HARDWARE IMPLEMENTATION OF TUNABLE HETERODYNE BAND-PASS FILTERS

By

ASAD AZAM

Bachelor of Science

Oklahoma State University

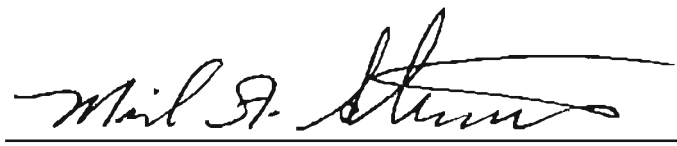
Stillwater, Oklahoma

1999

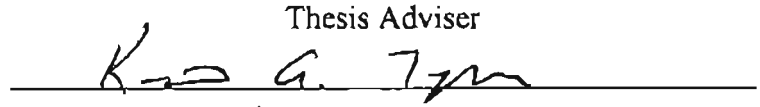
Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
August, 2001

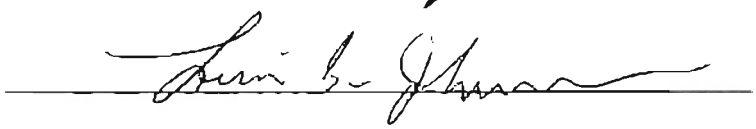
HARDWARE IMPLEMENTATION OF TUNABLE HETERODYNE BAND-PASS FILTERS

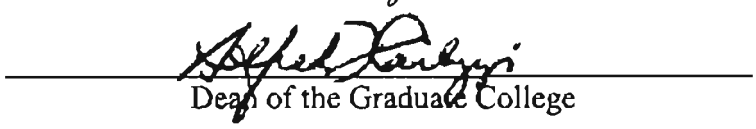
Thesis Approved:



Thesis Adviser







Dean of the Graduate College

PREFACE

Modern wireless and satellite communication systems make use of spread-spectrum modulation concepts such as Frequency-hopping spread-spectrum (FHSS) and Direct sequence spread-spectrum (DSSS). The spread-spectrum modulation method inherently possesses anti-jamming and anti-interception properties due to the fact that the narrowband information signal is spread over a wide range of frequencies, masking the information-bearing signal as noise. Despite these properties, these communication channels can be severely corrupted by high-powered narrowband interference signals generated by local FM or AM transmitters which may cause complications when detecting the information signal at the receiver. Therefore, the communication system is made more efficient with the use of signal processing techniques for narrowband interference attenuation. Control systems is another area where the presence of narrowband interference signal due to mechanical resonance can be responsible for causing distortion in information signal.

Any Band-pass, High-pass or a Low-pass filter may be converted into a tunable filter through the use of new Tunable Heterodyne Band-pass Filter concept in which the frequency of the heterodyne signal is adjusted thereby creating the effect of translating the entire transfer function of the fixed filter in frequency. In this thesis, hardware implementation techniques and results of the new Digital Tunable Heterodyne Band-pass

filter is proposed that allows a prototype IIR or FIR filter to be shifted through the entire range of digital frequencies with a single parameter, the heterodyning frequency. The unique property of this new tunable filter is the range of tunability it possesses. With this technique, the fixed filter is tuned continuously using the concept of frequency translation. The images created by the heterodyne process are cancelled without the use of image canceling filters, which significantly contribute towards a hardware efficient design. In this thesis, simulation results are observed to illustrate the effects of having the fixed prototype filter as a band-pass, high-pass, low-pass or notch filter.

This thesis concentrates on the hardware implementation of the tunable heterodyne filter structure with a band-pass filter as the fixed prototype filter. Thus, simulation and experimental results show that if the fixed filter is a narrowband Band-pass filter, a much hardware efficient implementation can be achieved by using the new Tunable Heterodyne Band-pass filter to extract the narrowband interference from broadband communication or control systems as compared to the standard techniques used.

The proposed heterodyne filter is suitable both as a tunable filter or to be implemented with standard algorithms to design adaptive digital filters. The new structure proposed is composed of three main components which can be implemented using Field Programmable Gate Arrays (FPGA) or easily be retargeted for an Application Specific Integrated Circuits (ASIC) standard cell technology or custom designed for Very Large Scale Integration (VLSI) processes. A prototype system is implemented using a single chip Xilinx Virtex Series Field Programmable Gate Arrays (FPGA) and the simulation results are compared with the hardware data.

Dedication

To AMI AND ABU

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my research advisor, Dr. Michael A. Soderstrand for his intelligent supervision, constructive guidance, inspiration and friendship. His inspiring insight and incomparable patience guided me through my study and provided me with a valuable experience and knowledge. My sincere appreciation also extends to my committee member Dr. Louis G. Johnson whose guidance, assistance, encouragement, and friendship are also invaluable. I would also like to thank committee member Dr. Keith Teague for carefully reviewing my thesis and giving me his precious advice.

I am thankful to all my colleague researchers in the Digital Signal Processing and Communication (DSP&C) Lab for assisting and helping me on numerous occasions. I like to thank Ms. Regina Henry at the International Students and Scholars office for being a great friend and advisor. I would like to thank all my close friends for providing me with love, encouragement and support during the period of my studies.

To complete this acknowledgement, I would like to give my special thanks to my family for their patience and support. I like to thank my parents who have provided me with support and encouragement through their prayers and love. I like to thank my sisters, Nayyar and Nargis for their love and encouragement. I would like to thank my brother Asim for his support and guidance through my studies.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

Introduction and Thesis Organization

1.1 The Problem Statement

Tunable filters are widely used in applications where the frequency of interest varies over a period of time. Some of the applications lie in the field of communications and controls, where occurrence of narrowband interference is unpredictable and thus needs to be eliminated from the desired signal for an efficient system. In order to tune a filter over a range of frequencies, the coefficients of the filter have to be changed for every tunable frequency. This would require a much larger hardware structure to implement even for a moderate order filter with decent range of discrete tuning frequencies. A novel technique of heterodyne filtering allows us to tune a high-pass or a low-pass filter continuously over the range from DC to Nyquist and to tune a band-pass filter continuously over the range from DC to Nyquist/2 or from Nyquist/2 to Nyquist. In this concept, the incoming signal is translated to the fixed filter frequency rather then moving the filter over the incoming signal.

Through the mirroring effect of poles and zeros and their conjugates, a high-pass or a low-pass filter can be converted into a tunable band-pass filter with the method of

frequency translation. To analyze the effects on the tunability of the heterodyne filter based on the category of prototype filter used in this structure, detailed simulations are required for all forms of common filters; high-pass, low-pass, band-pass and band-stop. However, this thesis only deals with the hardware implementation of Tunable Heterodyne filter structure [1], where the prototype filters chosen for this implementation were band-pass filters. For rapid prototyping, the hardware was implemented using a Xilinx Virtex family of Field Programmable Gate Arrays (FPGA). One of the key goals of the work reported in this thesis was to implement the theoretical core of the tunable heterodyne filtering concept in hardware and to show that the heterodyne technique is more efficient for continuous tuning of frequencies compared to the standard techniques.

The particular Tunable Heterodyne filter structure discussed in this thesis has three distinct parts, Splitter, Fixed Prototype Filter and Combiner. The Splitter and Combiner structures are basically the heterodyne units, where the Splitter circuit is responsible for translating the incoming signal to the fixed filter frequency and the Combiner circuit translates the filtered response from the fixed filter structure to its base band.

The fixed prototype band-pass filter can be chosen as either an FIR or IIR filter. Applications for this type of tunable or adaptive filter can be found in many areas including embedded control and communications systems. For example, in broadband communication and control systems, where it is extremely important not to filter out the important information data, this type of filter can be used to attenuate narrow-band interference with minimum degradation of the broadband signal. Thus the need for high performance filters with sharp transition bands is essential in these applications in order

to prevent degradation of the broadband signal due to large transition bands associated with typical digital filters. Therefore, in this thesis, we focus on using IIR filters, which yield sharp transition bands, but are also extremely hard and tricky to tune.

In order to be able to target this design to various FPGA or ASIC technologies, the behavioral code for the circuit needed to be written in Hardware Description Languages (VHDL & Verilog). While this VHDL or Verilog code must be written with some target-specific technology (in our case, targeted specifically for the Xilinx Virtex FPGAs), as much as possible we have tried to develop a target-free code that can be easily be retargeted to other FPGAs or to ASIC designs.

The theory of Tunable Heterodyne Filters is based on complex mathematics and has been used in many communication systems; one such application is Single Side Band (SSB) Modulation. Most of these communication structures are built using analog devices [13]. This project deals with the implementation of Tunable Heterodyne Band-pass Filter for Digital Signal Processing applications.

1.2 Introduction

The importance of Digital Signal Processing (DSP) has been significantly exhibited over the years with advances in broadband communications. Several issues plague the modern communication systems, namely narrowband interferences in broadband Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK) signals.

Narrowband interferences are usually generated from oscillators that are local to the area where the receiver is located. These signals have strong energy confined in a

narrow-band at a specific frequency and are effective enough to make it extremely hard for the receivers to detect the wide-band signal. With this effect on signals around it, narrowband interference causes problems especially in the mobile communication industry whereby the received signal will be corrupted due to the system failure and low performance. Thus, to improve the system performance, signal processing techniques are applied to eliminate these interferences [2].

The basic idea behind the digital heterodyne filter structure is to develop a tunable band-pass filter in the digital domain. Utilizing the heterodyne property of signal translation, the signal is modulated with a carrier frequency in the digital domain to shift the interference frequency to the center frequency of the fixed filter. The signal is then filtered and retranslated through the process of heterodyning back to its base-band. The filter frequency can be situated at any place in the frequency spectrum, thus a relationship between the heterodyning frequency and the incoming interference frequency with respect to the center frequency of the band-pass filter needs to be established. The tunable heterodyne filter can be made adaptive by using an adaptive algorithm such as LMS, in which the heterodyning frequency can be calculated with prior knowledge of interference frequency and fixed filter frequency.

The newly proposed filter implementation was discovered based upon the heterodyne process in which the prototype filter could be adjusted or "tuned" to the appropriate frequency in order to attenuate narrowband interferences. The heterodyne process itself can be defined as the generation of new frequencies by mixing two or more signals in a nonlinear device. The proposed heterodyne filter implementation cancels the

effect of all the undesired images created during the heterodyne processes without using image-canceling filters.

Although theoretically the proposed digital filter has been proven to perform as intended, the emphasis presented here is towards the hardware implementation of the "Tunable" Heterodyne Filter which was carried out on the Xilinx Virtex Series FPGA. The Xilinx FPGA was chosen as a suitable test environment, which will eventually lead to the design being hardwired into an ASIC or VLSI process.

1.3 Thesis Organization

In chapter 2, previous work and research in the area of narrowband interference suppression from wide-band communication has been presented. A brief overview of Spread-Spectrum communication system is provided and the problems related to narrowband interference are pointed out. It is analyzed and researched, as to which technique our Tunable Filter is comparable to. This project has emerged from the heart of the communication systems, therefore the point of applications for the other similar structures are addressed. Then this chapter goes into a brief description of complex numbers and signals. This chapter forms a basis for the understanding of the underlying mathematics in this application.

In chapter 3, introduction to the Tunable Heterodyne Filter structure is elaborated. Each component of this filter is separately analyzed through mathematical expressions, which are derived at each node of the filter [8]. A detailed analysis in simulation is performed to show the tuning characteristics of different categories of the prototype filters, when implemented as fixed filters in the tunable heterodyne filter structure.

In chapter 4, a similar analysis of Tunable Heterodyne Band-pass Filter is used, but this time instead of explaining with mathematical equations, a graphical signal analysis is done to explain the workings of this structure.

In chapter 5, an introduction of FPGA systems and their architecture, along with a brief description of the synthesis process and method is presented. Terminologies built in this chapter will be extensively used in the successive chapters.

In chapter 6, hardware for the Tunable Heterodyne Band-pass filter is constructed using Xilinx FPGAs. This chapter elaborates the design process and synthesized circuitry. Implementation topics and issues included in this chapter are, Sine/Cosine wave generator, Signal to Noise Ratios, Multipliers and Prototype filter designs.

In chapter 7, the results that were obtained from Matlab simulation and FPGA hardware implementation, are presented. This chapter also shows the effect of having a higher order IIR filter to improve the system response. This chapter goes into a brief detail of our testing techniques and experimental setup. We also compare and estimate hardware requirements between the standard technique and the tunable heterodyne filtering technique.

Finally in chapter 8, we conclude the discussion of tunable heterodyne band-pass filters and set forth the goals and ideas for more future work.

Chapter 2

Background

2.1 Introduction

Digital Signal Processing has its applications embedded in a vast number of disciplines. Introduction of Signal processing in the field of communication has enhanced the quality of information transferred from one place to another. One of the fields of communication where signal processing has made a dominant effect is wide-band communication. In this chapter a brief overview of the broadband communication techniques and their associated problems are analyzed. Some of the material presented in this chapter deals with the research work done in the field of narrowband interference suppression from broadband signals and related techniques that are comparable to the one presented in this thesis. Signal Processing applied in this thesis is not only applied in narrowband interference in the field of communications but also in control systems.

2.2 Spread-Spectrum Communication

In this section, a brief description of Spread-Spectrum Communication is given along with some of the problems associated with this technique. The concept of Spread-

Spectrum is to encode a relatively narrowband information signal in such a way that the information is spread over a broad band of the frequency spectrum. This can be accomplished by several well-known techniques. For example, Direct-Sequence Spread-Spectrum (DSSS) is accomplished by mixing the input signal with a sequence of numbers. The signal to be transmitted is first multiplied by a pseudo noise to spread the information signal over a bandwidth of B. In this manner the power spectral density of the information signal is spread to about 100-1000 times the bandwidth of the original signal, thus making the information appear as noise. This process is shown in figure 2.1, where the PSD is the Power Spectral Density.

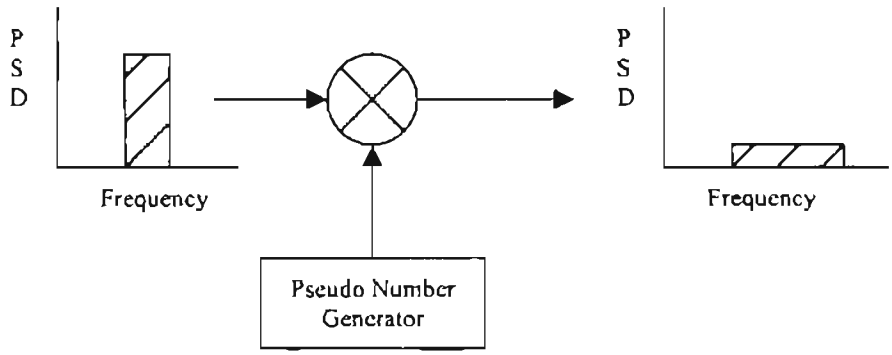


Figure 2.1 Spread-Spectrum generated at transmitter end.

To retrieve the information data from the noise-like spread-spectrum signal, the receiver has to use the same sequence of pseudo numbers. Through this process the data is recovered since there is very little correlation between the pseudo sequence generated by other users and the sequence used to retrieve the data. At the receiver end, the noise like data is multiplied by the same sequence. This process is shown in figure 2.2.

The advantage of spreading the information signal over a wide range of frequencies is that it is hard to detect the information data, which appear as noise in the communication channel and thus making it extremely difficult for deliberate jammers to distort or intercept the information. One of the advantages to this scheme is when the noise-like spread-spectrum information signal corrupted with a narrowband interference signal over the channel is multiplied by the pseudo sequence at the receiver end, the power of the narrowband is spread over a wide band of frequencies. Thus, this spread of narrowband would appear as noise to the retrieved information signal. This process is shown in figure 2.3, where the shaded area is the narrowband interference.

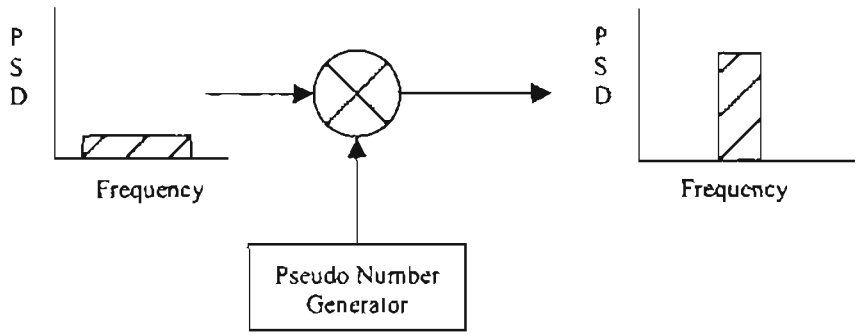


Figure 2.2 Information retrieved at receiver end.

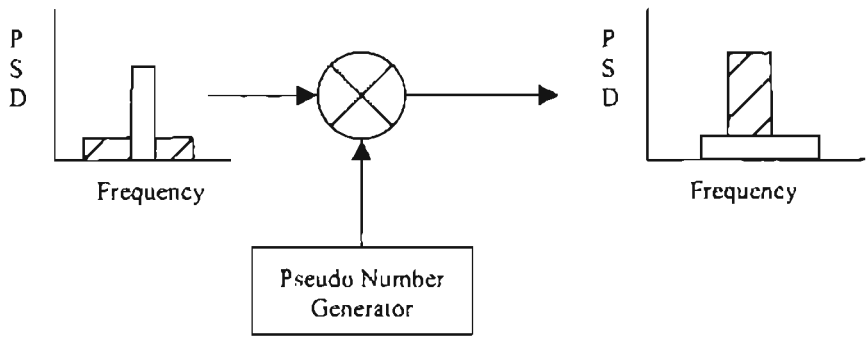


Figure 2.3 Information and narrowband noise at receiver end.

Although theoretically it seems that the narrowband interference can be suppressed and spread down into the noise level at the receiver end, in practice these narrowband interferences have high power associated with them, which would saturate the receiver amplifier, and thus would cause system failure. These strong narrowband signals generated by AM and FM radio transmissions within the communication systems make it impossible to detect the spread-spectrum signals [9][10][11][12]. Introduction of signal processing techniques for interference suppression is applied prior to detecting the signal, where narrowband interference suppression techniques are applied to eliminate them and consequently to enhance the performance of spread-spectrum communications [1][2].

2.2.1 Narrowband Interference Suppression Techniques

There are a number of research groups that are working to find novel techniques to eliminate narrowband interference from broadband communication systems without corrupting the information data. One of the techniques is to implement an adaptive algorithm to filter out the narrowband frequencies. These algorithmic techniques use transversal or tapped delayed filters to adjust the filter response to the frequencies of the narrowband interference. Since the thermal noise and the spread-spectrum signals are wideband processes, their future values cannot be predicted, whereas in the case of a narrowband signal, they can be predicted from their past behavior [2]. Therefore, the narrowband interference can be subtracted from the original signal. To eliminate the interference from the system, the tap weights of the filter are adapted until the mean squared error is minimized [1][4][5][6][7]. Another technique used for the suppression of

narrowband interference is the transform method. This technique consists of a Fourier Transform, a multiplier and an Inverse Fourier Transform [2].

The method implemented in this thesis closely resembles the adaptive algorithm technique described above, as the digital filter described here can be made adaptive by placing it in an adaptive algorithm. Since the coefficients of the transversal filter are not fixed, therefore variable multipliers are utilized, which consume a lot of silicon area. Also in the technique using the Fourier Transform method for narrowband interference elimination, considerable hardware is required to implement the Fast Fourier Transform and inverse Fourier transform. The tunable heterodyne band-pass filter proposed in this thesis can replace the traversal filter in the adaptive algorithm such as Least Mean Squared (LMS) [8] and can be tuned to any frequency from DC to $\pi/2$ or $\pi/2$ to π with a single parameter, the heterodyne frequency.

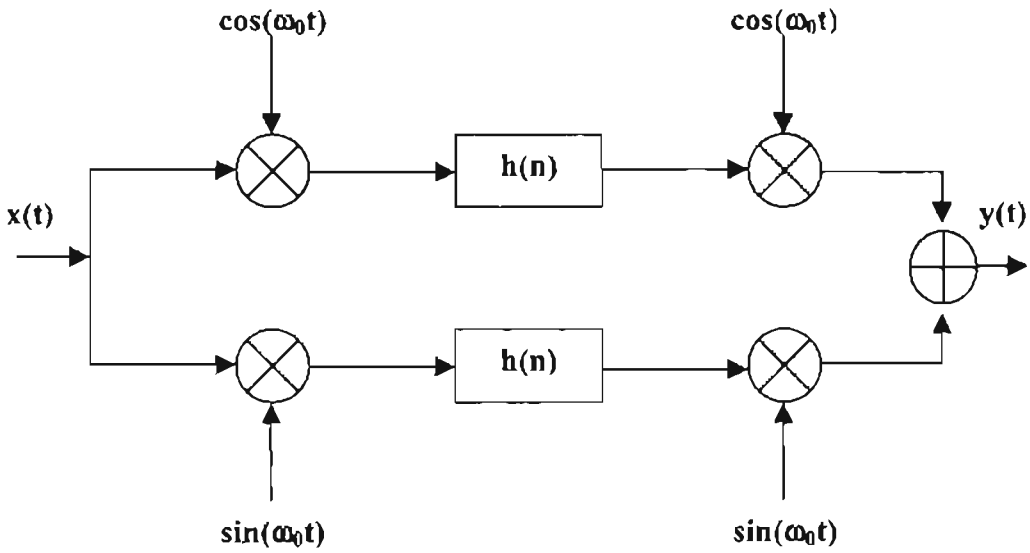


Figure 2.4 The Basic Block of Heterodyne Unit.

2.2.2 Structural Issues

The concept of the heterodyne filter is drawn from classical communications theory. The structural configuration shown in figure 2.4, which we will use to implement the tunable band-pass filter, is employed in many communication systems for various different applications. Most of these applications are designed as analog circuits in transmitters and receivers for the communication channels. The most common application found for this technique is Single Side Band (SSB) modulation Radio Transceivers [13][14]. In these implementations, low-pass filters are used where the heterodyning frequency translates the incoming signal down to the low pass filter frequency for filtering and demodulation. Our tunable heterodyne band-pass filter, on the other hand, uses this communication concept to implement a unique filtering scheme, which can be tuned with a single parameter and can be adapted when used in an adaptive algorithm [8]. Using a digital scheme allows us to construct identical filters, which are essential in eliminating all the images generated due to the heterodyne process.

2.3 Tunable Heterodyne Band-pass Filter

The tunable heterodyne band-pass filter is a novel idea, which uses the concept of the heterodyne process to translate the incoming signal from base-band up to the fixed filter frequency [8][15]. The concept of tunable filtering by frequency translation has its roots embedded in communication theory. The basic idea behind the Tunable Heterodyne Filter is very similar to the concept of super-heterodyne used for radio transmission, where the signal coming into the system is heterodyned or mixed with a tunable heterodyne frequency which translates the signal to a fixed intermediate frequency (IF),

thus translating the signal to a new, but fixed, frequency spectrum. In the case of the super heterodyne receiver, the IF is filtered and then demodulated to produce the audio or video output. In our new heterodyne filter, the signal is translated by modulation with the heterodyne frequency to a fixed intermediate frequency (IF) of the filter and then a second heterodyne process brings the signal back to the original frequency, but it is now filtered by the action of the fixed IF filter. While the IF filter is fixed, the frequencies of the signals being filtered vary depending upon the heterodyne frequency. The result is a hardware efficient tunable filter with a single tuning parameter. In the tunable heterodyne filter structure, a fixed filter is implemented with constant coefficients, which is mathematically proven to give the same results as those obtained from a much more complicated tunable filter implemented with variable coefficients.

2.4 Complex Mathematical Analysis

A real signal is composed of imaginary and real components, that is if we view a real signal as a vector, then one of its constituent lies on the real axis and the other on the imaginary axis. We can represent a complex number as shown in figure 2.5

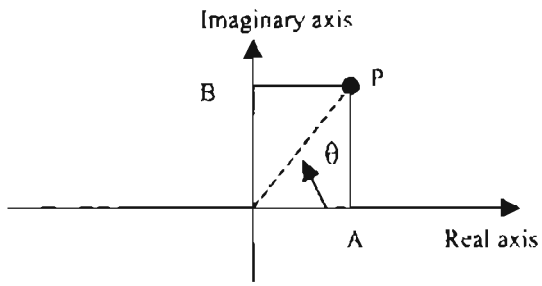


Figure 2.5 Complex number representation.

where 'P' is a complex number, represented in rectangular form by P = a + jb. The magnitude of this complex number is given by, $M = (a^2 + b^2)^{1/2}$ and the angle is given by, $\theta = \tan^{-1}(b/a)$. If we see this form as a vector instead of a point, we get a phasor representation or rotating vector. Figure 2.6 shows this effect.

The phasor shown in figure 2.6 has both real and imaginary parts and can be expressed as $e^{jfot} = \cos(2\pi fot) + j \sin(2\pi fot)$. Thus if the phasor is rotating as time progresses, we get a spiral shape helix coming out of the page. The projection of the helix on the real/time plane is a cosine wave and its projection on to the imaginary/time plane is the sine wave, as shown in figure 2.7.

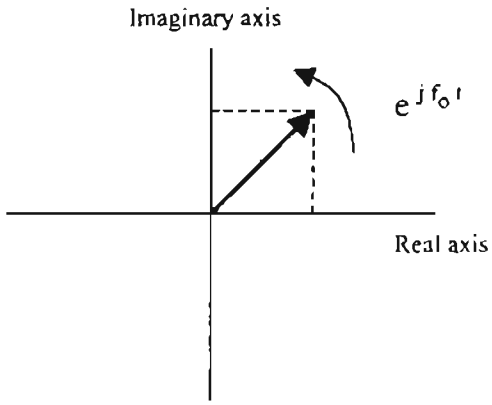


Figure 2.6 Complex Phasor representation.

Therefore, to represent a phasor in rectangular form we require two sinusoids, a cosine for the real part and a sine for the imaginary part. If we have two phasors, which are complex-conjugate pairs, rotating in a counter direction with respect to each other, the real components of the two complex-conjugate phasors would add constructively and the imaginary parts of the two complex-conjugate phasors would cancel each other out, thus producing a real sinusoid [16][17].

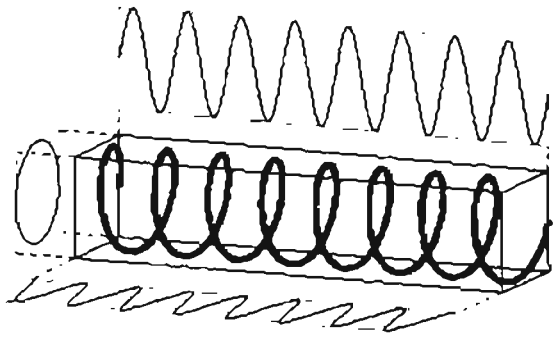


Figure 2.7 Helix formed by rotating Phasor in time.

In communication systems, real signals are represented as band-limited signals that have an equal amount of positive and negative frequencies. These signals behave in a similar way to the phasor representation above. Thus to make use of the property of rotating conjugate pair cancellation to obtain a purely real signal, the incoming signal to the Tunable Heterodyne filter is distributed in two branches, real (cosine) and imaginary (sine). Thus the results of the second heterodyne units are added to cancel all the images without the use of image canceling filters.

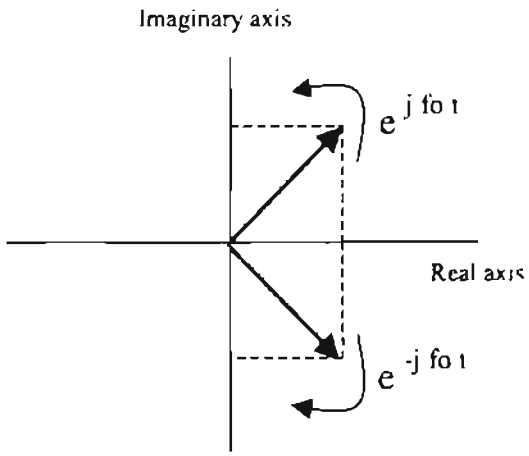


Figure 2.8 Conjugate pair rotating Phasor.

2.5 Field Programmable Gate Arrays (FPGA)

To implement the structure shown in figure 2.4 we used Xilinx FPGA. FPGAs are programmable devices that can be configured and reconfigured to provide circuit designers the flexibility of designing a system on a chip within the comfort of their lab. We used Xilinx Virtex 800 family FPGAs, which have some added features that allow having a better and more efficient system design. To synthesize the design behaviorally, we described the hardware using hardware description language (VHDL), which is then used in the Synplicity Synthesis tools to generate a Netlist file. The Netlist file created by the synthesis tools is then mapped to a Virtex library using the Alliance place and route tools. After mapping, the components are then placed and routed to create a configuration bit file, which can be downloaded into the Xilinx FPGA through the parallel port.

2.6 Hardware Implementation of the Tunable Heterodyne Filter

The implementation of the Tunable Heterodyne Band-pass filter was done using the VHDL (VHSIC Hardware Description Language) behavioral code. There are three parts to this structure (1) the Splitter, which translates the signal up to fixed filter frequency, (2) the Prototype IIR Filter, which is the fixed frequency filter used to extract the narrowband interference from the incoming signal, and (3) the Combiner which translates the signal back to the base-band and adds the output of the two branches, thus yielding a image free narrowband interference frequency. The transfer function of Tunable Heterodyne Band-pass Filter can be shown to be [8]:

$$H(z,\omega_h) = H(ze^{j\omega_h}) + H(ze^{-j\omega_h}) \qquad (2\text{-}1)$$

where, ω_h is the heterodyne frequency. Since the final transfer function of the tunable heterodyne filter is the sum of translated transfer functions of the fixed filter, some limitation is imposed on the types of filters that can be tuned using this structure. High-pass, low-pass and band-pass filters are the best candidates for implementation with this technique. There are a few unique band-pass filters, which can be converted into a notch filter and thus made tunable with this heterodyne technique. An example of such a filter is a simple 2nd order band-pass filter, with the following transfer function:

$$H(z,\omega_h) = \frac{(1-\alpha)}{2}\frac{(1-z^{-2})}{(1+\alpha z^{-2})} \qquad (2.2)$$

2.7 Simulation and Experimental Results

To simulate the behaviour of the tunable heterodyne band-pass filter, we made use of Matlab tools. The results obtained from both the simulation and hardware show the feasibility of tunable heterodyne filtering technique. It is apparent from the results that for the extraction of narrowband interference signals, we would require an extremely narrowband band-pass filter. As a consequence of this statement, we require a high order band-pass filter with sharp transition bands to avoid elimination and distortion of the information signal contained in the spread-spectrum signal. Thus an estimate comparison can be devised to show that it would be more hardware efficient to implement a tunable heterodyne filter for narrowband interference suppression.

Chapter 3

Tunable Heterodyning Filter Concept

3.1 Introduction

The notion of heterodyne filtering stems from communications theory and is embedded in many analog wireless applications. Heterodyning (figure 3.1) is another term for frequency translation, which is the process of mixing (multiplying) an incoming signal with a heterodyning frequency to shift the entire signal to a new frequency spectrum. The following equations give the Fourier transform of the cosine and sine functions:

$$\cos(\omega_0 t) \leftrightarrow 1/2 \quad [d(\omega - \omega_0) + d(\omega + \omega_0)]$$

$$\sin(\omega_0 t) \leftrightarrow 1/2j \quad [d(\omega - \omega_0) - d(\omega + \omega_0)]$$

Therefore, multiplying by an input signal $x(t)$ where $x(t) \leftrightarrow X(\omega)$:

$$x(t) * \cos(\omega_0 t) \leftrightarrow 1/2 \quad [X(\omega - \omega_0) + X(\omega + \omega_0)]$$

$$x(t) * \sin(\omega_0 t) \leftrightarrow 1/2j \quad [X(\omega - \omega_0) - X(\omega + \omega_0)]$$

The incoming signal which is considered as a real band-pass signal centered at a certain frequency (ω) when multiplied with the local oscillator (LO) operating at a fixed

intermediate frequency (ω_0), translates the input signal to the new spectral locations of $(\omega - \omega_0)$ and $(\omega + \omega_0)$, as shown in figure 3.2.

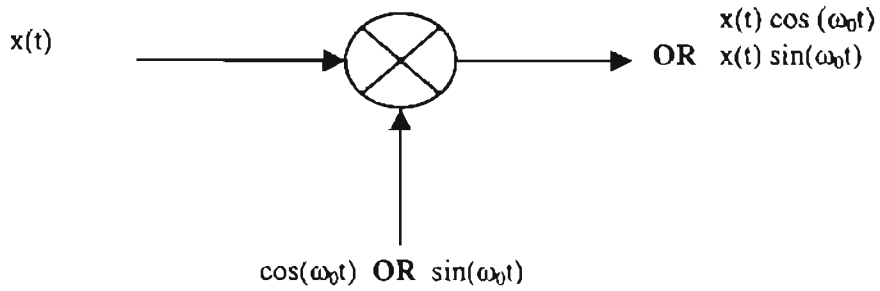


Figure 3.1 Heterodyne Structure with "ω_0" as heterodyning frequency.

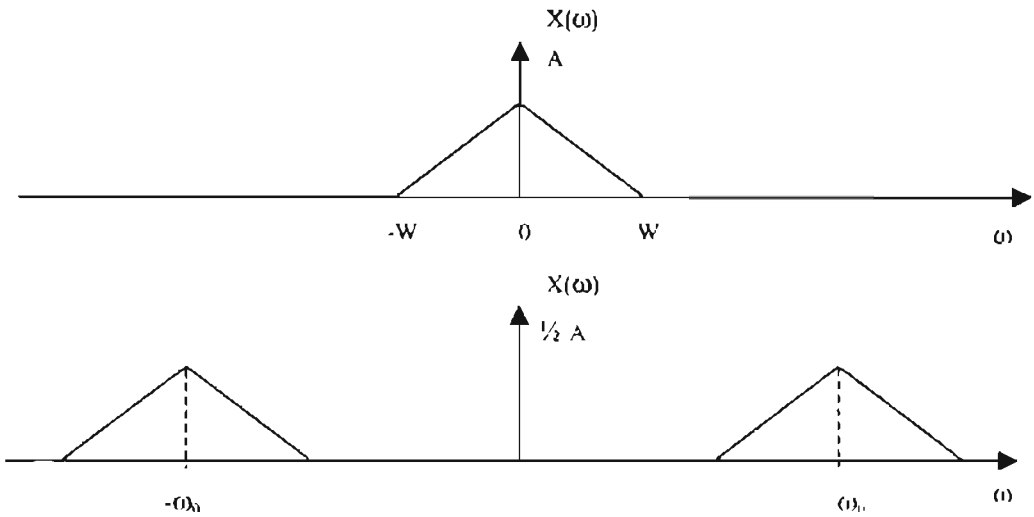


Figure 3.2 The Heterodyned Signal.

The incoming signal for the tunable heterodyne band-pass filter is assumed to be a broadband signal that is corrupted with narrowband interference during its path from the transmitter to the receiver. The heterodyning frequency is selected to shift the input signal to a new frequency spectrum, thus assuring that the narrowband interference falls within the pass-band frequencies of the fixed filter. This technique works perfectly in getting the signal to the fixed filter frequency where the filtering occurs. The problem with the

technique of heterodyning is the generation of images whenever two frequencies are multiplied. Although the images formed in the first stage of heterodyning are eliminated because of the selection of a narrowband of frequencies by the band-pass filter, when the signal is translated back to it's base-band with a second stage of heterodyne, images are generated again. In order to cancel these images, we have two branches which are 90^0 out of phase from each other and are discussed in detail later.

3.2 Basic Tunable Heterodyne Filter Structure

The Basic Heterodyne Tunable Filter Block is divided into three parts, performing distinct operations: 1) Splitter 2) Band-Pass Filter 3) Combiner

3.2.1 Splitter

The Splitter is a direct digital up-converter that multiplies the input signal by sine and cosine at the heterodyne frequency to create the intermediate frequency quadrature signals. The input for this application will be an 8-bit 2's complement fixed-point audio signal sampled at 48kHz.

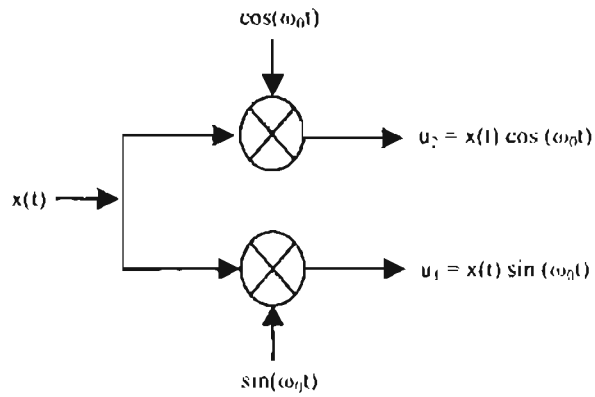


Figure 3.3 The Splitter Structure.

The Splitter is the first part of the Tunable Heterodyne Band-pass filter. It splits the incoming signal $x(t) \leftrightarrow X(f)$ into real and imaginary parts thus producing two outputs, input signal $x(t)$ multiplied with cosine and with sine, respectively. Both the sine and the cosine frequency is the heterodyning frequency and is the same for all the heterodyning units.

If we assume $X(f)$ as the incoming signal in frequency domain, equations 3.1 and 3.2 show the Fourier Transform of the results when the incoming signal is convolved with the Fourier transform of the sine and the cosine signals, respectively. As we can see from equations 3.1 and 3.2, the input signal is split into two images and each image is translated to a new spectral position, located at +/- ω_0, where ω_0 is the heterodyning frequency. The power of the incoming signal is equally split between each of the images.

$$U_1(z) = X(z) * \Im[\sin(\omega_0 n)] = \frac{1}{2j}\left[X(ze^{-j\omega_0}) - X(ze^{+j\omega_0})\right] \qquad (3.1)$$

$$U_2(z) = X(z) * \Im[\cos(\omega_0 n)] = \frac{1}{2}\left[X(ze^{-j\omega_0}) + X(ze^{+j\omega_0})\right] \qquad (3.2)$$

Both the branches produce the images in the same spectral locations but are 90^0 out of phase due to the imaginary term 'j' in equation 3.1. This defines the cosine branch as the real (In-phase) branch and sine branch as the imaginary (Quadrature) branch.

3.2.2 Prototype Filter

The next part of the Tunable Heterodyne Band-pass Filter is the Fixed Filter, which is required to isolate the narrowband interference. Both the branches have identical filters with the same transfer function. These filters are required to extract the

narrowband interference signal in both the channels. It is essential to have identical filters to cancel the images. This can be done easily with digital filters, whereas it is extremely difficult to produce identical analog filters. Since analog filters are implemented with discrete components which are greatly dependent on many factors which effect their values, this technique is perfectly suitable for implementation in the digital domain.

In this thesis, the fixed digital filter is implemented as a band-pass filter but can be replaced by a narrow low-pass or a high-pass filter that generates the tunable band-pass filter. Tunable heterodyne filters are suitable for tuning IIR filters, which are extremely difficult to tune but bear with them many advantages that would require a very high order FIR filter. Therefore, in this technique, it is suitable to have a narrowband band-pass IIR filter with sharp transition bands. Also due to the targeted applications, the requirement for having a linear phase is non-critical and thus makes IIR filters a more desired choice. Since the filter is fixed, it gives an advantage to the technique of heterodyning to have very high order filters, which can be designed with a constant coefficient multiplier using techniques such as Canonic Sign Digits (CSD) or Dempster-McLeod (DM) [19][20]. Implementing the heterodyne structure with higher order filters does not have as significant an effect on the hardware complexity when compared to the standard technique of constructing a tuning band-pass filter. To validate the theoretical aspects of the tunable heterodyning filter technique, a prototype second-order band-pass filter was designed and implemented using FPGAs. The filter was chosen to be centered at quarter of the sampling frequency (f_s / 4), where f_s is the sampling frequency. The filter equation is given by

$$H(z) = \frac{(1-\alpha)(1-z^{-2})}{2(1+\alpha\, z^{-2})}$$

$$where\ \ \alpha = 7/8 = 0.875$$

$$Then\ , H(z) = \frac{(0.125)\left[1-z^{-2}\right]}{2\left[1+(0.875)z^{-2}\right]}$$

$$\frac{Y(z)}{X(z)} = \frac{0.0625\ -0.625\ z^{-2}}{1+0.875z^{-2}}$$

$$\left[1+0.875z^{-2}\right]Y(z) = \left[0.0625\ -0.0625\ z^{-2}\right]X(z)$$

$$Y(z) + 0.875\, z^{-2}Y(z) = 0.0625\ X(z) - 0.0625\ z^{-2}X(z)$$

$$Y(z) = 0.0625\ X(z) - 0.0625\ X(z)z^{-2} - 0.875\, Y(z)z^{-2} \qquad (3.3)$$

$$Let\ \ k = 0.0625\ \ and\ \ r = 0.875\ , Then$$

$$Y(z) = kX(z) - kX(z)z^{-2} - rY(z)z^{-2} \qquad (3.4)$$

The heterodyne frequency is chosen to translate the interference signal to the band-pass filter frequency. Figure 3.4 shows the insertion of Band-Pass Filters in both the channels.

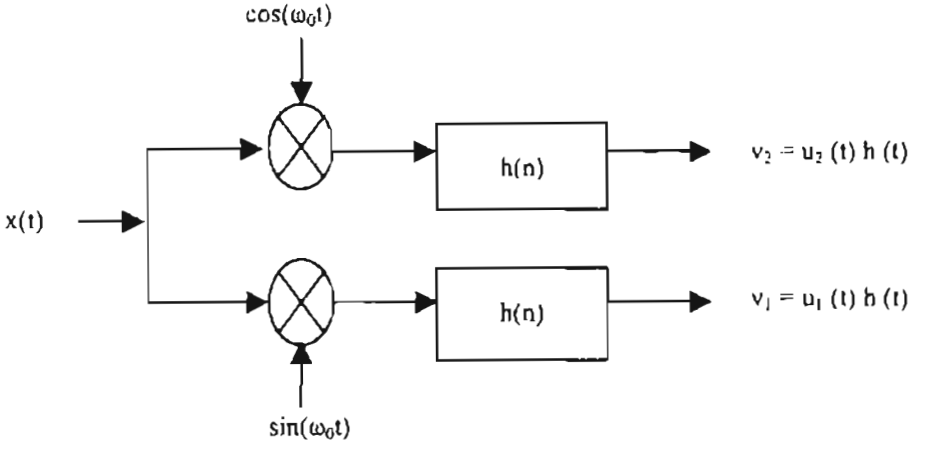


Figure 3.4 Splitter and Band-pass filter configuration.

The transfer function of the filter is:

$$H(z) = \frac{B(z)}{A(z)}$$

Equations 3.5 and 3.6 show in the form of Fourier Transform, the effect after the output of the Splitter is multiplied by the transfer function of the prototype filter. $U_1(z)$ and $U_2(z)$ are taken from equations 3.1 and 3.2, respectively.

$$V_1(z) = U_1(z)H(z) = \frac{1}{2j}\left[X(ze^{-j\omega_0}) - X(ze^{+j\omega_0})\right]H(z) \qquad (3.5)$$

$$V_2(z) = U_2(z)H(z) = \frac{1}{2}\left[X(ze^{-j\omega_0}) + X(ze^{+j\omega_0})\right]H(z) \qquad (3.6)$$

The Splitter circuit translates the input signal to the fixed frequency of the band-pass filter, which allows the band-pass filter to select out the interference from the input signal.

3.2.3 Combiner

The final part of the tunable heterodyne filter is the Combiner structure. The Combiner is a direct digital down-converter that multiplies the intermediate frequency quadrature signals (one signal by sine and the second signal by cosine) thus bringing back the signal to its base-band. Since both the branches in the tunable heterodyne filter structure are out of phase, the summation of both the channels cancels the effect of all the images created by the heterodyning process. Figure 3.5 shows the structure for the combiner circuit.

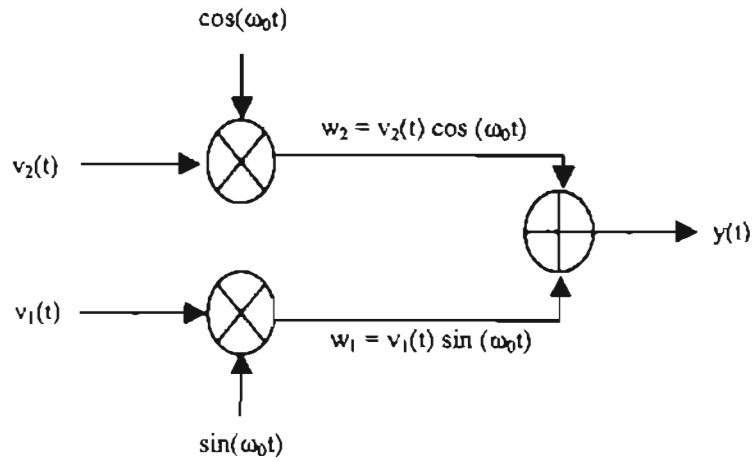


Figure 3.5 The Combiner Structure.

Equations 3.7 and 3.8 show the effect of convolving the output of the prototype filters in both channels with the Fourier transform of the sine and the cosine functions.

$$W_1(z) = V_1(z) * \Im[sin(\omega_0 n)] = \frac{1}{2j}\left[V_1(ze^{-j\omega_0}) - V_1(ze^{+j\omega_0})\right] \qquad (3.7)$$

$$W_2(z) = V_2(z) * \Im[cos(\omega_0 n)] = \frac{1}{2}\left[V_2(ze^{-j\omega_0}) + V_2(ze^{+j\omega_0})\right] \qquad (3.8)$$

Substituting the $V_1(z)$ and $V_2(z)$ terms from equation 3.5 and 3.6 respectively, we get the results shown in equations 3.9 and 3.10. Since $V_1(z)$ contains imaginary terms when convolved with another imaginary term, the imaginary terms cancel out and the equation becomes real. As we can see, there are no imaginary terms present in either of the equations. Since both the channels are real and have images produced in the same spectral locations but with different signs, when these terms are added, the images cancel each other out. This is the reason the transfer function of both of the prototype filters is required to be identical, otherwise the images would not cancel out completely and an image canceling filter needs to be implemented to extract the desired response.

$$W_1(z) = \frac{1}{4}\left[X(z) - X\left(ze^{+2j\omega_0}\right)\right]H\left(ze^{+j\omega_0}\right) + \frac{1}{4}\left[X(z) - X\left(ze^{-2j\omega_0}\right)\right]H\left(ze^{-j\omega_0}\right) \quad (3.9)$$

$$W_2(z) = \frac{1}{4}\left[X(z) + X\left(ze^{+2j\omega_0}\right)\right]H\left(ze^{+j\omega_0}\right) + \frac{1}{4}\left[X(z) + X\left(ze^{-2j\omega_0}\right)\right]H\left(ze^{-j\omega_0}\right) \quad (3.10)$$

Equations 3.11 and 3.12 show the response of the complete structure.

$$Y(z) = W_1(z) + W_2(z)$$

$$Y(z) = \frac{1}{2}\left[H(ze^{+j\omega_0}) + H(ze^{-j\omega_0})\right]X(z) \quad (3.11)$$

$$H(z,\omega_0) = \frac{Y(z)}{X(z)} = \frac{1}{2}\left[H(ze^{j\omega_0}) + H(ze^{-j\omega_0})\right] \quad (3.12)$$

The transfer function equation 3.12 for the tunable heterodyne filter is given in terms of ω_0, thus reflecting that poles and zeros are located at a new position as a function of ω_0. The transfer function of the tunable heterodyne filter gives the effect of the filter being moved or translated, but in reality the incoming signal was translated to the fixed filter. Thus if we express the "z" terms in form of $z = e^{j\omega}$, then $ze^{+/-j\omega 0}$ is translated in the frequency domain and is given by $e^{j(\omega +/- \omega 0)}$.

3.3 System Transfer Function Analysis for various Prototype Filters

In this section we will develop and analyze the characteristics of tunable heterodyne filter's transfer function with different types of prototype filters. To vigorously analyze the output response, we chose three different types of filters; High-

pass, Low-pass and Band-pass. Since the final transfer function of the tunable heterodyne filter is the sum of the translated transfer function of prototype filter (equation 3.12), notch filters cannot be used in this structure of a tunable heterodyne filter. It can be shown by this analysis that the range of tunability of the heterodyne filter is relative to the type of prototype filter used as the fixed filter. In this chapter, all the simulation from this point onwards will be considered to have a sampling frequency of 48000Hz and heterodyning frequency (ω_0) of 10000Hz.

3.3.1 High-pass Prototype Fixed Filter

Through the use of frequency translation, we can show that a high-pass filter can be converted into a tunable band-pass filter. Elaborating equation 3.12 in the following form can show the effect of frequency translation:

$$H(z) = \frac{1}{2}\left[H(e^{+j\omega}e^{+j\omega_0}) + H(e^{+j\omega}e^{-j\omega_0})\right]$$

$$H(z) = \frac{1}{2}\left[H(e^{j(\omega+\omega_0)}) + H(e^{j(\omega-\omega_0)})\right] \qquad (3.13)$$

A 6th order high-pass filter is taken as an example to show the consequence of each component in equation 3.13. Figure 3.6 shows the magnitude response of the 6th order high-pass filter and figure 3.7 shows the corresponding zero/pole plot.

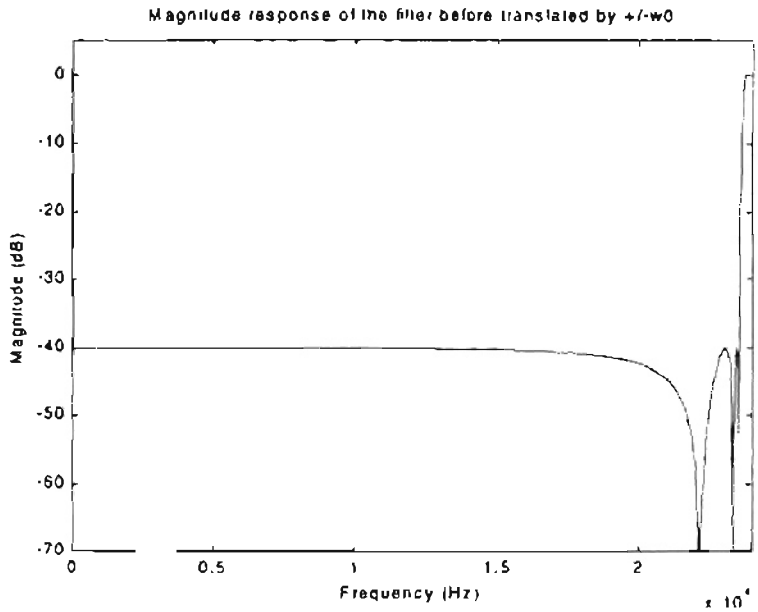


Figure 3.6 Magnitude response of original high-pass filter

The response of the transfer function $H(e^{j(\omega + \omega_0)})$ is obtained when the zero and pole locations of the fixed filter have been shifted by $+\omega_0$. This result shows that although the incoming signal is translated to the fixed filter, in the final response of the tunable heterodyne filter, it appears as if the filter has been translated to the signal. Figures 3.8 and 3.9 show the plot of the magnitude response of the translated transfer function and the relocated zeros/poles, respectively. From figure 3.7 we can see that the zeros and poles to the right of Nyquist frequency on the unit circle have their conjugates on the left side of the Nyquist frequency, thus giving a band-pass filter centered at Nyquist frequency. When we rotate these poles and zeros by $+\omega_0$ as shown in figure 3.9, we see that the center frequency of the band-pass filter is shifted to the left of the Nyquist frequency to (Nyquist + ω_0). Similarly, figures 3.10 and 3.11 show the magnitude response and the zeros/poles plot for the transfer function when shifted by $-\omega_0$.

respectively. Since the Nyquist is at 24kHz, translation by $+\omega_0$ centers the band-pass at 34kHz and translation by $-\omega_0$ centers the band-pass at 14kHz.

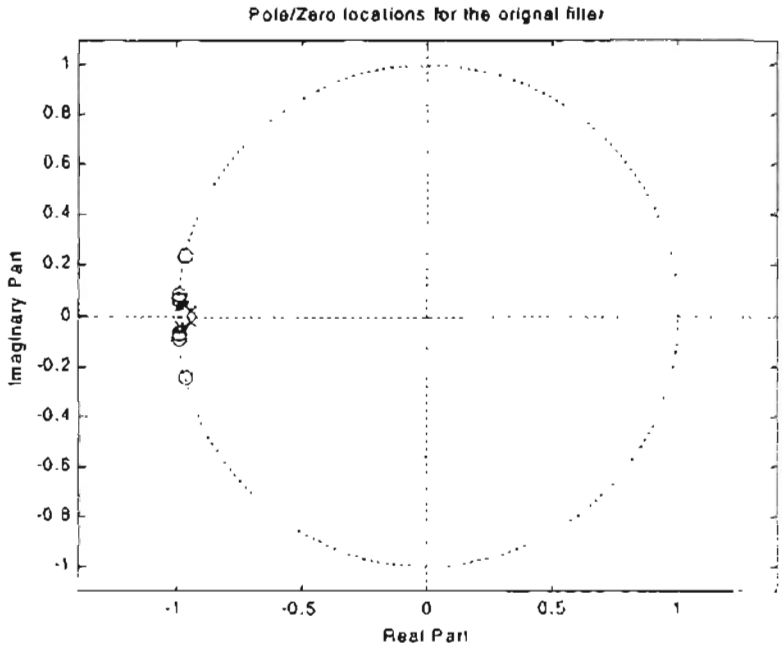


Figure 3.7 Zero/pole diagram for the 6th order high-pass filter

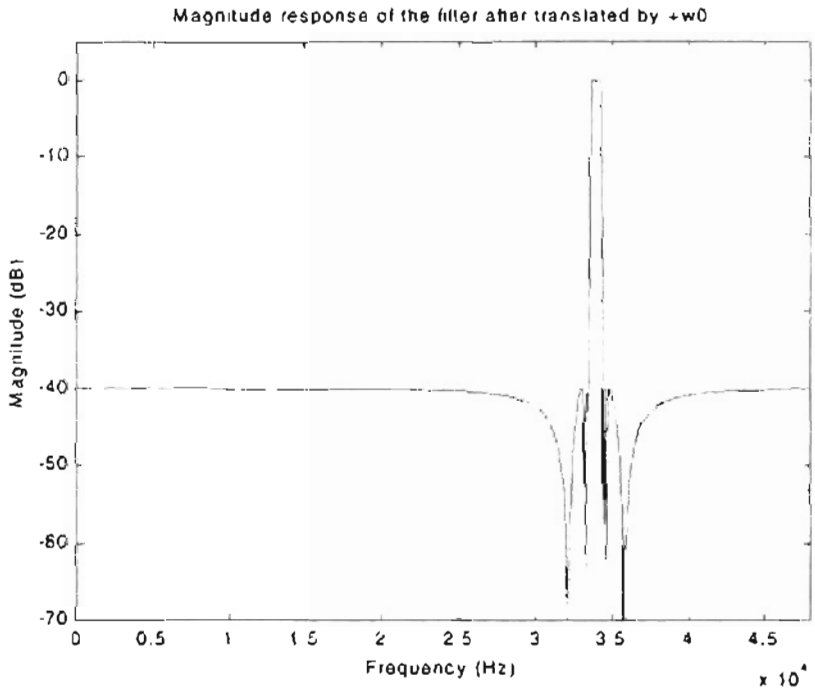


Figure 3.8 Magnitude response of the filter transfer function shifted by $+\omega_0$.

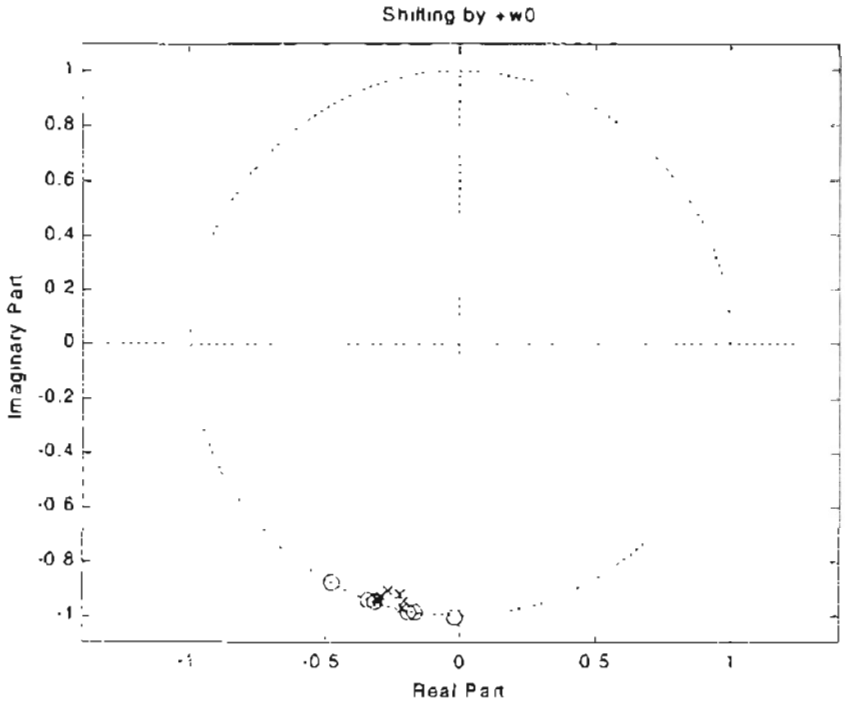


Figure 3.9 Zero/pole plot for the transfer function shifted by $+\omega_0$.

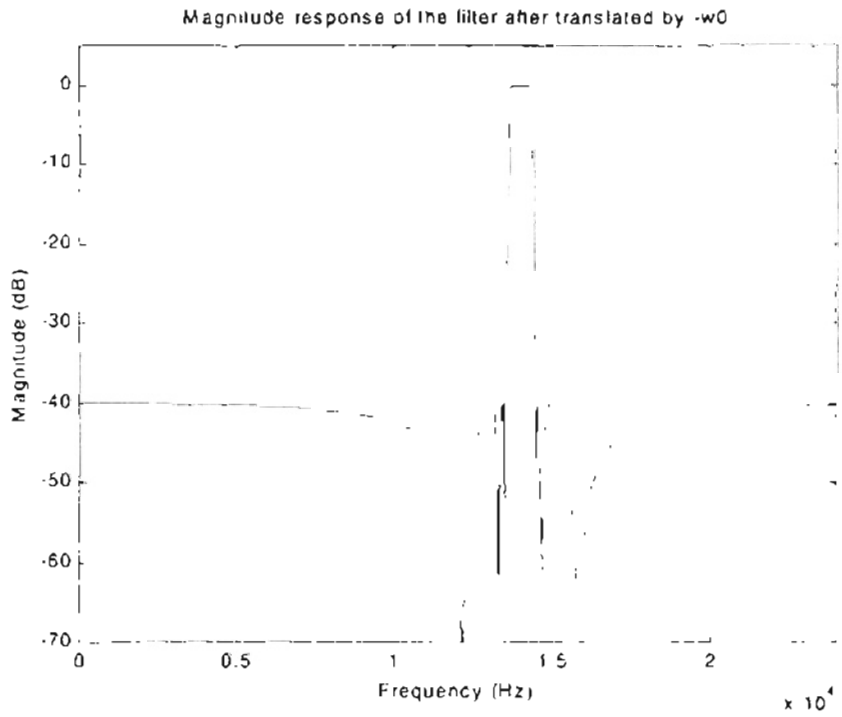


Figure 3.10 Magnitude response of the filter transfer function shifted by $-\omega_0$.

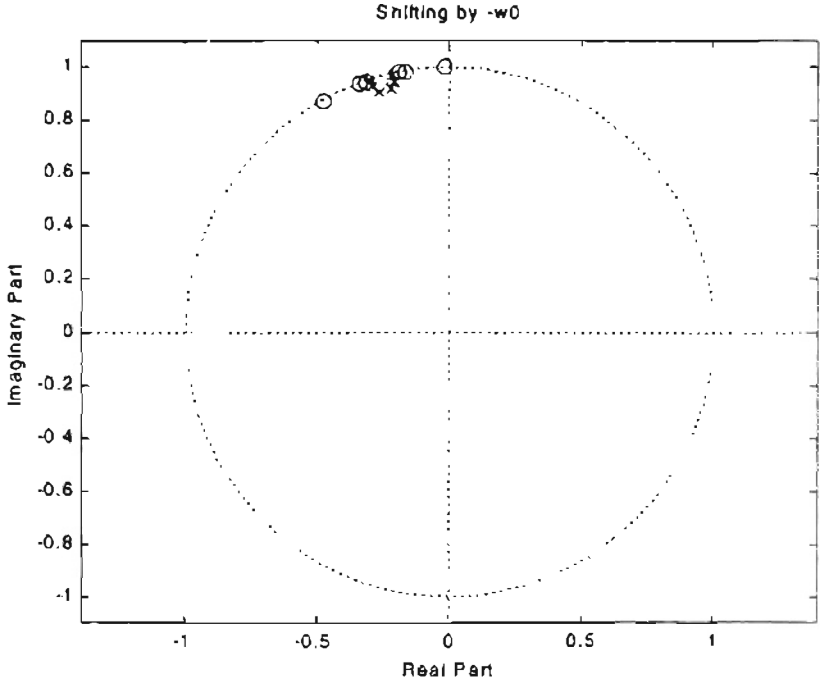


Figure 3.11 Zero/pole plot for the transfer function shifted by $-\omega_0$.

Now finally we add the two transfer functions, as is done in equation 3.13. We get the magnitude response and zeros/poles locations as shown in figure 3.12 and 3.13, respectively.

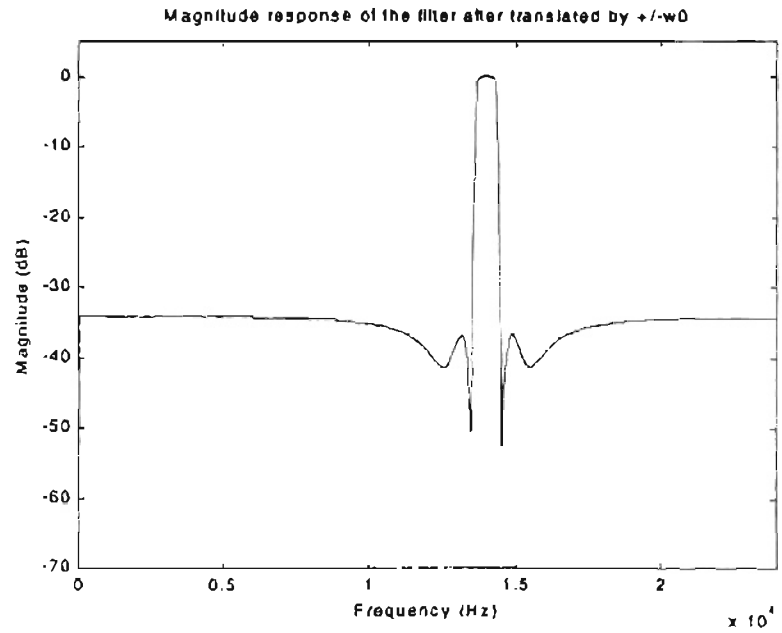


Figure 3.12 Magnitude response of the tunable heterodyne filter.

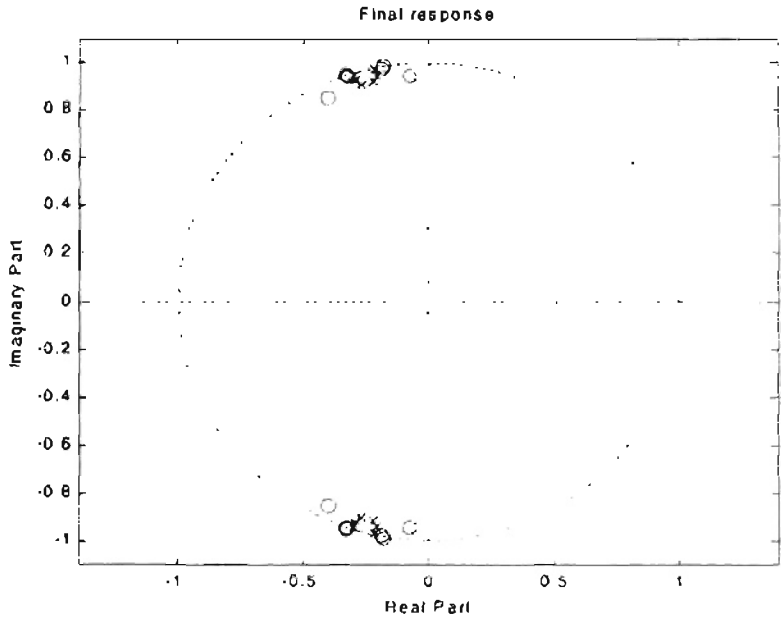


Figure 3.13 Zero/pole plot for the tunable heterodyne filter transfer function.

The final transfer function gives a 12^{th} order band-pass filter centered at frequency ($F_S/2$ - f_0), where F_S is the sampling frequency and f_0 is the heterodyning frequency. Thus this band-pass filter, which is generated from a high-pass filter through the process of frequency translation, can be tuned continuously from DC to the Nyquist frequency.

3.3.2 Low-pass Prototype Fixed Filter

The characteristics of the output response of the tunable heterodyne filter with a low-pass filter used as a fixed prototype filter are similar to the results obtained when a high-pass filter is used as a fixed prototype filter. Therefore we go through the similar exercise as we did with the high-pass filter and observe the results obtained when a 6^{th} order low-pass filter is used. The magnitude response and the zero/pole plot for the low-pass filter is given in figures 3.14 and 3.15, respectively.

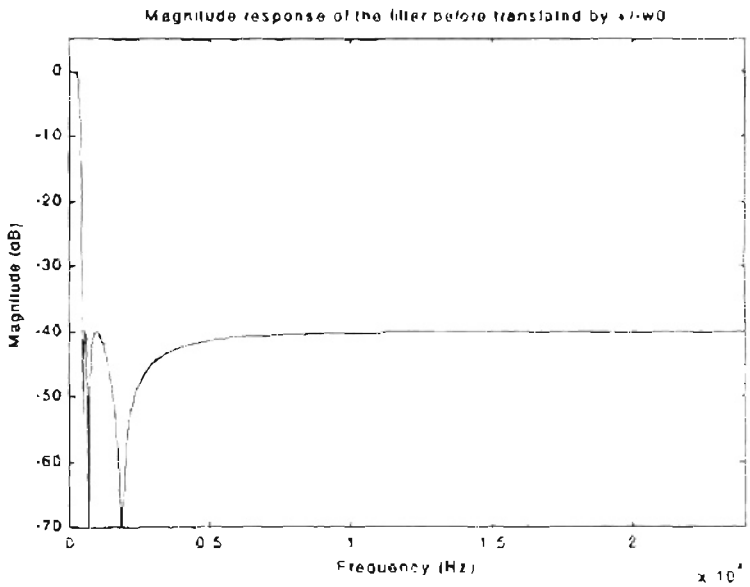


Figure 3.14 Magnitude response of original Low-pass filter

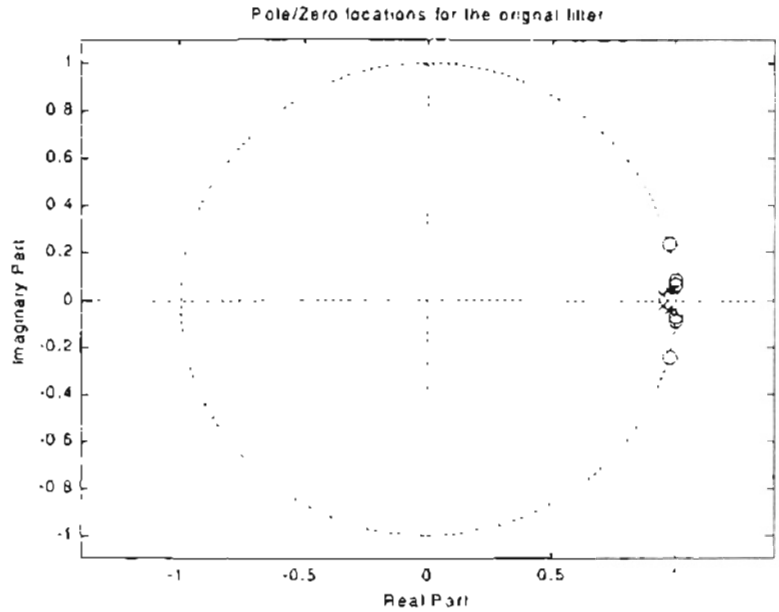


Figure 3.15 Zero/pole diagram for the 6th order Low-pass filter

Figures 3.16 and 3.17 show the responses when the transfer function of the fixed low-pass filter is shifted by $+\omega_0$. Similarly, figures 3.18 and 3.19 shows the responses when the transfer function of the fixed low-pass filter is shifted by $-\omega_0$.

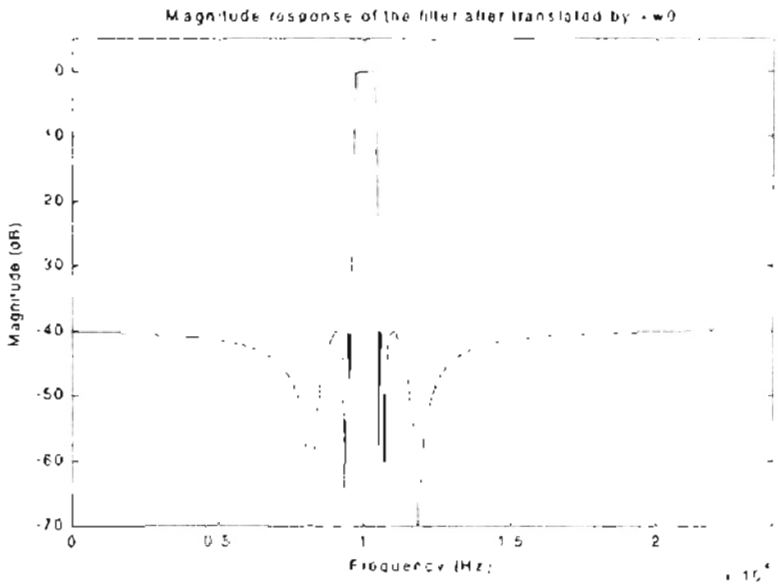


Figure 3.16 Magnitude response of the filter transfer function shifted by $+\omega_0$.

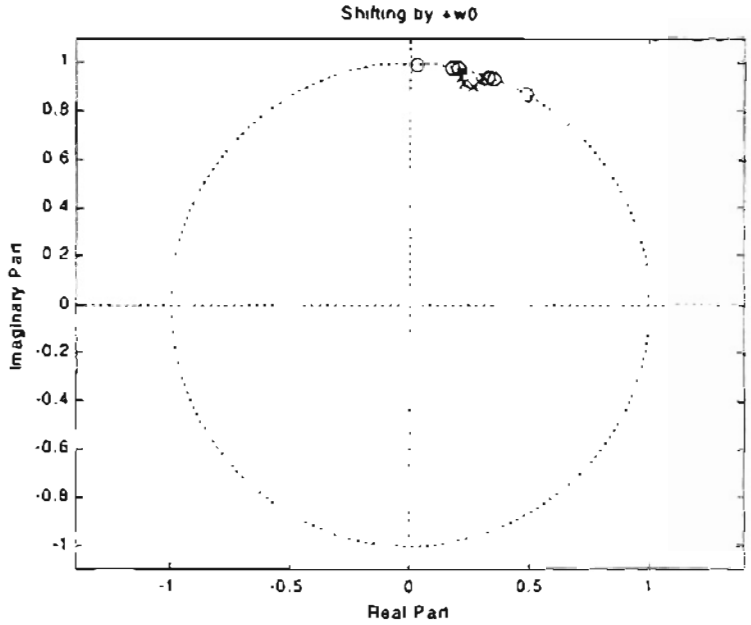


Figure 3.17 Zero/pole plot for the transfer function shifted by $+\omega_0$.

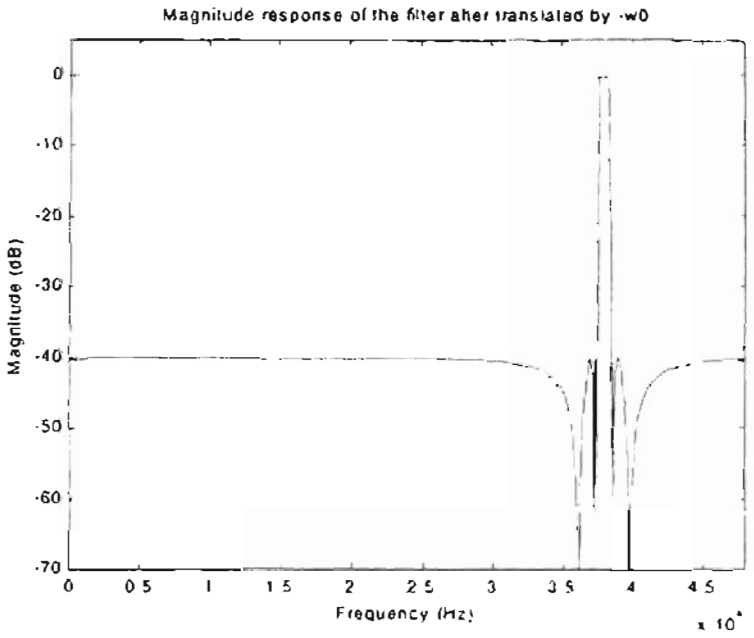


Figure 3.18 Magnitude response of the filter transfer function shifted by $-\omega_0$.

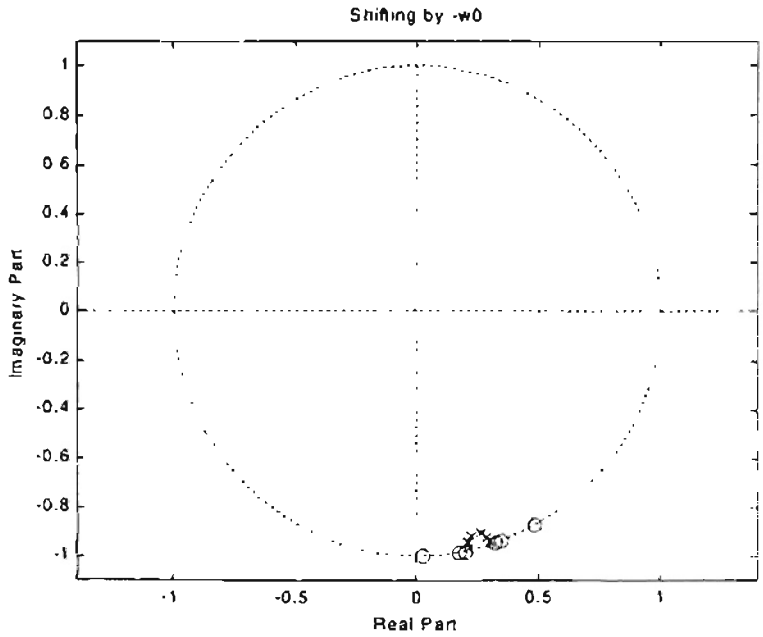


Figure 3.19 Zero/pole plot for the transfer function shifted by $-\omega_0$.

Due to the zeros and poles conjugate mirroring effect, the low-pass filter response produces an effect of having a band-pass filter centered at DC and therefore translation by $+\omega_0$ centers the band-pass at 10kHz and translation by $-\omega_0$ centers the band-pass at 38kHz. Comparing these results with the response of high-pass filter, we see that the only difference between the responses of the high-pass filter and the low-pass filter is where the resulting band-pass filter is centered. When we add these two translated transfer functions, we get the responses given in figure 3.20 and 3.21.

Again a 12^{th} order band-pass filter is obtained which is centered at frequency (ω_0), where ω_0 is the heterodyning frequency. Thus this band-pass filter, which is generated from a low-pass filter through the process of frequency translation, can be tuned continuously from DC to the Nyquist frequency as well.

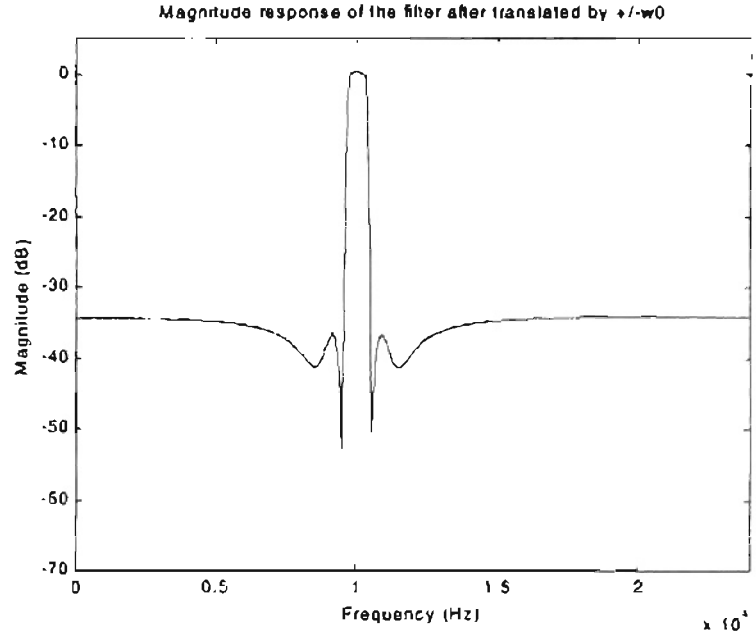


Figure 3.20 Magnitude response of the tunable heterodyne filter.

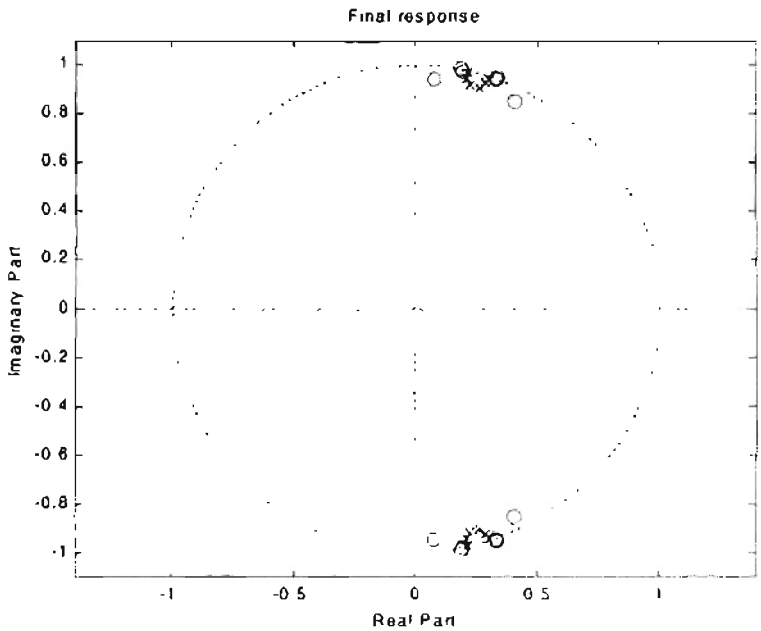


Figure 3.21 Zero/pole plot for the tunable heterodyne filter transfer function.

3.3.3 Band-pass Prototype Fixed Filter

In the discussion above we have converted a 6^{th} order high-pass and a 6^{th} order low-pass filter to a 12^{th} order tunable band-pass filter. In this sub-section we will develop

and analyze the results when a 6^{th} order band-pass filter is used as a fixed prototype filter in the tunable heterodyne structure. The type of filters used as prototype filter in the tunable heterodyne filters depends on the application where the structure is employed. The magnitude response of the band-pass filter is given in figure 3.22 and its corresponding zeros/poles plot is given in figure 3.23.

We go through the same procedure to rotate the poles and zeros of this band-pass filter by $+\omega_0$ which gives us the corresponding magnitude response shown in figure 3.24 and the zeros/poles plot shown in 3.25. Similar to high-pass and low-pass filters, we rotate the zeros/poles location by $-\omega_0$ to get the magnitude response and zero/pole plot as shown in figures 3.26 and 3.27, respectively.

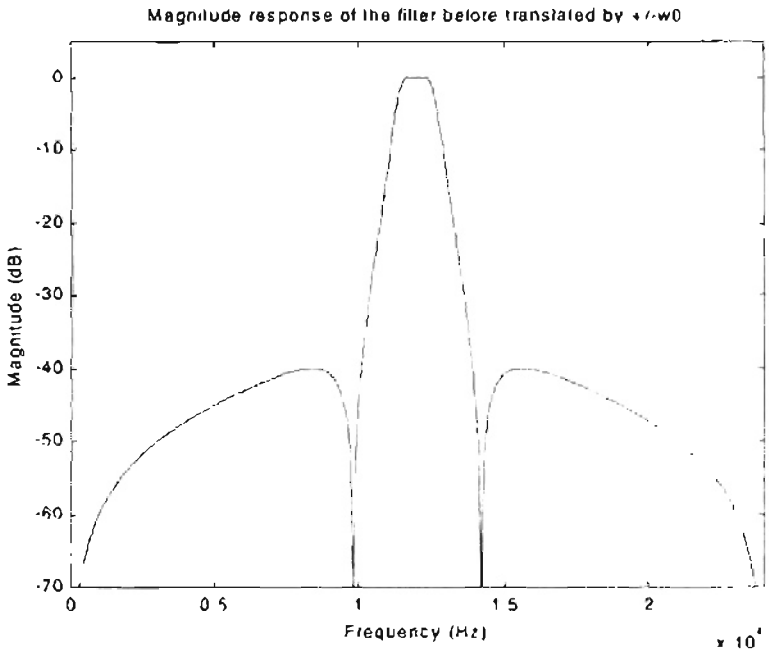


Figure 3.22 Magnitude response of original Band-pass filter

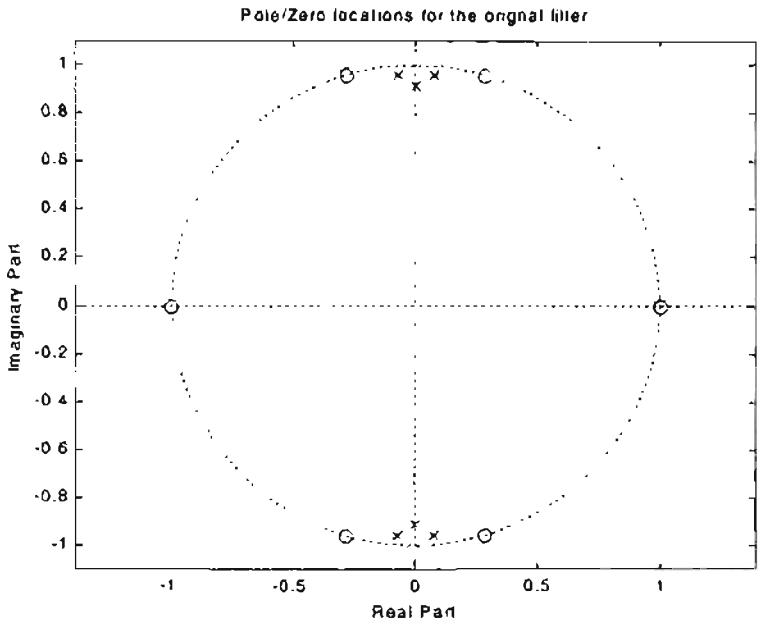


Figure 3.23 Zero/pole diagram for the 6th order Band-pass filter

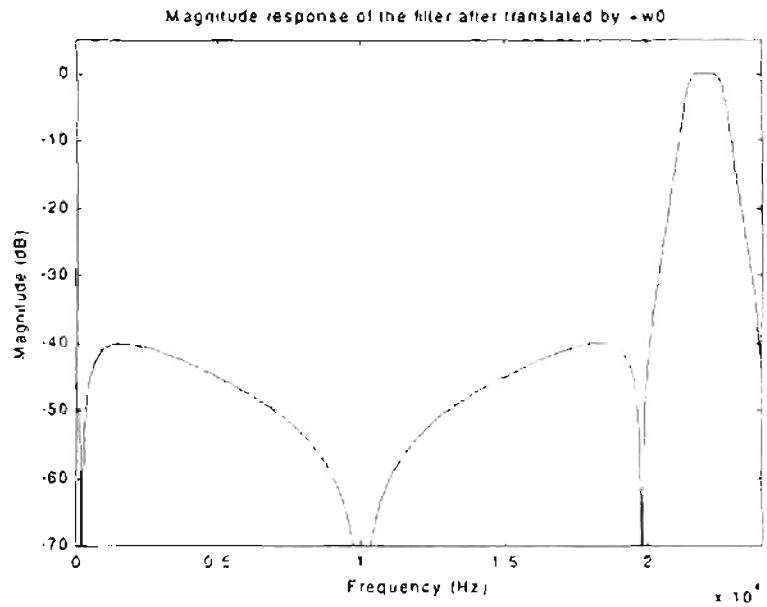


Figure 3.24 Magnitude response of the filter transfer function shifted by $+\omega_0$.

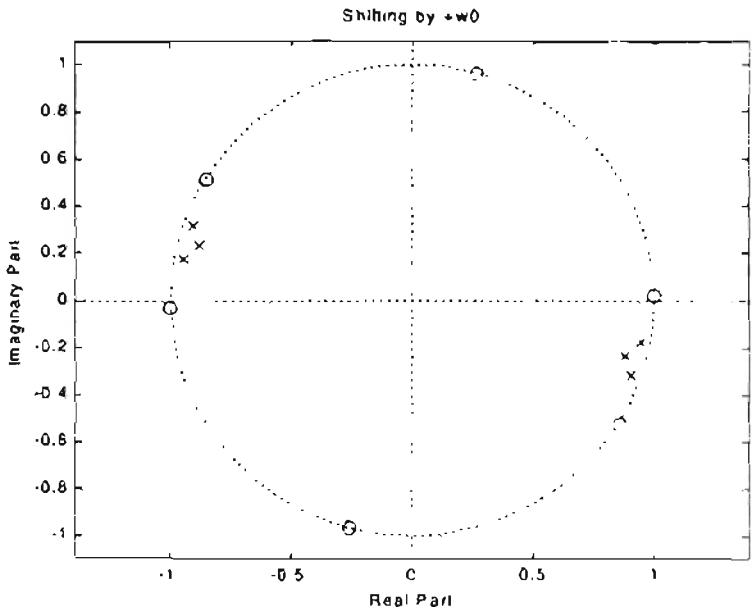


Figure 3.25 Zero/pole plot for the transfer function shifted by $+\omega_0$.

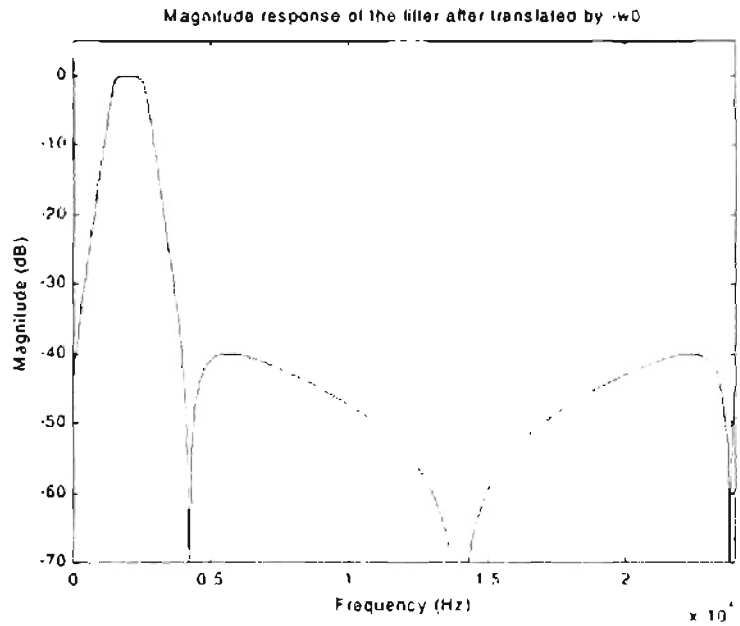


Figure 3.26 Magnitude response of the filter transfer function shifted by $-\omega_0$.

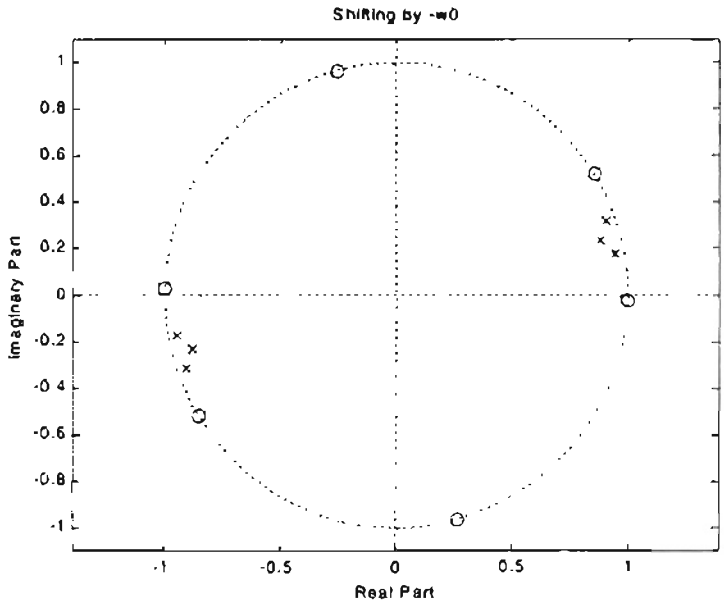


Figure 3.27 Zero/pole plot for the transfer function shifted by $-\omega_0$.

The fixed prototype band-pass filter is centered at a quarter of the sampling frequency ($F_S/4$). Therefore, when we translate the zeros and poles of the band-pass filter to $+/-\omega_0$ (where the heterodyning frequency f_0 is set at 10000Hz) and add these translated transfer functions as in equation 3.13, we get two band-pass filters; one centered at 2000Hz and the other one at 22000Hz. The final magnitude response and corresponding zero/pole plot for the tunable heterodyne filter is shown in figure 3.28 and 3.29, respectively.

As a consequence of having two band-pass filter responses at the output of the tunable heterodyne filter, the tuning range for this tunable heterodyne structure with the band-pass filter as prototype filter is split into two regions. Therefore we have to apply a constraint to the incoming interference frequency, thus limiting the tuning range from either DC to Nyquist/2 or Nyquist/2 to Nyquist frequency.

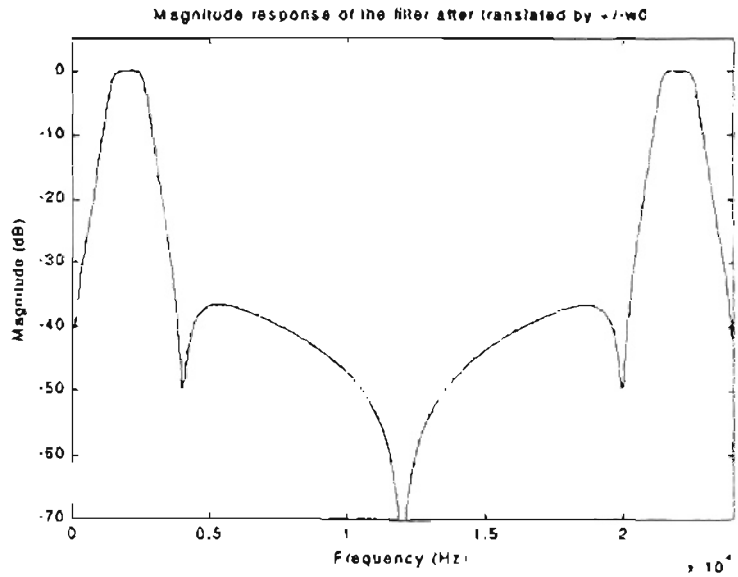


Figure 3.28 Magnitude response of the tunable heterodyne filter.

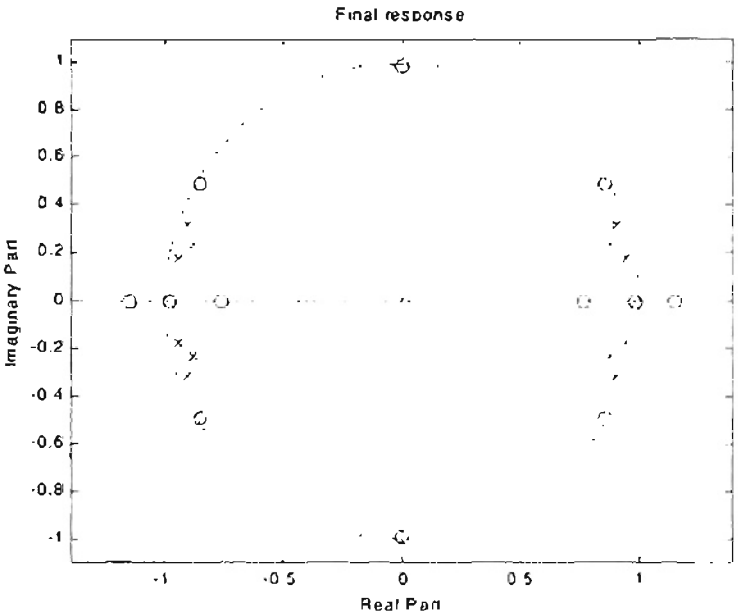


Figure 3.29 Zero/pole plot for the tunable heterodyne filter transfer function.

3.3.1 Band-stop (Notch) Prototype Fixed Filter

Since the transfer function of the tunable heterodyne filter (given in equation 3.13) is a sum of the two translated filter transfer functions, a notch filter cannot be made tunable with this structure. Figure 3.30 shows the magnitude response of a notch filter and figure 3.31 shows the magnitude response of the tunable heterodyne final output when a notch filter is utilized as the prototype filter. The output response of the tunable heterodyne filter structure in figure 3.31 shows that the notches are only 3dB deep, thus rendering the output response to appear as almost an all-pass filter.

One way to get around this problem is to have a band-pass filter that has a unique property where it can be converted into a perfect notch filter. But this characteristic for particular band-pass filters is extremely rare and limited. A special second order filter with the following transfer function can be converted into a stop-band filter

$$H(z) = \frac{(1-\alpha)}{2}\frac{(1-z^{-2})}{(1+\alpha z^{-2})} \qquad (3.14)$$

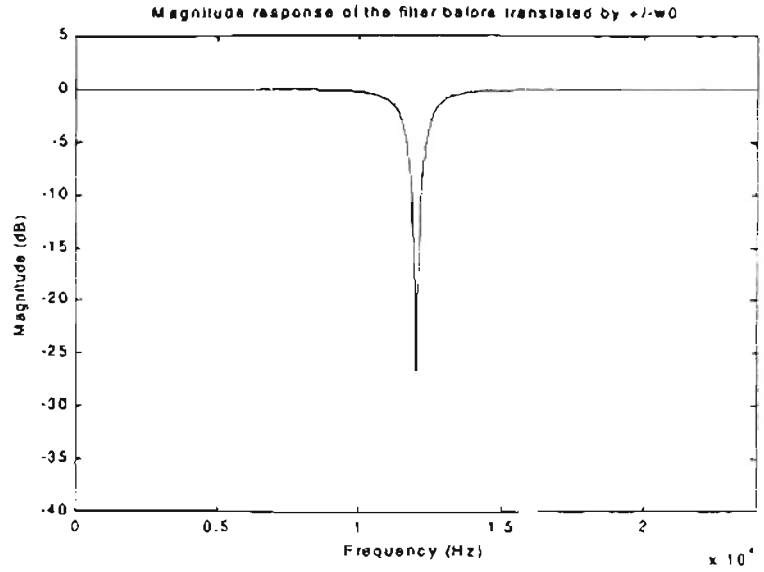


Figure 3.30 Magnitude response of original Stop-band filter

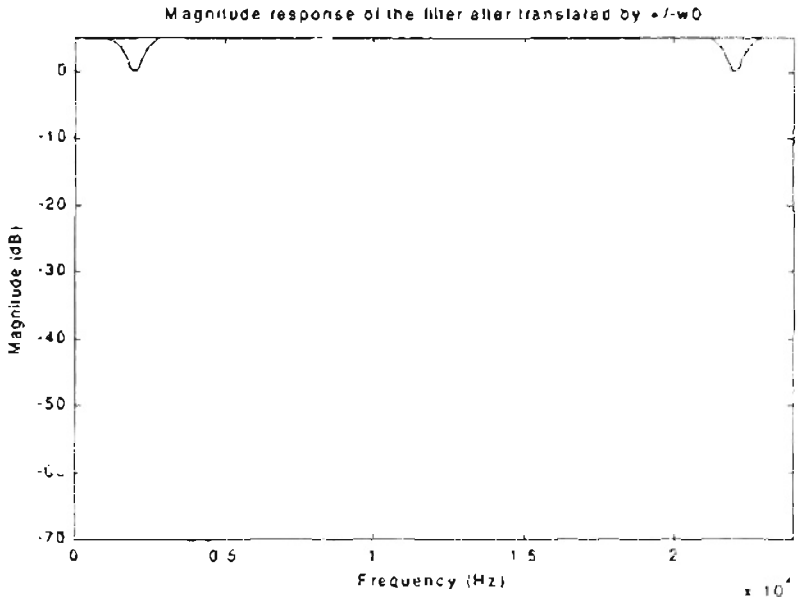


Figure 3.31 Magnitude response of the tunable heterodyne filter.

The magnitude response of the band-pass filter and the tunable heterodyne filter is shown in figures 3.32 and 3.33, respectively. Now if we subtract '1' from the tunable heterodyne filter transfer function we get:

$$H'(z,\omega_0) = H(z,\omega_0) - 1$$

$$= \frac{B(z,\omega_0)}{A(z,\omega_0)} - 1$$

$$= \frac{B(z,\omega_0) - A(z,\omega_0)}{A(z,\omega_0)}$$

$$= \frac{B'(z,\omega_0)}{A'(z,\omega_0)} \qquad (3.15)$$

Where H'(z, ω_0) is the new transfer function with zeros defined as B'(z, ω_0) and poles defined as A'(z, ω_0). Figure 3.34 shows the magnitude response of new transfer function. As can be seen from this figure we have similar results for the band-pass filter but this time we have two notches placed at 2000Hz and 22000Hz.

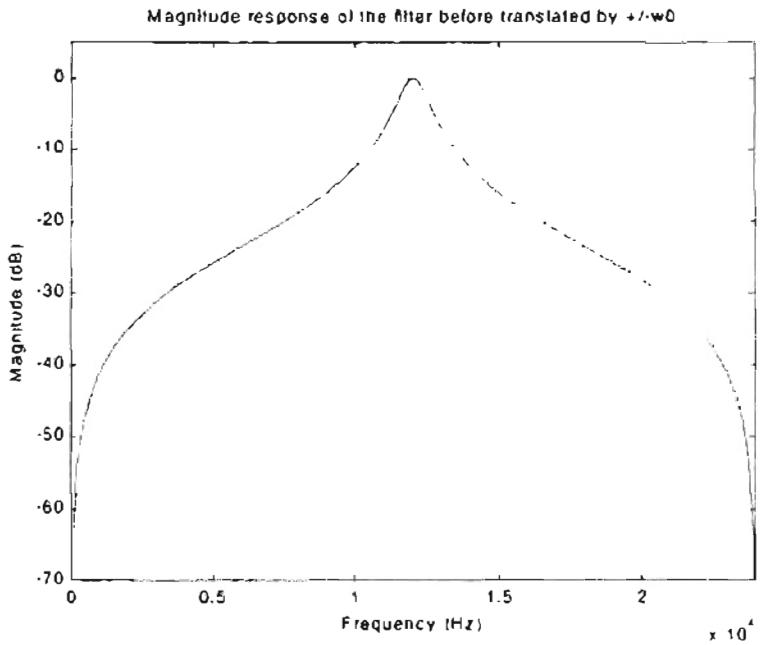


Figure 3.32 Magnitude response of 2nd order Band-pass filter

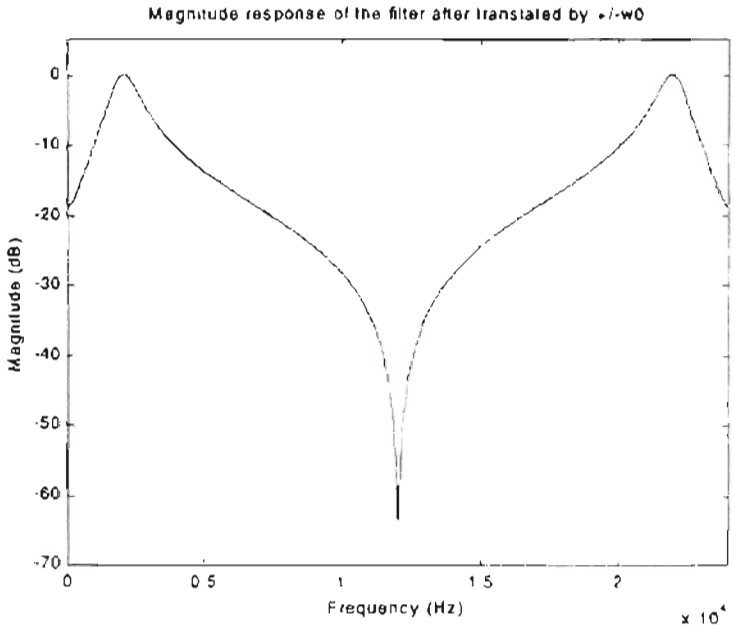


Figure 3.33 Magnitude response of the tunable heterodyne filter.

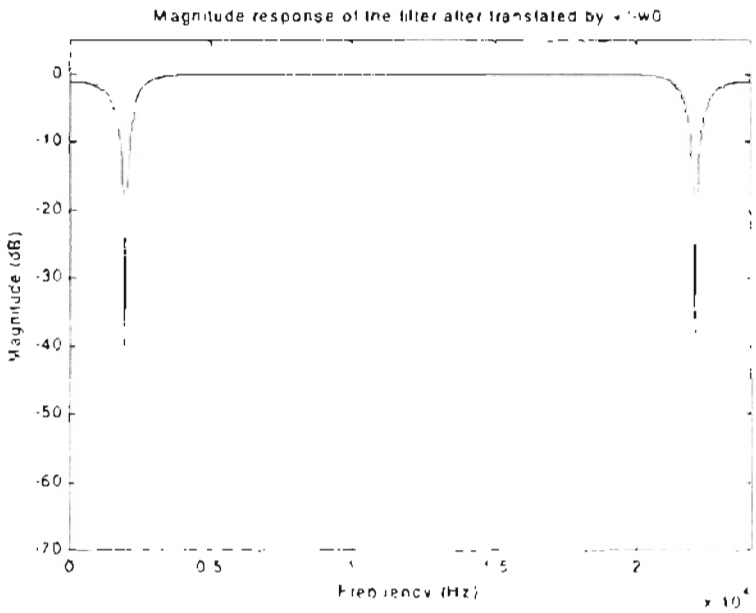


Figure 3.34 Magnitude response of the modified tunable heterodyne filter.

To show the effect when a band-pass filter that does not have the characteristic to be converted into a notch filter, we take the example of 6th order band-pass filter

described in sub-section 3.3.3. Figure 3.35 shows the magnitude response of the filter if we insert a 6^{th} order filter transfer functions in equation 3.14.

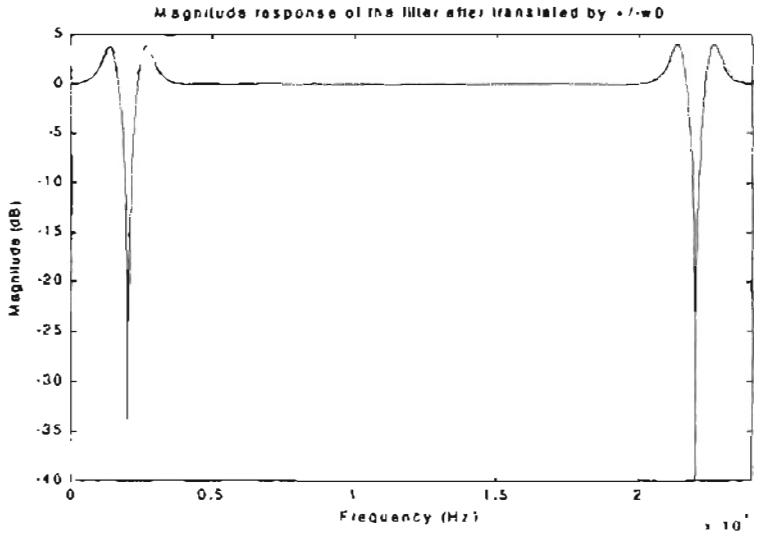


Figure 3.35 Magnitude response of distorted notch filters

The poles and zeros plot corresponding to the magnitude response shown in figure 3.35 is shown in figure 3.36.

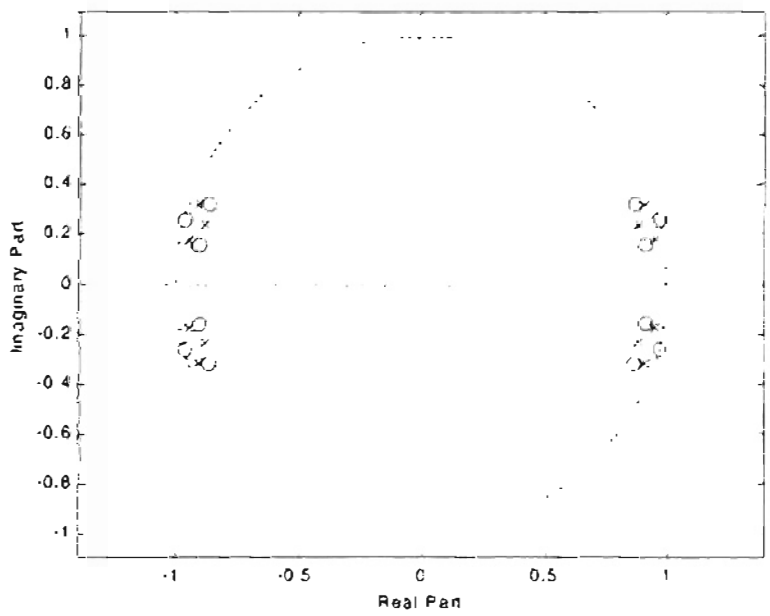


Figure 3.36 Zeros/poles plot of notch filter converted from band-pass filter

Thus the results show that if the specifications do not allow 4dB ripples at either edge of the notch, then this type of notch filter cannot be used. This matter can be further probed to find some unique way to produce distorted band-pass filter response, which yields a perfect tunable notch filter.

3.3.4 Tuning Ranges and Heterodyne Frequency Relationships

Summary

From the preceding analysis and results we can conclude that a band-pass, high-pass or a low-pass filter can be tuned easily with a single parameter, the heterodyning frequency. Notice that the range of tunability depends on the type of fixed prototype filter used in the tunable heterodyne filter structure; high-pass, low-pass or band-pass. Another factor that depends on the category of the prototype fixed filter used in the heterodyne structure is the heterodyne frequency. The interference signal coming in the tunable heterodyne filter is translated to the fixed filter frequency by the process of heterodyning. Thus there exists a relationship between the interference frequency and the fixed filter frequency. This relationship is used to calculate the heterodyning frequency required to bring the interference signal to the fixed filter. The following table summarizes the range of tuning frequencies for the different types of prototype filter. This table also illustrates the relationship between the interference signal frequency and the tuning heterodyne frequency.

	Low-pass Filter	High-pass Filter	Band-pass Filter
Tuning Range	From DC to Nyquist Frequency	From DC to Nyquist Frequency	1) From DC to Nyquist/2 Frequency OR 2) From Nyquist/2 to Nyquist Frequency
Heterodyne Frequency Relationships	$F_H = F_I$	$F_H = F_S/2 - F_I$	1) $F_H = F_F - F_I$ If, $F_I \leq F_S/4$ OR 2) $F_H = F_I - F_F$ If, $F_I > F_S/4$

Note It is assumed for the band-pass filter that the center of the pass-band is located at quarter of the sampling frequency
FH = Heterodyne Tuning Frequency
FS = Sampling Frequency
FI = Interference Frequency
FF = Fixed Filter Center Frequency

Table 3.1 Tuning range and heterodyne frequency relationships.

Chapter 4

Signal Analysis for Tunable Heterodyning Filter

4.1 Introduction

Figure 4.1 shows the block diagram of the Tunable Heterodyne Band-pass filter.

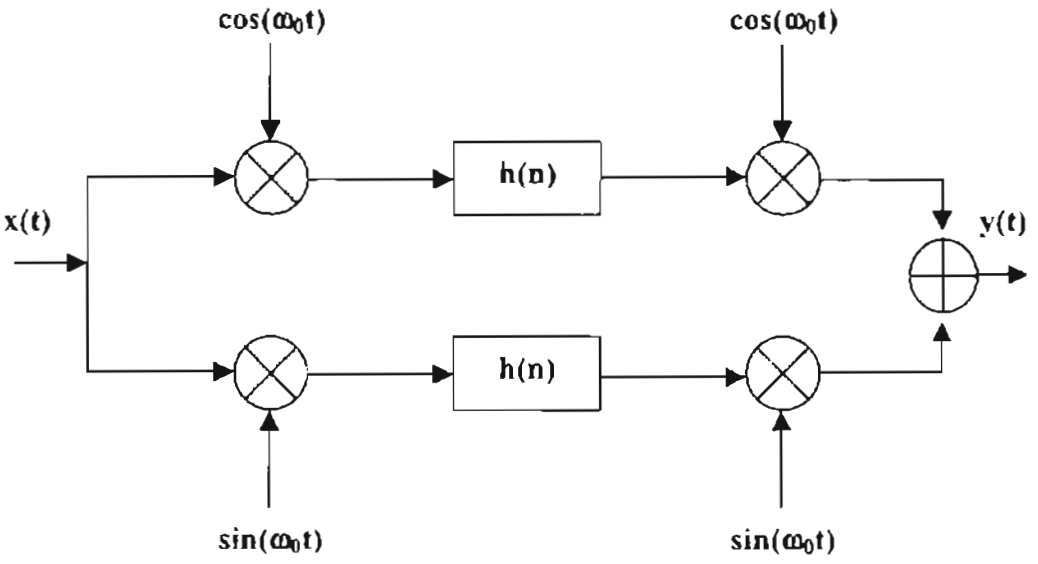


Figure 4.1 Block Diagram of Tunable Heterodyne Band-pass Filter.

Real signals are composed of an infinite number of frequencies limited to a bandwidth of 'W'. All the real signals have equal amounts of positive and negative frequencies. Therefore, for signal x(t) we would have spectrum |X(f)| = |X(-f)|. This is shown in figure 4.2, where the three dimensional real signal spectrum is shown at (+/- fo) frequency. The real (in-phase) component of the spectrum is shown on the real plane and the imaginary (Quadrature) component of the real signal spectrum is shown on the imaginary plane [17]. Terms, In-phase and Quadrature signal are used to signify the fact that the imaginary and real components of the real signals are 90^0 out of phase with respect to each other.

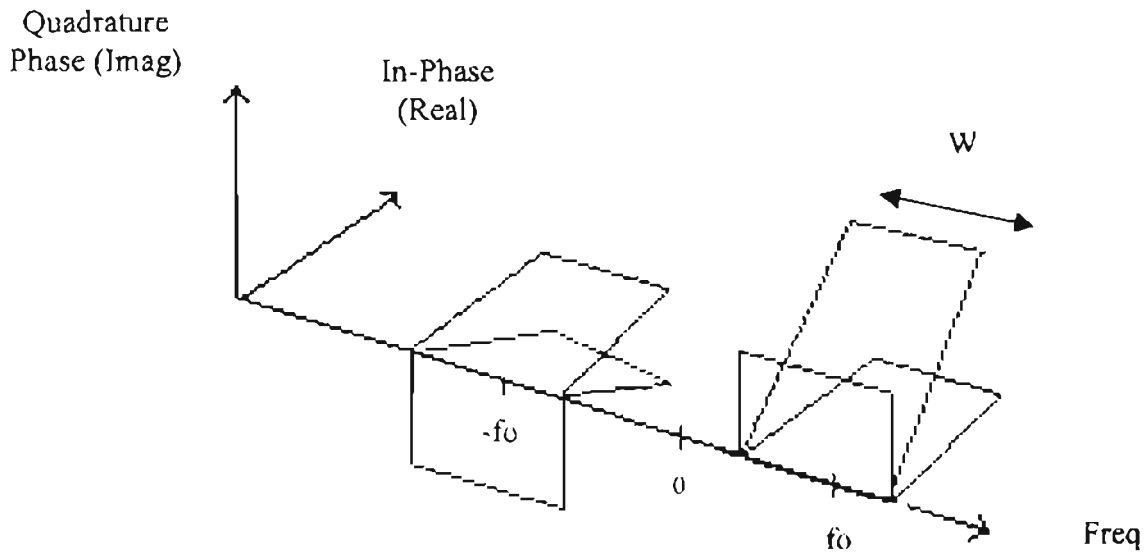


Figure 4.2 3D Representation of Real Signal Spectrum.

4.2 Signal Analysis of Tunable Heterodyne Band-pass Filter

4.2.1 Signal Response of In-phase and Quadrature Signals

The concept of tunable heterodyne band-pass filter utilizes the tricks of complex mathematics to eliminate the images generated through the process of heterodyning. This process yields a hardware efficient tunable band-pass filter with the ability to tune continuously across the range from DC to Nyquist frequencies in the case of high-pass and low-pass prototype filters and from DC to Nyquist/2 or Nyquist/2 to Nyquist in the case of band-pass prototype filters. To illustrate the concept used to develop the foundation of this theory, the following case can be used. We assume an input to our tunable heterodyne band-pass filter structure (figure 4.1) is a band limited signal of 'W' width centered at frequency f_C with narrowband interference at frequency f_i shown in figure 4.3. Figure 4.3 shows the real signal spectrum X(f) with both positive and negative frequencies centered at +/- f_c respectively.

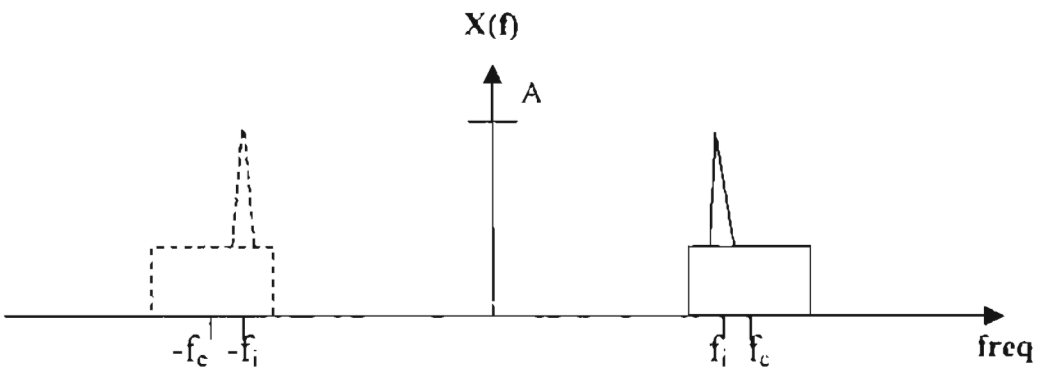


Figure 4.3 Wide-band signal with narrowband interference.

In the Splitter circuit, the input signal is distributed in two separate channels, where one of the branches multiplies the input signal with cosine and in the other channel the input signal is multiplied by sine. The frequency for both the sine and cosine signals' generation is set as the heterodyne frequency. For future reference we would identify the channel with cosine mixers as the 'In-phase' channel and channel with sine mixers as the 'Quadrature' channel, since both are 90^0 out of phase from each other [17].

Figure 4.4 shows the output of the first heterodyne unit in the "In-phase" channel. Both positive and negative signals are translated to plus and minus the heterodyne frequency f_r. Since the cosine function does not constitute an imaginary term, the spectrum is plotted on the real axis. This first heterodyne process translates the whole input spectrum to two different frequencies. Our goal in this part of the circuit is to translate the center frequency of the narrowband interference signal to the center frequency of the fixed band-pass filter. We set the tuning heterodyne frequency in such a way that the sum of the mixer frequency and the center frequency of the incoming narrowband interference equates to the center frequency of the fixed band-pass filter. The relationship to determine the heterodyne frequency for a different type of prototype filters is given in table 3.1. Thus, in this example it is assumed that the center frequency of the band-pass filter is equal to the sum of interference frequency (f_i) and heterodyne frequency (f_r). As a consequence of the heterodyning process, two images are formed in the positive frequencies and two images are formed in the negative frequencies. The images of the signal produced due to the $e^{j2\pi f_{01}}$ would add to the interference frequency (f_i) and images produced due to $e^{-j2\pi f_{01}}$ would subtract from the interference frequency

(f_i). Since the heterodyne process generates two replicas of the real signal, the spectral power of the images is half that of the real signal.

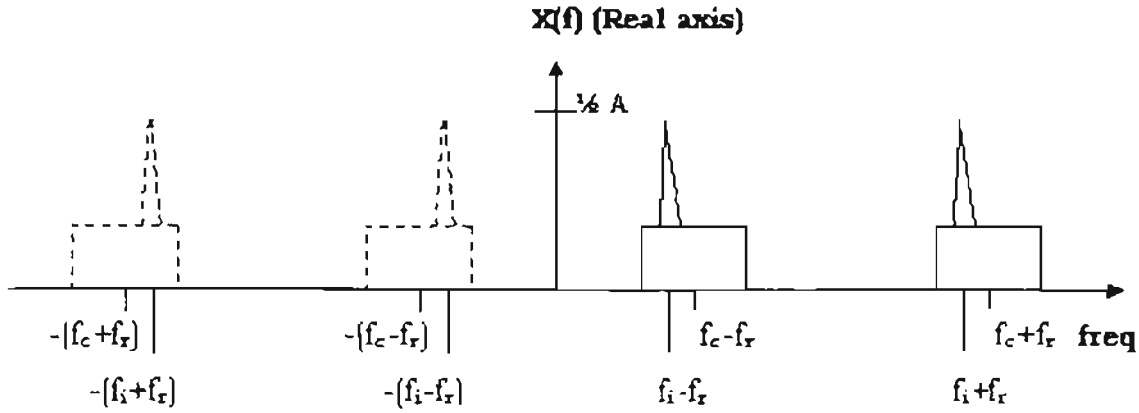


Figure 4.4 Translation of input signal using Cosine Function.

We take the output of the Splitter circuit and pass it through the fixed prototype filter. This prototype filter selects the narrowband interference and attenuates all the other frequencies. This result is illustrated in figure 4.5.

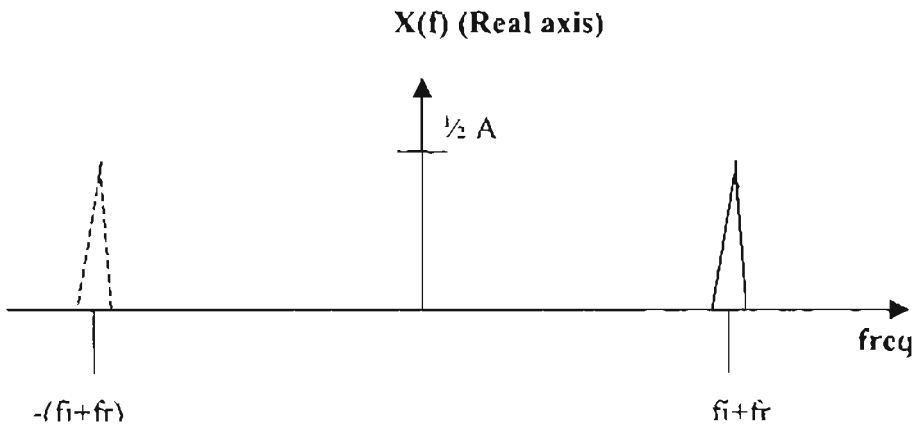


Figure 4.5 Band-pass Signal (I) in In-Phase Channel.

Now we turn our attention to the Quadrature channel where a similar process occurs but this time instead of a cosine signal, we multiply the input signal by the sine. Since the multiplication is with sine function, which constitutes imaginary terms, we represent all the waveforms on the imaginary axis. Due to the negative sign present in the sine term ($-j\, e^{j2\pi f_0 t}$), the image generated at $+f_0$ is always in the opposite direction of the image produced at $-f_0$. This is shown in figure 4.6, where images are flipped around the axis.

The output from the Quadrature channel is fed into the prototype filter in similar fashion as in the In-phase channel. The result of the filtered signal is shown in figure 4.7.

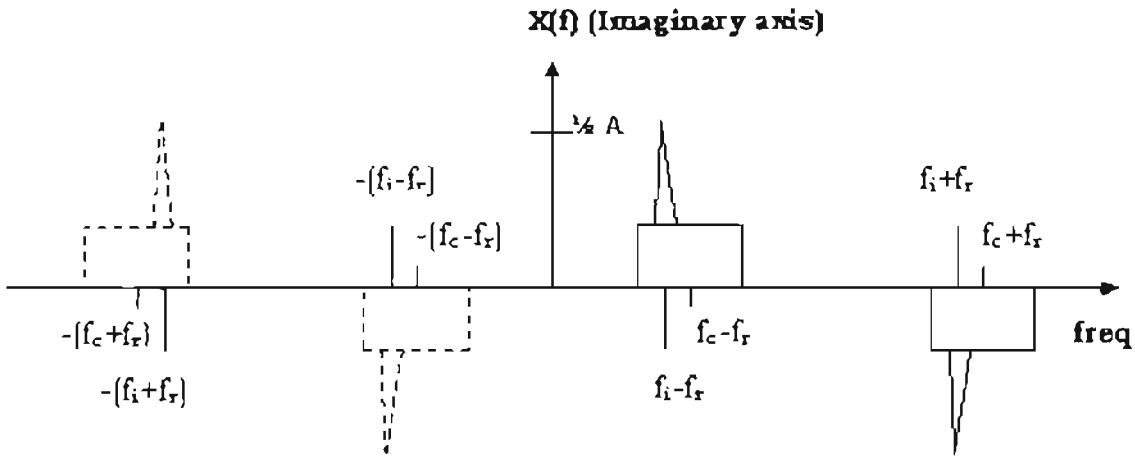


Figure 4.6 Translation of input signal using Sine Function.

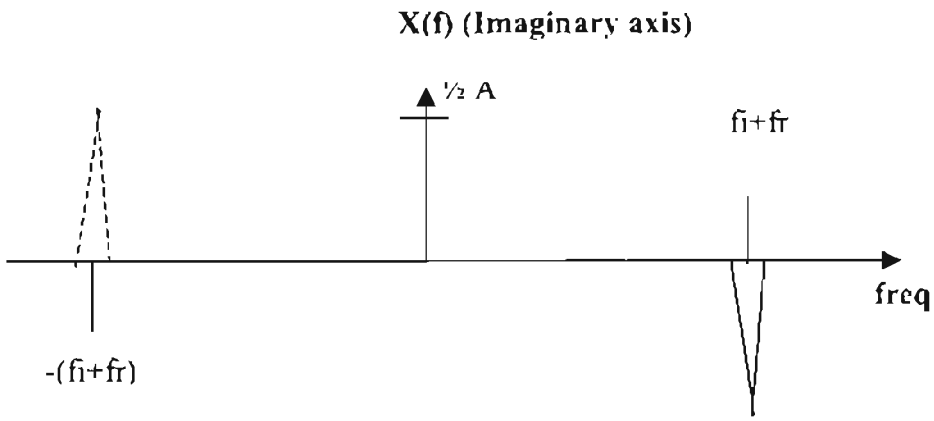


Figure 4.7 Band-pass Signal (Q) in Quadrature Channel.

4.2.2 Signal Response after Combiner Circuit

Now the In-phase (I) and Quadrature (Q) signals go into the combiner circuit. The In-phase signal in figure 4.5, is produced after the prototype filter is heterodyned again in the combiner circuit to bring the translated signal back to the base-band. Similar to the first heterodyne process, the images of the heterodyne signal are produced at plus and minus the heterodyne frequency. Figure 4.8 illustrates the output of the second heterodyne process in the In-phase channel.

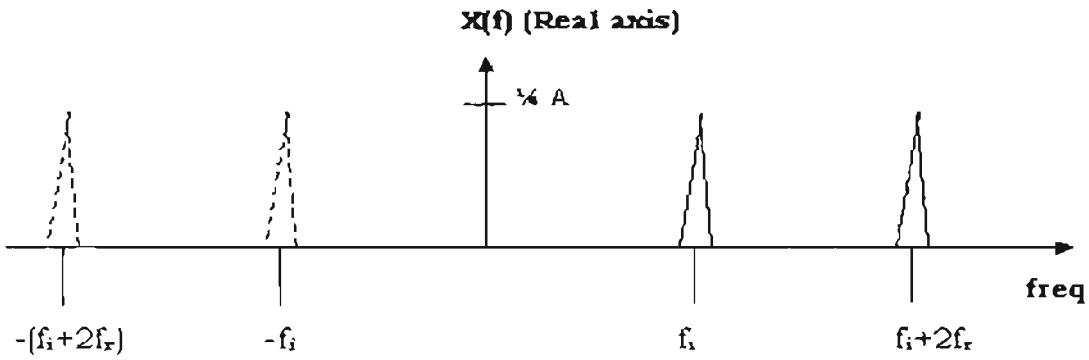


Figure 4.8 In-phase Signal (I) to the Combiner Circuit.

The Quadrature channel is a little more complicated then the In-phase channel since we are multiplying an imaginary term, at the output of the prototype filter with another imaginary term (sine signal), which results in j^2 terms. As a consequence of this multiplication, the j^2 term gives us –1. This has two effects: 1) Due to the multiplication by j term, all the imaginary terms are rotated by 90^0 thus resulting in all real terms 2) The j^2 (–1) factor flips the images around their axis. Figure 4.9 shows the results when the Quadrature signal coming out of the prototype filter is only multiplied by j. Note that the

imaginary terms have a phase shift of 90^0 and thus is plotted on a real axis rather then an imaginary one [17].

Now we multiply the other terms of sine function. Since there is a negative sign in the sine function, terms would not be in the same direction. This is shown in figure 4.10.

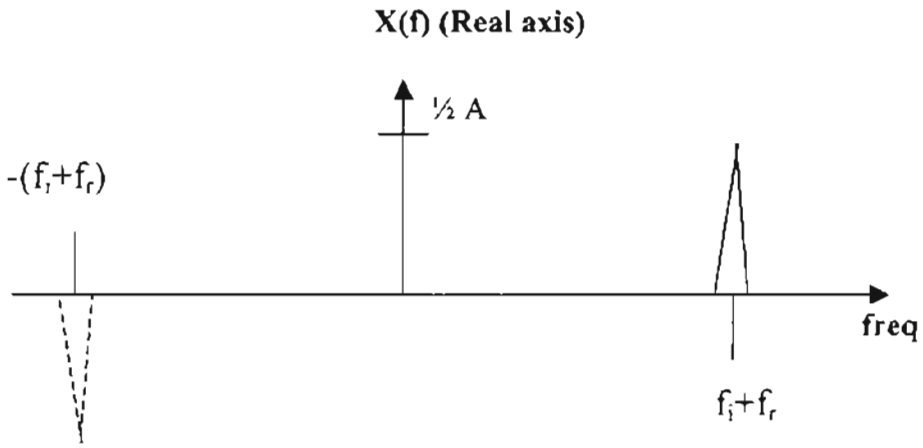


Figure 4.9 Multiplying Q signal by j.

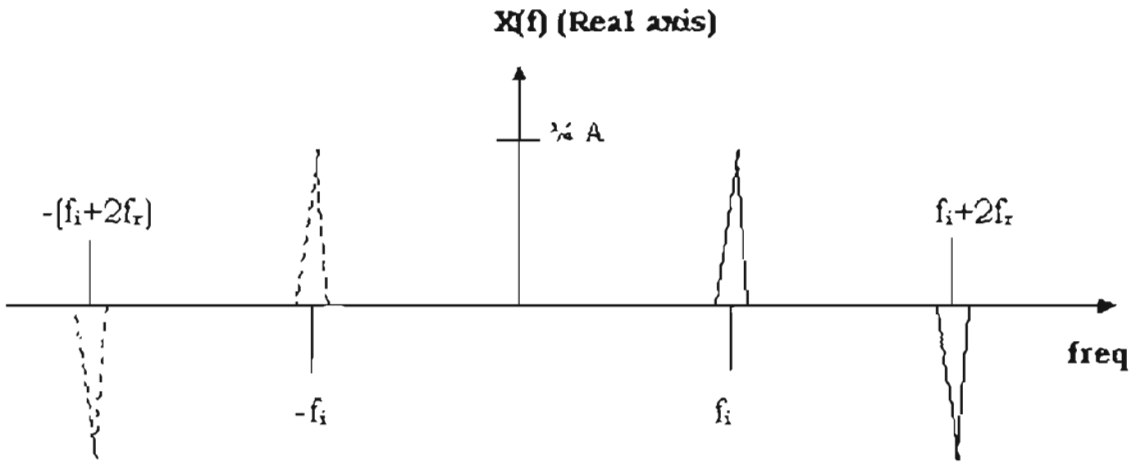


Figure 4.10 Quadrature Signal (Q) to the Combiner Circuit.

Now we have the two outputs from the combiner circuit, which are then added together to cancel all the undesired images. Therefore, we add figures 4.8 and 4.10 to yield the output of the tunable heterodyne band-pass filter. Figure 4.11 shows the final result, which is the narrowband interference frequency at f_i.

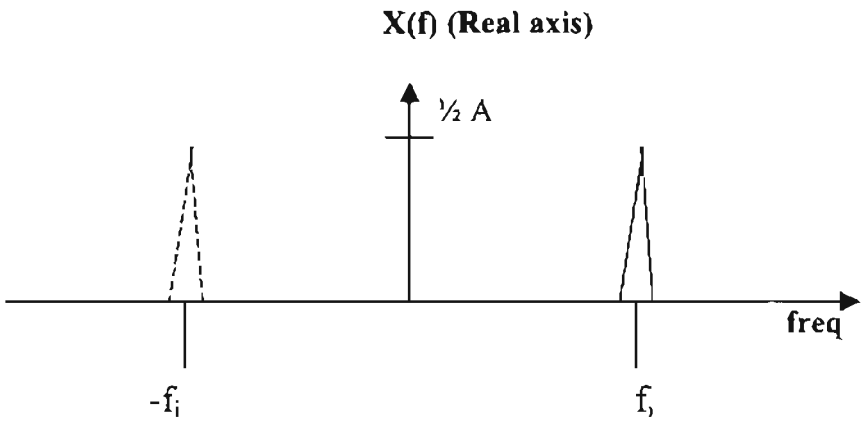


Figure 4.11 Final output from Tunable Heterodyne Band-pass filter.

Chapter 5

Field Programmable Gate Array (FPGA)

5.1 Introduction

The Field Programmable Gate Array (FPGA) is a uniform structured VLSI chip that can be configured and reconfigured for different designs. It allows for a wide variety of applications to have a quick and cheap means of implementing digital designs. FPGA's were chosen to implement the Tunable Heterodyne filter for reasons of rapid prototyping and the ability to configure and reconfigure for different prototyping filters.

There are 4 types of FPGAs on the market today: the symmetrical array, the row-based FPGA, the hierarchical PLD, and the sea-of-gates. All these devices allow for designs to be configured and tested without the long development cycle associated with many custom VLSI designs of this nature and complexity. A relatively quick download process replaces all test pattern generation, mask making, wafer fabrication, packaging and testing associated with design implementation. Given the quick turnaround time and the dynamic capability that comes from using FPGAs, it is much easier to verify the correctness of a design and also much less expensive to fix the error.

However, the greater size of FPGA logic also means that its logic function is smaller when compared to a gate array of similar size, which is composed of the denser mask, programmed logic. Current FPGAs are about ten times less dense than the equivalent mask programmed devices, so FPGAs are considerably more expensive when used to implement a specific logic function. This greater size for equivalent logic function also leads to its interconnect being longer and slower - about twice as slow as a mask programmed gate array.

Field Programmable Gate Array (FPGA) is used in many domains where quick and relatively efficient results are required. FPGAs are applied in prototyping and emulation, where it acts as a bridge to aid software development in parallel to hardware. In certain circuits, designers use FPGA as glue logic to fill a communication gap between two or more systems. This technique is especially attractive because it gives extremely quick results, as compared to developing an ASIC or custom design a chip. Another discipline where emergence of FPGA is becoming apparent is Custom Computing, which has a number of reconfigurable chips on the same platform along with a memory and controller unit. These boards are plugged into the workstation or PCs to accelerate computationally intensive tasks. Due to the flexibility of reconfiguring the device and the comparatively short time required for configuration, FPGAs are the best choice for custom computers [21][24]. This is also the reason FPGAs are the preferred choice for the Digital Signal Processing applications.

To implement the hardware for the tunable heterodyne band-pass filter, we chose the Xilinx Virtex FPGA family chip. There are a number of features in Virtex FPGAs which can be used in parallel to develop an efficient hardware implementation. In the

next section, a brief overview of the architectural structure of the Xilinx Virtex FPGA is introduced to acquaint the readers with the terminology used in chapter 6.

5.2 Xilinx FPGA Architecture

FPGAs are a form of programmable logic that, in general, have a higher gate-count and allow for much greater flexibility than most other programmable logic devices. Therefore, FPGAs range from coarse grain (Larger logic units) to fine grain (Smaller logic units). Most FPGAs on the market today are SRAM (Static Random Access Memory) based, meaning that they can (*and must*) be reconfigured each time they power-up. To make an FPGA stand alone, we need to store the bit stream configuration file in an EPROM that downloads and configures the FPGA every time the power turns on.

Essentially, the XILINX FPGA chip is considered as coarse architecture and contains three main components: The Configurable Logic Block (CLB), the programmable interconnect and finally the Input/Output block (IOB). The Virtex chip, which is used to implement our structure, has an on chip RAM (Read Access Memory) component called Block RAM. The CLBs are connected through vast and versatile programmable interconnect routing resources that can support highly complex patterns and are abundantly surrounded by Input/Output Blocks (IOBs) which connect the FPGA to the outside world. In most of the Xilinx FPGA families, CLBs are the unit structure. But in the case of Virtex family, Slices are the unit structures. CLBs are wired to each other and to other wire segments using programmable interconnects which are essentially configurable switches or SRAM. These switches open or close depending on the voltage

applied to their gate and are laid out in matrix fashion in order to provide the much-needed flexibility in routing [21][24].

5.2.1 Configurable Logic Block (CLB) or Slices

To implement our designs, we made use of Vitex 800 family FPGAs. The basic virtex FPGA logic building block is called a Slice. There are two Slices in one CLB and each slice consists of two four input Function Generators also referred to as Look-up Tables (LUT), two flip-flops and control and carry logic. The presence of the Functions Generators enables the CLB to implement a wide variety of combinational logic functions depending on the number of inputs. These function generators can also be used as Read-Only Memory (*ROM*) or Random-Access Memory (*RAM*). The carry logic produces a fast propagation carry chain dedicated to arithmetic logic, which yields long delays due to long chains of carry and borrow signals. The carry chain for the dedicated arithmetic logic is independent of normal routing resources. This greatly increases the performance of adders, subtractors, multipliers and counters in the design. The multiplexers or the control logic in the CLBs aid the routing of the design to the appropriate cells in the logic block while the flip-flops allows for efficient implementation of pipelined designs and shift registers. Figure 5.1 below shows an example of a CLB [22].

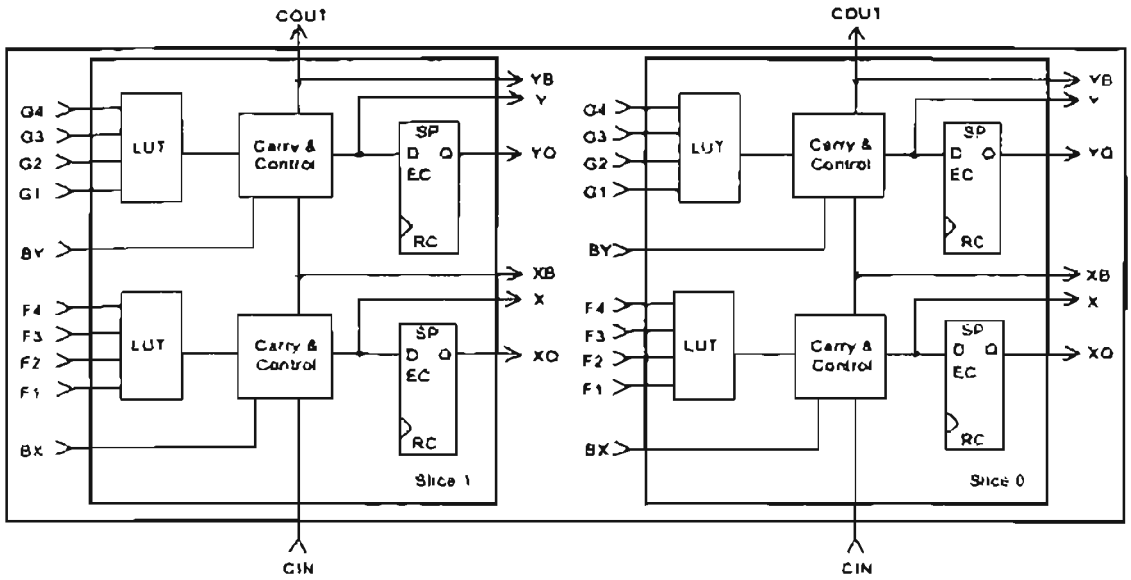


Figure 5.1 Virtex Family Configurable Logic Block (CLB).

5.2.2 Programmable Interconnect

All internal connections of an FPGA are composed of metal segments with programmable switching points and matrices to implement a desired routing. There are several different programmable interconnect configurations depending on the type of device being used. There are usually up to five (5) different interconnect types available and they are distinguished by the length of their metal segments. The single-length lines connect the switching matrices that are located in every row and column of CLBs. The double-length lines consist of a grid of metal segments, each twice as long as the single-length lines: they run past two CLBs before entering a switch matrix. Double-length lines are grouped in pairs with the switch matrices staggered, so that each line goes through a switch matrix at every other row or column of CLBs. The horizontal and vertical single-length and double-length lines intersect at a Programmable Switch Matrix (PSM). Shown in Figure 5.2 below is an example of a PSM [22].

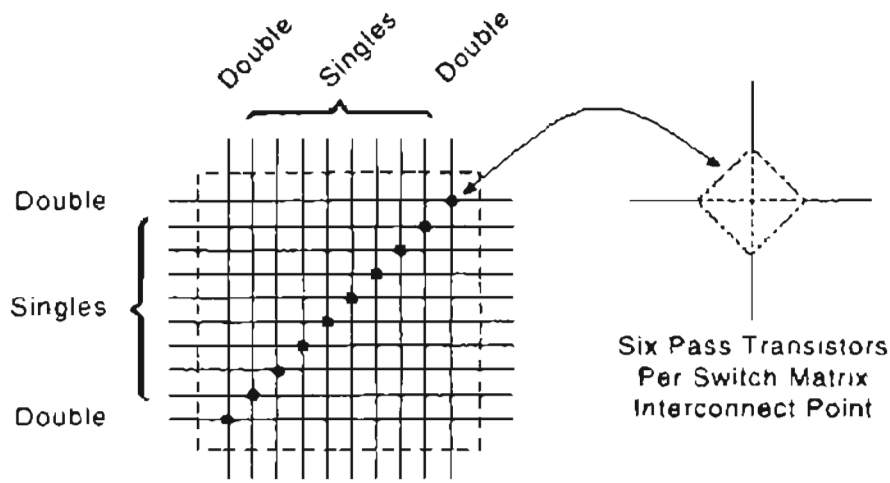


Figure 5.2 Programmable switch matrix.

5.2.3 I/O Blocks

IOBs provide the interface between the external package pins and the internal logic of the FPGA. Each IOB controls one package pin and can be configured for Input, Output or bi-directional signals. Figure 5.3 shows the simplified block diagram of an IOB [22].

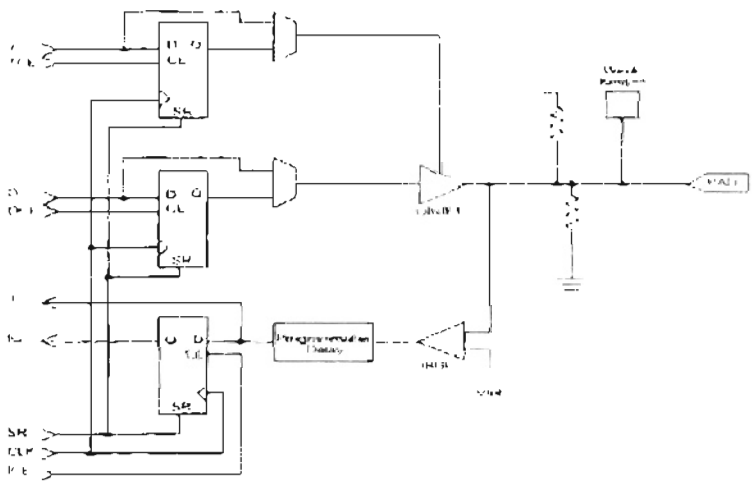


Figure 5.3 Simplified Block diagram of a Virtex Series IOB.

5.3 FPGA Design Flow

A generalized FPGA development flow is shown in figure 5.4. This flow is similar to an Application Specific Integrated Circuit (ASIC) design flow with a standard cell library. The design entry for the FPGAs is in the form of either behavioral code using Hardware Description Languages (like VHDL and Verilog) or Schematic capture. Synthesis tools are used to fabricate and optimize the design described with these HDL languages. Using this synthesized circuit of the design, it is then targeted toward a specific technology library, which uses the components available in the library to map the components required in the design with the ones that are available in the library. Once all the components are mapped to the circuit file generated by the synthesis tool, these components are placed and routed. Implementation of this design in actual hardware is done by downloading the configuration file into the FPGA using downloading tools.

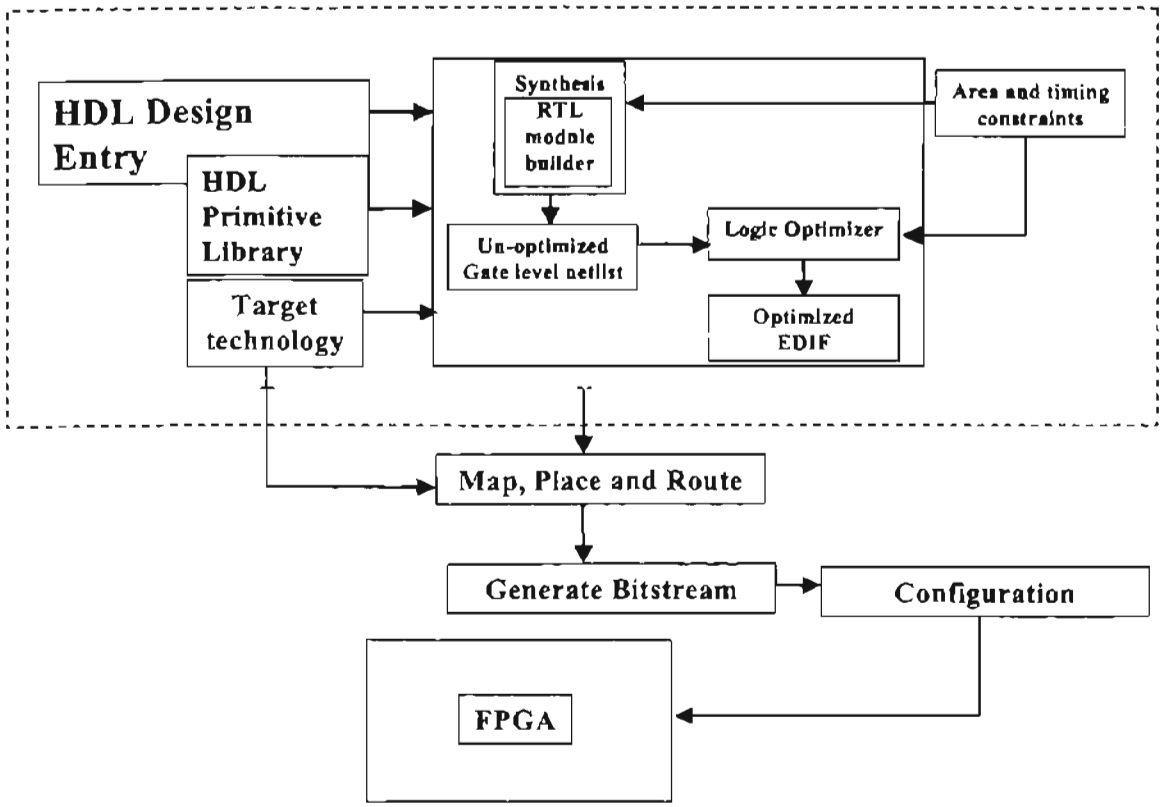


Figure 5.4 FPGA Design Flow Block Diagram

5.3.1 Synthesis

Synthesis is the process of converting a behavioral description of a design using hardware description language (like VHDL, Verilog etc.) or other means of design definition into gate level netlist [23]. Netlist is a file that is generated by the synthesis tool to implement the functionality of the design described by the designer. The contents of these files include instantiation of predefined circuit modules linked together through wire connections. These predefined elements are specific to a certain vendor to which the design is linked to a targeted standard cell library

5.3.2 Synthesis Flow

A block diagram from [23] shows the process of a synthesizer program to optimize the circuit description in the best possible way.

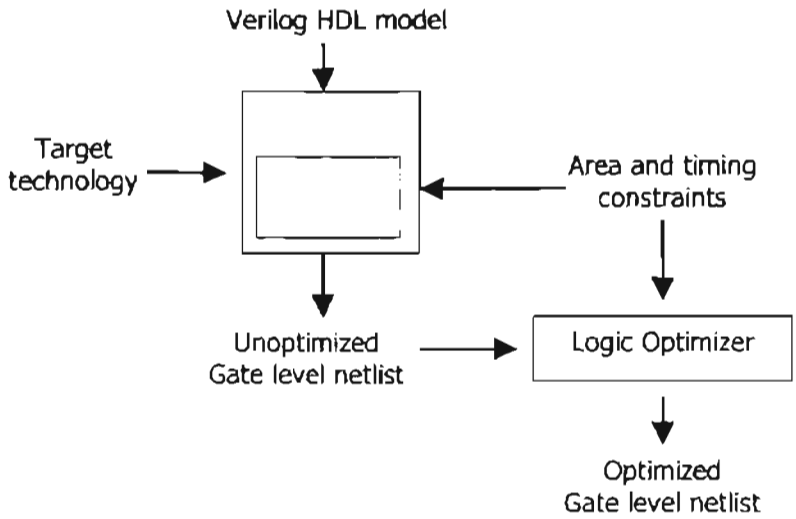


Figure 5.5 Synthesizer optimizing flow diagram.

Figure 5.5 shows the flow of synthesis process, where the Netlist is optimized to meet the constraints (timing or area) the designer has specified. The Netlist created depends heavily on the coding style adapted in the hardware description languages. To fully utilize the synthesizer's optimizing abilities, it is of utmost importance that the HDL designer write the code more effectively [23].

5.3.3 What is Synthesized?

The synthesizer tool is sensitive to the way the code is written and the way the model is described for the circuit. The manner in which the instructions are placed in the code makes a profound difference in how the optimizer will generate the Netlist [23]. Design can be implemented according to the requirements defined in the design specifications. Sometimes designs are opted to be more area efficient and sometimes designs are opted to be more speed efficient. Designers have to make the decision to model their design in the light of this tradeoff to yield the most efficient design. Introducing techniques to intensify resource sharing and parallelism in the HDL design code would give two different optimizing criteria to the synthesizer. Resource sharing is usually used to make the design more area efficient by reducing the module duplication as much as possible, commonly at the expense of increased delay in the system. On the other hand, by sacrificing area, designers can try to decrease the delay factor by introducing more parallelism in their designs.

5.3.4 Implementation

The design developed using the hardware description language and synthesized using a synthesizing tool, now can be implemented in hardware using the EDIF (Electronic Design Interchange Format) file produced by the synthesizer. The design is mapped to the certain technology library and the components are placed and routed to complete all the connections in the design. The Alliance tool provided by Xilinx does the place and route and creates an implementation file with extension <filename>.bit. This file is downloaded into the FPGA using various methods, depending on the application.

Chapter 6

Hardware Implementation of the Basic Block

6.1 Hardware Implementation of the Splitter Circuit

The splitter circuit is shown as a part of Tunable Heterodyne Band-pass filter Basic Block, in the rectangular dotted box of figure 6.1. It consists of two components: Local Oscillator (LO) circuit or the Sine/Cosine generator and a frequency Mixer or a Four Quadrant Multiplier.

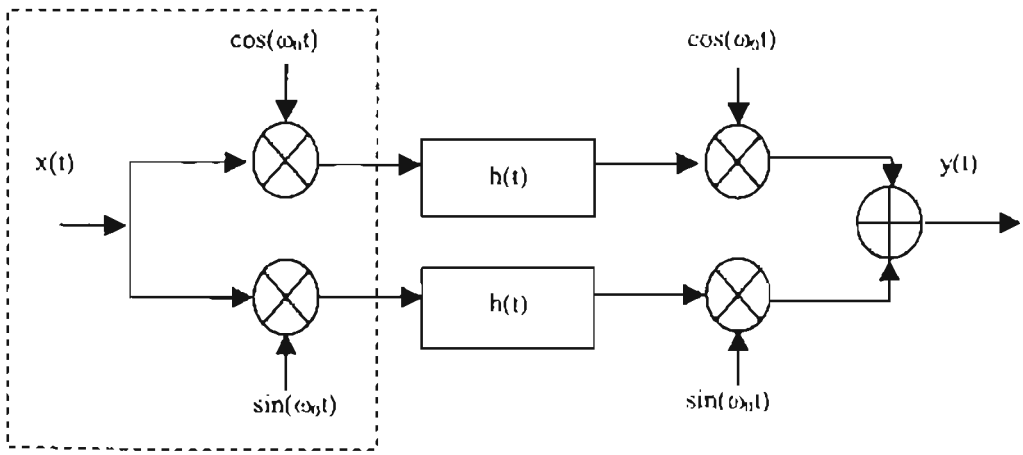


Figure 6.1 Tunable Heterodyne Filter Basic Block.

The Splitter circuit is commonly found in Analog Communication structures, where incoming signals are translated to an Intermediate Frequency by modulating the signal with a local oscillator (LO). In heterodyne receivers, the incoming signal is tuned by adjusting a local oscillator to translate the signal to a band-pass amplifier or to DC where the desired signal is extracted from the original incoming signal. The local oscillator's frequency is set so that the difference between its frequency and the desired signal frequency results in the Intermediate Frequency (IF) of the band-pass amplifier or cancel each other out to translate the signal down to DC. In the case of heterodyne receivers, the mixers used to translate the incoming signal are Analog non-linear Multipliers.

6.1.1 Sine/Cosine Function Generator Circuit Design

The Sine/Cosine generator basically produce $\sin(2\pi f_0)$ and $\cos(2\pi f_0)$ simultaneously. There are various methods that can be used to implement this type of structure, one such example is a CORDIC algorithm, which calculates the trigonometric function through the use of rotating phasors. For the implementation of the Tunable Heterodyne Band-pass Filter structure in this project, a method utilizing the Look-up tables (LUT) was used. In this implementation, the pre-calculated values of the sine and cosine waves are stored in a LUT and then accessed by using an address generator. The expression to calculate the theta for the Sine/Cosine look-up table is [25]:

$$\theta = \frac{2\pi * n}{2^{input\_width}} \tag{6.1}$$

Where the input width is given by:

$$n = \#\ \text{of samples per period}$$

$$\text{input\_width} = \log_2(n) \tag{6.2}$$

The theta (θ) calculated, represents all the values in radians for n number of samples, needed to represent a sine or cosine wave in one period. This angle is used to calculate the Sine/Cosine values for each sample. These magnitude values of the Sine/Cosine samples are then converted into 8-bit, twos complement numbers, which are stored in the Look-up tables.

Figure 6.2 shows the structure for the sine and cosine generator, where the values of a full wave are stored in four Look-up Tables. Although sine and cosine waves are out of phase by 90^0, both share the same magnitude values shifted by a quarter of the wave. A straightforward mechanism is used to obtain all possible values needed to produce full periods of both Sine/Cosine waves simultaneously. A six bit counter serves two purposes, one to generate the address for the look-up table and the other to generate control bits for the multiplexers. The two control bits switch the output of the multiplexers in complete sync with the Look-up table address generation.

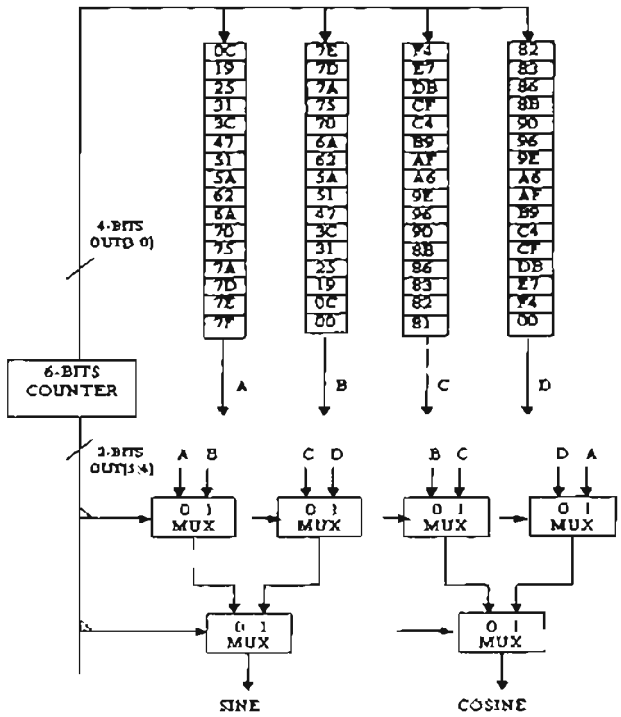


Figure 6.2 Sine/Cosine Generator Hardware Structure.

The two outputs of the Sine/Cosine generator are 90^0 out of phase. To complete one period, we have to sample 2^6 values, therefore the clock frequency of the counter (f_{ROM}) has a relationship with the required clock (f_{REQ}) which is given by the following expression:

$$f_{ROM} = f_{REQ} * 2^{input\_width} \qquad (6.3)$$

The required frequency (f_{REQ}) is the heterodyning frequency, with which we wish to translate the signal. Thus if the required frequency (f_{REQ}) is 1000Hz, then the counter frequency (f_{ROM}) needs to be 64000Hz.

6.1.2 Signal to Noise Ratio For Sine / Cosine Generation

Since we are trying to generate Sine/Cosine functions in the digital domain, we are faced with the problem of quantization error. This error is due to the limited number of both samples in a period and bits used to represent the magnitude of Sine and Cosine functions. The problem caused due to the limited number of points in a period is not as significant as the effect caused by quantizing the magnitude values. Figure 6.3, 6.4 and 6.5 show the magnitude response of FFT of 64, 512 and 1024 point sine waves with infinite precision magnitude generated using Matlab program, respectively.

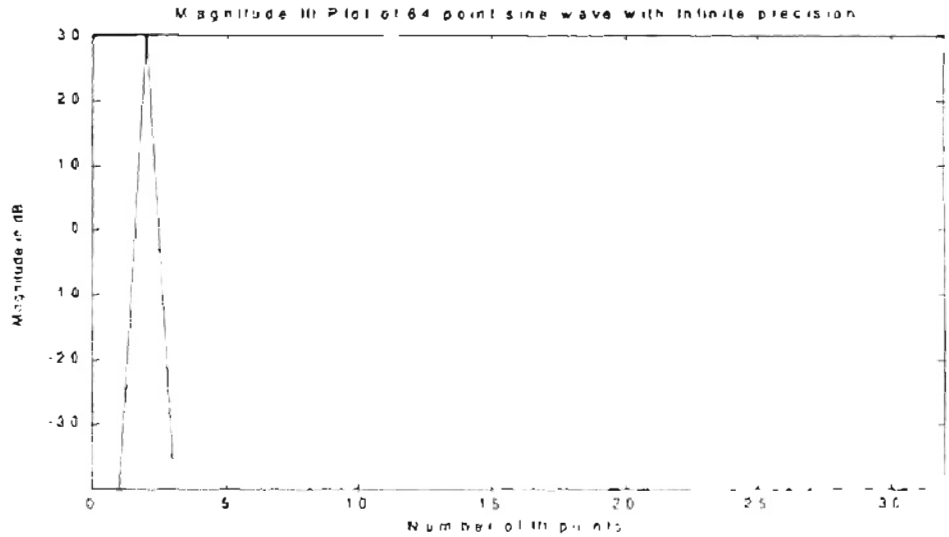


Figure 6.3 Magnitude response of 64-point waveform.

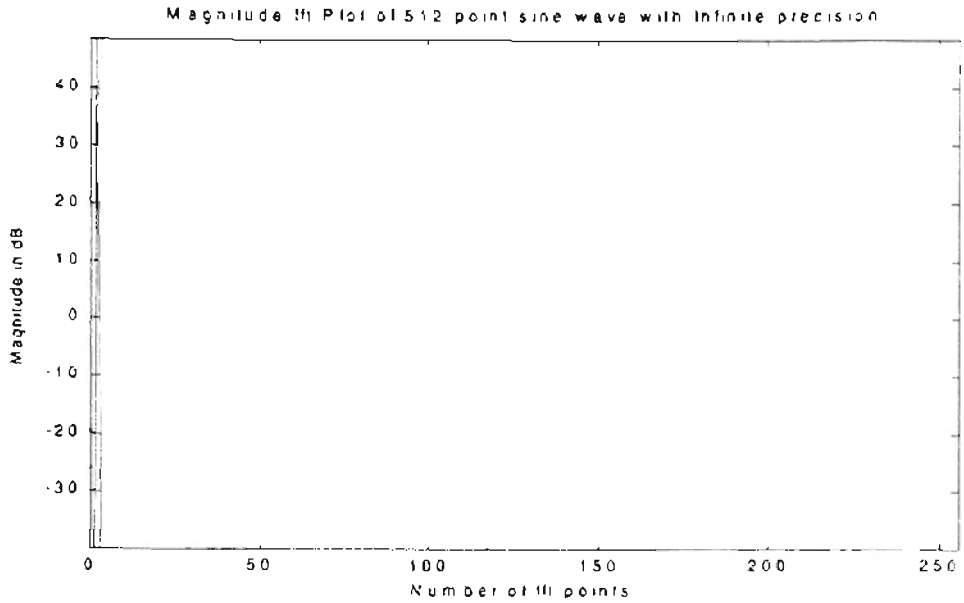


Figure 6.4 Magnitude response of 512-point waveform.

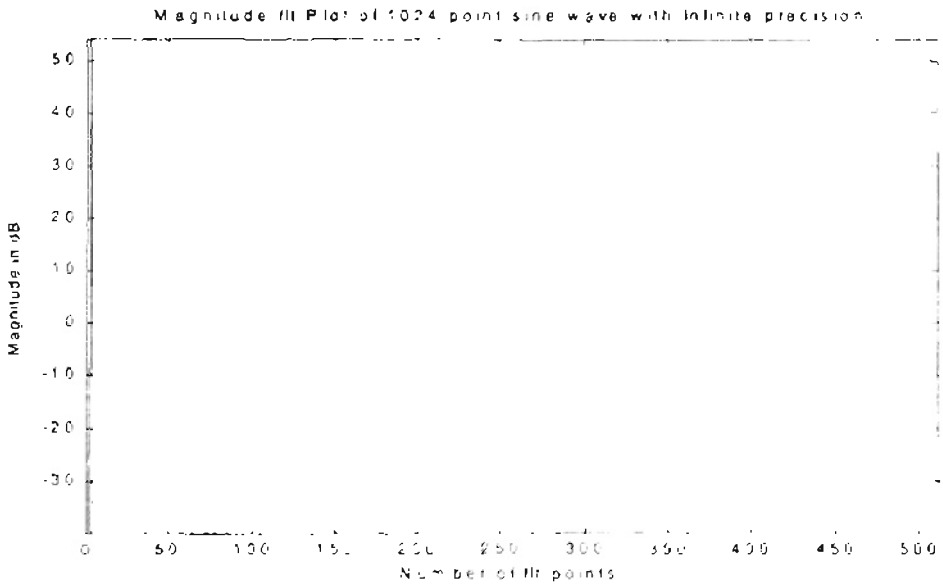


Figure 6.5 Magnitude response of 1024-point waveform.

It is seen from these figures that the noise is almost negligible in all the cases. The signal to noise ratio (SNR) is approximately 308 dB and is almost the same for all three

plots shown above. Therefore, we can see the number of points in a period has very little effect on the SNR.

Now we will analyze the effects of quantization of output bits, which defines the limit of the magnitude approximation, made against the real signal. We implement a 64 point sine wave, since it does not contribute significantly to noise generation and also it is relatively hardware efficient. Figures 6.6, 6.7 and 6.8 illustrate the effect of noise generation due to quantization of output to 6, 8, 16-bits, respectively.

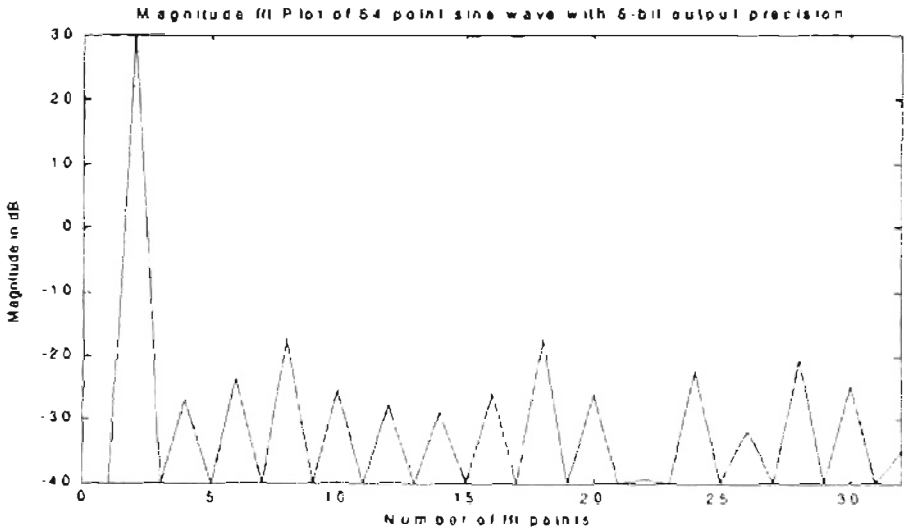


Figure 6.6 Magnitude response of 64 point, 6-bits output sinusoid.

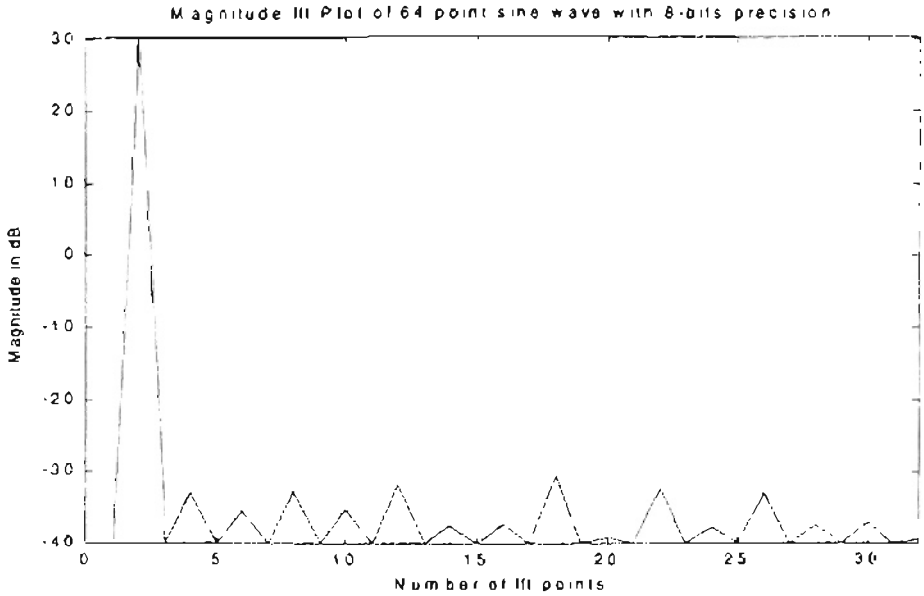


Figure 6.7 Magnitude response of 64 point, 8-bits output sinusoid.

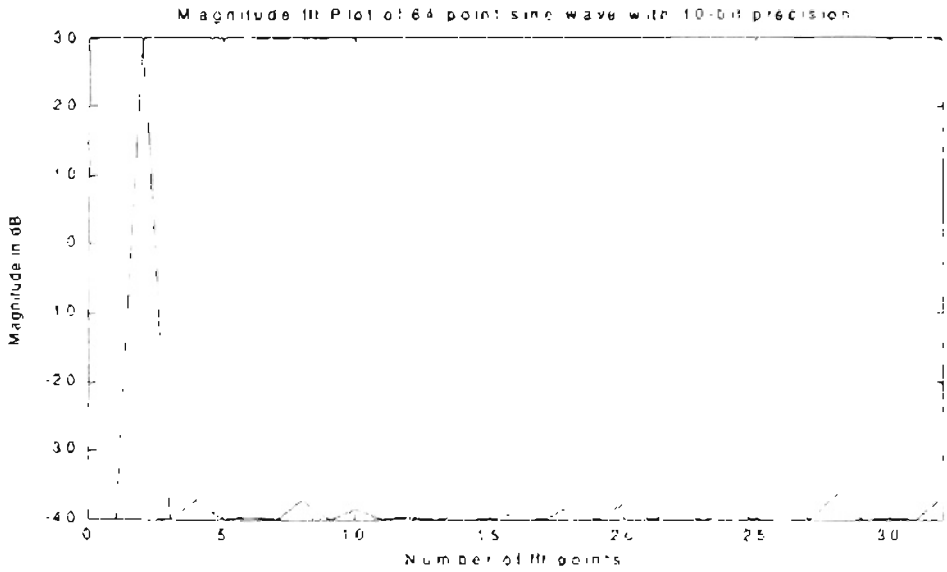


Figure 6.8 Magnitude response of 64 point, 10-bits output sinusoid.

Number of Output Bits	SNR (dB)
6-bits	42 dB
8-bits	58 dB
10-bits	70 dB

Table 6.1 Signal to Noise Ratio for various length outputs.

Table 6.1 shows the SNR for 64-point sinusoid with various word length output. It is seen that the quantization effect has introduced a significant amount of noise in the system. Although 6-bit output has a SNR of 42dB, which is considered sufficient to represent a signal but to avoid the effect of additive noise from other sources, which could deteriorate the output waveform, an 8-bit output word length is chosen in our system.

6.1.3 Synthesized Circuit of Sine/Cosine Generator

Syplicity Tools from Synplify were used to synthesize the sine/cosine generator's VHDL source code. The quantized magnitude values of the Sine and Cosine waveform were stored in a ROM table. This VHDL code would synthesize differently depending on the target technology. If the design is targeted towards ASIC or MOSIS technology, then this ROM table would infer a ROM memory device. In the case of Xilinx Virtex FPGA technology, these values are stored in a Look-up table (LUT). Since each LUT in the Virtex Chip is capable of storing sixteen one-bit values, the estimated number of LUTs used to store these values is about 24. Thus the four tables of values would take at least

16 Slices or 8 CLBs just to produce the ROM tables. These ROM tables could be stored in the Blocked Memory of the Virtex FPGA, but this was avoided since it would have required instantiating a specific component from the Virtex library, thus resulting in a tool dependent code. Another scheme to produce a Sine/Cosine table would require two ROM tables and two adders rather than four ROM tables. This case was not synthesized in real hardware and may or may not yield a more optimized circuit. The counter used in the design is merely a binary incrementer, which also serves as control bits for multiplexers. The VHDL files for the Sine/Cosine generator are: sin_cos.vhd, counter16.vhd. Figure 6.9 shows the synthesized circuit of the 6-bit counter produced by the Synplicity Synthesis tools.

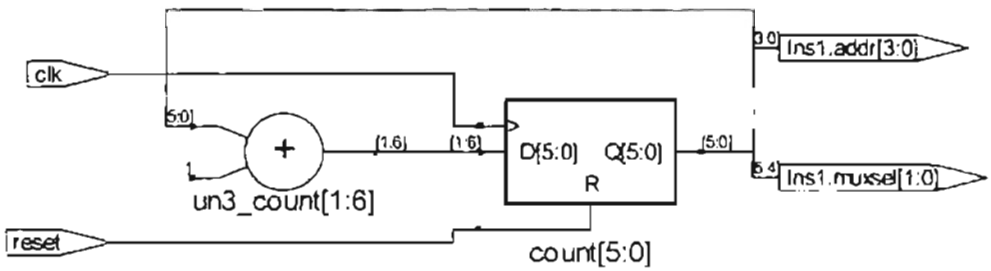


Figure 6.9 Synthesized circuit for 6-bit counter.

Figure 6.10 shows the full structure of Sine/Cosine generator circuit, synthesized by Synplicity Synthesis tools. Notice that "counter16" is used as a component in this figure.

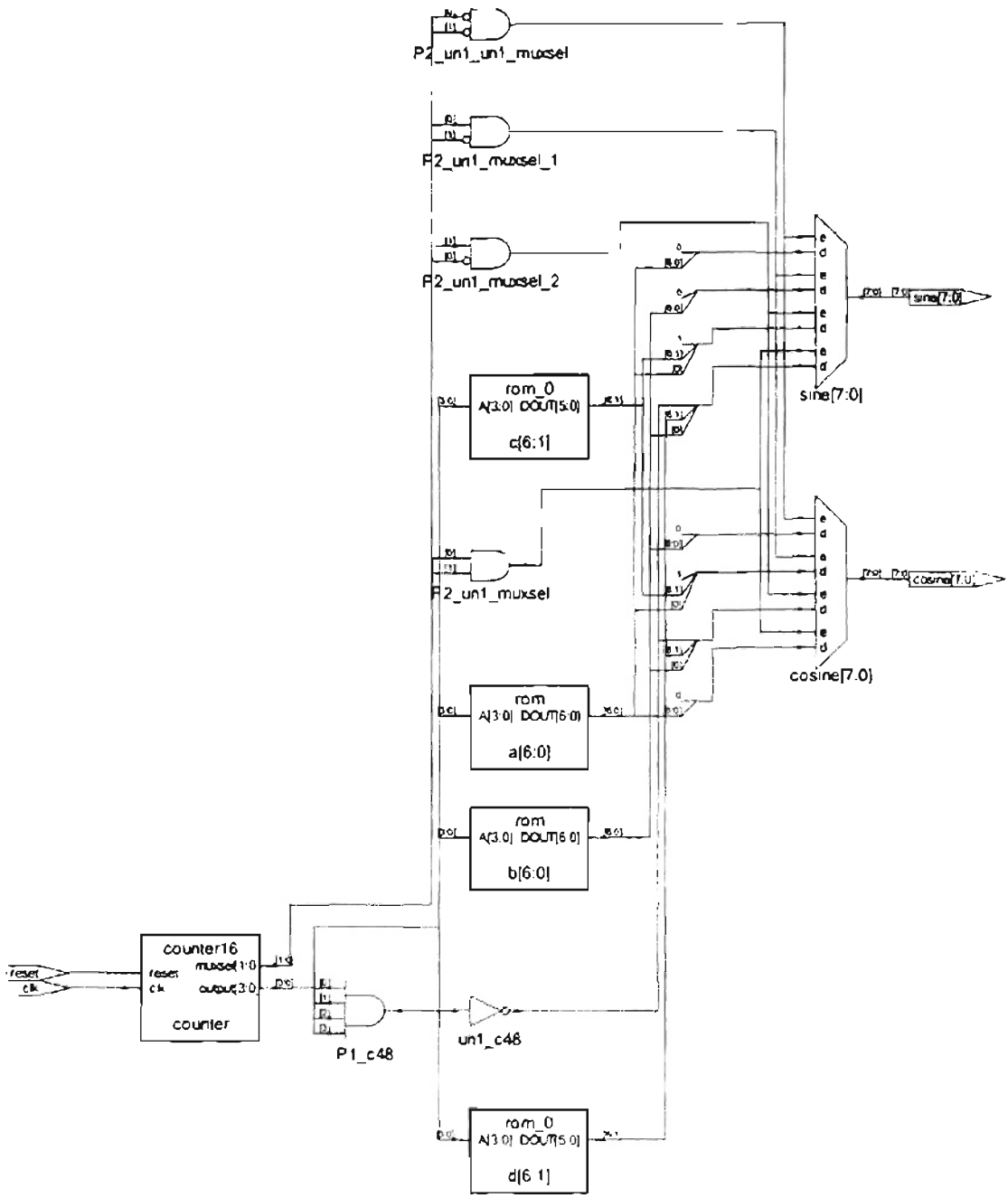


Figure 6.10 Synthesized circuit for Sine/Cosine Generator.

6.1.4 Quadrature Mixers Circuit Design

The next part of the Splitter circuit is the Quadrature mixers, which multiply the output of the Sine/Cosine generator with the incoming signal. These circuits can vary depending on the type of technologies the design is linked towards. For example, for VLSI technologies it is best to implement a Booth Multipliers using the Booth algorithm. For ASIC and FPGA technologies, it is best to design a sequential multiplier or combinatorial multipliers. For this project a combinatorial 2's complement 8x8 multipliers were implemented.

Signed magnitude is a number system in which negative numbers are represented as normal binary numbers with a '1' used as an extra bit in the most significant bit position. This signifies that the number is negative. Since the input to our structure is assumed to be a 2's complement 8-bit samples, therefore we need to implement a two's complement multiplier. To implement a 2's compliment multiplier, three extra adders along with some control logic was inserted to convert a simple add and shift signed magnitude multiplier circuit into a 2's complement multiplier. The control logic for the three extra adders was produced from the most significant bits of the two inputs. Thus a one in the most significant bit would reflect that the input number is a negative number. If the incoming number is negative, then a 2's compliment of that input is taken and fed into the signed magnitude multiplier. Again, depending on the most significant bit of the two inputs, the output is adjusted to be a negative or positive value. Figure 6.11 shows the multiplier circuit.

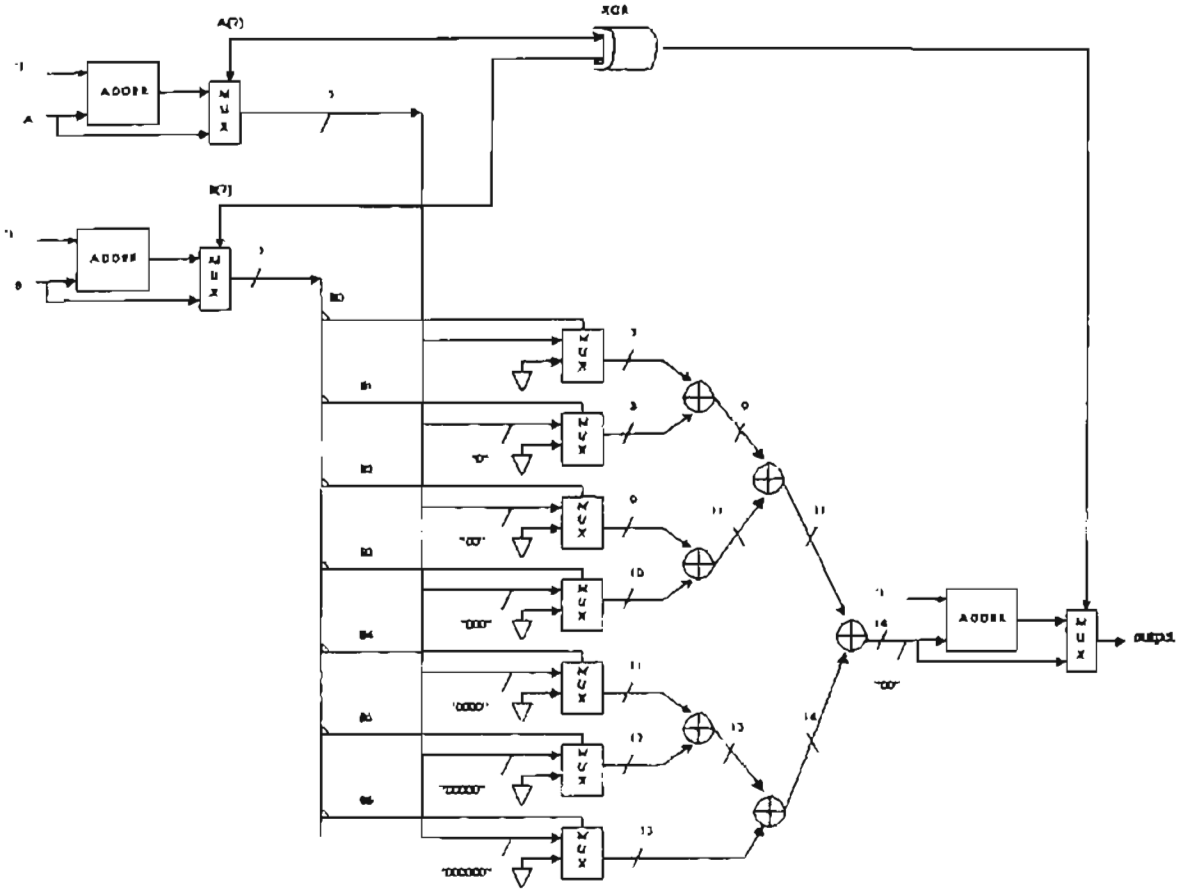


Figure 6.11 Two's Complement 8x8 Multiplier Circuit.

6.1.5 Synthesized Circuit of the Quadrature Mixer

To implement 8-bit x 8-bit signed magnitude multiplier, we require at least 6 adders to add all the partial products along with the carry generated from each stage. In addition to these adders we have three extra adders to implement this structure for 2's complement multiplication. The VHDL source codes to implement this multiplier are: two_comp.vhd, mult8x8.vhd, two_comp_out.vhd.

The input is a two's complement number which is converted into appropriate representation of a signed magnitude number by two_comp file. The VHDL file mult8x8 is used to implement the multiplier function for the signed magnitude values. The output of this structure is adjusted to be a negative or positive number with the two_comp_out VHDL file according to the most significant bit of both the inputs. The synthesized circuits for all three two_comp, two_comp_out , mult8x8 and mult_top circuits are shown in figures 6.12, 6.13 6.14 and 6.15, respectively.

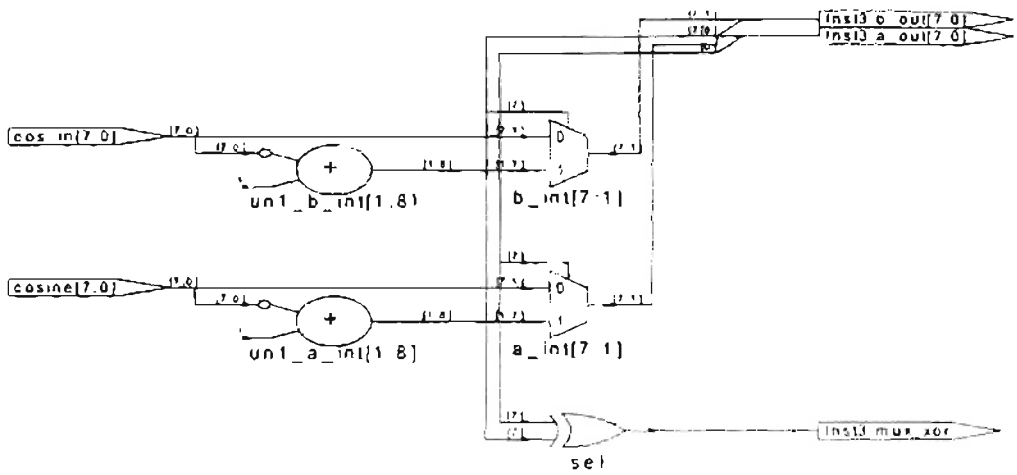


Figure 6.12 Synthesized Circuit to adjust the inputs.

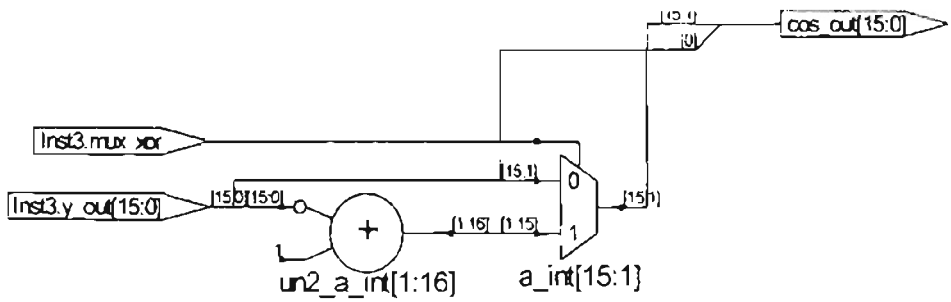


Figure 6.13 Synthesized Circuit to adjust the output.

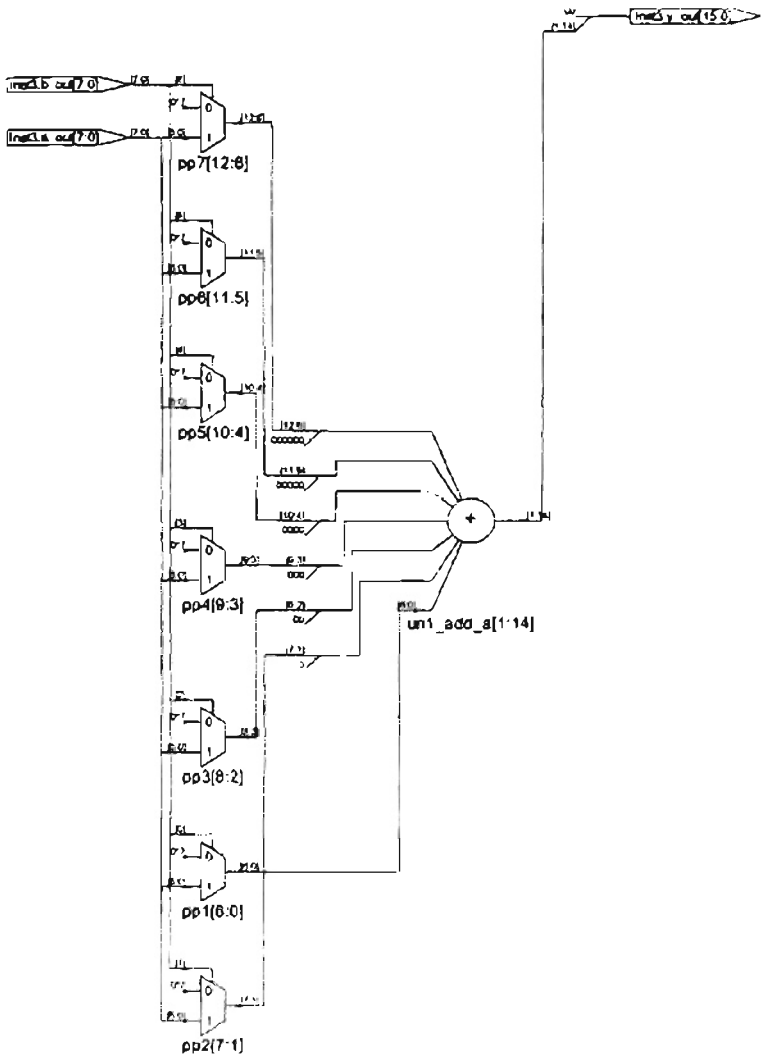


Figure 6.14 Synthesized Circuit Partial Product Summation.

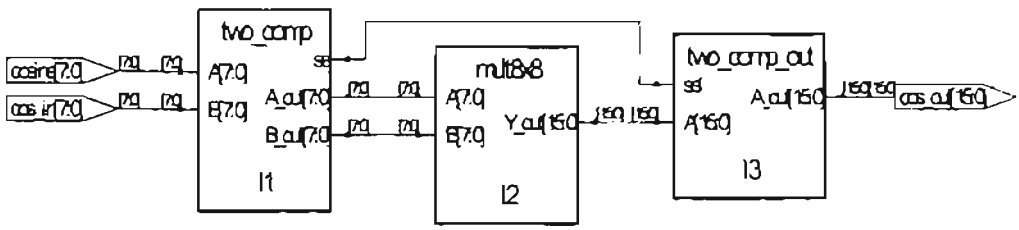


Figure 6.15 Synthesized Circuit for the 8x8 Multiplier.

6.1.6 Complete Splitter Circuit

Two components, Sine/Cosine Generator and Quadrature Mixers, are combined together to form a Splitter circuit. The splitter circuit takes the input signal and splits it into two branches, In-phase and Quadrature. One of the branches multiplies this incoming signal by the sine function generated by the Sine/Cosine generator and the other branch multiplies the incoming signal by the Cosine function. Therefore, the Splitter has basically two input signal buses and two output signal buses. In this case the input signals are joined together. Another input to the Splitter circuit is the heterodyning frequency (f_R), which is provided to the ROM address generator.

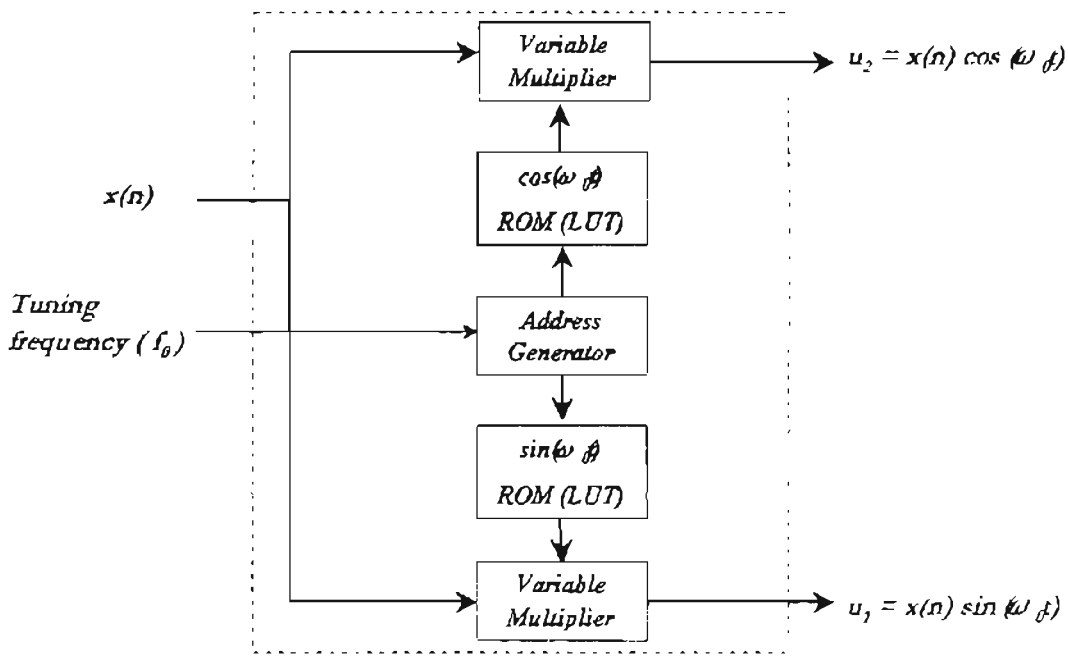


Figure 6.16 Block Diagram of the Splitter Circuit.

6.1.7 Synthesized Circuit of Splitter

The synthesized circuit for the whole splitter is shown in figure 6.17. The VHDL files used to implement this structure are the same as the combinations of all the files listed above: counter16.vhd, sin_cos.vhd, two_comp.vhd, two_comp_out.vhd, mult8x8.vhd, mult_top.vhd, mixer_cir.vhd.

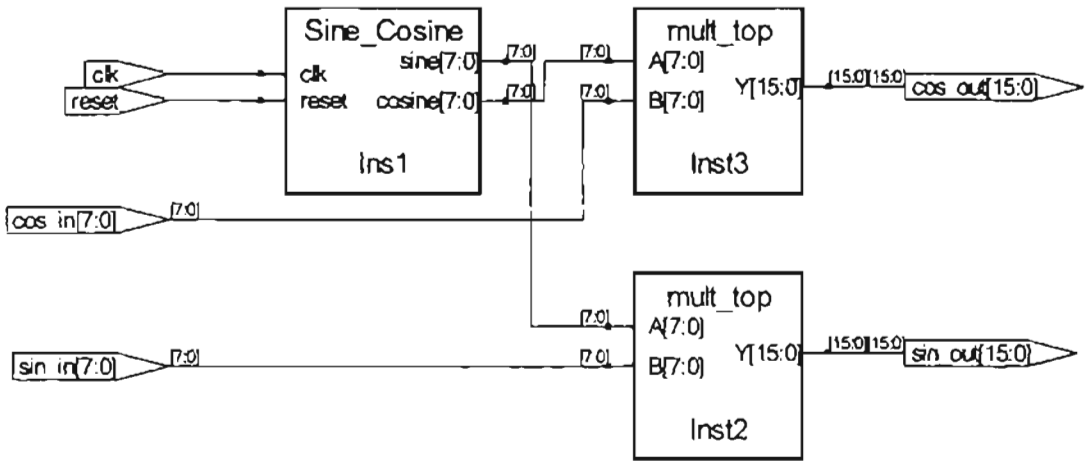


Figure 6.17 Synthesized Circuit for Splitter.

There are two-multiplier circuits instantiated where one of the input of each multiplier is the input signal from outside the circuit and the other is either sine or cosine, which are the outputs of the Sine/Cosine generator. In the case of the Splitter, the two inputs are connected in the main VHDL file.

6.1.8 FPGA Resource Utilization in Splitter Circuit

Table 6.2 shows the utilization of Slices, LUT and Registers for each component in the splitter circuit using a Virtex 800 family FPGA chip. There are two multipliers and one Sine/Cosine generator implemented in the Splitter Circuit.

	Xilinx Implementation		
	Slices	**LUT**	**Reg**
Sine/Cosine	30	53	6
Multiplier	53	100	0
Splitter	139	253	6

Table 6.2 FPGA Resource Utilization in the Splitter Circuit.

6.2 Hardware Implementation of Prototype 2nd Order IIR Filter

For the prototype band-pass filters used in the Heterodyne Structure, we chose a 2nd order IIR Band-pass filter with center frequency fixed at a quarter of the sampling rate. Any order band-pass filters depending on the application can replace these band-pass filters. The general equation describing a second order Band-pass filter is given in equation 6.4.

$$H_{BP}(z) = \frac{1-\alpha}{2} \frac{1-z^{-2}}{1-\beta(1-\alpha)z^{-1}+\alpha z^{-2}} \tag{6.4}$$

where $\beta = \cos(w_0)$ and w_0 = center frequency of the band-pass filter. The 'β' factor determines where the filter is located in the frequency spectrum and the α factor

determines the width of the pass-band. The center frequency for the fixed filter was chosen as a quarter of the sampling frequency, thus $w_0 = f_S / 4$, which, in radian terms, is located at $\pi/2$ and hence resulted in the 'β' value to be zero. The 'α' factor is chosen with two considerations: one is that the pass-band of the band-pass filter needed to be narrow and the other is to keep the coefficients as close to the power of two as possible, thus reducing the amount of hardware used to implement these coefficients. Thus, after substituting the values for 'α' and 'β', equation 6.4 becomes:

$$H_{BP}(z) = 0.0625 \frac{(1 - z^{-2})}{(1 - 0.875z^{-2})} \tag{6.5}$$

From equation 6.5 we get the values of zero coefficients 'b' and feedback coefficients 'a' and plot the magnitude response for the filter using the Matlab program. Figure 6.18 and 6.19 are the magnitude responses of the filter and zplane showing the poles and zeros, respectively.

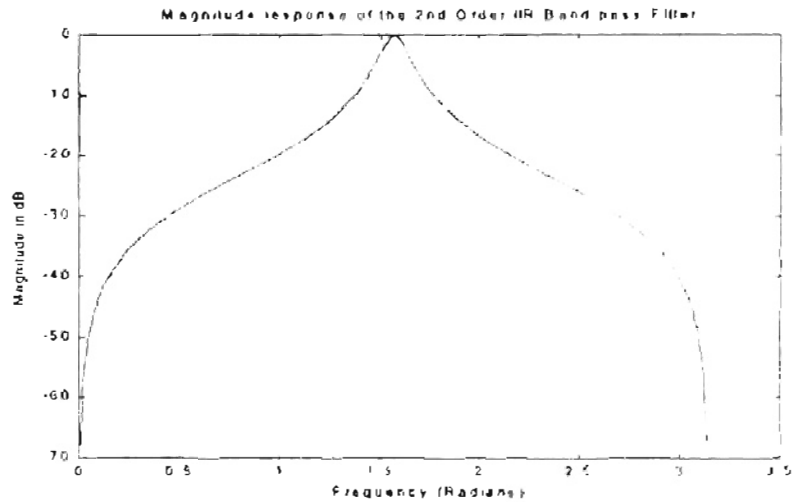


Figure 6.18 Magnitude response of 2nd Order IIR Band-pass Filter.

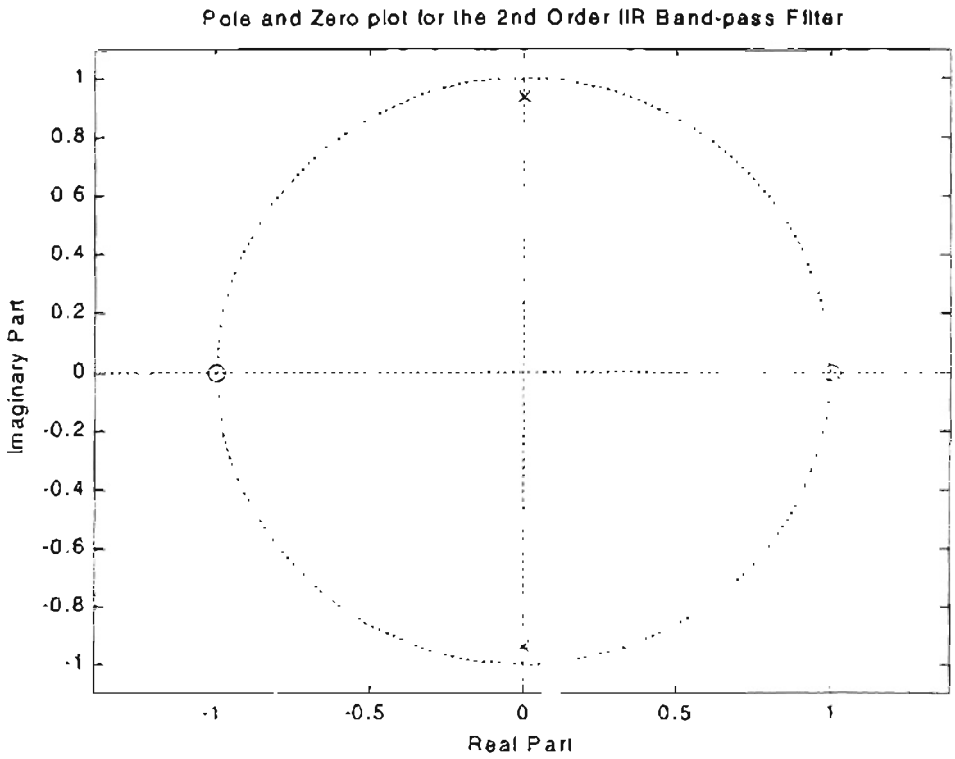


Figure 6.19 Poles and Zero Plot for 2nd Order IIR Band-pass Filter.

Since the coefficient were carefully chosen to be represented with 8-bits, therefore the problem of scaling did not arise in this filter design. The uniqueness about this band-pass filter is that it can be transformed into a band-stop filter by just subtracting the input from the output. Thus,

$$H_{SB}(z) = \frac{B_{BP}(z)}{A_{BP}(z)} - 1$$

$$= \frac{B_{BP}(z) - A_{BP}(z)}{A_{BP}(z)}$$

$$= \frac{B_{SB}(z)}{A_{SB}(z)}$$

where B_{SB} and A_{SB} are stop-band filter's zero and feedback coefficients, respectively. Magnitude response and pole and zero plot for the notch filter are shown in figures 6.20 and 6.21, respectively.

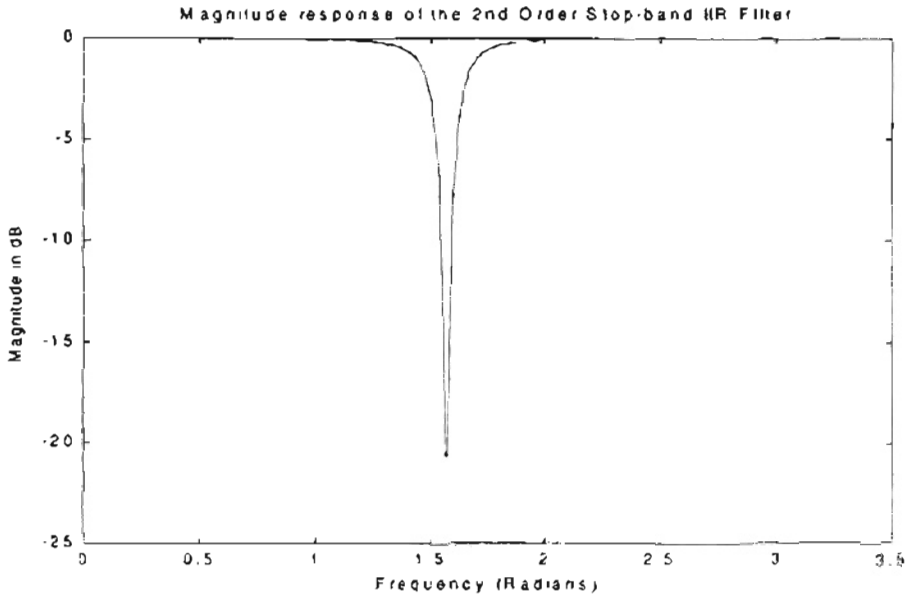


Figure 6.20 Magnitude Response of 2nd Order IIR Stop-band Filter.

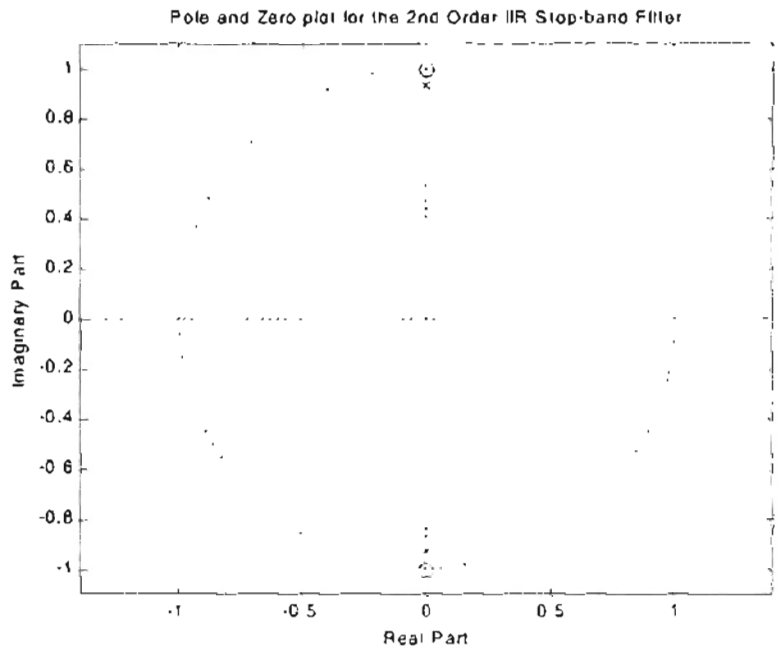


Figure 6.21 Pole and Zero plot for 2nd Order IIR Stop-band Filter.

Since the transfer function of the Tunable Heterodyne filter results in an add to the transfer function of translated filters, the notch filter could not be used in structure. Since not all the band-pass filters can be converted into a notch filter, this b pass filter is unique with respect to this property. The input of the heterodyne filter c be subtracted from the output of the heterodyne filter to give an effect of a tuning n filter. Since only a very small number of unique band-pass filters can be converte notch filters, this heterodyne filter is specifically used for tuning band-pass filters. A structure "Fully Tunable Heterodyne Filter" eliminates this problem at the expens more hardware. The tunable heterodyne band-pass filter implemented in this th project is a basic block in the Fully Tunable Digital Heterodyne IIR Filter [26].

The structure used to implement the fixed frequency Band-pass filter was Dire Form II. This structure is shown in figure 6.22.

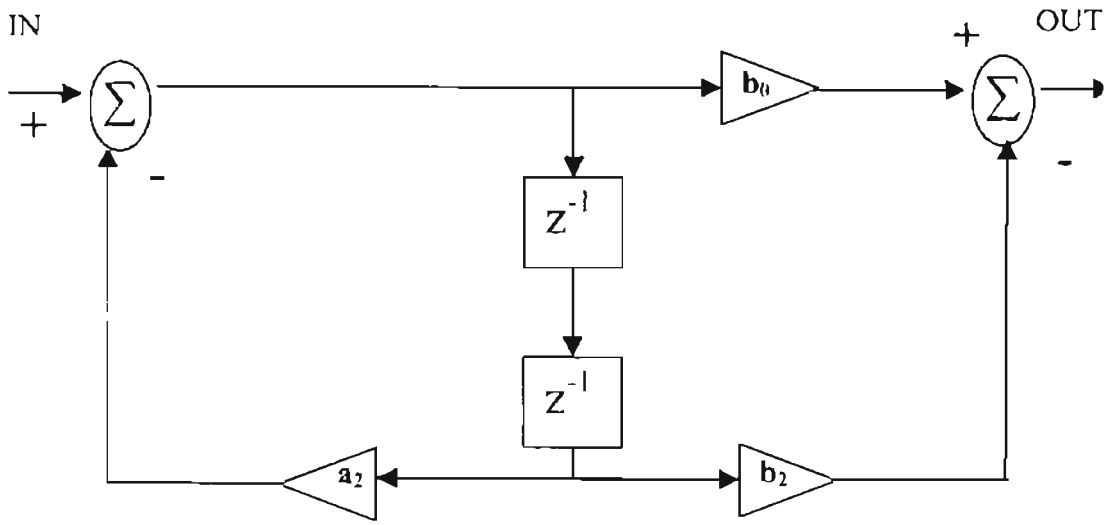


Figure 6.22 Direct Form II structure of 2^{nd} order Band-pass IIR Filter.

Since the coefficients of the filters are known and they do not need to be changed during the tuning process, therefore add and shift procedure is used to implement the constant coefficient multipliers. The coefficients of the band-pass filters are,

$b_0 = 0.0625$

$b_2 = -0.0625$ $\qquad\qquad$ $a_0 = 0.875$

Since the 'b' coefficients have the same magnitudes therefore we implement the negative sign in the actual hardware (as shown in figure 6.8) and treat the coefficients as unsigned numbers. The 8-bit two's complement representation of 0.625 and 0.875 given below,

$0.0625_d = (0.0\,0\,0\,1\,0\,0\,0)_b$

$0.875_d = (0.1\,1\,1\,0\,0\,0\,0)_b$

This is a fixed-point 8-bit representation in binary digits, whereas the binary point is shown to illustrate the fact that the coefficients are fractions. Thus in the final answer, it is required to keep a consistent representation so that the binary point convention used in the design does not change. The implementation of the 'b' coefficients is relatively easy, since they don't require any extra hardware and can be built by only right shifting the incoming input by four, hence dividing the incoming signal by 16. The 'a' coefficient requires more hardware to be implemented, since there are three non-zero numbers in it. Using binary multiplication, this coefficient can be implemented using two 11-bit 2's complement adders. Using Canonical Signed Digits as explained in the chapter 2, we can implement the 'a' coefficient with a reduced number of adders. The following is the CSD representation of the 'a' coefficient,

$0.875_d = 1.00\text{-}10000$

The CSD representation of the coefficient requires only one adder rather than the two implemented as shown in figure 6.23.

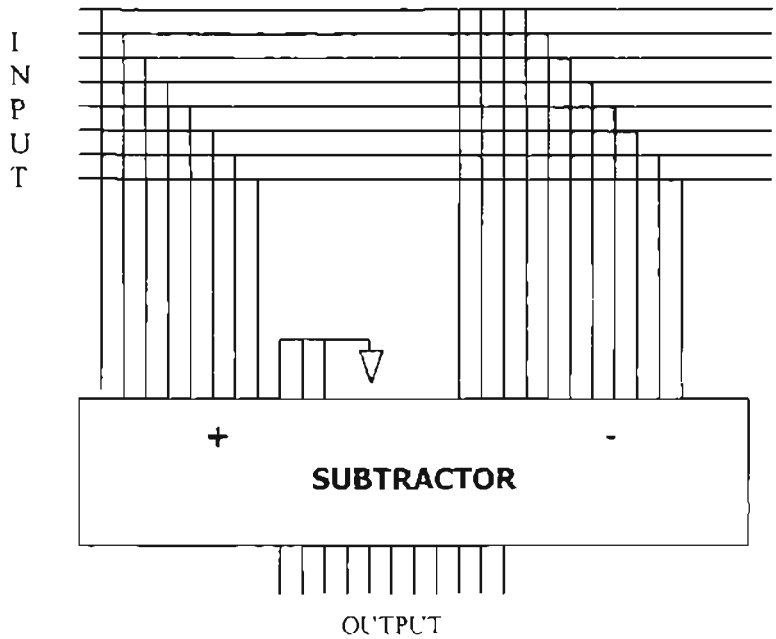


Figure 6.23 'a' Coefficient Implementation Structure.

6.2.1 Synthesized Circuit of 2nd Order Band-pass IIR Filter

Direct Form II structure configuration was used to implement the prototype 2nd order Band-pass IIR filter. We have to be careful when designing an IIR filter, since the truncation of bits is inevitable in these filters and may cause feedback coefficients to change the location of poles, rendering the system unstable. In this structure all the internal nodes are arranged to have an 8-bit wide data path. Figure 6.24 shows the synthesized circuit for this filter. The VHDL file used to implement this structure is: iir.vhd.

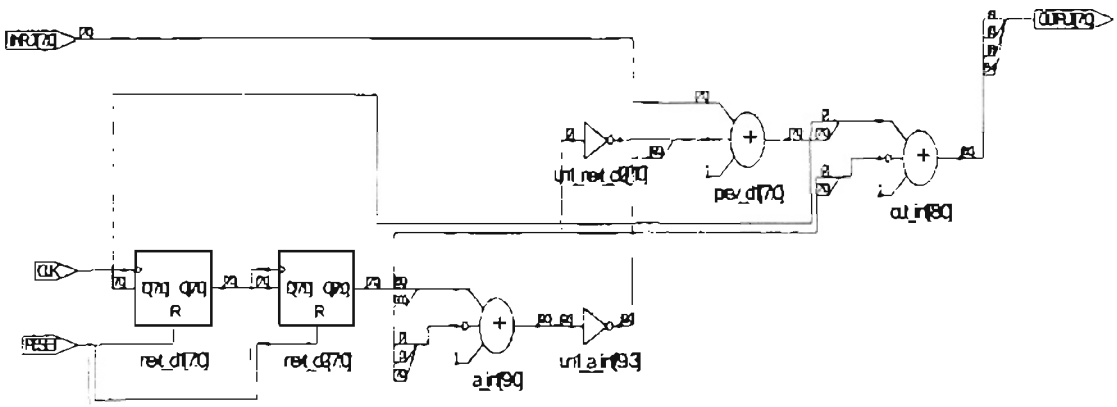


Figure 6.24 Synthesized Circuit of IIR Filter.

6.2.2 FPGA Resource Utilization in Filter Circuit

Table 6.3 shows the utilization of Slices, LUT and Registers for the IIR Filter circuit using a Virtex 800 family FPGA chip.

	Xilinx Implementation		
	Slices	**LUT**	**Reg**
Filter	18	26	16

Table 6.3 FPGA Resource Utilization for the Filter Circuit.

6.3 Hardware Implementation of Combiner Circuit

The combiner circuit is implemented exactly the same way as the Splitter circuit. One difference between the combiner circuit and the splitter circuit is that there is an extra adder that adds the two outputs from the multipliers. Another difference between the two structures is that the Combiner takes two separate inputs whereas a splitter needs only one. The same heterodyning frequency was used in the combiner circuit to produce the same frequency sine and cosine waveforms.

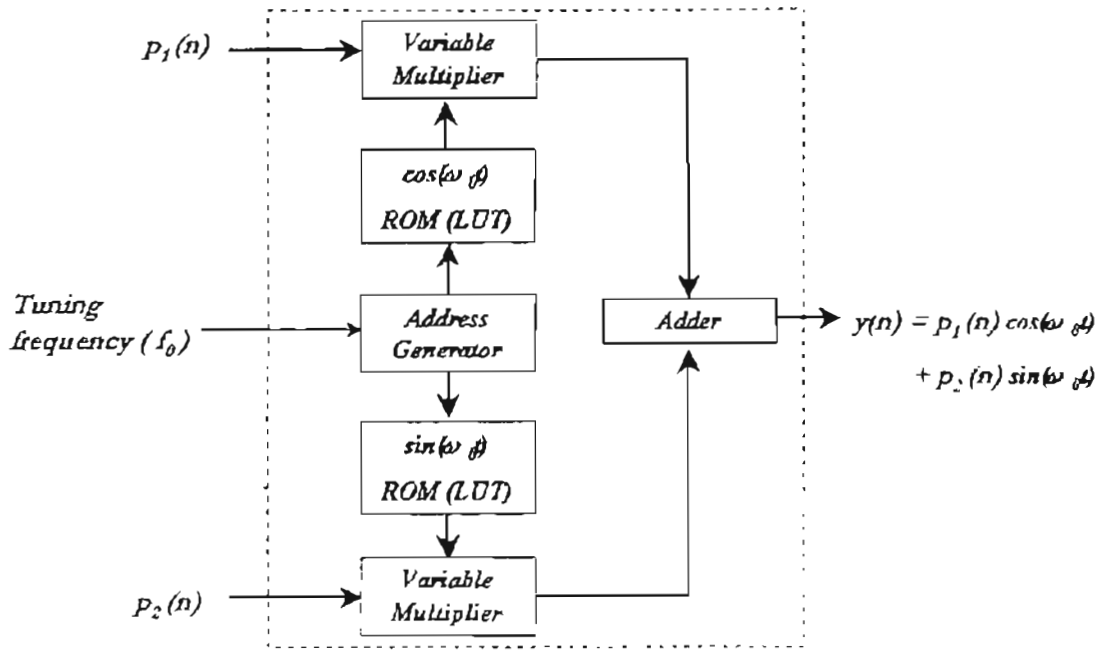


Figure 6.25 Block Diagram for Combiner Circuit.

The VHDL code for Splitter and Combiner Circuits was generalized, so that they both can be used as a component in these circuits whereas the modifications to attain required structures were done in the main VHDL file. Since both the Splitter and Combiner circuits share similar hardware, in the future only one structure can be used in

an application where further area optimization is required at the expense of an extra number of clock cycles.

6.3.1 Synthesized Circuit of Combiner

The synthesized circuit for the Combiner circuit is shown in figure 6.26. Same VHDL files were used to implement this structure as Splitter circuit: counter16.vhd, sin_cos.vhd, two_comp.vhd, two_comp_out.vhd, mult8x8.vhd, mult_top.vhd, mixer_cir.vhd.

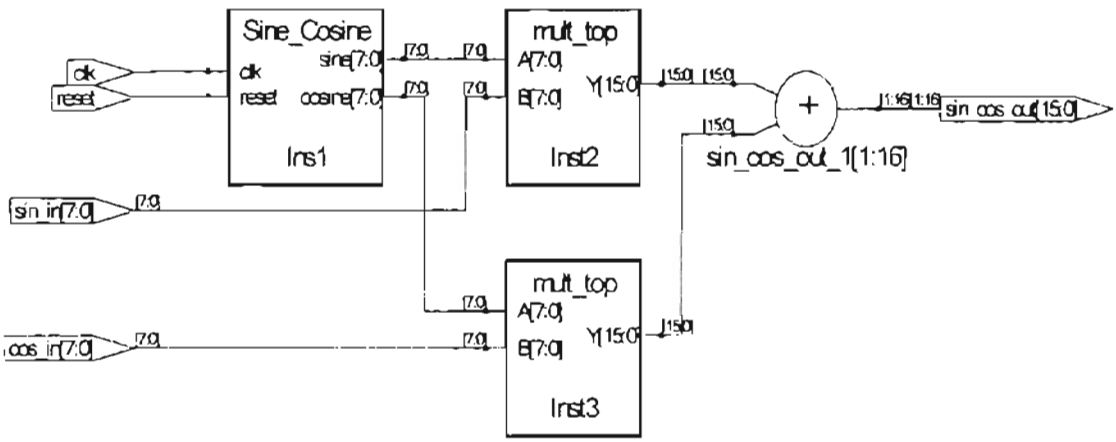


Figure 6.26 Synthesized Combiner Circuit.

6.3.2 FPGA Resource Utilization in Combiner Circuit

Table 6.4 shows the utilization of Slices, LUT and Registers for each component in the combiner circuit using a Virtex 800 family FPGA chip. Two multipliers and one Sine/Cosine generator are used to implement the Combiner Circuit.

	Xilinx Implementation		
	Slices	LUT	Reg
Sine/Cosine	30	53	6
Multiplier	53	100	0
Combiner	142	256	6

Table 6.4 FPGA Resource Utilization in the Combiner Circuit.

Comparing the resource utilization table for the Splitter (Table 6.1) and Combiner (Table 6.4), we can see that both the circuit require almost same number of slices and LUTs, where the combiner has slightly more slice count due to an extra adder.

6.4 Hardware Implementation of Prototype 6th Order IIR Filter

To completely eliminate narrowband interference from the broadband communication signal without introducing any kind of distortion, we require high performance filters. Typically these filters have a narrow pass-band and sharp transition bands, to reduce the effect of filtering on the adjacent information data. From the simulation and experimental results (chapter 7), it can be seen that the second order filter designed as a prototype filter does not give impressive results. A higher order filter needs to be designed to show that better performing filters would yield improved results at the output of the tunable heterodyne filter.

The details regarding the design and implementation of 6th order Chebyshev II band-pass filter are presented in appendix A. Figure 6.27 shows the magnitude response

of this filter produced in Matlab. As is seen from the figure, the pass-band for the filter is narrow (approx. 500Hz) and the transition bands are sharp, compared to the second order filter shown in figure 6.18.

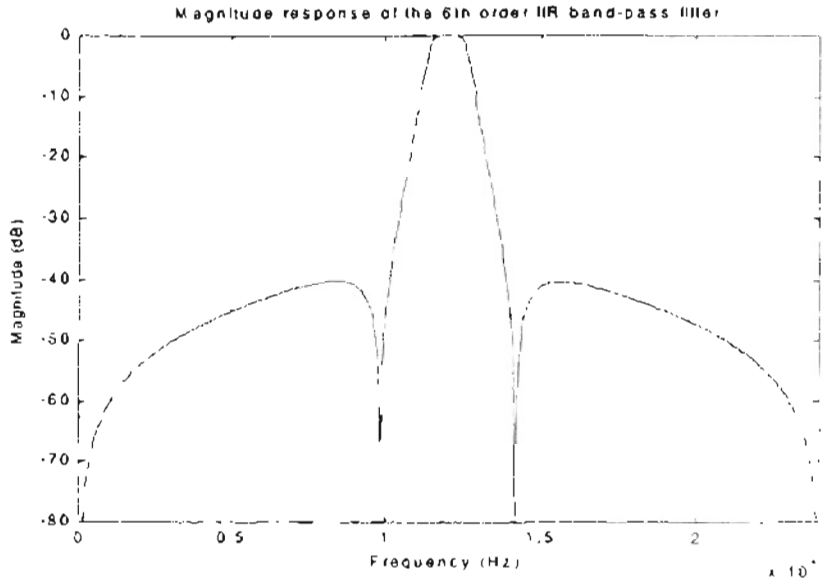


Figure 6.27 Magnitude response of the 6th order Chebyshev band-pass filter.

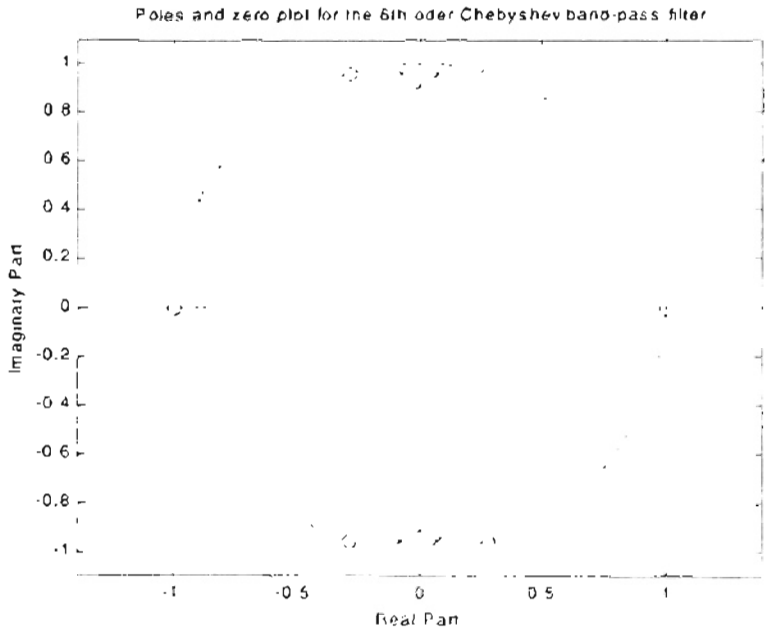


Figure 6.28 Poles and zeros of the 6th order Chebyshev band-pass filter.

Since this 6th order filter has a larger structure, we have to make a design tradeoff between the performance of the filter and the amount of hardware utilized. The filter was constructed using a cascaded structure scheme to reduce the effect of quantization error produced due to bit truncation. In this type of structure, each pole and its conjugate are realized independently from other poles, thus confining the effect of quantization on poles to be local in each cascade stage [28].

6.4.1 Synthesized Circuit of 6th Order Band-pass IIR Filter

Each stage in this configuration is effectively a second order filter, cascaded in series to yield a final desired response. Figure 6.29 show the top-level cascaded band-pass structure. The detailed structure used to construct these stages is given in appendix A.

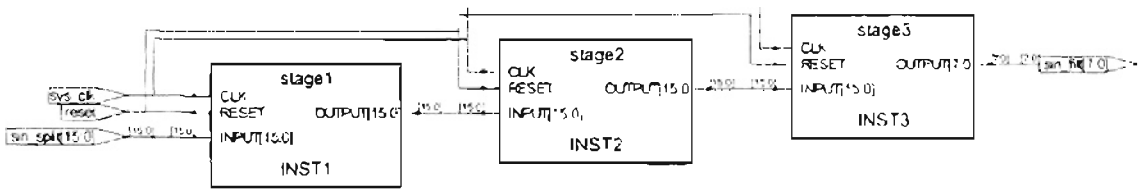


Figure 6.29 Synthesized cascaded IIR band-pass filter.

6.4.2 FPGA Resource Utilization in 6th Order Filter Circuit

Table 6.5 shows the utilization of Slices, LUT and Registers for the 6th order IIR band-pass filter circuit using a Virtex 800 family FPGA chip.

	Xilinx Implementation		
	Slices	**LUT**	**Reg**
Filter	334	478	48

Table 6.5 FPGA Resource Utilization for the 6^{th} order Filter Circuit.

Comparing the hardware utilization for the second order filter (table 6.3) and the 6^{th} order filter (table 6.5), we notice that considerably more hardware is required to implement the 6^{th} order filter, thus trading off hardware for performance.

6.5 FPGA Resource Utilization of Complete Heterodyne Filter

Two heterodyne filter structures were designed; one with a 2^{nd} order IIR filter as a prototype filter and the second with a 6^{th} order IIR filter as a prototype filter. Table 6.6a and 6.6b, illustrate the hardware utilization for both of these structures. It can be seen from the tables that the hardware required for implementing tunable heterodyne filter with 6^{th} order band-pass filter is about 3 times the hardware used to implement with 2^{nd} order band-pass filter. Thus we have utilized more hardware to obtain better performance at the output of the heterodyne filter structure.

Components	Xilinx Implementation			
	Heterodyne with 2nd order Filter			
	Splitter	Filter	Combiner	Total
Slices	139	18	142	293
LUT	253	26	256	491
Reg	6	16	6	43

Table 6.6a Hardware utilization for heterodyne structure with 2nd order filter.

Components	Xilinx Implementation			
	Heterodyne with 6th order Filter			
	Splitter	Filter	Combiner	Total
Slices	139	334	142	952
LUT	253	478	256	1403
Reg	6	48	6	107

Table 6.6b Hardware utilization for heterodyne structure with 6th order filter.

Chapter 7

Simulation and Experimental Results

7.1 Simulation

Simulation results to observe the response of the tunable heterodyne band-pass filter were obtained by using Matlab tools. Since this structure implementation uses a tunable band-pass filter, the expected signal at the output of the tunable filter is the narrowband interference frequency. This tunable structure can replace a transversal filter in an adaptive narrowband interference suppression circuit, where the interference frequency can be subtracted from the original incoming signal by adapting the heterodyne frequency. In the case of an adaptive heterodyne filter, the expected input signal is composed of a spread-spectrum signal, thermal noise and the interference signal. Since the value for the interference signal can be predicted from its behaviour because its a narrowband process [3], the heterodyning frequency parameter can be adapted to translate the interference frequency to the fixed filter frequency. Eventually, as the error converges to zero, the filter is locked to the narrowband interference, which then can be subtracted from the original signal.

Although the real spread-spectrum signal is composed of many frequencies, for the purpose of simulation and hardware experiments, a two-tone signal test was used. In this experiment, two known frequencies are used as input to the tunable heterodyne filter. One of the frequencies is characterized as narrowband interference frequency, which even in a real world signal appears as a pure sine or cosine tone. The other frequency is the desired frequency used to observe and analyze the behaviour and the accuracy of the heterodyne filter.

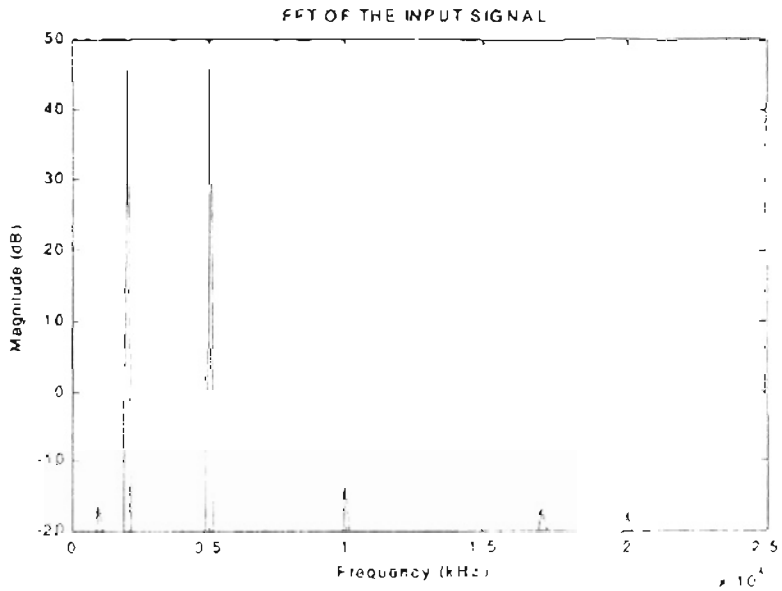


Figure 7.1 Input Spectrum for the Heterodyne Filter.

To analyze and confirm the theoretical results for the tunable heterodyne filter, we step through each node in the filter structure and see its response in simulation. These results are then compared to the theoretical expectations. The Fast Fourier Transform of the input signal is shown in figure 7.1, where the input spikes are located at two frequencies, 2000Hz and 5000Hz.

The frequency of 2000Hz was chosen as the interference frequency. This two-tone signal is presented at the input of the Splitter circuit, where in one of the channels it is multiplied by the cosine function and in the other branch it is multiplied by the sine function. The frequency of sine/cosine functions, which is the heterodyne frequency, is carefully chosen in this experiment to translate the interference frequency to the fixed filter frequency (table 3.1). Since we are using a 2^{nd} order band-pass filter discussed in chapter 6, the center frequency of the filter is set to be a quarter of the sampling frequency. In this simulation example we are using 48kHz as our sampling frequency. Therefore the center frequency of the fixed filter is located at 12kHz. The interference frequency is chosen to be 2000Hz, therefore, to translate the center frequency of this signal to the center frequency of the band-pass filter, we set our heterodyning frequency to be 10000Hz. As a result of this, the frequencies produced after the Splitter circuit are 8000Hz and 12000Hz; 5000Hz and 15000Hz, as shown in figure 7.2.

The modulated signals have half the magnitude of the original signal, as shown in figure 7.1. The absolute value of the input signal is 46dB, whereas the magnitude of the modulated signals from sine branch and cosine branch are 40dB and 40dB, respectively.

The output of the splitter circuit goes into the band-pass filter in both the channels. Figure 7.3 shows the FFT of both the channel outputs.

The filter is centered at 12kHz, where the highest peak is shown. Since this is a 2^{nd} order filter and does not have sharp transition bands, we can see the peaks at frequencies other than the 12kHz. This effect can be reduced if a higher order filter with sharp transition bands has been used to filter out the interference frequency.

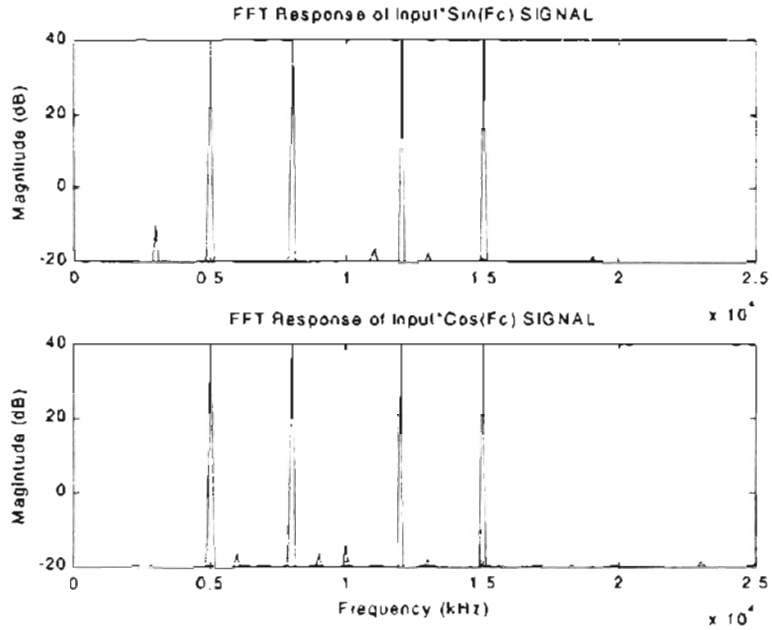


Figure 7.2 FFT magnitude response at the outputs of Splitter two branches.

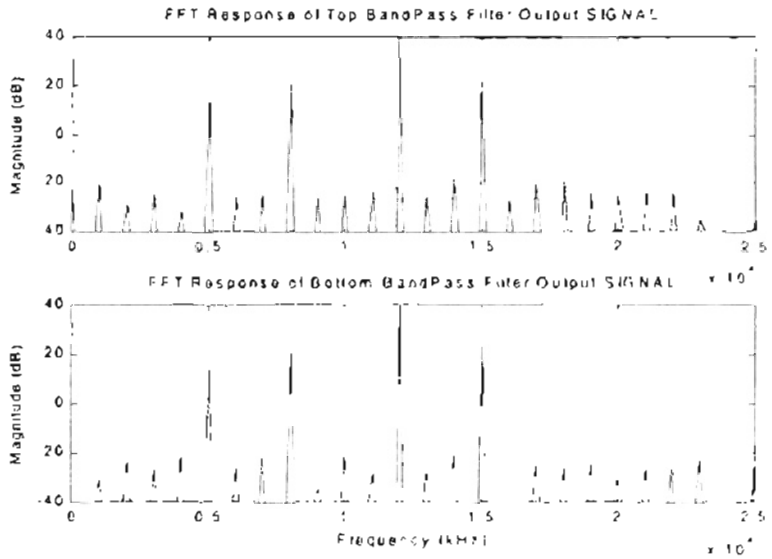


Figure 7.3 FFT magnitude response at the outputs of Filters.

This result shows the importance of having a higher order filter for narrowband interference, which can be done fairly easily in tunable heterodyne band-pass filter technique as compared to the transversal filter, where it would be impractical to realize a

very high order IIR filter. The results obtained by using a high performance filter are shown later. The output of the filters has the maximum amplitude of 40dB, at 12kHz frequency. The magnitude of the other frequencies is observed to be: at 5000Hz the magnitude is 14dB, at 8000Hz the magnitude is 20dB and at 15000Hz the magnitude is 24dB. This can be compared with the magnitude response of the band-pass filter given in figure 6.18. It should be noted that the magnitude of the frequencies in figure 7.3 depends on the magnitude response of the band-pass filter.

The output of the filter is then heterodyned again to bring the interference signal back to its base-band. The outputs shown in figure 7.4 are the outputs of the two channels.

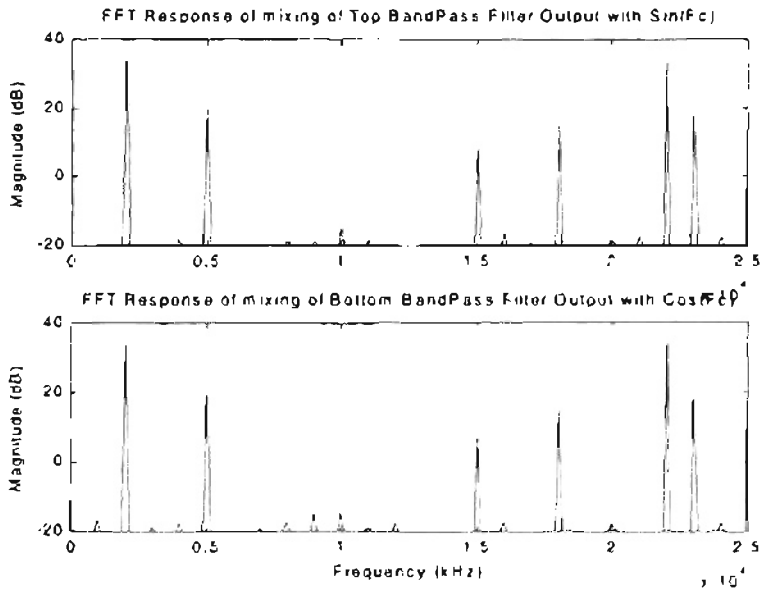


Figure 7.4 FFT magnitude response at the final mixers.

As we can see, the interference frequency (2000Hz) is brought to the base-band, but still at this time there are images that need to be cancelled out. The magnitude of the

output signal has once again been halved because of the heterodyning process, thus yielding 35dB and 35dB magnitudes in the sine and the cosine channel respectively.

The final output is then obtained by adding the two outputs from the heterodyning multipliers. Figure 7.5 shows the final output of the Tunable Heterodyne Band-pass filter.

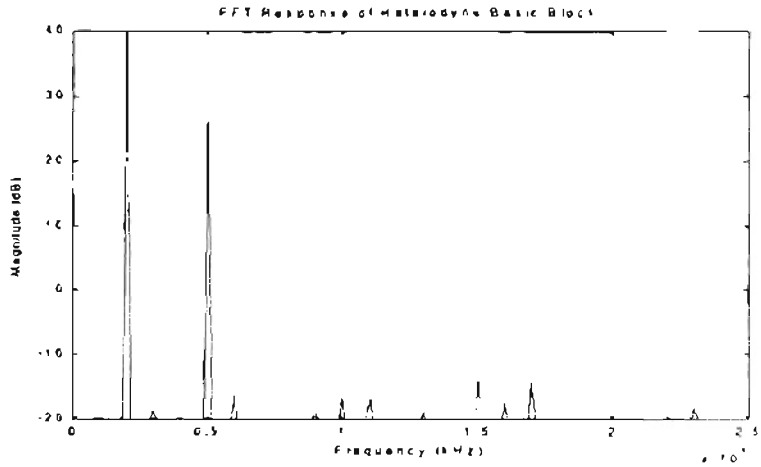


Figure 7.5 FFT magnitude response of Tunable Heterodyne Band-pass Filter.

Although the only output we required is the interference frequency at 2000Hz, the 5000Hz frequency has leaked in the final response due to quantization error and crude band-pass filter response. From figure 7.5, we observe that the difference in magnitude of the two spectra is about 15dB. A better response of the Tunable Heterodyne filter can be shown if we use a higher order band-pass filter, with sharp transition bands.

To make a comparison in simulation and to show the results of inserting a higher order filter in the heterodyne structure, a sixth order IIR band-pass filter is chosen. The magnitude response of the filter is given in figure 7.6. As can be seen from this figure, the filter has sharp transition bands and is also centered at 12KHz.

Again we go through the same exercise to see the output response of the tunable heterodyne band-pass filter using the sixth order filter. Since the output of the Splitter is not affected by the insertion of the 6^{th} order filter, we start our analyses from the output of the filter unit in each channel of the heterodyne filter structure.

Figure 7.7 shows the response, which is comparable to figure 7.3. Although it appears that even using a higher order filter we are still getting all the other frequencies, the magnitudes of other frequencies are well attenuated compared to the spike at 12kHz. The magnitude of the signal at the output of the filter is almost the same as the input (40dB), thus having a negligible loss of magnitude response. The magnitudes of all the other frequencies appearing in figure 7.7 are below 0dB, thus making the difference of 40dB between the signal of interest and other frequencies. This is again defined in the 6^{th} order band-pass filter specifications where the stop-band attenuation is 40dB.

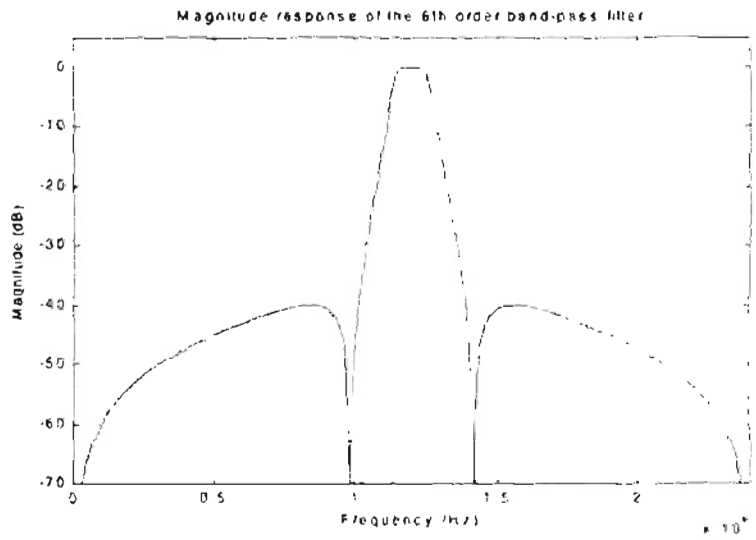


Figure 7.6 Magnitude response of 6^{th} Order Band-pass IIR Filter.

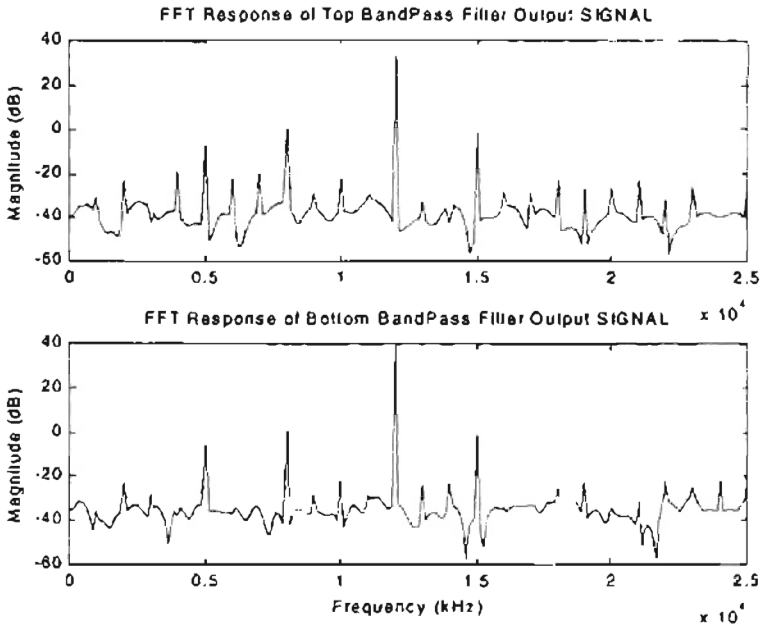


Figure 7.7 FFT magnitude response at the outputs of 6th order Filters.

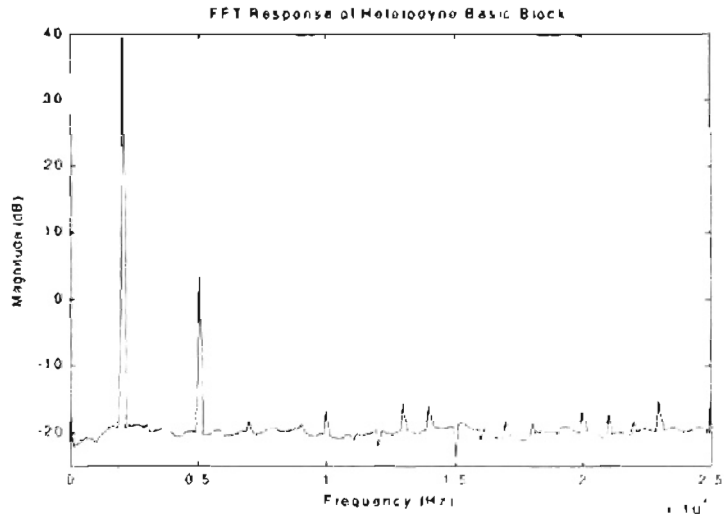


Figure 7.8 FFT magnitude response Heterodyne Filter, 6th order Band-pass Filter

Hence in the same manner we can see the final output of the heterodyne filter after the two channels had been added together. Figure 7.8 shows the results of the output response of the tunable heterodyne band-pass filter with a higher order filter, compared to

figure 7.5. The output Signal is approx. 40dB, whereas the frequency at 5kHz is about 40dB below the 2kHz frequency.

7.2 Experimental Results

The Tunable Heterodyne Band-pass Filter structure with the 2nd order band-pass filter was implemented using Xilinx Virtex 800 family FPGA. Each structure was implemented separately and then the whole system was put together in a single Virtex 800 Chip. This section gives a slight overview of our design, testing setup, and then an analyses of the results obtained from each section as well as the whole system and compares them with the results obtained through simulation.

7.2.1 Experiment Setup

Since our implementation is in the digital domain, it was required to have the analog signal generated outside by a function generator to be digitized in order for it to be used in the circuit. The output of the filter is converted from digital into an analog signal. A proto-board with Xilinx Virtex 800 FPGA was used as a testing platform. Along with the Virtex FPGA it also had an Analog to Digital (ADC) and Digital to Analog (DAC) converters inside a 20-bit CODEC chip. To interface between the FPGA chip and the CODEC chip, a CODEC handshaking module needed to be implemented in the FPGA. This interface between the two chips was done by instantiating a code written in VHDL. The CODEC chip took an analog input through a stereo "in" jack and converted that into a digital bit stream, which was then serially transferred to the FPGA. It also took a serial bit stream from the FPGA chip and converted this data into an analog signal, which went to the stereo "out" jack. The VHDL interface code converted the serial bit stream from

the CODEC, into a parallel 20-bit bus to be used by our design. It also converted the parallel out of our design into a serial bit stream, which was sent to the CODEC chip [18]. A block diagram of the interface is shown in figure 7.9.

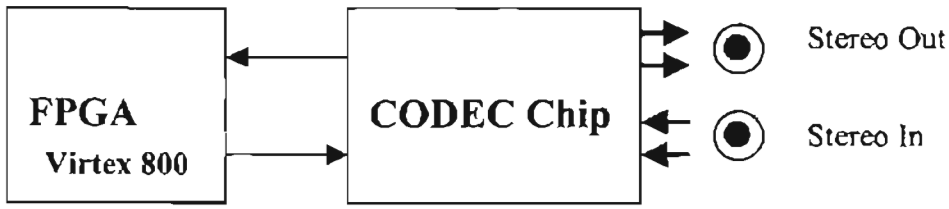


Figure 7.9 CODEC and FPGA interface with external signals.

The stereo "in" signal was connected to the function generators producing sinusoids at two different frequencies. These frequencies came in as stereo left and right channels, which were multiplied inside the FPGA chip to develop a two-tone signal. Stereo "out" was connected to the spectrum analyzer and/or digital oscilloscope. A block diagram of this configuration is shown in figure 7.10.

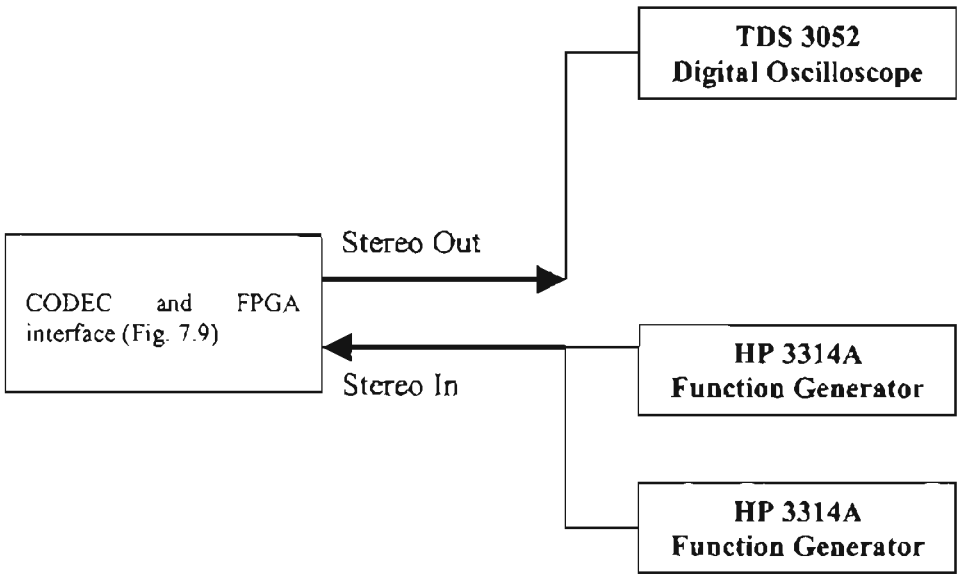


Figure 7.10 Data Acquisition setup, stereo in/out jack ports.

7.2.2 Hardware Results for Heterodyne Filter using 2nd Order Filter

Each component of the tunable heterodyne filter was tested for its functionality. The sampling rate used in all the experiments was 48kHz. In these experiments various combinations of the left and right channel for the stereo cable were used.

7.2.2.1 Splitter Experimental Results

To compare our experimental data with the simulation results obtained from Matlab, we used the same input frequencies for the two-tone test, 2000Hz and 5000Hz. Similarly, we chose 2000Hz as our interference frequency and 5000Hz as our analyzing frequency. All the nodes in the Tunable Heterodyne filter were kept at the 8-bit precision.

Since we had to truncate the width of some of the nodes to 8-bits, this introduced quantization error, which appeared as noise at the output.

Two tones at frequencies 2000Hz and 5000Hz can be produced by either two analog sine waves and adding them up in analog domain, or two frequencies, 1500Hz and 3500Hz, which when digitally multiplied, yield frequencies at 2000Hz and 5000Hz.

Figure 7.11 shows the output of the Splitter circuit using Xilinx Virtex 800 FPGA. Both the channels in the Splitter circuit produce the same results as figure 7.11.

Since our interference frequency is set at 2000Hz and the fixed filter is centered at 12000Hz, we translate the interference signal to 12000Hz by setting the Heterodyning frequency at 10000Hz. As the sine and cosine functions are digitally synthesized, according to equation 6.3, we set our ROM clock rate at 640KHz.

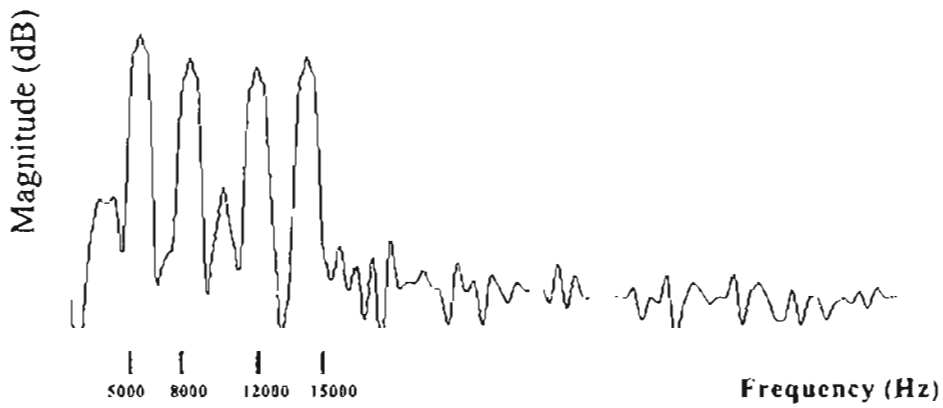


Figure 7.11 Hardware results of the Splitter circuit.

7.2.2.2 2nd Order Band-pass Filter Experimental Results

Varying the frequency for the input signal of the filter and using the oscilloscope to record the output response, we were able to obtain the magnitude response of the 2nd order IIR Band-pass filter. To compare the response to the Matlab simulation result of the IIR band-pass filter, we convert the absolute values of output voltage verses the input voltage into decibels. Figure 7.12 shows the result obtained from hardware superimposed on the Matlab response. The difference in the real hardware response maybe caused by the fewer number of points taken over the frequency sweep and/or data acquisition error translated from the generation of quantization noise.

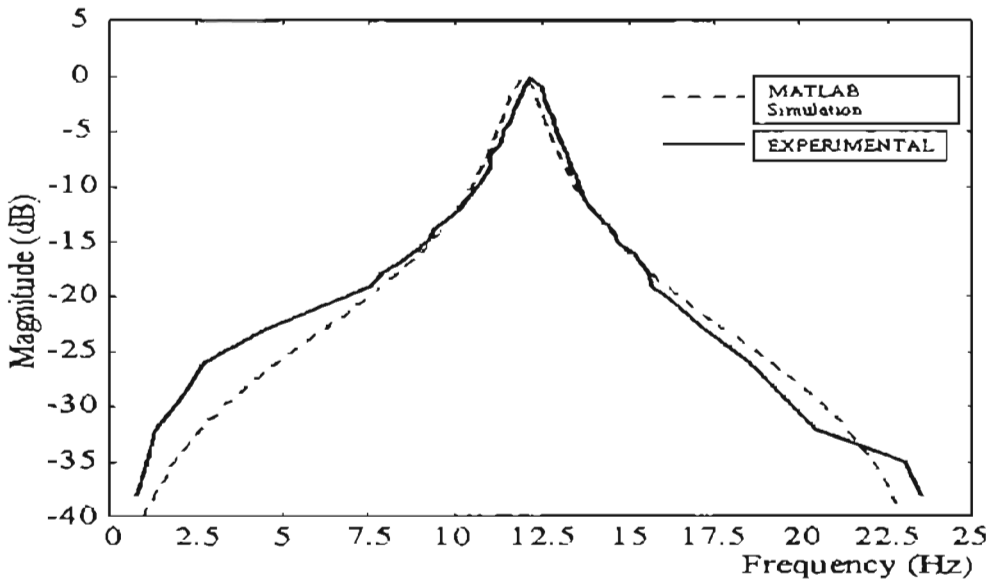


Figure 7.12 Superimposed results of IIR filter from hardware and Matlabs.

Figure 7.13 shows the response after the filters in each channel of the heterodyne filter structure. As shown in this figure, due to the crude response of the 2nd order band-

pass filter, the images of the other frequencies can also be seen. But these frequencies are almost 20dB below the desired frequency.

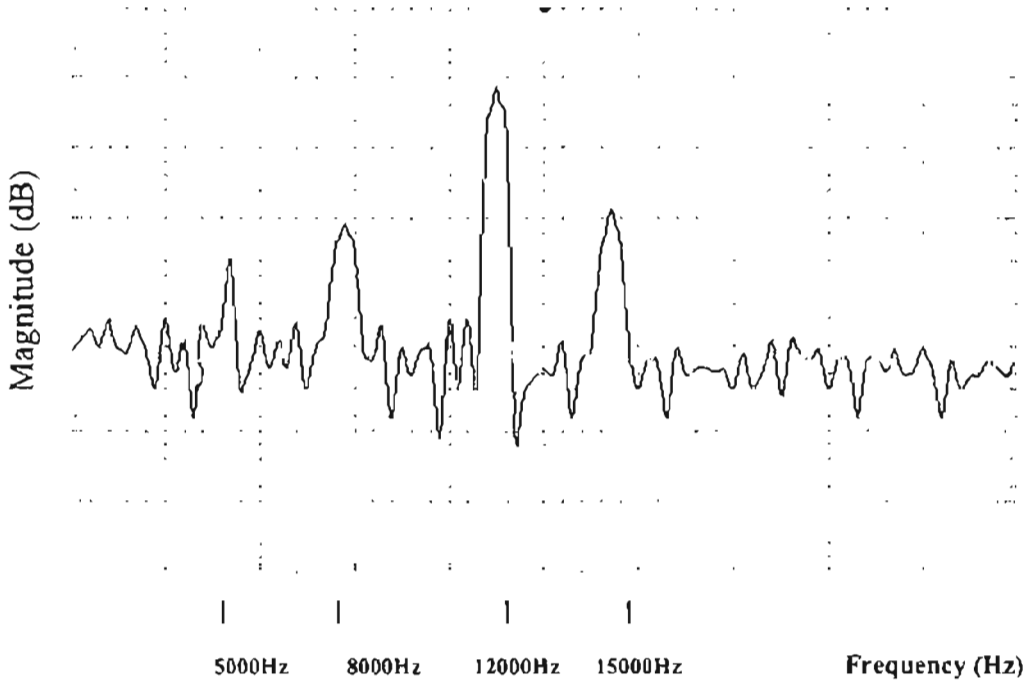


Figure 7.13 Hardware response after the Band-pass filter in Heterodyne Structure

7.2.2.3 Combiner Experimental Results

The outputs of the filters are then fed into the Combiner circuit, where they are mixed with the heterodyne frequency to bring the filtered signals back to the base-band. Figure 7.14 shows the results from the hardware.

There is a small amount of leakage frequencies appearing at the output of the tunable filter, which is again owing to the fact of crude filter response. But we can see that the interference frequency, which is the desired frequency at the output of this

structure, is about 15dB more than the other frequencies. Inserting a higher order filter, with sharper transition bands, can solve this problem.

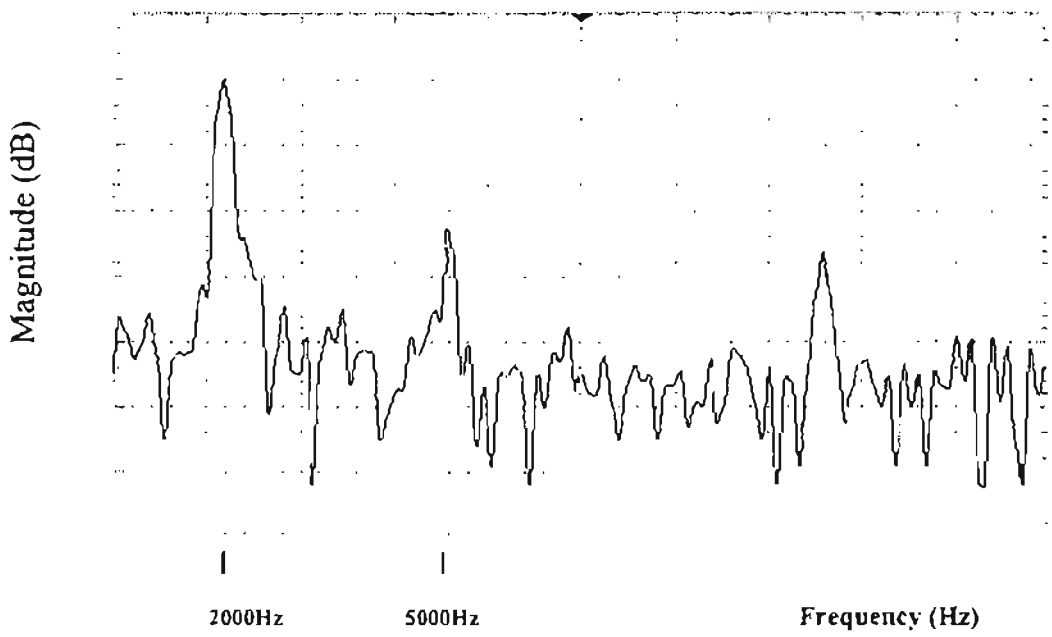


Figure 7.14 Hardware results of the output of the Tunable Heterodyne Filter.

7.2.3 Hardware Results for Heterodyne Filter using 6th Order Filter

Similarly we go through the structure of the heterodyne filter, but this time the 6th order band-pass filter has been inserted rather than the 2nd order filter, as described in the previous section. Again the Splitter results are not affected by using a different type of filter, therefore we will start our analyses at the output of the fixed filters in the heterodyne filter structure.

7.2.3.1 6th Order Band-pass Filter Experimental Results

Similar to the procedure for obtaining the magnitude response of the 2nd order filter described in previous section, we vary the frequency for the input signal of the filter and used the oscilloscope to record the output response to get the magnitude response of the 6th order IIR Band-pass filter. To compare the response to the Matlab simulation result of the 6th order IIR band-pass filter, we convert the absolute values of output voltage verses the input voltage into decibels. Figure 7.15 shows the result obtained from hardware superimposed on the Matlab response. The difference in the real hardware response may be caused by the fewer number of points taken over the frequency sweep and/or data acquisition error translated from the generation of quantization noise.

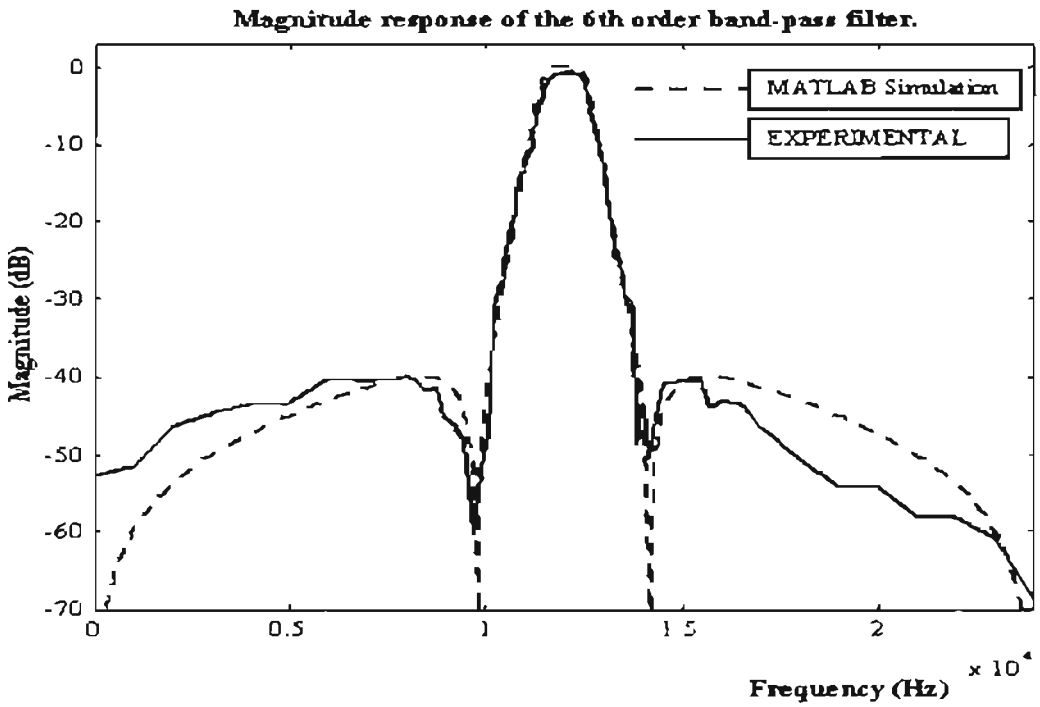


Figure 7.15 Superimposed results of 6th order IIR filter from hardware and Matlabs.

Figure 7.16 shows the response after the filters, in each channel of the heterodyne filter structure. Comparing this result to the one obtained from the heterodyne structure with the 2nd order filter, we see that the results have improved significantly. As shown in this figure, due to the relatively sharp transition bands of the 6th order band-pass filter, the images of the other frequencies are attenuated significantly.

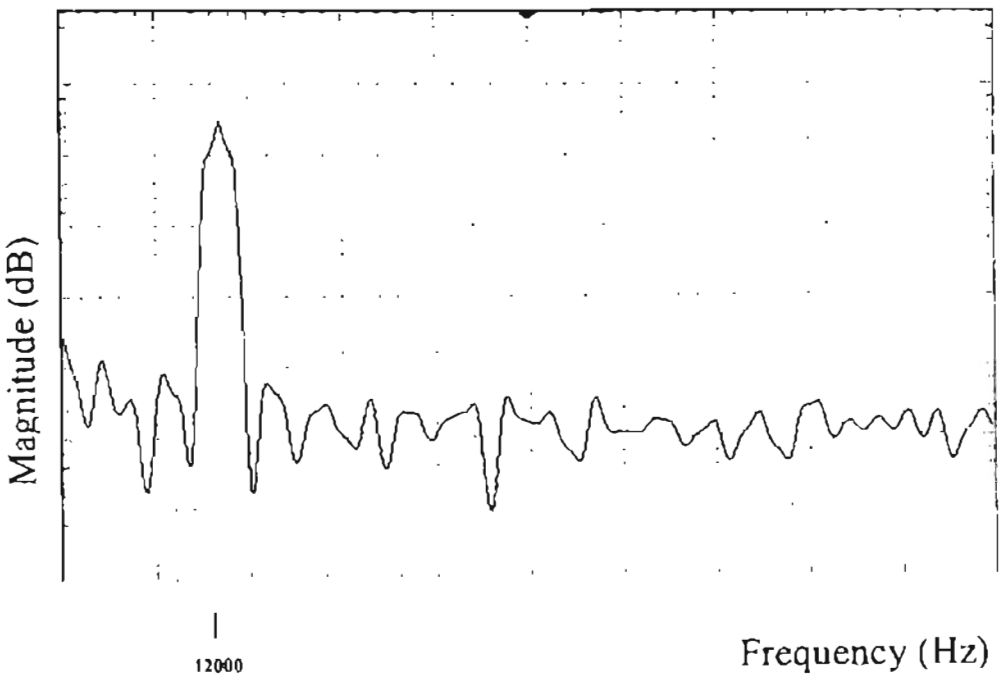


Figure 7.16 Hardware response after the 6th order Band-pass filter

7.2.3.2 Combiner Experimental Results

The outputs of the filters are then fed into the Combiner circuit, where they are mixed with the heterodyne frequency to bring the filtered signals back to the base-band. Figure 7.17 shows the results from the hardware. Again there is a slight leakage of

frequency at 10KHz. This can be due to a synchronization problem with the main system clock or/and quantization error effect.

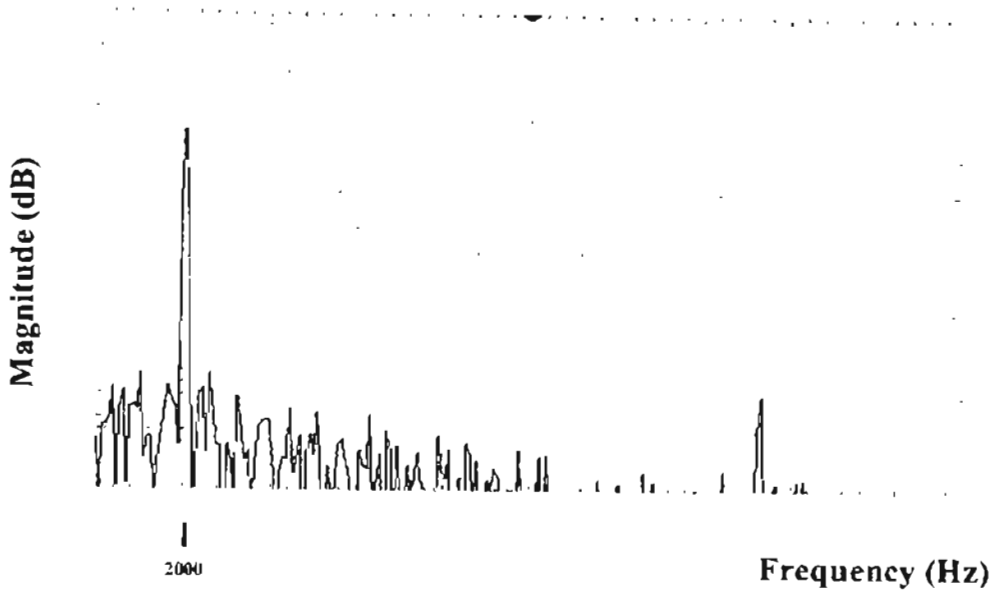


Figure 7.17 Hardware results of the Tunable Heterodyne Filter with 6th order filter.

7.3 Hardware Resource Comparison between Standard and Heterodyne Techniques

The technique of the tunable heterodyne filter is much more hardware efficient with a higher order filter compared to the standard technique of a tuning filter. What we characterize as standard technique is a tunable filter that can be tuned by adjusting the weights of the coefficients. Just to keep all the comparisons within a similar structure, we chose Direct Form II structure (as shown in figure 7.15) in both the techniques.

Thus to compare these techniques, we assume an Nth order Direct Form II, IIR filter. Lets assume the standard technique can tune to M distinct position, where M is the

number of ROM locations for each coefficient, that stores N+1 8-bit filter coefficients for each of the M discrete frequencies that the filter can be tuned to. Table 7.1 estimates the hardware resources in terms of order (N) and tuning positions (M), used to implement both the techniques in Virtex 800 FPGA. Since the tunable heterodyne filter can be tuned continuously from DC to Nyquist/2 or Nyquist/2 to Nyquist, the final term for that is in terms of order (N) only.

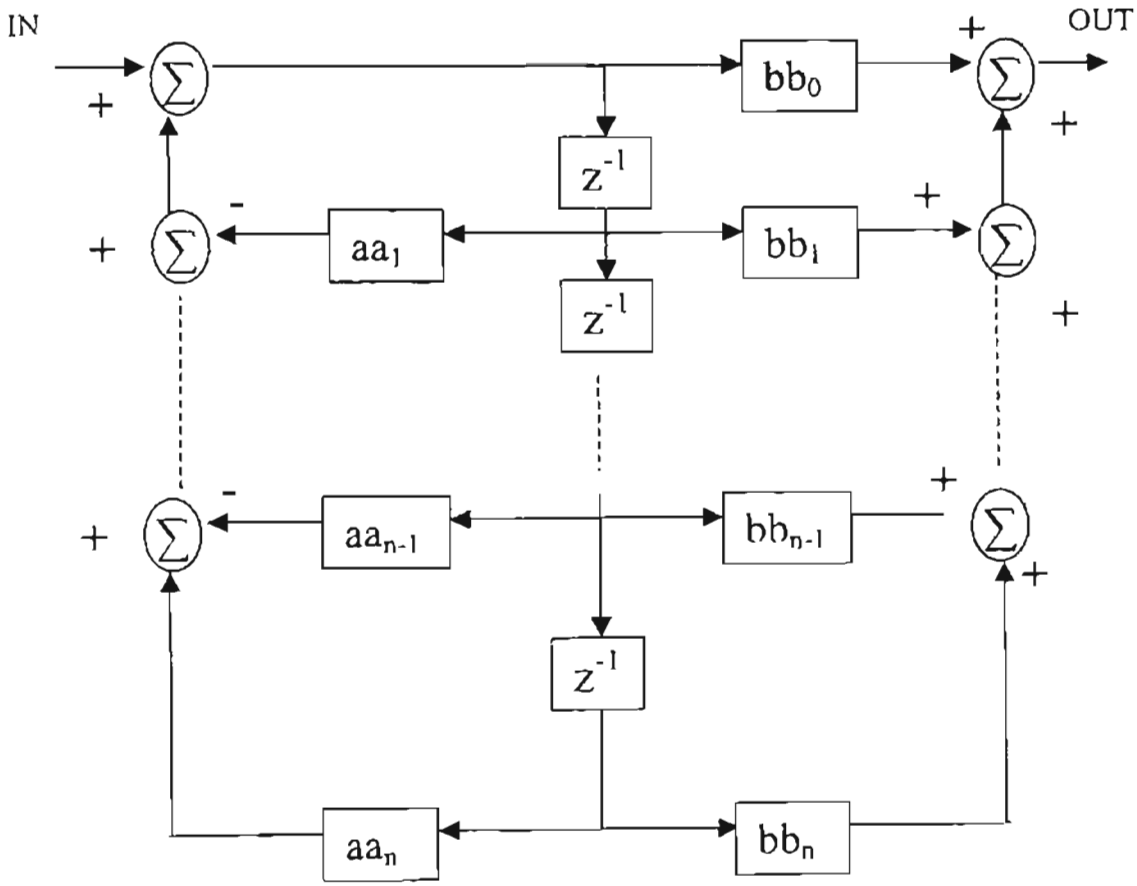


Figure 7.18 Direct Form II IIR filter structure

Component	CLB's Per component	Heterodyne Filter	CLB's in Heterodyne Filter	Standard Filter	CLB's in Standard Filter
Adders	8	4N+1	32N+8	2N	16N
Delays	4	2N	8N	N	4N
Fixed-Coefficient Multipliers	4	4N-4	16N-16	0	0
Variable-Coefficient Multipliers	21	4	84	2N-1	42N-21
64-Bit ROM's	9	8	72	(2N-1)M/8	9(2N-1)M/8
TOTAL			56N+148	62N+9(2N-1)M/8-21	

Note: N is the order of the IIR filter and M is the number of frequencies that the standard filter can be tuned to. The heterodyne filter is continuously tunable while the standard filter can only be tuned to M distinct frequencies.

Table 7.1 Comparison of Heterodyne Tunable Filter to Standard Tunable Filter

To get the same amount of hardware as the tunable heterodyne filter, we equate the two final equations and thus can obtain the number of tuning position the standard filter can be tuned to. Equation 7.1 gives this expression.

$$M = \frac{8(169 - 6N)}{18N - 9} \qquad (7.1)$$

Plugging in the order number, we can see that for higher order filters, the tunable heterodyne filter out performs the standard tunable filter's technique. Figure 7.16 shows the number of tuning frequencies that the standard tunable filter can be tuned to, for various order filters. Therefore if the number of tuning frequencies (M) exceeds the number shown on top of each bar for that particular order filter, then the tunable heterodyne filter technique would be better.

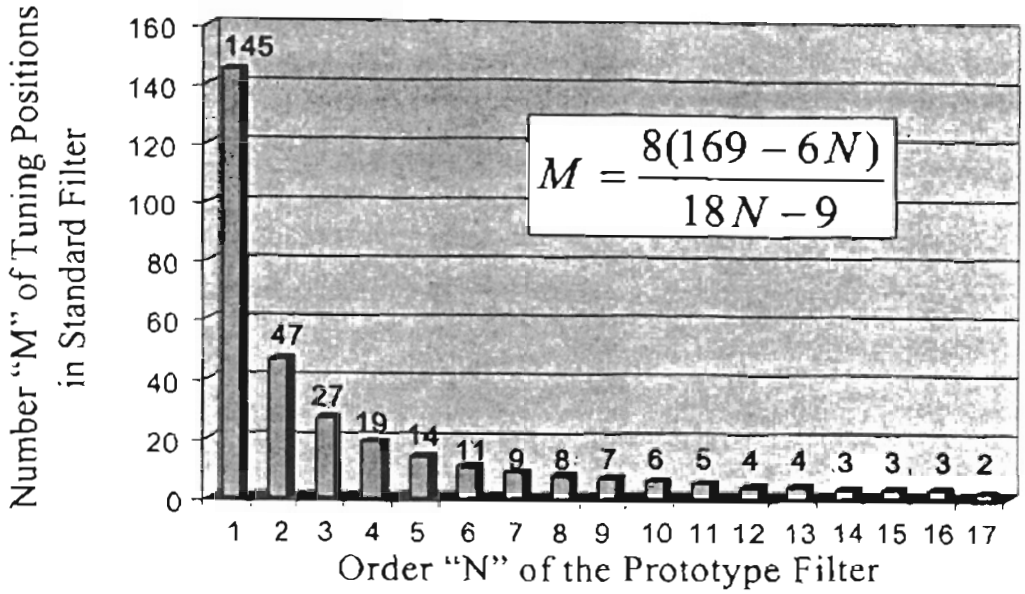


Figure 7.19 Number of tuning position against the order of the filter.

Since we need variable multipliers to implement the standard technique, as the order of the filter is increased to where a reasonable filtering can be obtained for the applications described in chapter 2, the hardware resources needed to build these structures goes up. A highly optimized IIR filter can be realized by using Canonic Signed Digits (CSD) or Dempster & Macleod (DM) to implement constant coefficient multipliers [19] used in Tunable Heterodyne filter technique.

Chapter 8

Future Work

8.1 Alternative Structures

The structure discussed in this thesis has been demonstrated for tuning band-pass filters. We have concentrated on demonstrating the feasibility and the hardware complexity of this structure with various tunable band-pass filers. We have not attempted in this thesis to develop other tunable structures. We have noted that in one very special case, that of a second-order band-pass filter with unity gain in the pass-band, the band-pass filter can be converted to a notch filter. This hints at the possibility of developing other tunable structures from this basic structure. However, the summing nature of the prototype filters in this structure presents a challenge to implementing anything other than band-pass filters. Thus one possible topic for future research would be developing design criteria for tunable filters other than band-pass filters using this structure. This would fall beyond the scope of this thesis, but could result in some very useful structures in addition to the band-pass structure proposed in this thesis.

A different heterodyne filter structure is proposed in [26] which defines a new architecture with which any type of filter would be suitable to tune. The basic block used

in that structure is the Tunable Heterodyne Band-pass filter presented in this thesis report. Therefore if the application requires a tunable band-pass filter, then the structure presented in this thesis would be an ideal choice. On the other hand, if the application can only use tunable notch filters, then a fully tunable heterodyne filter would appear to be a better choice. The design of filters using the filter structure of [26] would be another topic for work beyond the scope of this thesis.

Another potential project would be to implement the hardware for the fully tunable heterodyne filter. The solution to having any type of tunable filter comes at the expense of more hardware requirement. It would be a worthwhile task to build this new structure in hardware for comparison purposes. This would be a very good follow-up topic to the present thesis.

The implementations of the sine and cosine functions in this thesis were done using look-up table LUT techniques. However, this operation could be replaced by the CORDIC (COordinate Rotation DIgital Computer) algorithm that calculates the angle of sin and cosine by using phasors [27], thus yielding $\sin(\theta)$ and $\cos(\theta)$. This algorithm appears to be more efficient in VLSI technologies and some other FPGA technologies than it is in the Xilinx FPGAs. An implementation of Tunable Heterodyne Band-pass filter or Fully tunable Heterodyne filter in VLSI technology would give a good comparison for an optimized implementation of tunable filters using either LUTs or the CORDIC Algorithm. This could be the source of several new projects based upon the work of this thesis.

BIBILOGRAPGY

1. Tero Ojanpera and Ramjee Prasad, "Wideband CDMA for Third Generation Mobile Communications", London, England: Artech House, 1998.
2. Pickholtz, R.L., Milstein, L.B. and Schilling, D.L. "Spread spectrum for mobile communications", Vehicular Technology, IEEE Transactions on , Volume: 40 Issue: 2 , May 1991 Pg: 313 –322.
3. Milstein, L.B., "Interference Rejection Techniques in Spread Spectrum Communications", Proceedings of the IEEE , Volume: 76 Issue: 6 , June 1988 Pg: 657 –671.
4. F. M. Hsu and A. A. Giordano, "Digital whitening techniques for improving spread-spectrum communications performance in the presence of narrow-band jamming and interference", IEEE Trans. Commun., vol. COM-26, pp. 209-216, Feb, 1978.
5. R. A. Iltis and L. B. Milstein, " Performance analysis of narrow-band interference rejection techniques in DS spread-spectrum systems", IEEE Trans. Commun., vol. COM-26, pp. 209-216, Feb. 1978
6. J. W. Ketchum and J. G. Proakis, "Adaptive algorithms for estimating and suppressing narrow-band interference in PN spread-spectrum systems," IEEE, Trans. Commun. ,vol. COM-30, pp. 913-924, May 1982.

7. L. Li and L. B. Milstein, " Rejection of narrow-band interference in PN spread-spectrum systems using traversal filters," IEEE, Trans. Commun. , vol. COM-30, pp925-928, May 1982.

8. Karl E. Nelson and Michael A. Soderstrand, "Adaptive Heterodyne filters (AHF) for detection and attenuation of narrowband signals," IEEE International Conference on Acoustics Speech and Signal. Processing, Seattle, WA, June 1998.

9. M.A. Soderstrand, T.G. Johnson, R.H. Strandberg and H.H. Loomis, Jr., "Suppression of Multiple Narrow-Band Interference Using Real-Time Adaptive Notch Filters," IEEE Transactions on Circuits and Systems, Vol. 44, No. 3, March 1997, pp. 217-225.

10. R.H. Strandberg, M.A. Soderstrand and H.H. Loomis, "Elimination of Narrow-Band Interference Using Adaptive Sampling Rate Notch Filters," Proceedings 28th IEEE Asilomar Conference on Signals Systems and Computers, Pacific Grove, CA, October 1992, pp. 861-865.

11. R.H. Strandberg, J.C. Le Duc, L. G. Bustamante, V.G. Oklobdzija and M.A. Soderstrand, "Efficient Realizations of Squaring Circuit and Reciprocal used in Adaptive Sample Rate Notch Filters," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, Vol 14, No. 3, December 1996, pp. 303-310.

12. L.G. Bustamante and M.A. Soderstrand, "Switched-Capacitor Adpative Sample Rate Filter," Proceedings 40th IEEE International IEEE Midwest Symposium on Circuits and Systems, Sacramento, CA, August 1997.

13. Donald K. Weaver, Jr., "A third method of generation and detection of single side-band signals," Proceedings of IRE, vol. 44, no. 12, pp.1703-1705, Dec. 1956.

14. Asad A. Abidi, "Direct-conversion radio transceivers for digital communications," IEEE Journal of Solid State Circuits, vol. 30, no. 12, pp 1399-1410, 1995.

15. K.E. Nelson and M.A. Soderstrand, *AdaptiveFiltering Using Heterodyne Frequency Translation,* Proceedings 40th IEEE International Midwest Symposium on Circuits and Systems, Sacramento, CA, August 1997.

16. I.A. Glover and P. M. Grant, "Digital Communications," Harlow, England, Prentice Hall, 1998.

17. Richards Lyon, "Quadrature Signals: Complex, But not complicated," http://www.dspguru.com/info/tutor/quadsig.htm , November 2000.

18. XSV Board V1.0 Manual, "How to install and use your new XSV Board", Xess Corporation. United State, 1999-2000.

19. Darren S. Jue, "Optimal Design of Canonic Signed Digit IIR Digital Filters," Master's Thesis, University of California, Davis, CA, 95616, 1996.

20. A.G. Dempster and M.D. Mcleod, "IIR digital filter design using minimum adder multiplier blocks," IEEE Trans, Circuit and Systems, vol. 45, no. 6, pp. 761-763, June 1998.

21. M.J. Sebastian Smith, "Application-Specific Integrated Cicuits,", Addison-Wesley, 1999, ISBN 0201500221.

22. Xilinx, "The Programmable Logic, Data Book 2000," San Jose, CA, 2000

23. J. Bhasker, "Verilo HDL Synthesis, Apractical Primer," Star Galaxy Publishing, Allentown, PA, 1998. ISBN 0-9650391-5-3.

24. Michael Barr, "Programmable Logic: What's it to Ya?," Embeded Systems Programming, June 1999.

25. Xilinx Product Specification, "Sine/Cosine Look Up Table V1.0.2," Logicore, San Jose, CA, 95124, October, 1999.

26. K.E. Nelson and M.A. Soderstrand, "Full Tunable Digital Heterodyne IIR Filter," IEEE International Conference on Acoustics Speech and Signal Processing, 1999.

27. Yu Hen Hu, "CORDIC- based VLSI architecture for Digital Signal Processing," IEEE Signal Processing Magazine, July 1992.

28. A. Oppenheim and R. Schafer, "Discrete Signal Processing," Upper Saddle river, New Jersey: second edition, Prentice Hall, 1999.

29. Sanjit K. Mitra, "Digital Signal Processing, *A computer-based approach*," New York, NY : McGraw Hill, 1998.

Appendix A
IIR Filter Design and Implementation

This appendix presents the design and implementation of a sixth order IIR filter using Matlab and Xilinx Virtex FPGA respectively, in a tutorial manner. This filter is used as a fixed filter in the tunable heterodyne band-pass filter structure introduced in this thesis report. The first part of this appendix goes through the filter design procedure using Matlab from the specifications defined to generate a narrowband band-pass filter with sharp transition bands. The second part of the appendix illustrates the design techniques and procedures used to implementation this structure in Xilinx FPGA. The comparison between the results obtained from the hardware and Matlab simulations is given in chapter 7.

A1. Computer Aided IIR Filter Design

The specification for the band-pass filter was chosen so as to have a narrow band-pass frequency filter with sharp transition bands, which is very important in order to have effective filtering of a narrowband interference signal from the spread-spectrum signal. Usually the interference frequency in the broadband signal appears as a sine tone and

only occupies a specific frequency, thus having its spectral power concentrated in a narrowband. The purpose of developing this filter is to prove that a better system response can be obtained by using a narrowband filter, and consequently this result is compared to the system response obtained from the second order filter designed initially. Figure A.1 shows the representation of typical magnitude specifications required to obtain the desired response.

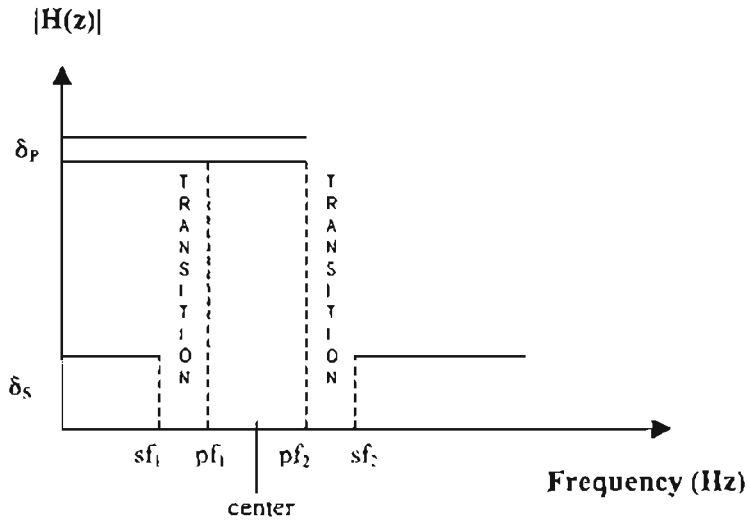


Figure A.1 Band-pass filter critical specifications

The specification values chosen to design this filter are:

center = 0.5;
pf_1 = center*0.98;
pf_2 = center*(1/0.98);
sf_1 = center*0.80;
sf_2 = center*(1/0.80);
δ_P = 0.01 dB;
δ_S = 40 dB;

In Matlab, the Nyquist frequency is referenced as one, therefore to center the band-pass filter at a quarter of the sampling frequency, the center is set at 0.5. The filter type chosen to meet these specifications is Chebyshev II filter, which is monotonic in pass-band and equi-ripple in the stop-band. The choice of pass-band and stop-band frequencies was made to have a narrowband band-pass filter while keeping the order of the filter relatively manageable to implement in hardware. The pass-band ripple (δ_P) is chosen to be 0.01dB and stop-band ripple (δ_S) is chosen as 40dB.

Matlab function of '*CHEB2ORD*' is used to determine the order of the band-pass filter to meet the specification of the filter described above. The Matlab function is:

[N, Wn] = CHEB2ORD(Wp, Ws, Rp, Rs)

where: $W_P = [pf_1\ pf_2]$
$W_S = [sf_1\ sf_2]$
$R_P = \delta_P$
$R_S = \delta_S$

This function returns a number N, which is half the order number (N - order / 2), so to obtain the real order of the filter we have to multiply N by 2. In our case the value of the number 'N' returned was 3, thus making the order of our filter to be 6. This filter order is the minimum requirement needed to meet the specifications defined. This function also returns the stop-band frequency edges 'Wn'.

These values are used in another Matlab function to get the transfer function for the band-pass filter.

[B,A] = CHEBY2(N, Rs,Wn)

This function returns the transfer function in form:

$$H(z) = B(z) / A(z)$$

where the B(z) and A(z) are coefficients for the band-pass filter and are elaborated in the following expression:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4} + b_5 z^{-5} + b_6 z^{-6}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4} + a_5 z^{-5} + a_6 z^{-6}} \qquad (1)$$

Due to the feedback coefficients ($a_i(z)$), IIR filter suffer greatly from quantization error. This error is caused by the truncation of bits after each calculation. To reduce the effect of this error, a cascaded structure was used to implement this filter. In this structure the transfer function is broken down into second order equations and then cascaded in an optimum manner to obtain the required response. Thus as opposed to the direct method, where all the poles need to be realized in the same structure, in cascaded structure each pair of complex-conjugate poles is realized independently of all the other poles. This reduces the error since all the poles and zeros are adjusted independently of the other poles and zeros in the system [28].

Using a Matlab function of,

[SOS,G] = TF2SOS(B,A)

we can generate an optimum set of second order state (SOS) equations from the transfer function (TF) equation obtained in equation 1. This function returns a 6 x L matrix of form:

$$SOS = \begin{array}{cccccc} b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{21} & a_{0L} & a_{1L} & a_{2L} \end{array}$$

where L is the number of stages in the cascaded structure. This function also returns the gain 'G' which is included in the first section of the cascaded structure. Each stage is cascaded in the same order as given in the SOS matrix [29]. Since the band-pass filter is a sixth order filter, we get three second order equations as follows:

SOS =

Columns 1 through 3

1.00000000000000	-0.00000000000000	-1.00000000000000
1.00000000000000	-0.56715697967715	1.00000000000000
1.00000000000000	0.56833652474461	1.00000000000000

Columns 4 through 6

1.00000000000000	0.00058832361840	0.83431427342031
1.00000000000000	-0.14485243409168	0.92108023901739
1.00000000000000	0.14607795305133	0.92108392096242

G =

0.00703896232741

The transfer functions for the three 2nd order filters are given as:

$$H_1(z) = \frac{0.00703896232741 - 0.00703896232741 z^{-2}}{1 + 0.00058832361840 z^{-1} + 0.83431427342031 z^{-2}}$$

$$H_2(z) = \frac{1 - 0.56715697967715 z^{-1} + z^{-2}}{1 - 0.14485243409168 z^{-1} + 0.92108023901739 z^{-2}}$$

$$H_3(z) = \frac{1 + 0.56833652474461 z^{-1} + z^{-2}}{1 + 0.14607795305133 z^{-1} + 0.92108392096242 z^{-2}}$$

Note that the gain has been included in the first stage of the cascade structure. These sections are cascaded in following manner.

$$H(z) = H_1(z) * H_2(z) * H_3(z)$$

The individual magnitude response of $H_1(z)$, $H_2(z)$ and $H_3(z)$ is shown in figure A.2, A.3 and A.4, respectively.

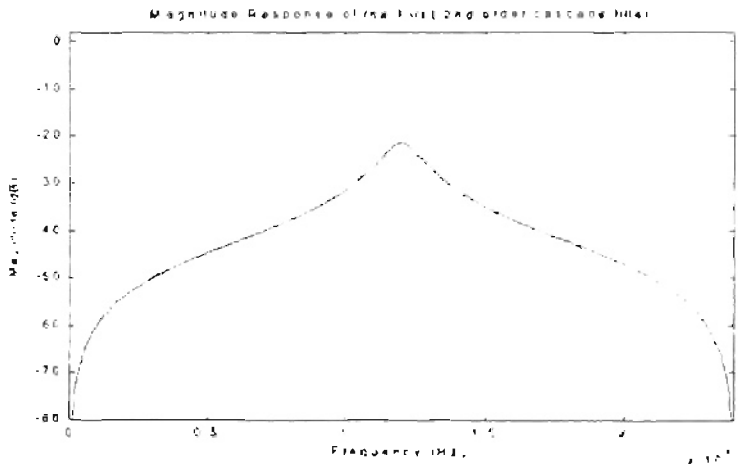


Figure A.2 Magnitude response of the first cascade section.

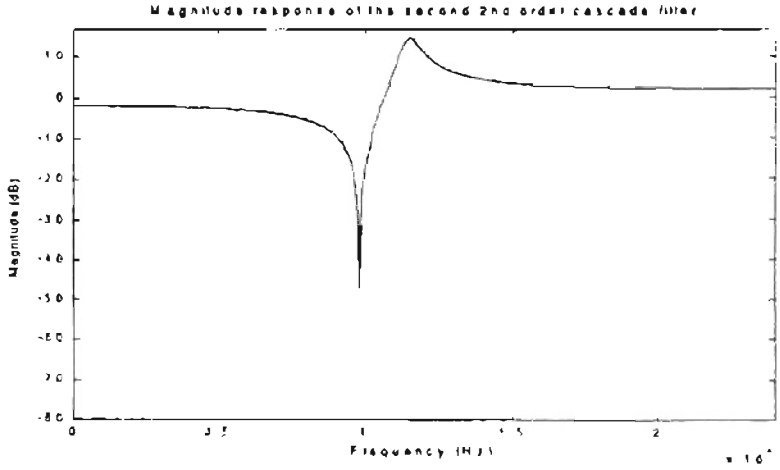


Figure A.3 Magnitude response of the second cascade section

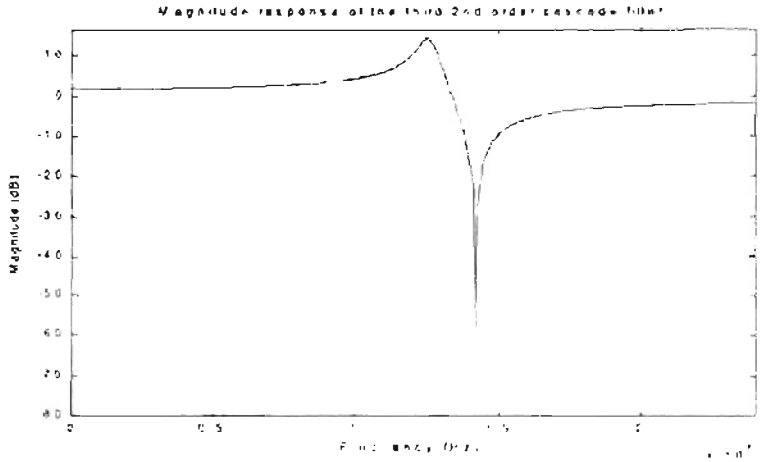


Figure A.4 Magnitude response of the third cascade section

Three 2nd order structures are cascaded to obtain the same response as the complete band-pass filter. Since multiplication in the frequency domain is convolution in the time domain, to see the response of the total filter, the impulse response of each stage is convolved with the impulse response of the following stage and is shown in figure A.5 [29].

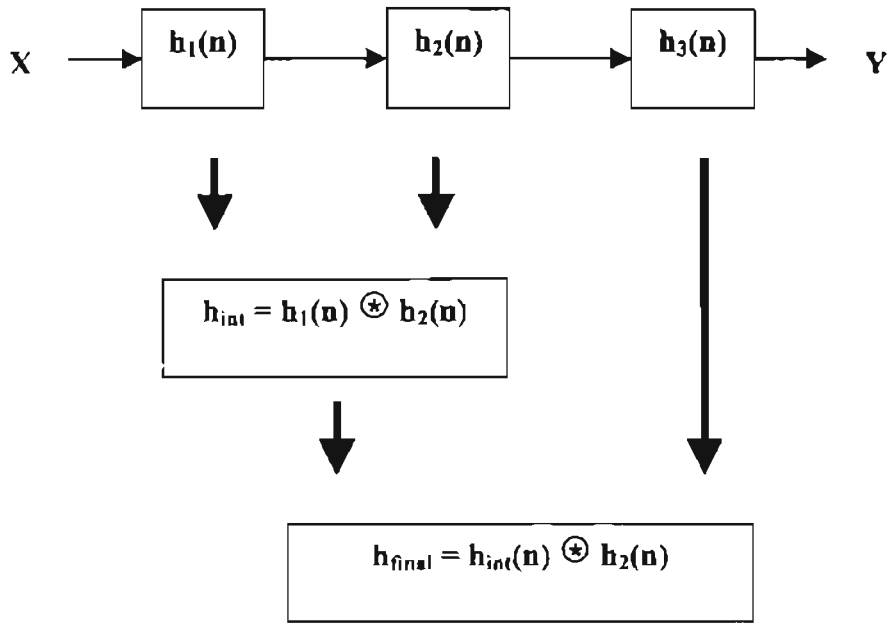


Figure A.5 Cascading three 2nd order filters

The impulse response of each section is obtained from the Matlab function:

[F w] = IMPZ(B,A)

Convolving can be done with the following function:

C = CONV(F_1 , F_2)

where F_1 and F_2 are the two impulse functions to be convolved, giving the output C. Figure A.6 shows the 6th order filter response obtained after all the sections have been cascaded together.

The coefficients produced by Matlab are represented with infinite precision, which is not practical to realize in actual hardware. We have to represent these coefficients with a finite number of bits. Therefore, at this time all the coefficients have infinite precision, thus presenting us with an ideal response of the filter. Next we go

through the exercise of truncating the feedback coefficients (poles) and scaling the zeros to make the filter coefficients realizable in hardware with a finite number of bits.

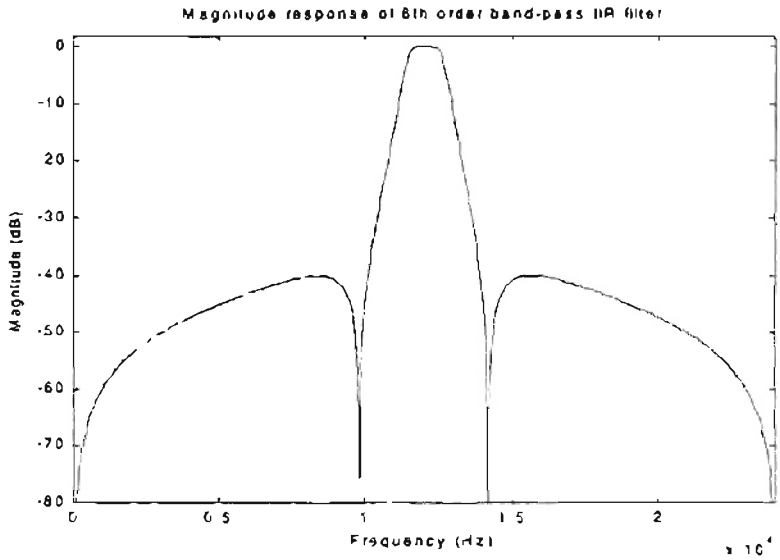


Figure A.6 Magnitude response of 6th order IIR band-pass filter.

Since the position of poles can be very critical and consequently, any alteration in their values can cause the poles to go out of the unit circle, thus rendering the system unstable, we avoid scaling the feedback coefficients. We perform bit truncation to limit the number of bits with which these feedback coefficients can be represented without causing the system to become unstable. The zero (numerator) coefficients can be scaled and truncated for two reasons: (1) to limit the number of bits required to represent each coefficient and (2) to avoid overflow in the adders. Rounding off the coefficients to B-bits is done in the following manner:

$$R\_FB = ROUND(FB_i * 2\text{^}B)/2\text{^}B$$

where B = number of bits and FB = feedback coefficient. The coefficients are considered as unsigned and the negative signs are implied in the cascaded structure adders. In this fashion we get the number representation in the following form:

COEFF = . X X X X X X X X

To perform scaling we divide all the coefficients by the maximum coefficient and then round it to B bits.

$$R\_ZB = ROUND(ZB_i / MAX(ZB) * 2 \wedge B) / 2 \wedge B$$

This process can alter the positions of zeros in the unit circle and as a result increases the magnitude response of the filter. To avoid having a scaling multiplier to bring the magnitude back to have a gain of one, we divide the zero coefficients by a number which is a factor of two, which requires no extra hardware and can be implemented by shift operation. Sometimes multiplying by the power of two does not accurately bring the magnitude back to 0dB, therefore to do this we multiply the zero coefficients by a fudge factor before rounding the bits. This process is shown as follow:

$$R\_ZB\_S = ROUND(Fudge * ZB_i / MAX(ZB) * 2\wedge B) / ((2\wedge B)*(2\wedge S))$$

where fudge = fudge factor and S = the scaling factor.

The S factor is a power of two scaling factor and can be implemented by a simple shift operation, therefore in the following calculations of coefficients, this factor will not be included and is implied only at the output of each stage of the cascade.

After rounding and scaling, we rewrite the second order filter transfer functions as shown below.

$$H_1(z) = \frac{0.76171875 - 0.76171875\,z^{-2}}{1 + 0.8359375z^{-2}}$$

$$H_2(z) = \frac{0.76171875 - 0.4296875z^{-1} + 0.76171875z^{-2}}{1 - 0.14453125z^{-1} + 0.921875z^{-2}}$$

$$H_3(z) = \frac{0.76171875 + 0.43359375z^{-1} + 0.76171875z^{-2}}{1 + 0.14453125z^{-1} + 0.921875z^{-2}}$$

The binary representation of these numbers is given in the table A.1. All the coefficients are implemented as unsigned numbers, where the signs of each coefficient are implemented in the 2nd order filter adders.

$H_1(z)$	rraa1_b1	1.0000000
	rraa1_b2	00000000
	rraa1_b3	.11010110
	rrbb1_b1	.11000011
	rrbb1_b2	00000000
	rrbb1_b3	.11000011
$H_2(z)$	rraa2_b1	1.0000000
	rraa2_b2	.00100101
	rraa2_b3	.11101100
	rrbb2_b1	.11000011
	rrbb2_b2	.01101110
	rrbb2_b3	.11000011
$H_3(z)$	rraa3_b1	1.0000000
	rraa3_b2	.00100101
	rraa3_b3	.11101100
	rrbb3_b1	.11000011
	rrbb3_b2	.01101111
	rrbb3_b3	.11000011

Table A.1 Binary representation of the coefficients

Since the filter has a fixed transfer function, we can use add and shift techniques to implement each coefficient. A better technique of constructing these constant coefficient multipliers is to use Canonic Signed Digit (CSD) number representation [19]. This number representation system promises to yield better or at the least, equal number of adders to implement these coefficients. In this scheme each coefficient (b) is represented by the following expression:

$$b = \sum_{i=0}^{n} s_i 2^i \qquad (2)$$

where $s_i \in \{0, 1, -1\}$

The CSD number system tries to reduce the number of non-zero elements by forcing no adjacent digits to be non-zero, thus having a maximum of (n+1)/2 non-zero digits in a n-bit number. Let us assume an 8-bit number in binary format shown below

B = 0 1 1 0 1 1 1 1

In this case we would require five adders to implement this number (111_d) with a simple add and shift technique. A CSD representation of this number is shown below:

C = 1 0 0 −1 0 0 0 −1

This gives same result (111_d) but utilizes only two adders to implement this in hardware, thus giving a savings of 3 adders [19]. CSD representations of all the filter coefficients are given in table A.2.

$H_1(z)$	rraa1_c1	1.0000000	1
	rraa1_c2	.0000000	0
	rraa1_c3	1.00-10-10-1	0.8359375
	rrbb1_c1	1.0-100010-1	0.76171875
	rrbb1_c2	.0000000	0
	rrbb1_c3	1.0-100010-1	0.76171875
$H_2(z)$	rraa2_c1	1.0000000	1
	rraa2_c2	.0010010 1	0.14453125
	rraa2_c3	1.000-10-10	0.921875
	rrbb2_c1	1.0-100010-1	0.76171875
	rrbb2_c2	.100-100-1	0.4296875
	rrbb2_c3	1.0-100010-1	0.76171875
$H_3(z)$	rraa3_c1	1.0000000	1
	rraa3_c2	.0010010 1	0.14453125
	rraa3_c3	1.000-10-10	0.921875
	rrbb3_c1	1.0-100010-1	0.76171875
	rrbb3_c2	.100-1000-1	0.43359375
	rrbb3_c3	1.0-100010-1	0.76171875

Table A.2 CSD representation of all the filter coefficients

Figure A.7 shows the magnitude response of the 6^{th} order filter, with finite number of bits representing each coefficient. The CSD representation of coefficients is used to implement these structures in Xilinx FPGA. Note that the magnitude of the band-pass filter is at almost 36dB. Now we divide the output of each cascaded structure by the scaling factor 2^S, which results in bringing the magnitude of the band-pass filter back to almost 0dB. The final magnitude response of the 6^{th} order band-pass filter is shown in figure A.8.

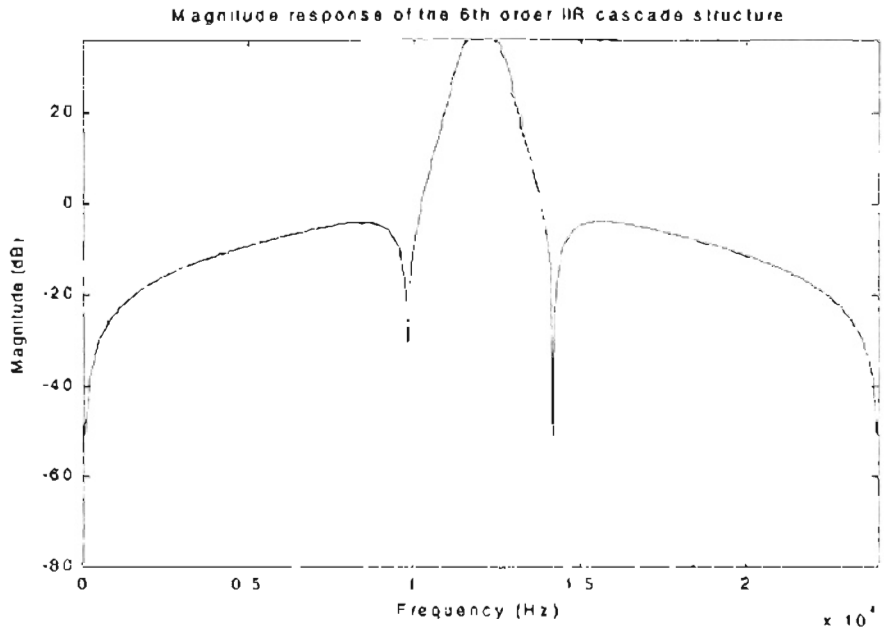


Figure A.7 Magnitude response of 6th order IIR filter with finite bits coefficients

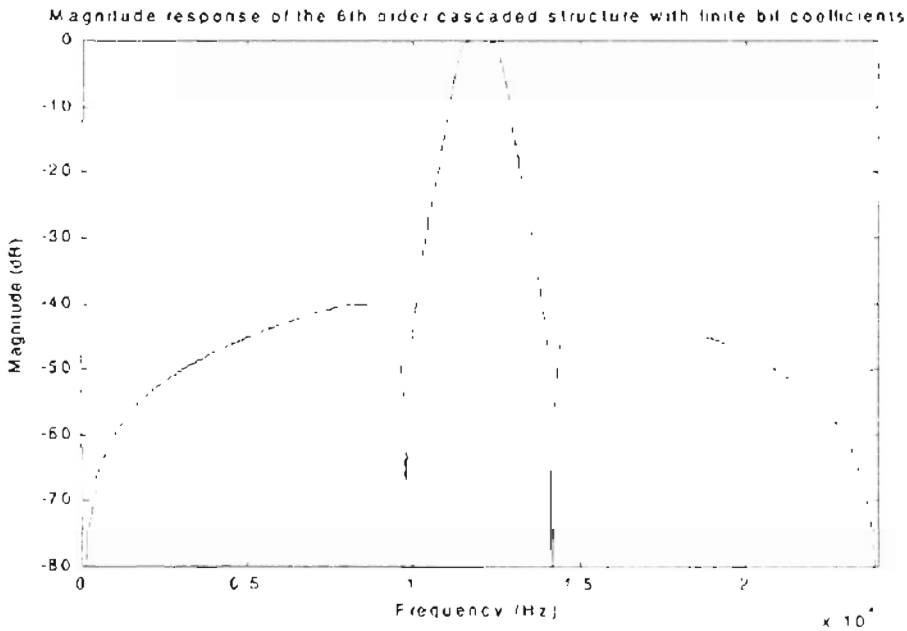


Figure A.8 Magnitude response of 6th order IIR filter with scaled output

A2. Hardware Implementation of 6th Order IIR Band-pass Filter

A Direct Form II structure implementation style was used to construct each 2nd order filter in the cascaded organization. Structures of $H_1(z)$, $H_2(z)$ and $H_3(z)$ are shown in figures A.9, A.10 and A.11, respectively.

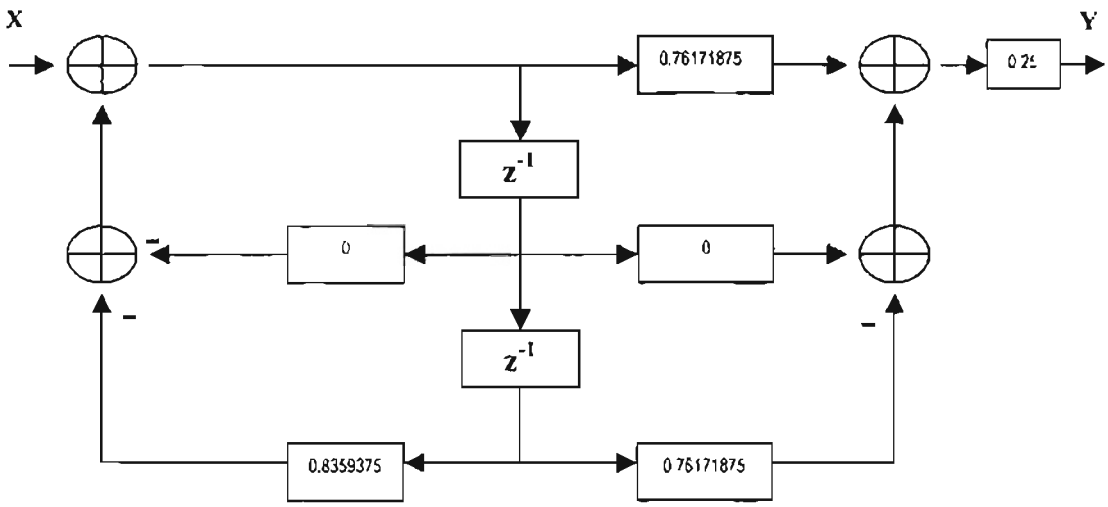


Figure A.9 Direct Form II structure of first section, 2nd order filter

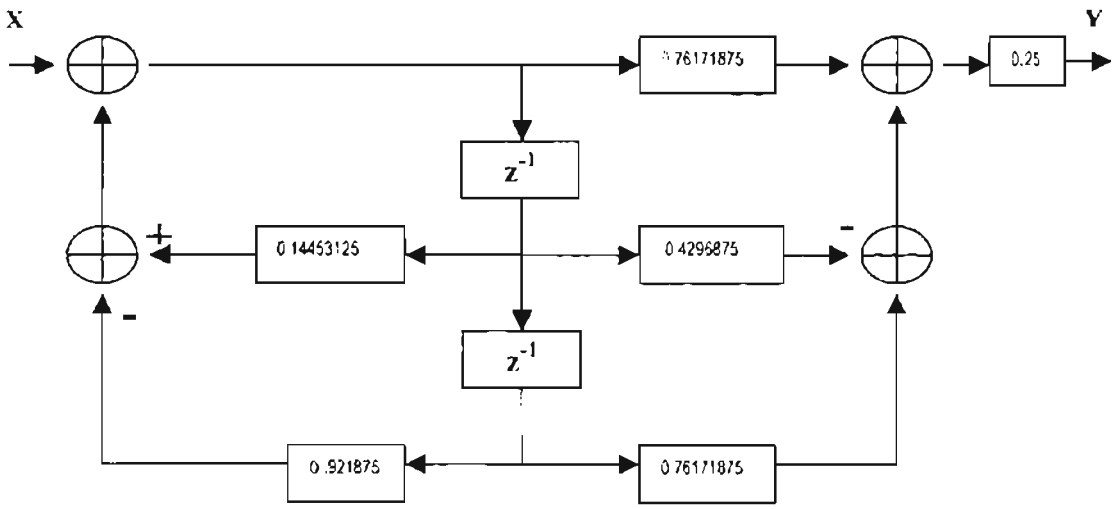


Figure A.10 Direct Form II structure of second section, 2nd order filter

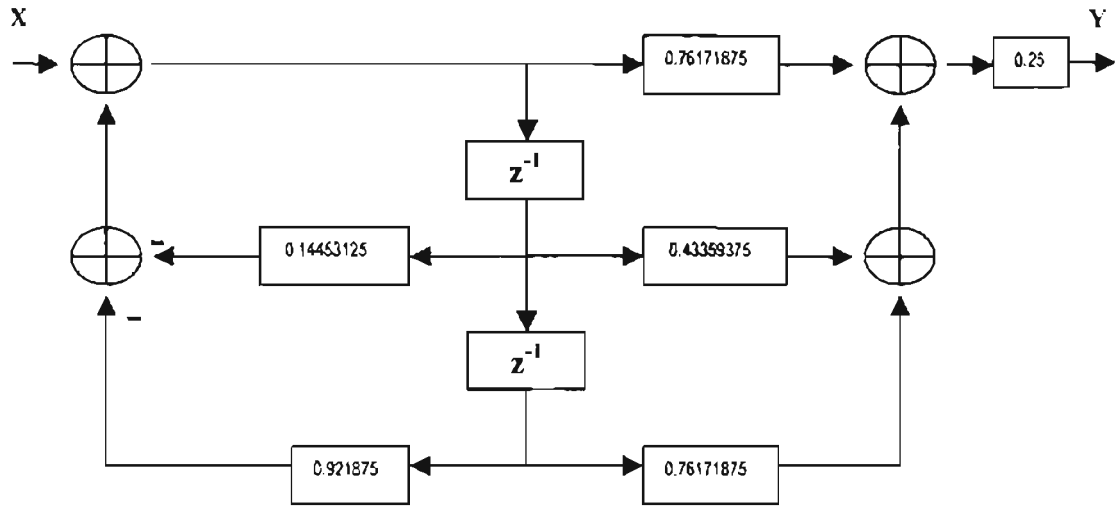


Figure A.11 Direct Form II structure of third section, 2nd order filter

Note that zero the coefficients have been included in figure A.9, which can be optimized either by the circuit synthesizer or can be left out, resulting in no connection. Also we divide the output of each structure by a scaling factor S (where S=4 in our case) to bring the magnitude response of the structure to 0dB. This configuration needs no extra hardware and can be done by two right shifts. The adders in all these structures are 16-bit wide and the delays are 8-bit wide, this technique is used to preserve as many bits as possible before truncation becomes eminent. To reduce the effect of quantization error, truncation is performed where it is essential. Since the outputs of the splitter circuit is 16-bits, therefore the input to all the 2nd order stages have been kept at 16-bits, and consequently the output of first two stages is also 16-bits. The output of the last stage has to be truncated to 8-bits since the input to the combiner quadrant multipliers is 8-bits wide. This configuration is shown in figure A.12.

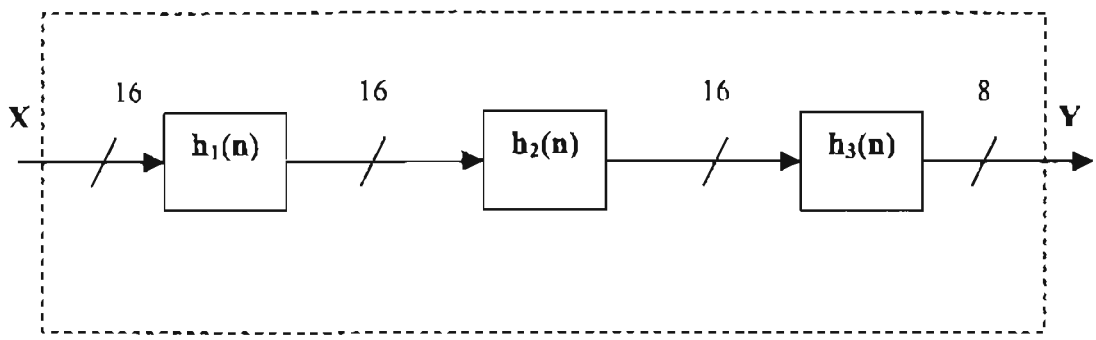


Figure A.12 Cascade structure with elaborated internal bus widths.

Now using table A.2 we can build the hardware for the constant coefficient multipliers. Since the implementation is done using fixed-point arithmetic, we have to make sure that all the binary points are aligned with each other. From table A.2 we can see that the maximum shift is 8-bits and also the input to our constant multipliers is 8-bits, therefore we will keep the outputs of each constant multiplier at 16-bits, with one bit of integer and 15-bits of fraction. Therefore, our binary point is after the most significant digit. Figure A.13 shows the multiplier structure for the feedback coefficient of the first cascade structure (aarr1_c3). Note, that a zero is appended at the least significant bit, to make the output 16-bit wide, to align the binary point with other constant coefficients multiplier structures. A.14 – A.18 shows the implementation structures for rest of the coefficients.

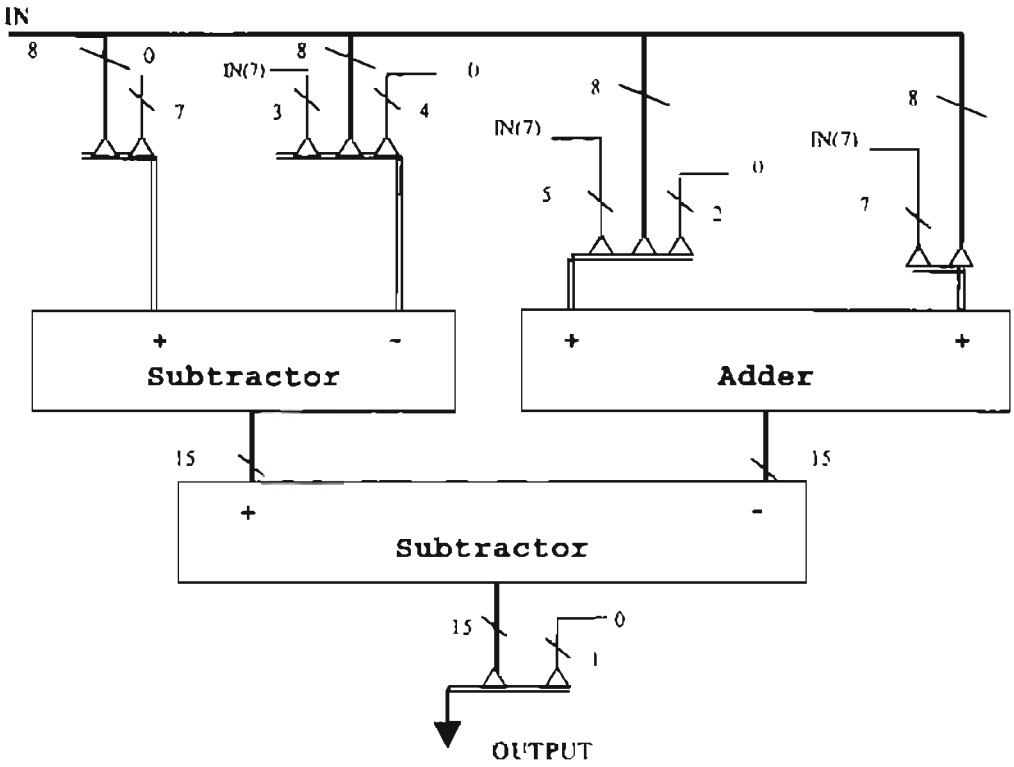


Figure A.13 'mul_c3' constant coefficient multiplier structure

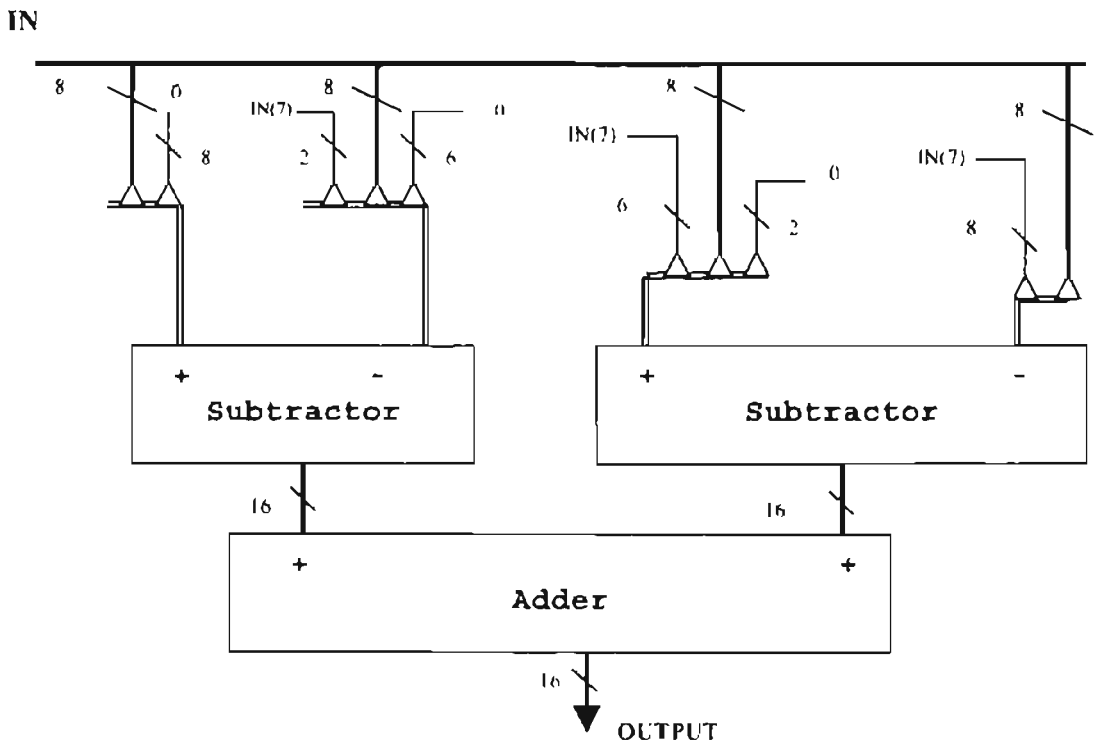


Figure A.14 'mul_c1' constant coefficient multiplier structure

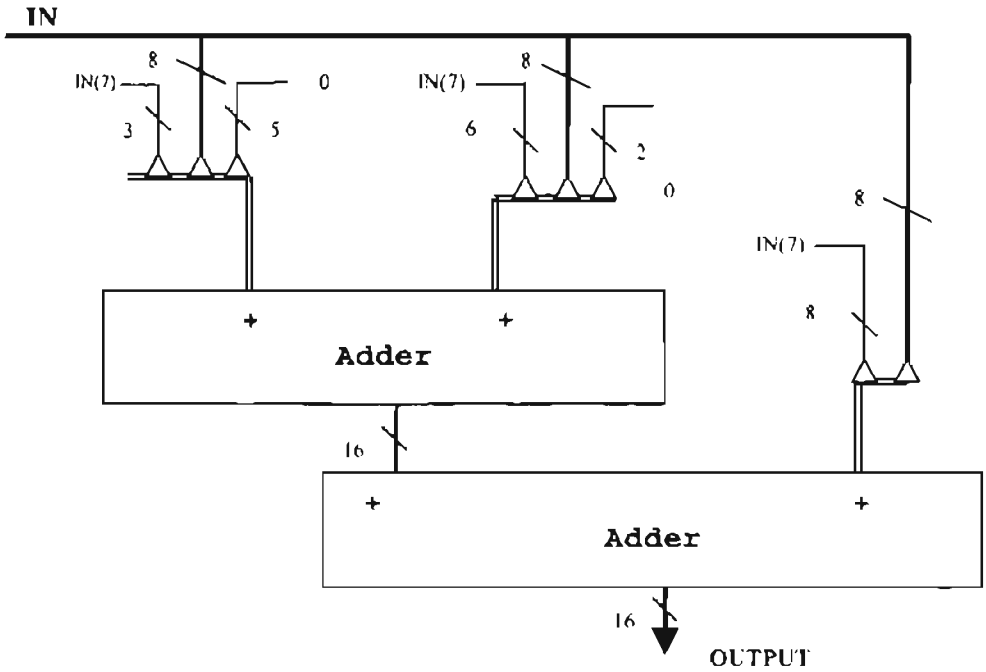


Figure A.15 'rraa2_c2' constant coefficient multiplier structure

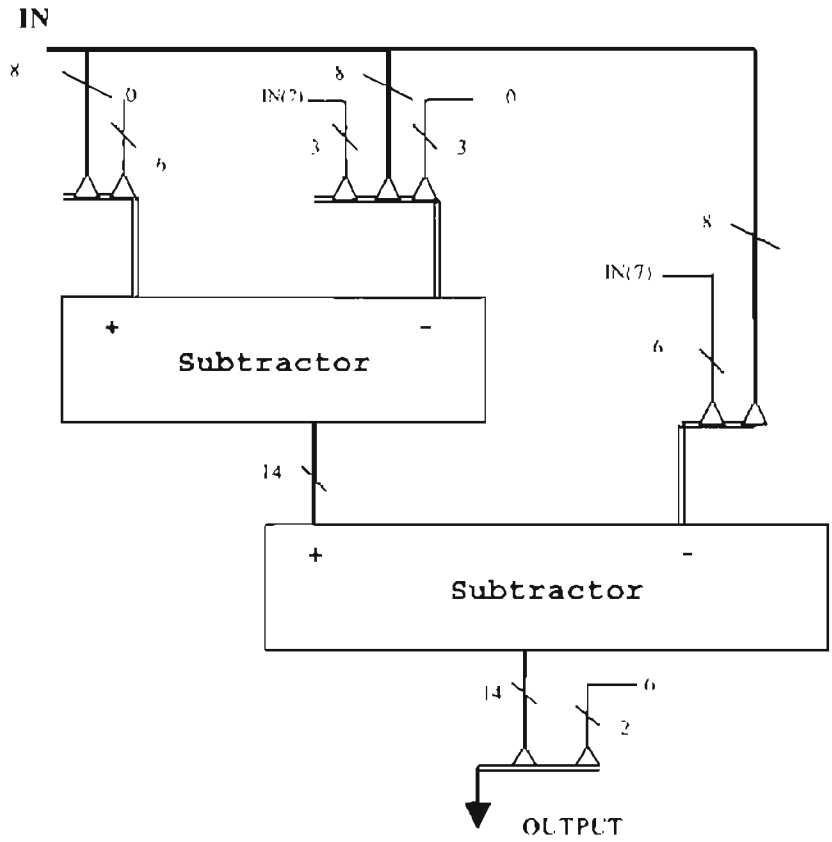


Figure A.16 'rraa2_c3' constant coefficient multiplier structure

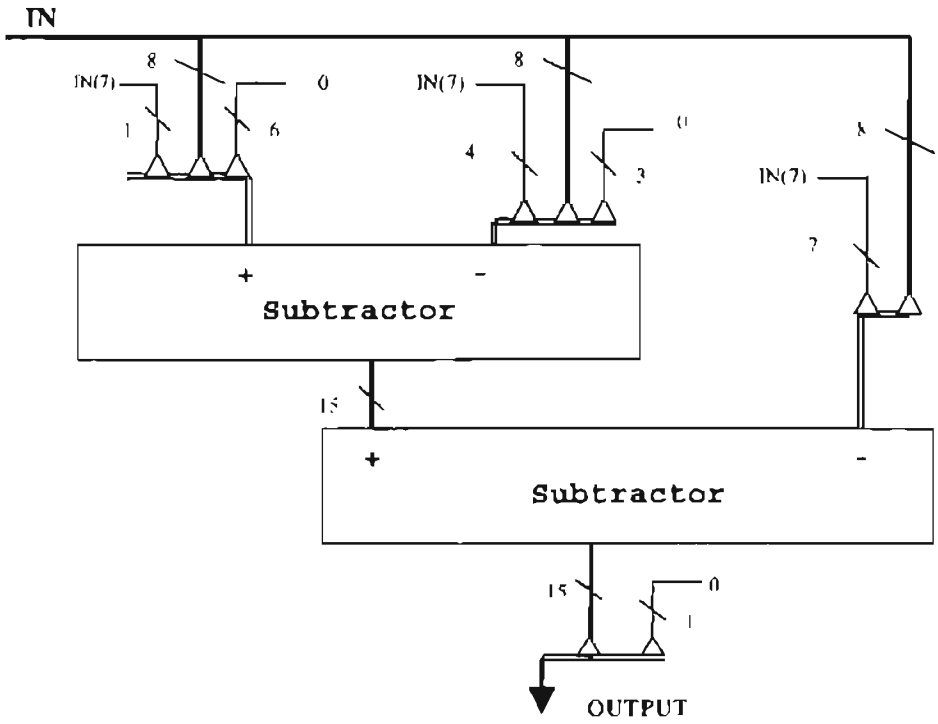


Figure A.17 'rrbb2_c2' constant coefficient multiplier structure

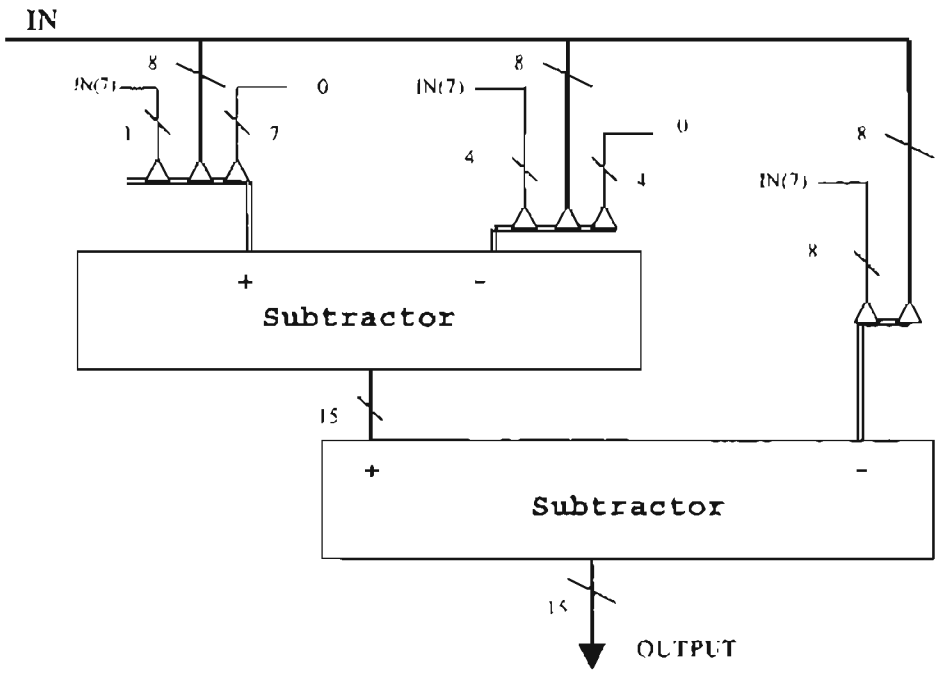


Figure A.18 'rrbb3_c2' constant coefficient multiplier structure.

VITA

Asad Azam

Candidate for the Degree of

Master of Science

Thesis: HARDWARE IMPLEMENTATION OF TUNABLE HETERODYNE BAND-PASS FILTERS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Karachi, Pakistan, on May 30th, 1974, the son of Azam and Anwar.

Education: Graduated from High school in Pakistan; received an associate degree from an affiliated school in Subang Jaya, Malaysia, as part of Oklahoma State University twinning program; received a Bachelor's of Science degree in Electrical Engineering with Computer option from Oklahoma State University, Stillwater, Oklahoma in May 1999. Completed the requirements for the Master of Science degree with a major in Electrical Engineering at Oklahoma State University in August, 2001.

Experience: During the Master degree worked as a Research and Teaching Assistant with the Department of Electrical Engineering at Oklahoma State University, 1999 to present. Worked as a summer intern employed by Quantum Corporation, Shrewsbury, MA, May 2000 to August 2000.