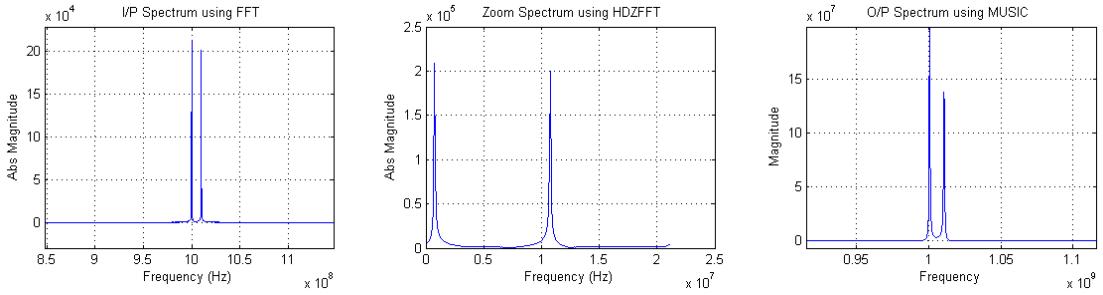


Figure 5.26: I/P, HDZFFT and MUSIC Spectrum for SNR = +30 dB

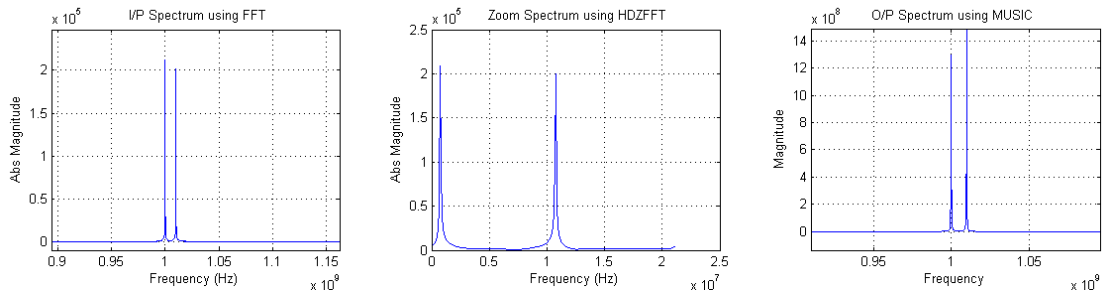


Figure 5.27: I/P, HDZFFT and MUSIC Spectrum for SNR = +35 dB

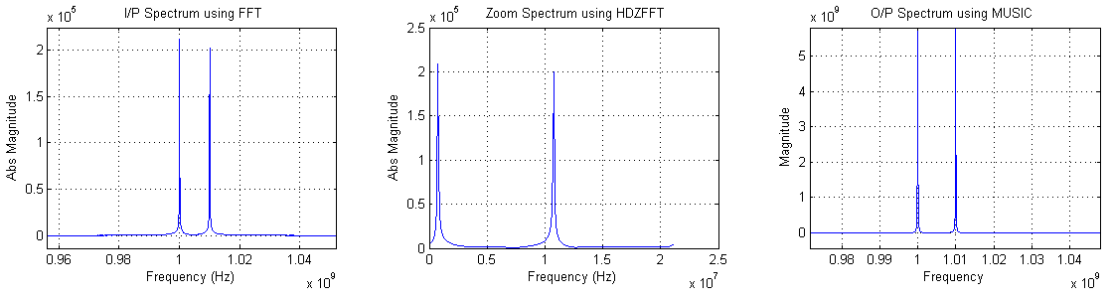


Figure 5.28: I/P, HDZFFT and MUSIC Spectrum for SNR = +40 dB

Table 5.2: HDZFFT vs MUSIC for M=4K

SNR(dB)	HDZFFT		MUSIC	
-10	F1=1000.31 MHz	F2=1010.19 MHz	F1=996.021 MHz	F2 not detected
0	F1=1000.31 MHz	F2=1010.19 MHz	F1=1005.58 MHz	F2 not detected
+10	F1=1000.31 MHz	F2=1010.19 MHz	F1=1005.25 MHz	F2 not detected
+20	F1=1000.31 MHz	F2=1010.19 MHz	F1=1004.92 MHz	F2 not detected
+25	F1=1000.31 MHz	F2=1010.19 MHz	F1=1004.26 MHz	F2 not detected
+30	F1=1000.31 MHz	F2=1010.19 MHz	F1=1008.54 MHz	F2 not detected
+35	F1=1000.31 MHz	F2=1010.19 MHz	F1=999.97 MHz	F2=1010.19 MHz
+40	F1=1000.31 MHz	F2=1010.19 MHz	F1=1000.31 MHz	F2=1010.52 MHz

5.3.2 Case2: M=4K,F1=1000 MHz,F2=1010 MHz,R=64,P=2

Simulation results for M= 4K is shown in figures 5.29 to 5.36.

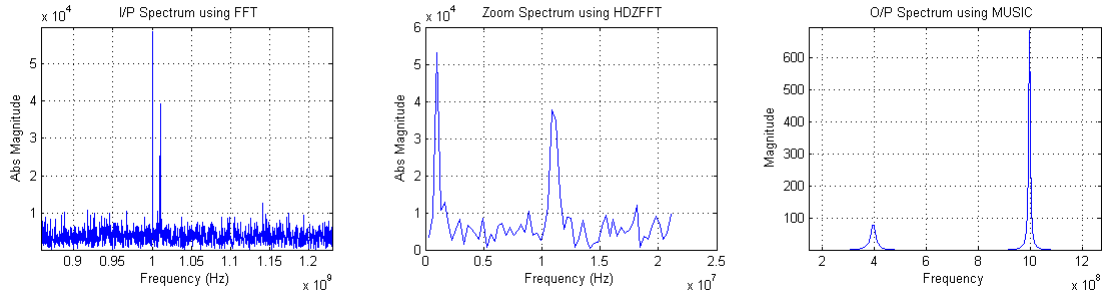


Figure 5.29: I/P, HDZFFT and MUSIC Spectrum for SNR = -10 dB

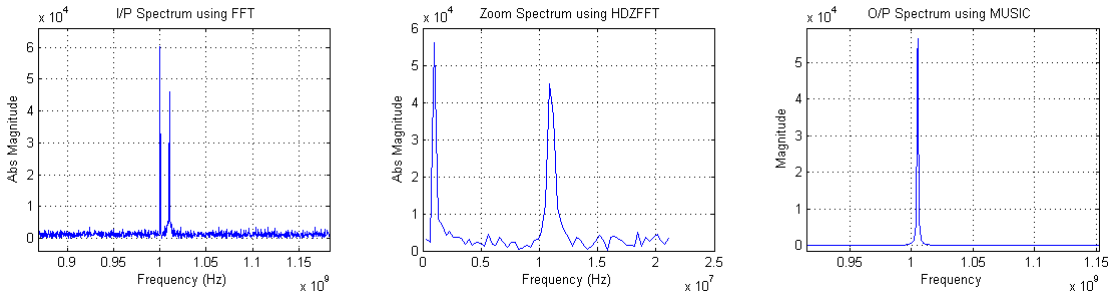


Figure 5.30: I/P, HDZFFT and MUSIC Spectrum for SNR = 0 dB

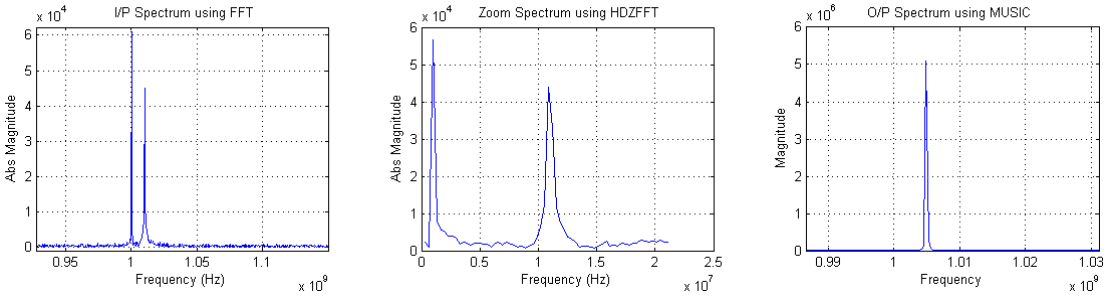


Figure 5.31: I/P, HDZFFT and MUSIC Spectrum for SNR = +10 dB

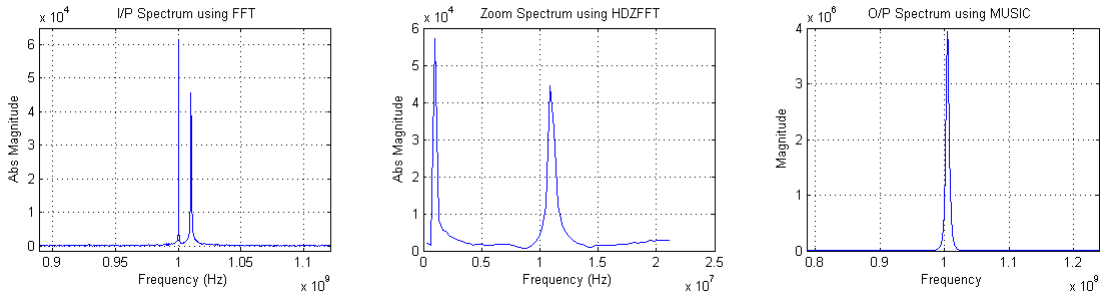


Figure 5.32: I/P, HDZFFT and MUSIC Spectrum for SNR = +20 dB

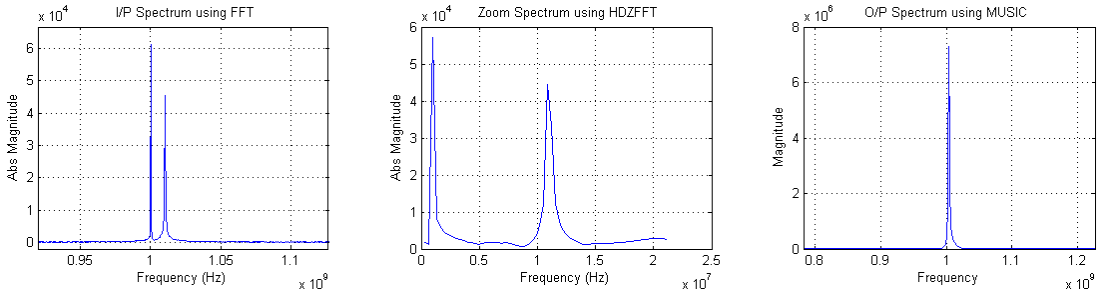


Figure 5.33: I/P, HDZFFT and MUSIC Spectrum for SNR = +25 dB

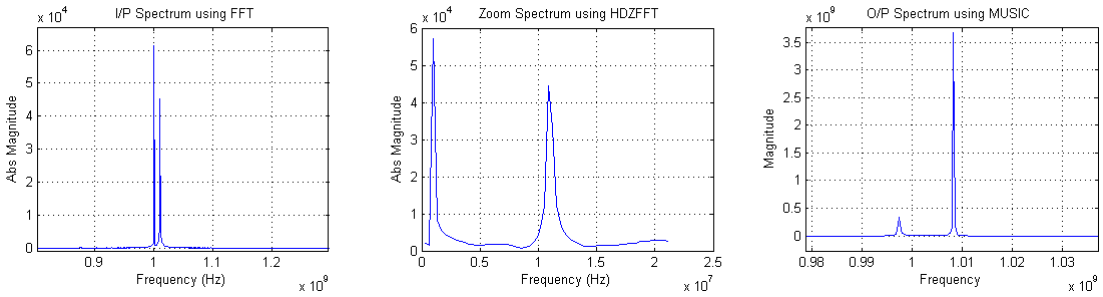


Figure 5.34: I/P, HDZFFT and MUSIC Spectrum for SNR = +30 dB

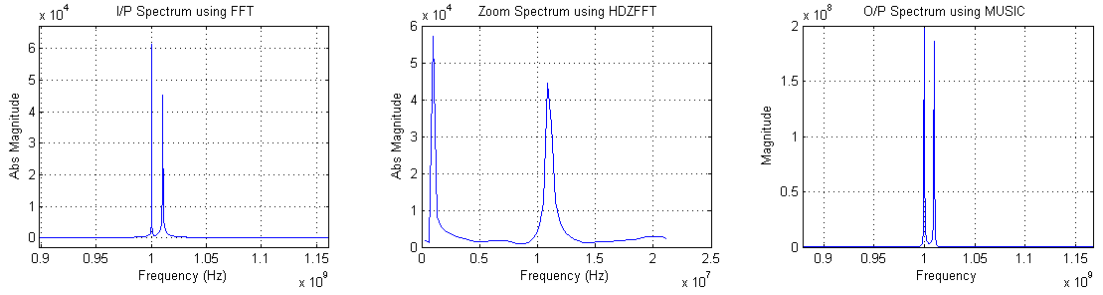


Figure 5.35: I/P, HDZFFT and MUSIC Spectrum for SNR = +35 dB

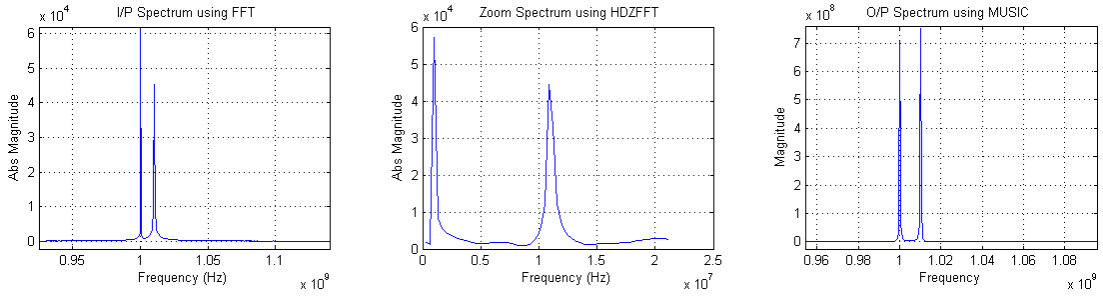


Figure 5.36: I/P, HDZFFT and MUSIC Spectrum for SNR = +40 dB

CHAPTER 6

Performance Analysis

6.1 Comparison with conventional FFT

The comparison between a conventional FFT and the Homodyne Zoom FFT is made by taking the number of multiplications involved for spectrum computation.

Conventional FFT:

- Number of multiplications:

$$M_{mul} = \frac{M}{2}\log_2 M = \frac{16384}{2}\log_2 16384 = 1,14,688 \tag{6.1}$$

Homodyne Zoom FFT:

- Primary FFT:

$$N_{mul} = \frac{N}{2}\log_2 N = \frac{512}{2}\log_2 512 = 2304 \tag{6.2}$$

- Homodyne Mixer: (2*M) = 2*16384 = 32,768
- Compensation FIR Filter: 2*(number of coefficients(M/D))= 2*(13*(16384/64))=6656
- Secondary FFT:

$$Z_{mul} = \frac{Z}{2}\log_2 Z = \frac{256}{2}\log_2 256 = 1024 \tag{6.3}$$

- DDS and CIC filter does not involve multiplications.
- Total number of multiplications: M_{mul} = 2304+32,768+6656+1024=42752

Saving in computation:

$$Saving = \frac{1,14,688 - 42752}{1,14,688} * 100 = 62.72\% \tag{6.4}$$

6.2 Comparison with MUSIC

The performance of Homodyne Zoom FFT is compared with MUSIC(Multiple Signal Classification) technique in this section. MUSIC is one of the popular method for high resolution spectrum estimation. The main difference between HDZFFT (Homodyne Zoom FFT) and MUSIC is, The MUSIC takes lesser number of samples to get high frequency resolution when compared to HDZFFT. But, The SNR requirements for MUSIC is very high compared to HDZFFT. We have seen in the simulations that HDZFFT works for -7 dB SNR for single signal to 0 dB SNR for five signals acting simultaneously at the input. But this is not the case with MUSIC. MUSIC requires more than 25 dB SNR in order to give faithful results. This SNR requirements are not suitable for EW applications, as the minimum of 0 dB SNR is required to process modern day LPI FMCW radar signals. Also the frequency estimation using MUSIC is computationally intensive. A brief description of MUSIC is discussed here. The basic idea of MUSIC is to separate signal from noise through eigen value decomposition.The steps involved, advantages, limitations of HDZFFT and MUSIC for the computation of frequency are briefly given below.

6.2.1 MUSIC

6.2.1.1 Steps involved

1. Let input signal is x(n) of size (1xM) consists of P complex exponentials in the presence of AWGN
2. Create R_{xx} matrix of size (MxM) using modified covariance method
3. Compute eigenvalues
4. Compute eigenvectors
5. Store eigenvalues in decreasing order
6. Number of largest eigenvalues gives the number signals,P
7. The eigenvectors corresponding to P largest eigenvalues span the signal space

8. Remaining (M-P) eigenvectors span the orthogonal space, where there is only noise
9. Estimate the function $P_{music}(f)$

$$P_{music}(f) = \frac{1}{\sum_{i=p+1}^{M} |e^H V_i|^2} \tag{6.5}$$

where,V_i=noise eigenvectors and

$$e = [1 e^{j\omega} e^{j2\omega} .... e^{j(M-1)\omega}]^T \tag{6.6}$$

10. Location of the P largest peaks of the estimation function gives frequency estimation for P signal components

6.2.1.2 Advantages

High Frequency resolution can be achieved with shorter data lengths when compared to Homodyne Zoom FFT.

6.2.1.3 Limitations

1. Accurate estimation of number of signals P is required in order to estimate $P_{music}(f)$
2. Even for a shorter data length, the order of R_{xx} matrix is high.
3. Peaks generated by MUSIC does not give amplitude information.
4. The technique works better only when SNR > 30 dB.
5. Frequency accuracy is a function of SNR and order of R_{xx} but not data dependant. The frequency resolution is severely affected by decreasing SNR <30 dB
6. Large computation involved.
7. If estimation of P is wrong then MUSIC gives erroneous frequencies.
8. MUSIC does not work well for too large or too small number of signals.
9. MUSIC does not work well for FMCW signals.

6.2.2 HDZFFT

6.2.2.1 Steps involved

1. Estimation of coarse frequency.
2. Homodyne mixing.
3. Decimation.
4. Zoom frequency estimation.
5. RF frequency calculation

6.2.2.2 Advantages

1. Detection of CW/FMCW signals with high frequency resolution
2. Capable of detecting various Frequency Modulations with triangular, negative ramp, positive ramp and hopping patterns of an FMCW Radar
3. Frequency Resolution F_r=82 KHz, Time Resolution T_r=12 us
4. Can handle Modulation Rate > 82 KHz/12 us
5. Modulation BW $\geqslant$ 164 KHz and $\leqslant$ 500 MHz can be tracked
6. The homodyne zoom fft is capable of detecting signals with 0 dB SNR in a multi signal environment with AWGN
7. The technique meets the requirements of Electronic Warfare by detecting CW/FMCW Radar signals which is very useful in estimating the FMCW Pattern in order to generate an accurate Jamming signal for Electronic Counter Measure (ECM)
8. Non availability of IP cores for more than 32K Point FFT makes Homodyne Zoom FFT very attractive for Electronic Warfare applications.
9. By using existing FFT cores Homodyne Zoom FFT can be quickly configured to analyse Electronic Intelligence (ELINT) acquired during trials.
10. No need to estimate the number of signals unlike in MUSIC.

6.2.2.3 Limitations

1. The HDZFFT works by detecting the signal with maximum power. This necessitates the desired signal power to be higher than the undesired signals by atleast 3 dB in this case.

2. frequency accuracy is data dependent unlike in MUSIC. Hence, to get higher frequency accuracy the length of data required is more.

3. Requires additional hardware for coarse frequency estimation, which is not required in a conventional zoom FFT technique.

4. Maximum modulation bandwidth is limited by ADC bandwidth.

5. Can not detect modulation rate <(82 KHz/12 us)

6.3 Performance Trade-Offs in HDZFFT

1. F_s vs Δ_{BW}: As frequency modulation bandwidth(Δ_{BW}) increases, the sampling frequency(F_s) required increases. And if F_s increases, the ADC band width required increases, this puts a limit on the receiver performance due to non availability of ADC with large band width more than 1 GHz.

2. F_s vs F_r: The frequency resolution (F_r) is a function of F_s and M (where M is the number of samples for FFT computation). As F_s increases, F_r becomes worse and by decreasing F_s, F_r becomes better. But, reducing F_s effects the modulation band width capability.

3. **M vs** F_r: To get high F_r, the required M is very high. But by increasing M the length of FFT increases which in turn increases the area and latency in FPGA. As discussed earlier Homodyne Zoom FFT is an efficient way of implementing long length FFT.

4. **N** vs S_{min}: The sensitivity of an EW receiver depends on the length of primary FFT, N. Sensitivity is higher for longer N and lower for smaller N. But, by taking longer N, area in FPGA increases.

5. $B_{\theta(n)}$ vs Δ_f in DDS: The frequency resolution of DDS (Δ_f) depends on phase width ($B_{\theta(n)}$) of accumulator in DDS. If $B_{\theta(n)}$ increases then Δ_f increases and vice versa. But, by increasing $B_{\theta(n)}$ depth of memory(LUT) increases.

6. **D vs Z**: The length of secondary FFT (Z) can be reduced by increasing decimation factor (D). But by increasing D, the decimation filter needs sharp cut-off frequency.This in turn increases the length of decimation filter and hence more area.

7. **SNR vs P**: From the simulation results shown in the previous chapters it is observed that SNR requirement is different for different number of signals

Table 6.1: SNR requirements in a multi signal environment

Signal environment	SNR required for detection
Single signal with AWGN	-7 dB
Two signals with AWGN	-5 dB
Three signals with AWGN	-3 dB
Five signals with AWGN	0 dB

at the input. To detect the required signal using Homodyne Zoom FFT in a multi signal environment the following SNRs are required at the input. The table 6.1 shows the SNR requirements in a multi signal environment with AWGN.

CHAPTER 7

Conclusions and Future Scope

7.1 Conclusions

The proposed Homodyne Zoom FFT or HDZFFT is an efficient way of implementing long length FFT for Electronic Warfare (EW) applications. The HDZFFT is useful for the estimation of frequency spectrum with high resolution with low SNR. Thus the modern day requirements of an EW receiver for the detection of LPI FMCW radars can be fulfilled by HDZFFT. The main difference between a conventional Zoom FFT and HDZFFT is the principle of *homodyne*. In a conventional Zoom FFT the information about the LO frequency is a known parameter. But,in case of HDZFFT it is derived from the incoming signal itself by using homodyne principle to meet EW requirements. Since the design involves high sampling rates of the order of GHz, the use of CIC filters for decimation makes the technique economical in terms of hardware.

The HDZFFT is very useful for EW applications because this can be quickly configured by using available IP cores, thus saves the development time. Also non availability of IP cores more than 32K point FFT makes this technique attractive. Since the multiplier is the costly element in FPGA, the implementation of HDZFFT is carried for both multiplier and multiplier less design.

The proposed HDZFT is capable of processing LPI FMCW radar signals with triangular, up ramp , down ramp and hopping patterns under multi signal environment with AWGN. The capability of high frequency accuracy of 82 KHz and low SNR of 0 dB meets the requirements of EW receiver. Also the modulation

rate of HDZFFT for the given specifications is 82 KHz/12 us which can track any frequency deviation > 82 KHz over a period of 12 us.

The propsed HDZFFT is compared with the most popular high resolution spectrum estimation technique, MUSIC in terms of SNR and it is observed from the simulations that MUSIC is not suitable for high sensitivity requirements of EW receiver, as the minimum SNR required is >30 dB to distinguish two signals separated by 10 MHz. Where as HDZFFT is able to distinguish these two signals at an SNR of -10 dB. This clearly shows the advantage of HDZFFT over MUSIC.

7.2 Future Scope

The proposed HDZFFT is capable of detecting LPI FMCW radar signals with high frequency resolution at low SNR. Though the proposed technique has many advantages in terms of hardware reconfigurability, this still requires large number of samples before decimation. So, the time resolution achieved with HDZFFT is limited by the length of samples acquired.(for example the time resolution for the present design is 12 us) Some of the advanced techniques like *ESPRIT* can be investigated to improve the time resolution by maintaining frequency resolution. However, the SNR requirements and hardware complexity needs to be addressed.

REFERENCES

[1] Chi-Hao Cheng,David M. Lin,Lihyeh L. Liou,James Tsui, "Electronic Warfare Receiver with Multiple FFT Frame Sizes", *IEEE Transactions on Aerospace and Electronic Systems* Vol. 48, NO. 4, October 2012, pp.3318 -3330

[2] Belal Al-Qudsi, Niko Joram, Axel Strobel and Frank Ellinger,"Zoom FFT for Precise Spectrum Calculation in FMCW Radar using FPGA", *IEEE Signal Processing 2*, Session W3B,2013,pp. 337-340.

[3] A.K.Singh, K. Subba Rao O Schmidt, Raymond E Franks, "Digital Receiver-based Electronic Intelligence System Configuration for the Detection and Identification of Intrapulse Modulated Radar Signals",*Defence Science Journal*, Vol. 64, No. 2, pp. 152-158,March 2014

[4] Dong Pei, Shuo Yang, Hongwu Yang, Quanzhou Wang, Manman Li, "High Efficient and Real Time Realization of Zoom FFT Based on FPGA," *International Conference on Computer Application and System Modeling*, IEEE, vol. 2, 2010, pp. 669-673.

[5] Philip E Pace, "Detecting and classifying Low probability of intercept radar ",Artech house, second edition, 2009

[6] Eugene B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation", *IEEE transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, NO. 2, AprilL 1981

[7] Ralph O Schmidt, Raymond E Franks, "Multiple Source DF Signal Processing An Experimental System",IEEE transactions on antennas and wave propagation, VOL. AP-34. NO.3, MARCH 1986.