FPGA IMPLEMENTATION OF AM/FM

Project report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Shruti Patial (121024)

Aditi Prabha Gautam (121025)

Dhruv Raj Singh Ujjawal (121027)

UNDER THE GUIDANCE OF

Dr. SHRUTI JAIN

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May , 2016

CONTENTS

DECLARATION

We hereby declare that the work reported in the B-Tech thesis entitled **" FPGA Implementation of AM/FM"** submitted at **Jaypee University of Information Technology, Waknaghat India,** is an authentic record of our work carried out under the supervision of **Dr. Shruti Jain and Dr. Neeru Sharma**. We have not submitted this work elsewhere for any other degree or diploma.

Shruti Patial (121024) [illegible]

Aditi Prabha Gautam (121025) Aditi

Dhruv Raj Singh Ujjawal (121027) [illegible]

Department of Electronics and Communication Engineering

Jaypee University of Information Technology, Waknaghat , India

Date: 24th May,2016

CERTIFICATE

This is to certify that the work reported in the B-Tech. thesis entitled **"FPGA Implementation of AM/FM**", submitted by **Shruti Patial (121024), Aditi Prabha Gautam (121025) and Dhruv Raj Singh (121027)** at **Jaypee University of Information Technology, Waknaghat , India,** is a bonafide record of their original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY JUIT * WAKNAGHAT *

[illegible]

[illegible]
25/5/16

(Signature of Supervisor)

Dr. Shruti Jain

Dept. of ECE, JUIT

24th May,2016.

(Signature of Co-Supervisor)

Dr. Neeru Sharma

Dept. of ECE,JUIT

24th May,2016.

ACKNOWLEDGEMENT

We would like to thank Prof. (Dr.) Sunil Bhooshan, Head of Dept. Electronics and Communication, for providing us the opportunity of working on a project.

We are highly indebted to Dr. Shruti Jain and Dr. Neeru Sharma for motivating and enlightening us for our project work. We thank you for being a constant support throughout, without whose valuable guidance and insights, this project would not be a complete one. We offer you our sincere gratitude to you for instructing and directing us through thick and thin.

We would also like to acknowledge Mr. Mohan Sharma for helping us in the labs. Thank you for being there and helping us in and out.

LIST OF ACROYNMS

ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
DAC	Digital to Analog Converter
DSP	Digital Signal Processor
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
IIR	Infinite Impulse Response
LPF	Low Pass Filter
SDR	Software Defined Radio

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The wireless market of the future will demand inexpensive hardware, expandability, interoperability, and the implementation of advanced signal processing functions—i.e. a software radio. The purpose of this study is to evaluate a low-cost Software Defined Radio (SDR). The AM/FM radio receiver will later be implemented using a Field Programmable Gate Array (FPGA). Using FPGA as the hardware platform, this approach brings together the flexibility of software and the speed of hardware to build a reconfigurable system, i.e. the behavior of the radio receiver can be changed by simply modifying the software without new hardware.

CHAPTER 1

LITERATURE SURVEY

1.1 Introduction

In the past radio receivers were designed with analog circuitry. This inherently has the same problems that all analog circuits have. That is, they are susceptible to temperature variations, electrical noise, component aging, and they are complicated and inflexible. Initially, as digital circuits and processors were developed, they were not useful for radio or any high frequency circuitry since they operated at low frequencies and their transistor density was not enough for the signal processing needed in receivers. However, recent advancement in semiconductor technology has made it possible to process high speed communication signals in radio systems using as much digital technology as possible. This makes the system very flexible and adaptive. Such a technology is called Software Defined Radio (SDR). With its re-configurability, the potential applications of SDR are numerous. One example is in wireless communication industry, where due to the many communication standards being used around the world, there is a need to build multimode handsets capable of connectivity irrespective of the underlying network technology used. SDR technology could be applied here as radio functions are implemented in software, multiple software modules implementing different standards can co-exist in a handset. An appropriate software module can be chosen to run depending on the network requirements.[1]

The aim of this project is to develop an SDR using MATLAB-Simulink and Xilinx system generator to implement design on Field Programmable Gate Array (FPGA) as the hardware platform. Unlike traditional analog receivers, which are built from analog components, design receivers convert analog signal into digital representation, and employ digital processing techniques to process the digital data for extracting the desired information. The extracted digital information (a radio channel in this case) is then converted back to analog platform for listening purposes.

In developing the system, we aimed to investigate different techniques for designing digital radio receivers, from which suitable techniques would be chosen for implementing the SDR. Several design models created in Simulink for implementing the receiver in FPGA would be developed and simulated, giving insight analysis into their performances as well as their efficiency in term of hardware usage.

Finally, these models would be translated into software (Hardware Description Language) used for synthesizing the system into a FPGA platform. The ultimate goal of this project will be having a fully working AM/FM Digital Receiver System residing in a FPGA board allowing users to tune in and listen to any of the available AM or FM channels.

Objectives[2]

- Global mobility—The need for transparency, i.e., the ability of radios to operate with some, preferably all, of these standards in different geographical regions of the world has fostered the growth of the software-defined radio concept.
- Compactness and power efficiency—The software-defined radio approach results in a compact and, in some cases, a power efficient design, especially as the number of systems increases, since the same piece of hardware is reused. to implement multiple systems and interfaces.
- Ease of manufacture—RF components are notoriously hard to standardize, however, digitization of the signal early in the receiver chain can result in a design that incorporates significantly fewer parts, meaning a reduced inventory.
- Ease of upgrades—A flexible architecture allows for improvements and additional functionality without the expense of recalling all the units or replacing the user terminals. Software-defined radio allows the new devices to interface seamlessly.

- Multi functionality—For instance, a Bluetooth-enabled fax machine may be able to send a fax to a nearby laptop computer equipped with a software-defined radio that supports the Bluetooth interface..

1.2 Overview

This project exemplifies to some extent the concepts, design and implementation of a broader area of digital technology called Software Defined Radio (SDR). This section presents a broad view overview of software Defined Radio including its background, applications, and a comparison of commonly used hardware platforms for implementing SDR.

Traditional analog radio receivers and transmitters consists of dedicated analog circuits for filtering, tuning and modulating/demodulating a specific type of waveform. These hardware based radio systems are inflexible and hard to modify if changes are to be made to its fundamental characteristics such as modulation/demodulation types. To make the system more flexible, SDR technology facilitates implementation of some of these functions in software. This results in reconfigurable software radio systems where changes to its fundamental characteristics can be made simply by modifying the software, whereas a complete hardware based radio system would require hardware modification in order to change these parameters.

SDR technology has potential applications in all areas of radio communications and broadcasting. To highlight the significance of SDR technology, consider the following problems faced by the wireless communications industry due to implementation of network infrastructure and terminals completely in hardware[3-4]:

- The constant evolution of wireless network standards from 2G to 3G and further to 4G problems for subscribers, network operators and equipment vendors. Users are forced to buy new handsets every time a new generation is deployed. Network operators face with the problem of migrating from one generation to another due to large number of subscribers using legacy handsets that may be incompatible with newer generation network.

- The air interface and link-layer protocols differ across different continents, for example, European wireless networks are mainly GSM/TDMA while US networks are predominantly CDMA. This problem has inhibited the deployment of global roaming facilities and thus causing inconvenience to subscribers who travel frequently between countries.

SDR technology promises to solve these problems as it enables implementation of radio functions in networking infrastructure and subscribers terminals as software modules running on generic hardware platforms. This relieves the cost and complexity of migrating network from one generation to another since the migration would only involve a software upgrade.

Furthermore, since radio functions are implemented in software, multiple software modules implementing different standards can co-exist in the equipments and handsets. An appropriate software module can be chosen to run (either by user or by the network) depending on the network requirements. This allows building of multi-mode handsets and equipment resulting in universal connectivity irrespective of the underlying network technology used.

As mentioned above, in Software Defined Radio technology, radio functions are implemented in software running on digital signal processing hardware platforms. The three most commonly used platforms are DSP's (Digital Signal Processor), ASIC's (Application Specific Integrated Circuit) and FPGA's (Field programmable Gate Array).

Digital Signal Processor does signal processing by fetching instructions and data from memory, does operations and stores the results back to memory, just like a regular CPU.

The difference between a DSP chip and a CPU chip is that a DSP chip usually has a block that does high-speed signal processing, especially a block called MAC (Multiply And Accumulate). By calling different routines in memory, a DSP chip can be reconfigured to perform various functions.

ASIC (Application Specific Integrated Circuit) is an integrated circuit that is designed to perform a fixed specific task. Examples of signal-processing specific ASIC's are DDC (Digital Down Convertor) chip and digital filter chips. The disadvantage of ASIC is that its functionalities are fixed and thus cannot be changed by the user.

FPGA (Field Programmable Gate Array) is able to perform any task by mapping the task to the hardware. One of the advantages of FPGA is its reconfigurability capability that ASIC does not have. Reconfigurability is a feature, which enables FPGA to realize any user hardware by changing the configuration data on a chip as many times as needed.

In Summary, with its many advantages, FPGA has become key components in implementing high performance digital signal processing systems. In this project for implementing an AM/FM Digital Radio Receiver.

1.3 Software Defined Radio (SDR)[5,23]

Software Defined Radio defines SDR technology as "radios that provide software control of a variety of modulation techniques, wide-band or narrow-band operation, communications security functions (such as hopping) and waveform requirements of current & evolving standards over a broad frequency range."

There are numerous definitions of Software Defined Radio in existence, all of which are not totally consistent with each other. The federal Communications Commission (FCC) defines SDR as "generation of radio equipment that can be reprogrammed quickly to transmit and receive on any frequency with a wide range of frequencies, using virtually any transmission format and any set of standards."

In contrary, the SDR Forum, as an international, non-profit organization promoting the development of SDR, offers a broader definition: "Software defined radio is a collection of hardware and software technologies that enable reconfigurable systems architectures for wireless networks and user terminals." One reason for the advent of several inconsistent definitions is probably due to the broad and complex nature of technology itself, and the variety of possible means for implementation of SDR systems.

Software defined radio (SDR) is receiving enormous recognition as the next evolutionary stage of wireless technology, getting support from governmental agencies as well as civil and commercial entities. The numerous benefits provided by SDR have created widespread interest, and the triumphal procession of software based radio systems now only remains a question of time.

1.3.1 Fundamentals of SDR :

Independent of what kind of radio we are talking, a radio can always be broken down into three fundamental components[6]:

1. RF Part
2. Signal Processing
3. User Interface

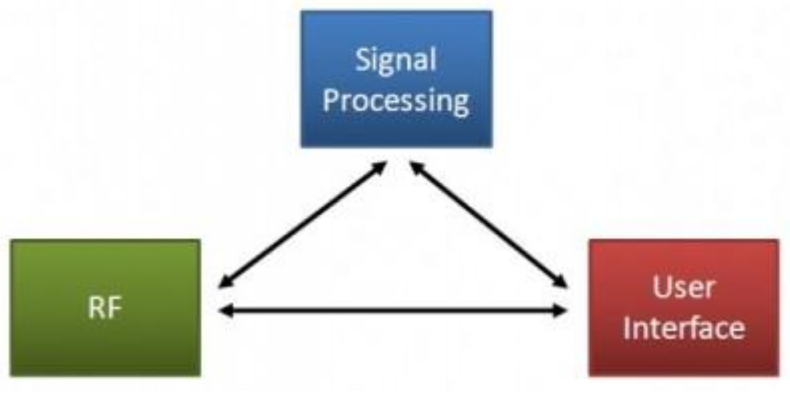


Figure 1.1 : The three basic Components of a Radio

All three components interact with each other.

- The **RF component** transforms high frequency signals into baseband signals or vice versa. Parts of the RF component are filters, mixers, amplifiers, oscillators, etc.

- The baseband signal is processed / generated in the **Signal Processing component**. In this component we will find functionality taking charge of modulation, demodulation, NF filtering, noise reduction, etc.

- The **User Interface** (UI) is the command and control unit of the radio. It’s the frontend to the (human) operator. The user interface might consist of buttons, knobs, LEDs or just a PC screen.

The Classic Radio : The classic radio is the kind of transceiver which we have in our shacks. It might be a FT1000MP, an IC706 or a Drake TR-7, just to mention a few of them. Over the last decades, the features and user interfaces have slightly changed, but the concept has remained the same: All-in-one-box.

All components are integrated in a single housing. With the increasing level of integration (and use of digital technologies) the **physical dependency** between the components increased as well. Even if some transceivers come with detachable control units (user interfaces), they are not interoperable with interfaces from other vendors. The schematic below illustrates the dependencies which come with this “All-in-one-box” concept of our today’s transceivers.

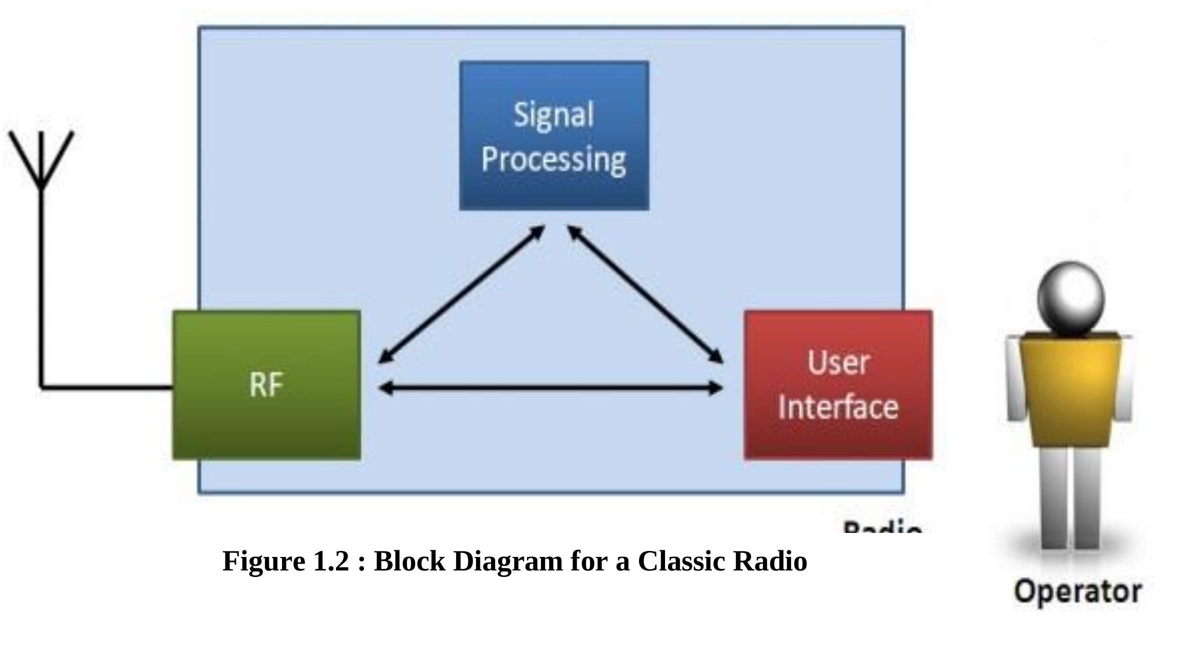


Figure 1.2 : Block Diagram for a Classic Radio

The SDR approach

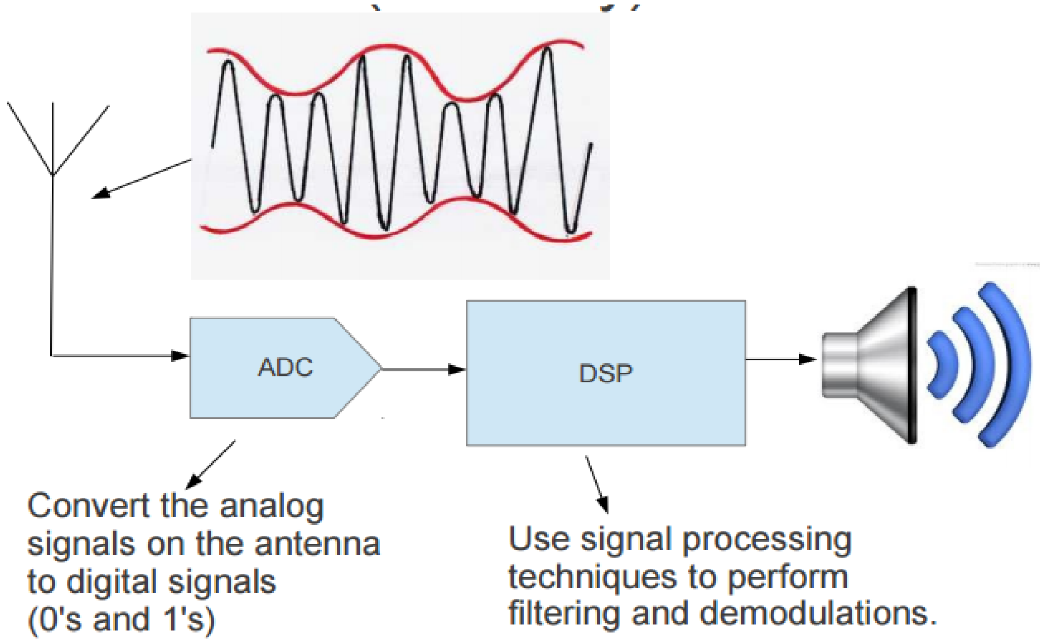
As the classic radio, the software defined radio consists also of the three fundamental components RF, Signal Processing and User Interface. However, the **tremendous difference** is the **physical** and **logical independency** between the three components.

Physical Independency means that each of the three components can be realized in an individual housing. There is no reason why the components would need to be located at same site. Geographically they could be easily separated by 10.000 miles or more without any degradation in performance.

Logical Independency means that each particular component can be designed and changed without affecting the other components. Especially the digital domain allows us to have various independent instances of the same component. Using the three components, the proposed schematic of the SDR is shown in Figure 1.4.

How does an SDR work?

Theoretically:


ADC
DSP
Convert the analog signals on the antenna to digital signals (0's and 1's)
Use signal processing techniques to perform filtering and demodulations.

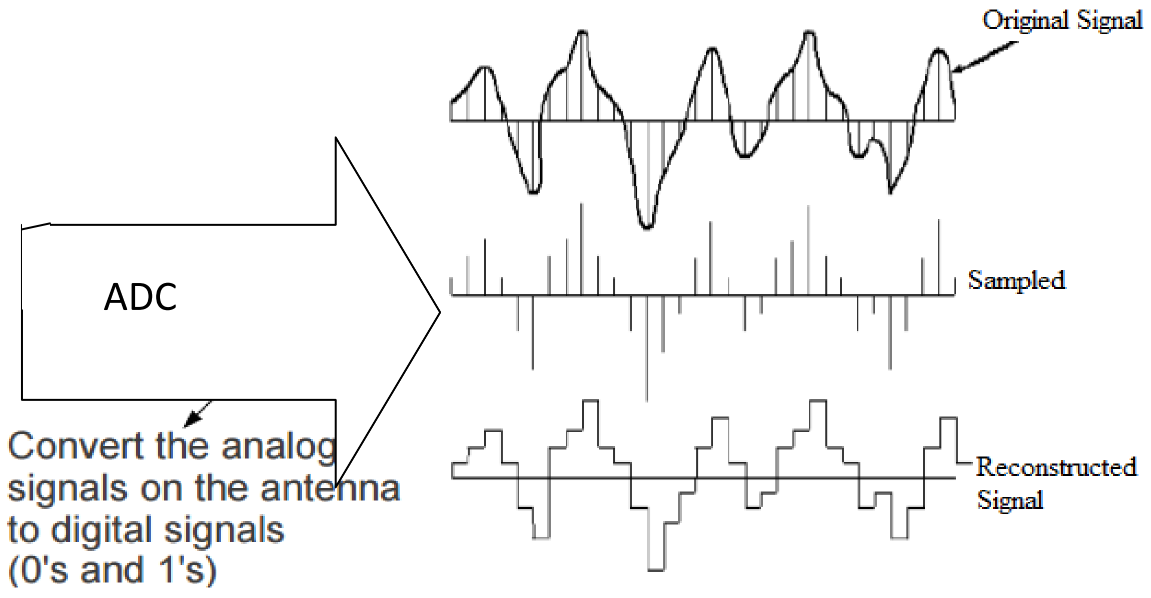

Original Signal
ADC
Sampled
Reconstructed Signal
Convert the analog signals on the antenna to digital signals (0's and 1's)

Figure 1.3 : Output of an Ideal SDR

Practically:

In practical implementation, a radio receiver can be broken down into a number of components and comprises of filters, A/D and D/A converters, mixer, local oscillator and an antenna as shown in Figure 1.4[7].

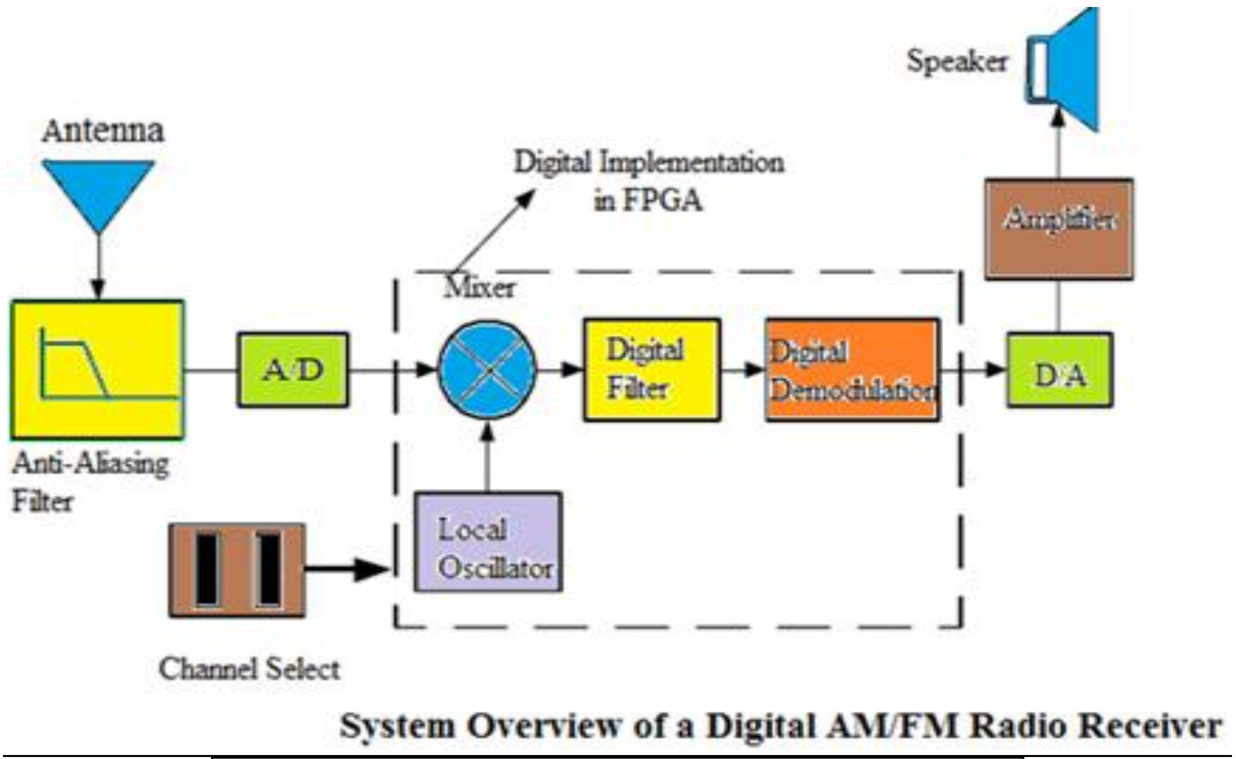


Figure 1.4 : Basic Block Diagram of a radio receiver

- **Antenna :** An antenna (plural antennae or antennas), or aerial, is an electrical device which converts electric power into radio waves, and vice versa. It is usually used with a radio transmitter or a radio receiver.

- **Anti Aliasing Filter :** Anti aliasing filter is filter which removes all signals out of required band so that the signal satisfies the sampling theorem. Aliasing refers to an effect which causes different signals to become indistinguishable when sampled. Nyquist's theorem says that any signal can be represented by discrete samples if the sampling rate is at least twice the bandwidth of the signal. For example, if an A/D converter samples analog input at 50

MHz, then input signal must have bandwidth of less than 25 MHz. Let's consider what happens if we violate the Nyquist sampling theorem. Figure 3 shows the frequency spectrum of a system sampling at frequency fs. Any signal below half the sampling frequency fs/2 such as the one at fo can be represented by digital samples. However, if input signal has components at higher frequency than fs/2, such as fa then after sampling, this component will be folded back into the [0,fs/2] region resulting in an image at frequency fs-fa. Henceforth, this image component cannot be distinguished from a true frequency which may be present at the same frequency.

- **A/D Converter :** An analog-to-digital converter (ADC, A/D, or A to D) is a device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude. The conversion involves quantization of the input, so it necessarily introduces a small amount of error. Furthermore, instead of continuously performing the conversion, an ADC does the conversion periodically, sampling the input. The result is a sequence of digital values that have been converted from a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal. The actual bandwidth of an ADC is characterized primarily by its sampling rate, and to a lesser extent by how it handles errors such as aliasing.

- **Mixer :** An electronic mixer is a device that combines two or more electrical or electronic signals into one or two composite output signals. The Digital Mixer is simply a 2 inputs multiplier which outputs the product of two digital samples. Digital output samples from the A/D are mathematically multiplied with digital samples of a sine (or cosine) signal from the local oscillator. The input samples from the A/D and sine (cosine) samples from the Local Oscillator are generated at the same rate fs (the sampling frequency of the A/D). The multiplication is sample-by-sample and thus the mixer produces samples at the same rate of fs. Note that unlike analog mixer which produces many unwanted mixer products, the digital mixer only produces the sum and the difference frequency signals.

- **Digital Filter :** In signal processing, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. Once the RF signal has been shifted down to DC level, a low pass filter can be used to filter out unwanted high frequency components. The filter accepts digital samples from the output of the Mixer at the sampling frequency, fs. It employs digital processing techniques to implement an FIR (finite impulse response) filter transfer function. The filter passes all frequencies from 0 up to a cutoff frequency equal to the bandwidth of the message signal of interest and rejects all frequency above that cutoff frequency.

- **Digital Demodulation :** Demodulation is the act of extracting the original information-bearing signal from a modulated carrier wave. A demodulator is an electronic circuit (or computer program in a software-defined radio) that is used to recover the information content from the modulated carrier wave.

AM Digital Demodulator : In amplitude modulation (AM), the message signal m(t) is impressed on the amplitude of the carrier signal Ac cos(wct). A conventional Amplitude Modulation (AM) signal is defined mathematically as[8]

$$s(t) = A_c[1 + a.m(t)]cos(2\pi f_c t) \qquad \text{(i)}$$

Where fc is the carrier frequency, |*m (t)*|<1, and a is the modulation index.

Function *m(t)* represents the baseband signal that contains the information to be conveyed. If *m(t)* is a deterministic signal with Fourier transform (spectrum) *M(f),* the spectrum of the AM signal *s(t)* is

$$s(t) = A_c[a.M(f - f_c) + \delta(f - f_c) + a.M(f + f_c) + \delta(f + f_c)]/2 \qquad \text{(ii)}$$

AM band frequency ranges from 526.5 KHz to 1606.5 KHz. To avoid aliasing, the ADC (analog-to-digital converter) needs to sample signal at sampling rate at least twice the highest frequency in the AM spectrum, thus a sampling rate of 4 MHz was chosen for the ADC to allow for some margin.

As mentioned previously, an ADC is used to digitize the analog signal received from the antenna into digital representation. S_{AM} (n) represents the digital samples of the modulated AM signal whose spectrum contains all AM channels. These samples are multiplied with digital sinusoidal sample

($cos(2\pi f_c n)$)of a local oscillator whose frequency is that of the channel of interest. In other words, the frequency of the sinusoidal signal generated by the local oscillator is dependent upon the channel selected by the user. The effect of this multiplication is to shift the spectrum of the AM signal down to DC level so that a low pass filter having a fixed cut-off frequency can be used to pass the baseband message of the radio channel of interest.
In digital sample representation, we have:

$$S_{AM} = A_c[1 + am(n)]cos(2\pi f_c n) \qquad \text{(iii)}$$

$$G(n) = A_c[1 + am(n)]cos(2\pi f_c n) = A_c[1 + am(n)]\,[cos(2\pi(2f_c)n) + 1]/2 \qquad \text{(iv)}$$

The low pass filter after the mixer is design to remove all but the baseband signals. Consequently, the output of the demodulator will be

$$S(n)_{baseband} = A_c[1 + am(n)]\;/2 \qquad \text{(v)}$$

which has an AC component proportional to the baseband message signal m(n). Thus we have successfully demodulated the AM signal.

FM Digital Demodulator : FM modulation is a type of angle modulated signal, a conventional frequency modulated signal is given by the following equation

$$S_{FM}(t) = A_c\, cos(2\pi f_c\ \ t + \phi(t)\ ) \qquad \text{(vi)}$$

in which the message signal m(t) is transferred as the rate of change of the phase and is defined by

$$\phi(t) = 2\pi K_f \int_{\infty}^{t} m(\tau)d\tau \qquad \text{(vii)}$$

where fc is the carrier frequency, k is the frequency deviation constant. Hence for FM modulation, it can be shown that the instantaneous frequency of the signal is changed by the message signal according to the following equation

$$f_i(t) = f_c + k_f m(t) \qquad \text{(viii)}$$

There is no direct connection between the spectrum of the message signal and the spectrum of the modulated signal. For a sinusoidal message signal like

$$m(t) = acos(2\pi f_m t) \qquad \text{(ix)}$$

The modulated signal can be expressed as

$$S_{FM}(t) = A_c cos[2\pi f_c t + \{K_f a(sin(2\pi f_m t))/f_m\}] = A_c cos(2\pi f_c t + \beta sin(2\pi f_m t)) \qquad \text{(x)}$$

where , $\beta = K_f a/f_m$ is the modulation index. In general, the actual bandwidth of FM modulated signal is infinite. However, the amplitude of the spectrum declines fast outside bandwidth around the carrier frequency.

The effective bandwidth of a FM modulated signal, which contains 98% of the signal power, where is the modulation index and fm is the highest frequency of the message signal.

$$B_c = 2(\beta + 1) f_m \qquad \text{(xi)}$$

- **Local Oscillator :** In electronics, a local oscillator (LO) is an electronic oscillator used with a mixer to change the frequency of a signal. This frequency conversion process, also called heterodyning, produces the sum and difference frequencies from the frequency of the local oscillator and frequency of the input signal. Processing a signal at a fixed frequency gives a radio receiver improved performance.

  The Local Oscillator generates digital samples of a sine (or cosine) wave having the same frequency as that of the desired radio channel. The output frequency of the local oscillator is dependent upon the channel selected by the user. The Local Oscillator produces a sinusoidal sample at exactly every output sample of the A/D converter, therefore it is driven by the same sampling clock of the A/D converter.

- **D/A Converter :** In electronics, a digital-to-analog converter (DAC, D/A, D2A or D-to-A) is a function that converts digital data (usually binary) into an analog signal (current, voltage, or electric charge). Unlike analog signals, digital data can be transmitted, manipulated, and stored without degradation, albeit with more complex equipment. But a DAC is needed to convert the digital signal to analog to drive an earphone or loudspeaker amplifier in order to produce sound (analog air pressure waves).

- **Amplifier :** An amplifier, electronic amplifier or (informally) amp is an electronic device that can increase the power of a signal. It does this by taking energy from a power supply and controlling the output to match the input signal shape but with a larger amplitude. In this sense, an amplifier modulates the output of the power supply to make the output signal stronger than the input signal. An amplifier is effectively the opposite of an attenuator: while an amplifier provides gain, an attenuator provides loss.

1.3.2 Features of SDR [9] Following are the key features of SDR technology:

- **Reconfigurability:** SDR allows co-existence of multiple software modules implementing different standards on the same system allowing dynamic configuration of the system by just selecting the appropriate software module to run. The dynamic configuration is possible both in handsets as well as infrastructure equipment. The wireless network infrastructure can reconfigure itself to subscriber's handset type or the subscriber's handset can reconfigure itself to network type. SDR technology facilitates implementation of future proof, multi-service, multi-mode, multi-band, multi-standard terminals and infrastructure equipment.

- **Universal Connectivity:** SDR enables of air interference standards as software modules and multiple instances of such modules that implement different standards can co-exist in infrastructure equipment and handsets. This helps in realizing global roaming facility. If the terminal is incompatible with the network technology in a particular region, an appropriate software module needs to be installed onto the handset (possibly over the air) resulting in seamless network access across various geographies. Further, if the handset used by the subscriber is a legacy handset, the infrastructure equipment can use a software module implementing the older standard to communicate with the handset.

- **Interoperability:** SDR facilitates implementation of open architecture radio systems. End-users can seamlessly use innovative third-party applications on their handsets as in a PC system. This enhances the appeal and utility of the handsets.

1.3.3 Advantages of SDR

The advantages of SDR are listed in the table below:

Table 1 : Advantages of SDR

Advantage	Explanation
Multi-operability	Support of multiple standards through multi mode, multi band radio capabilities.
Flexibility	Efficient shift of technology and resources.
Adaptability	Faster migration towards new standards and technologies through programmability and reconfiguration.
Sustainability	Increased utilization through generic hardware platforms.
Reduced Ownership Costs	Less infrastructure, less maintenance and easier deployment.

CHAPTER 2

Softwares Used for Implementation

We first start with the software implementation of an SDR in Simulink in MATLAB and then after generating the MATLAB code convert it into HDL using XILINX ISE System generator. We then burn the code on FPGA. The software's that will be used in implementation are discussed below:

2.1 **MATLAB 7.6.0**

Matrix Laboratory (MATLAB) is a programmable environment for algorithm development, data analysis, visualization and numerical computation. Using MATLAB, we can solve technical computing problems faster than with traditional programming languages, such as C, C++ and Fortran[10,11].

We can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. For a million engineers and scientists in industry and academia, MATLAB is the language of technical computing.

Key features of MATLAB include –

- High-Level language for technical computing.
- Development environment for managing code, files and data.
- Interactive tools for iterative exploration, design and problem solving.
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization and numerical integration.

- Tools for building custom graphical user interfaces.

MATLAB is the foundation for all products, including Simulink. We can extend MATLAB with add-on products for –

- Parallel Computing
- Math, Statistics and Optimization.
- Control System Design and Analysis.

We can use the MATLAB product family for -

- Communication Systems
- Control Systems
- Digital Signal Processing
- Embedded Systems
- FPGA Design
- Image and Video Processing.

2.1.1 <u>Simulink 7.1</u>

Simulink is an environment for multi-domain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let us design, simulate, implement and test a variety of time- varying systems, including communications, controls, signal processing, video processing and image processing. Simulink is integrated with MATLAB, providing immediate access to an extensive range of tools that let you develop algorithms, analyze and visualize simulations, create batch processing scripts, customize the modeling environment and define signal, parameter and test data [12].

Key features of Simulink include -

- Extensive and expandable libraries of predefined blocks
- Interactive graphical editor for assembling and managing intuitive block diagrams

- Ability to manage complex designs by segmenting models into hierarchies of design components
- Model Explorer to navigate, create, configure and search all signals, parameters, properties and generated code associated with your model
- Application programming interfaces (API) that let you connect with other simulation programs and incorporate hand-written code.

We can extend the Simulink environment with add-on products for -

- Simulation Graphics
- Control System Design and Analysis
- Signal Processing and Communications
- Code Generation

We can use the Simulink product family for -

- Model-Based Design
- Control Systems
- Digital Signal Processing
- Communications Systems
- Image & Video Processing
- Embedded Systems

With Simulink, we can quickly create, model and maintain a detailed block diagram of your system using a comprehensive set of predefined blocks. Simulink provides tools for hierarchical modeling, data management and subsystem customization, making it easy to create concise, accurate representations regardless of your systems complexity.

Simulink software includes an extensive library of functions commonly used in modeling a system. These include:

- Continuous and discrete dynamics blocks such as Integration & Unit Delay.
- Algorithmic blocks, such as Sum, Product and Lookup table.

- Structural blocks, such as Mux, Switch and Bus Selector.

You can customize these built-in blocks or create new ones directly in Simulink and place them into our own libraries. Additional blocksets (available separately) extend Simulink with specific functionality for aerospace, communications, radio frequency, signal processing, video and image processing and other applications. With Simulink, we can build models by dragging and dropping blocks from the library browser onto the graphical editor and connecting them with lines that establish mathematical relationships between the blocks. You can arrange the model by using graphical editing functions, such as copy, paste, undo, align, distribute and resize.

The Simulink user interface gives us complete control over what you can see and use onscreen. We can add your commands and submenus to the editor and context menus. You can also disable and hide menus, menu items and dialog box controls.

Simulink lets you organize your model into clear, manageable levels of hierarchy by using subsystems and model referencing. Subsystems encapsulate a group of blocks and signals in a single block. You can add a custom user interface to a subsystem that hides the subsystem's contents and makes the subsystem appear as an atomic block with its own icon and parameter dialog box. We can reuse the design components on multiple projects, easily maintaining audit and revision histories.

Organizing our models in this way lets us select the level of detail appropriate to the design task. For example, we can use simple relationships to model high level specifications and add more detailed relationships as you move toward implementation.

2.2 XILINX ISE System Generator 11.1

System Generator allows device specific hardware designs to be constructed directly in a flexible high level system modeling environment. In a System Generator design, signals are not just bits. They can be signed and unsigned fixed point numbers and changes to the design automatically translate into appropriate changes in signal types. Blocks are not just stand-ins for hardware. They respond to their surroundings, automatically adjusting the results they produce and the hardware they become. System Generator, allows designs to be composed from a variety of ingredients. Data flow models, traditional hardware design languages (VHDL, Verilog and EDIF) and functions derived from the MATLAB programming language can be used side-by-side, simulated together and synthesized into working hardware. System Generator simulation results bits and cycle accurately. This means results seen in simulation exactly match the results that are seen in hardware. System Generator simulations are considerably faster than those from traditional HDL simulators and results are easier to analyze. [13,14]

- Develops highly parallel systems.
- Provides system modeling and automatic code generation from Simulink and MATLAB (The Mathworks, Inc.)
- Integrates RTL, embedded, IP, MATLAB and hardware components of a DSP system.
- A key component of the XILINX DSP Targeted Design Platform.

System Generator for DSP is part of both the DSP and System Editions of ISE Design Suite. With System Generator for DSP, developers with little FPGA design experience can quickly create production quality FPGA implementations of DSP algorithms in a fraction of traditional RTL development times.

Key Features

- **DSP modeling -**
  Build and debug high performance DSP systems in Simulink using the Xilinx Blockset that contains functions for signal processing (e.g. FIR filters, FFTs), error

correction (e.g., Viterbi decoder, Reed Solomon encoder/decoder), arithmetic, memories (e.g., FIFO, RSM, ROM) and digital logic.

- **Automatic code generation of VHDL/Verilog from Simulink -**
  Implement behavioral (RTL) generation and target specific Xilinx IP cores from the Xilinx Blockset. System Generator also supports custom HDL through its HDL import flow.

- **Hardware co-simulation -**
  A code generation option that allows you to validate working hardware and accelerate simulations in Simulink and MATLAB. System Generator supports Ethernet (10/100/Gigabit) and JTAG communication between a hardware and Simulink.

- **Hardware/software co-design of embedded systems -**
  Build and debug DSP co-processors for the Xilinx MicroBlaze soft processor core. System Generator provides a shared memory abstraction of the HW/SW interface, automatically generating the DSP the bus interface logic, software drivers and software documentation.

CHAPTER 3

PROPOSED WORK

In this chapter we will discuss the block diagram using the simulink model. Simulink, developed by MathWorks, is a graphical programming environment for modeling, simulating and analyzing multi-domain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. Simulink is widely used in automatic control and digital signal processing for multi domain simulation and Model-Based Design.

After that we discuss the method for finding the filter orders for both AM and FM LPFs followed by the AM Transmitter and Receiver and FM Transmitter and Receiver. Finally the models are then implemented in FPGA .

3.1 **Basic Block Diagram**

The block diagram that we use for the SDR implementation [16,17] is shown in Figure 3.1 below.

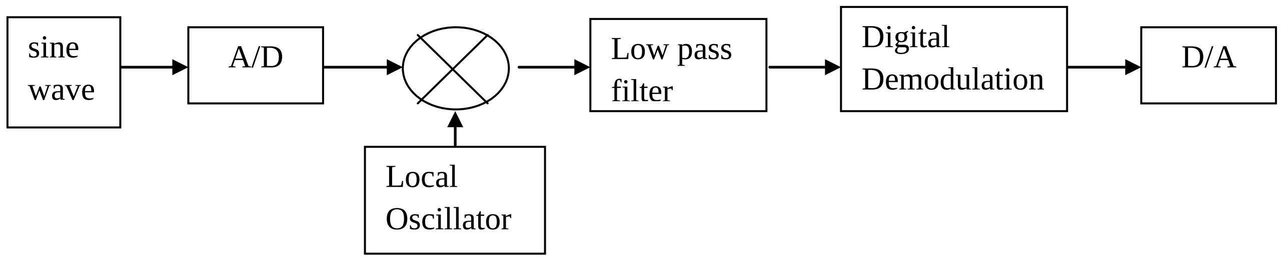


Figure 3.1 : Block Diagram for the SDR implementation of a sine wave

Explanation of each block and their corresponding results are given in the later sections.

3.1.1 Analog Filter

The block diagram along with the parameters for an analog filter is stated below:

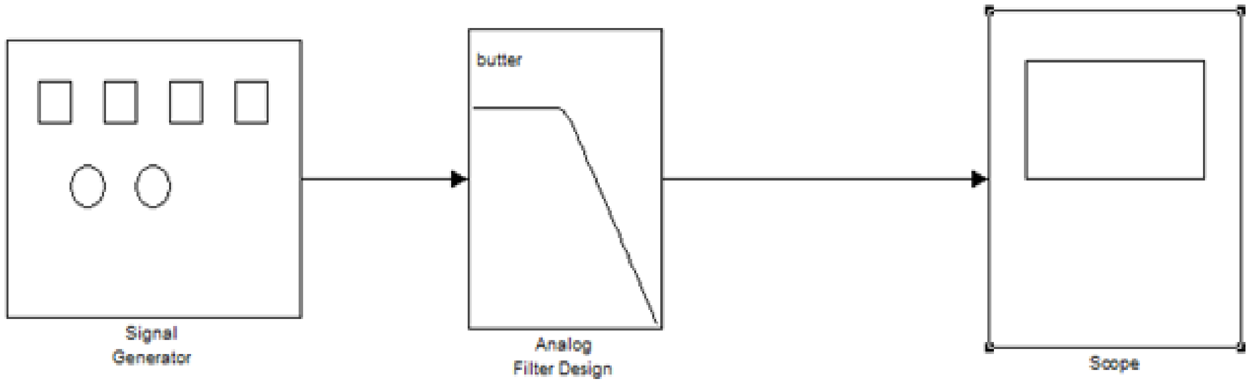


Figure 3.2 : Block Diagram for an analog filter

Function Block Parameters:

Design method:	Butterworth
Filter type:	Lowpass
Filter order:	5
Passband edge frequency(rad/s):	2*л*500

The Analog Filter Design block uses a state-space filter representation, and applies the filter using the State-Space block in the Simulink Continuous library.

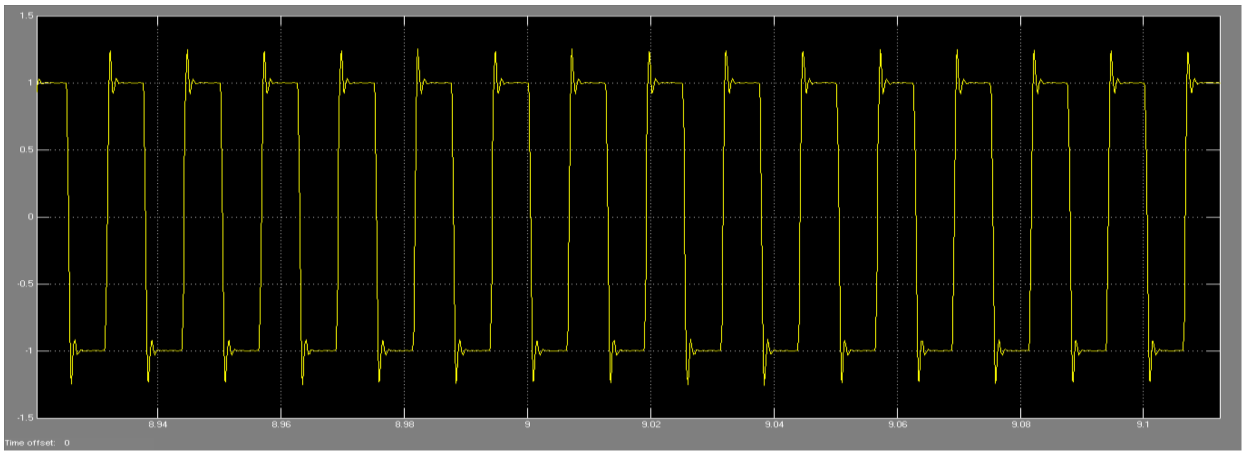


Figure 3.3 : Output of an analog filter

3.1.2 <u>A/D Converter</u>

The simulink block diagram for analog to digital conversion is :

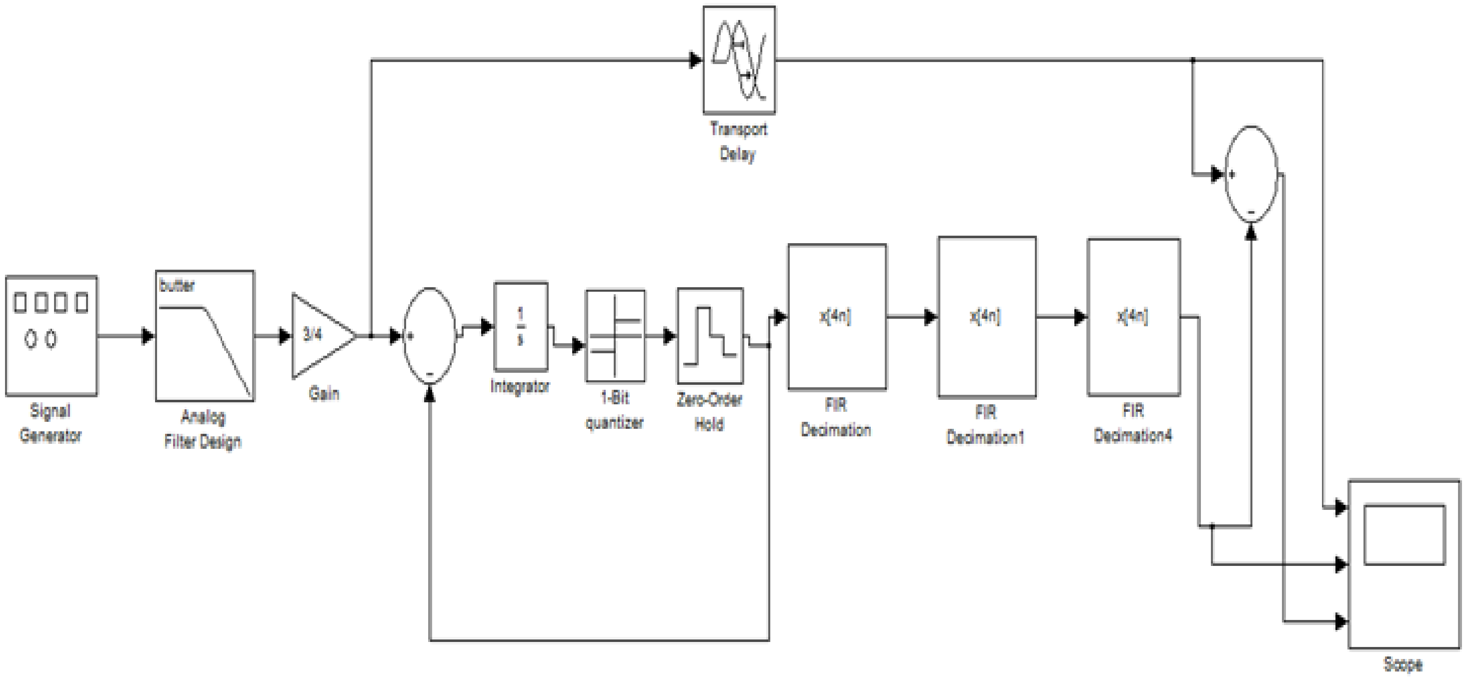
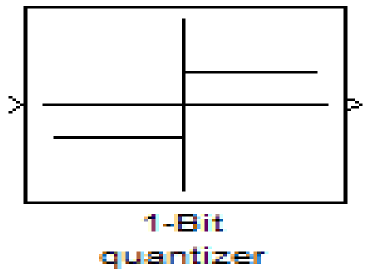


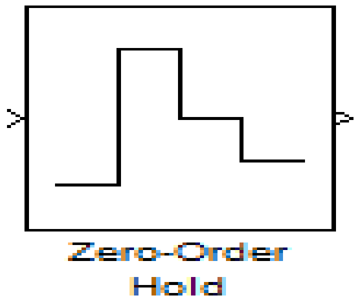
Figure 3.4.: Simulink model for A/D Converter

1-Bit quantizer and Zero-Order Hold:



A device or algorithmic function that performs quantization is called a quantizer.

Figure 3.5 : Symbol for 1-Bit quantizer



Convert an input signal with a continuous sample time to an output signal with a discrete sample time.

Figure 3.6 : Symbol for zero-order hold

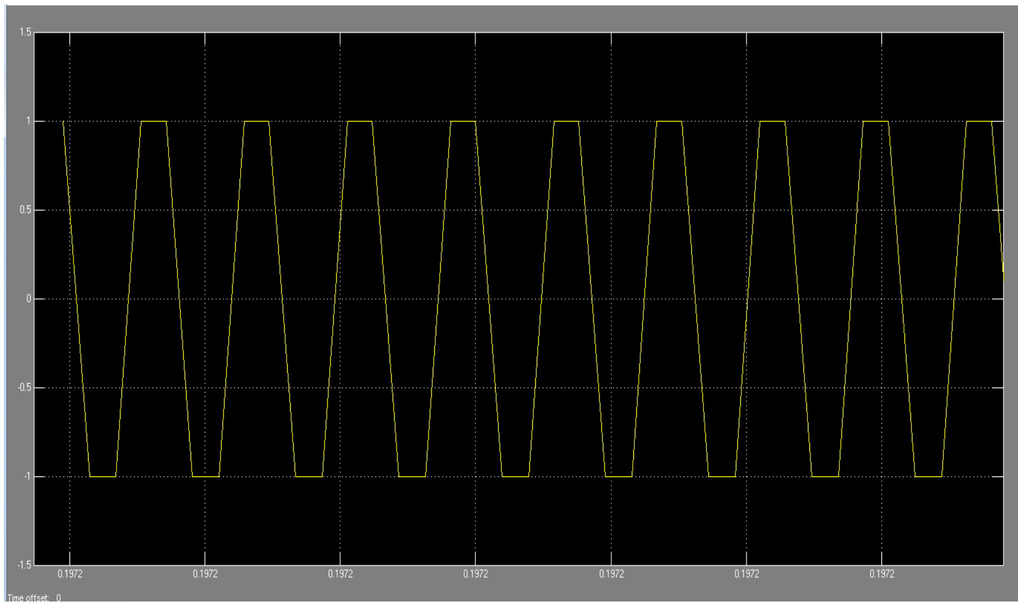


Figure 3.7 : Output after passing through 1-Bit Quantizer and Zero Order hold

3.1.3 <u>Decimator Design</u>

The simulink model for decimators is as shown in Figure 3.8.

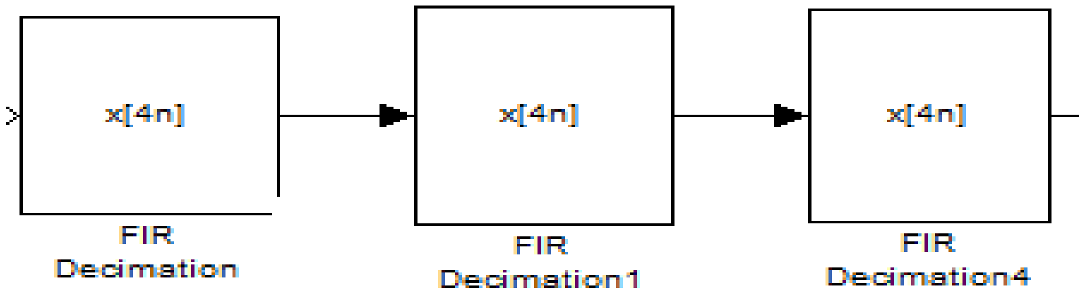


Figure 3.8 : Symbol for FIR Decimator

This model uses a cascade of three polyphase FIR decimators. This approach reduces computation and memory requirements as compared to a single decimator by using lower-order filters. Each decimator stage reduces the sampling rate by a factor of four. The latency introduced by the filters is used to set the appropriate 'Time Delay' in the 'Transport Delay' block. Due to the decimation operation the total latency introduced by the three filters is as follows: 16 (first filter) + 4*16 (second filter) + 16*16 (third filter) to give a final total delay of 336. The denominator of the 'Time delay' parameter is the base rate of the model (512 kHz).

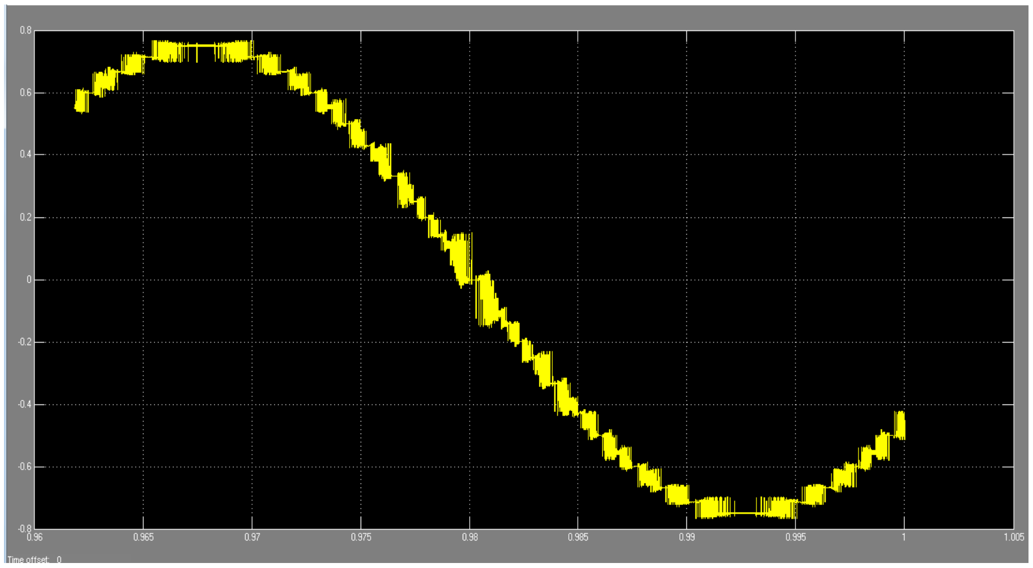


Figure 3.9 : Output after first decimation by a factor of 4

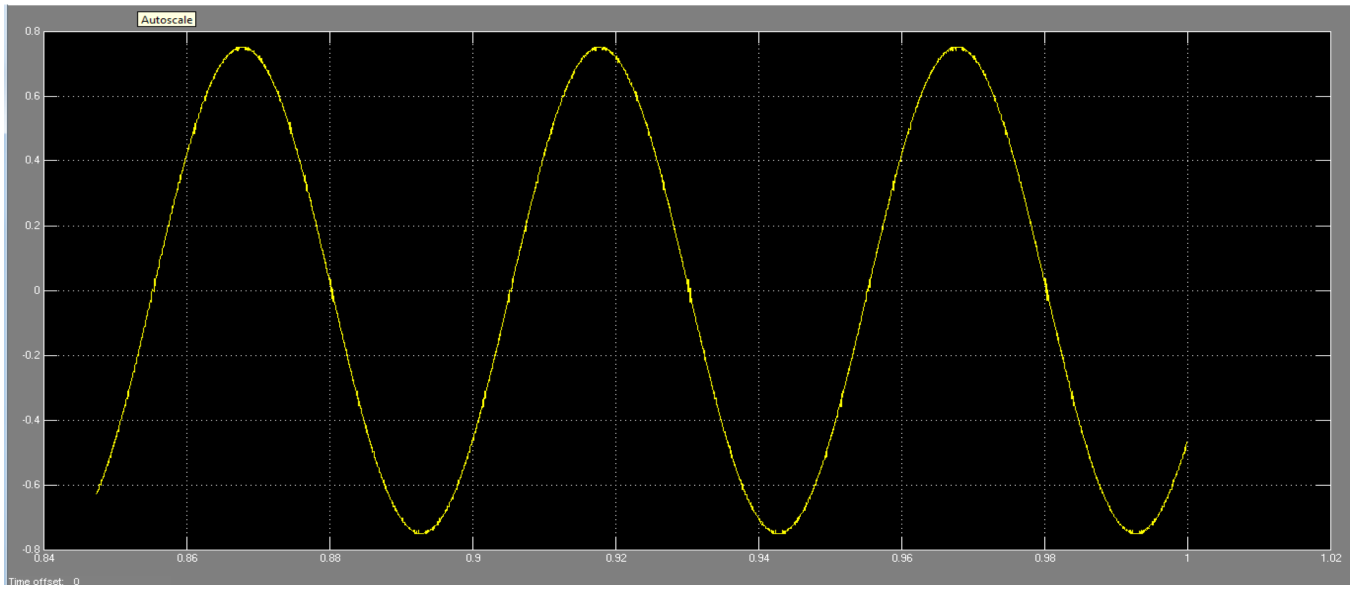


Figure 3.10: Output after second decimation by a factor of 4

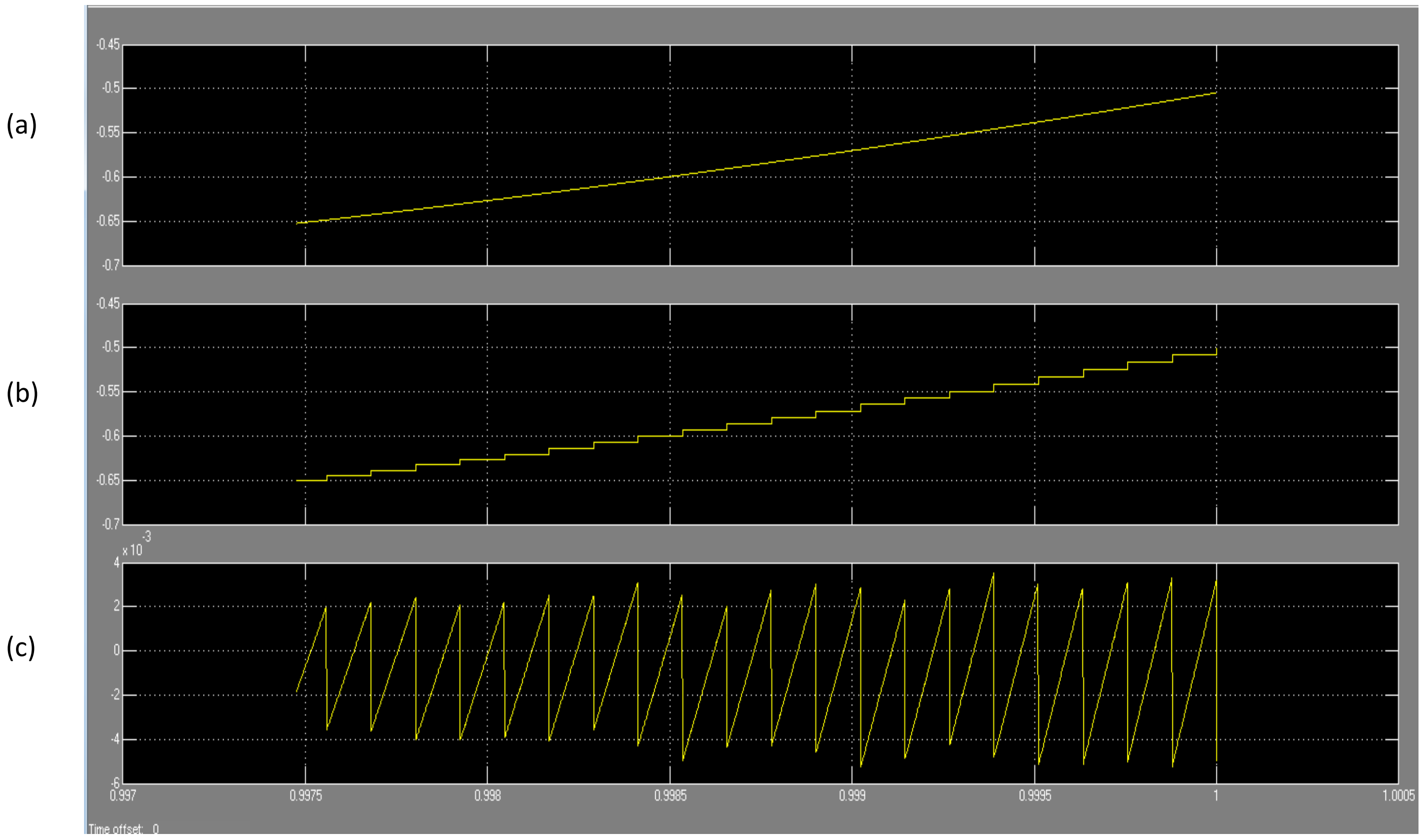
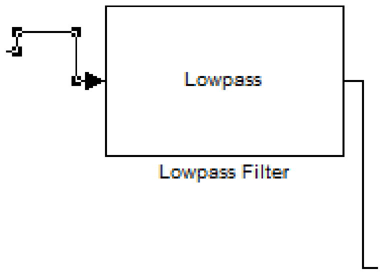


Figure 3.11 : Output for an amplitude modulated wave (a)original signal(delayed) (b)digitalized approximation (c) error

3.1.4 Low Pass Filter



The Lowpass Filter block independently filters each channel of the input signal over time using the given design specifications. the block designs a filter with the minimum order that the specified passband, stopband frequency, passband ripple, and stopband attenuation. Set these specifications using the corresponding parameters.

Figure 3.12 : Block Diagram for a Low pass filter

When passed through a Low pass Filter after mixing both outputs from the A/D converter and the carrier wave, the output is shown in figure 3.13

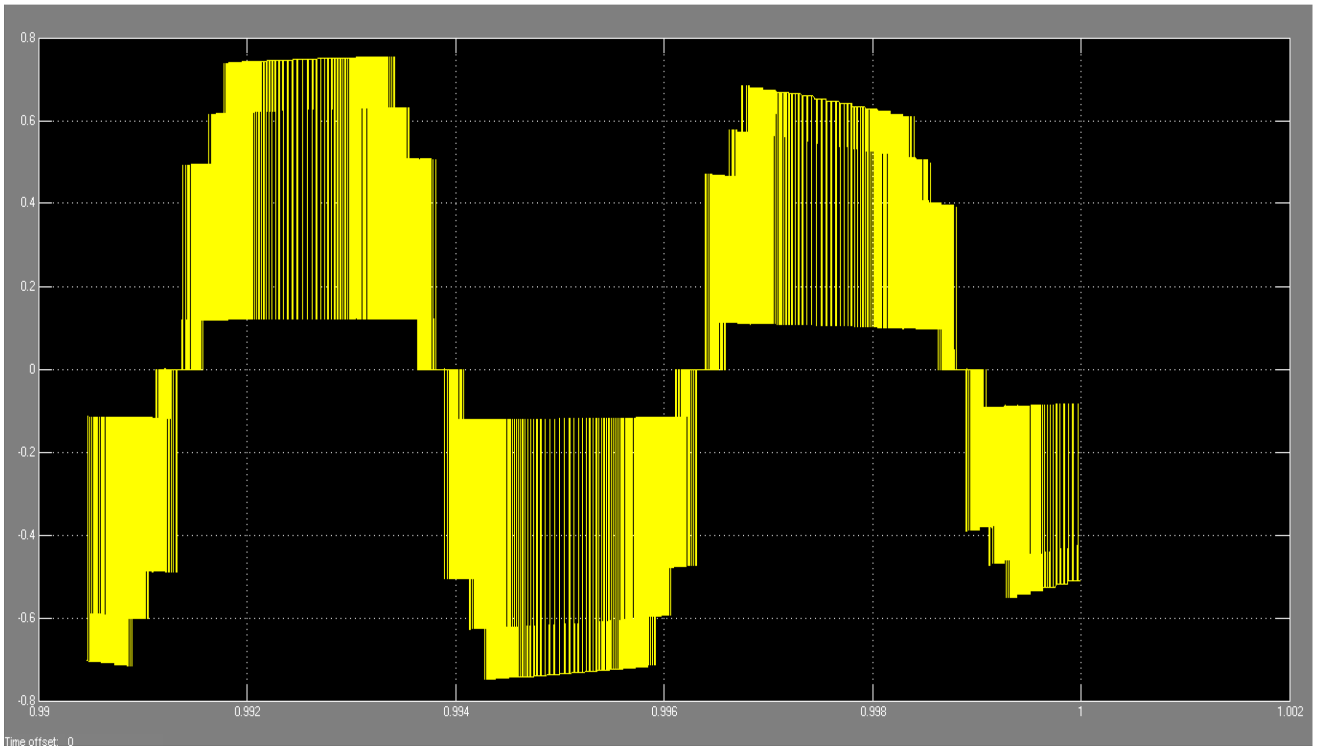


Figure 3.13 : Output of a Lowpass filter

3.1.5 Digital Demodulation and D/A conversion

The Simulink model for demodulation and digital to analog conversion is as shown:

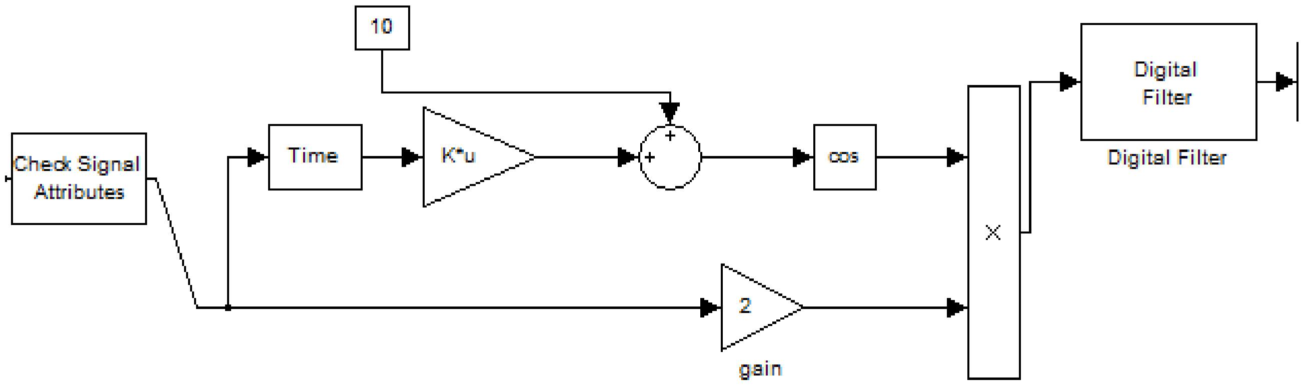


Figure 3.14 : Block Diagram for Demodulation and D/A conversion

3.2 Implementation of Low pass filter

Digital filters transform digital representation of analog signals to remove noise and unwanted signal components and to shape the spectral characteristics of the resulting signal. It operates on a finite precision digital representation of a signal. Two common architectures for the linear digital filters are:

a)Finite Impulse Response(FIR) filter

b) Infinite Impulse Response(IIR) filter

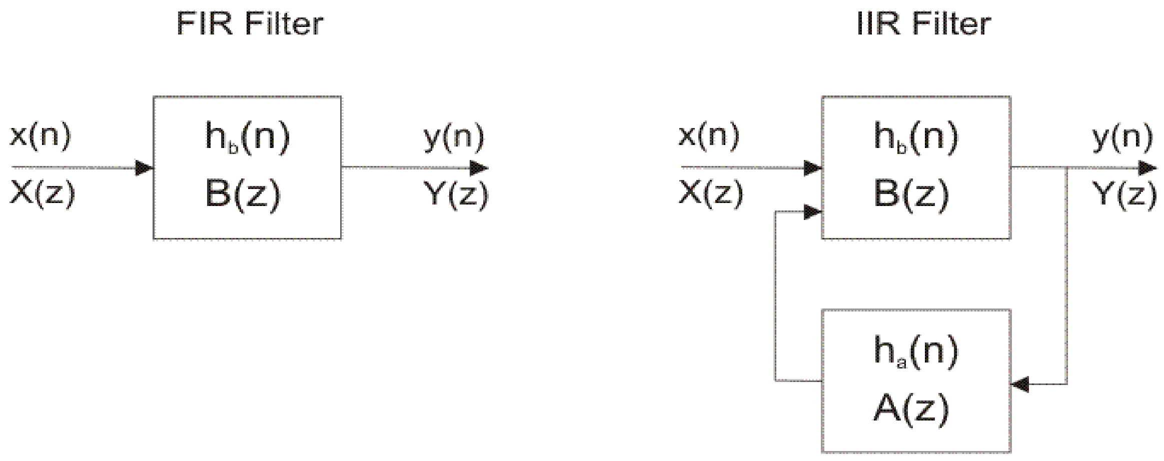


Figure 3.15 : Block Diagram of FIR and IIR filters

3.2.1 Implementation of AM-LPF:

An FIR filter is based on a feed-forward difference equation where feed-forward means that there is no feedback of past or future outputs to form the present output, just input related terms.

$$y[n] = \sum_{j=0}^{N} b_j \cdot x[n-j]$$

Feed-forward difference equation

Filter Specifications:

F_{pass}: Frequency range over which a filter passes signal energy

F_{stop}: Stop band edge frequency is the band of frequencies attenuated by a digital filter

A_{pass}:Attenuation in the pass band

A_{stop}: Attenuation in the stop band

Attenuation: an amplitude loss incurred by a signal

Filter Order: a number describing the highest exponent in the numerator or denominator of the z-domain transfer function of a digital filter.

Table 2 : AM Filter Specifications

AM Filter
$F_{pass:}$ 3 kHz
F_{stop}: 4.5kHz
$A_{pass:}$ 3dB
$A_{stop:}$30dB

Multistage Filtering Technique:

For both AM and FM receivers, the Low Pass Filter (LPF) plays a critical role in isolating adjacent channel interferences, and thus minimizing the noise at the output. The bandwidth of an AM channel is 9 kHz, thus a LPF having passband from 0-4.5 kHz is desired, whereas the bandwidth of FM channel is 200 kHz, thus the filter should have passband from 0-100 kHz. The following table shows the summary of filter requirements for AM and FM receivers.

In designing the FIR filter, we seek to minimize the filter order (or number of filter taps) while satisfying requirements specified above. FIR filter theory states that the filter order is approximately proportional to the input sampling frequency and inversely proportional to the width of the transition band. That is the higher the input sampling frequency, the higher the filter order, and the narrower the transition band, the higher the filter order.

Calculating Filter order

The filter order is calculated using the formula given below:

$$\text{Filter order (M)} \approx F_s/\triangle$$

$$\text{where} \quad \triangle F = F_{stop} - F_{pass}$$

Table 3 : AM Filter Order

Specification	**AM Filter**
Sampling frequency	4000kHz
F_{pass}	3kHz
F_{stop}	4.5kHz
Order	2670

This order is astronomically large due to very high input sampling rate and the narrow transition band. Therefore, a single FIR filter satisfying our requirements would be impossible to be realized in hardware. In order to overcome this problem we use Multistage filtering Technique so as to reduce number of filters.

In this technique, the single stage Filter of very high order is replaced by a chain of N stages of smaller order filters and down-sample operations. At each stage, the cutoff frequency of the filter is at the desired cutoff frequency of the original single filter, but its stopband frequency is set at half the new sampling rate (to the next stage).Therefore, the filter at each stage has lower input sample rate and broader transition band, resulting in a much smaller number of taps at each stage.

Table 4 : N-stage filter design

Filter	**Cutoff frequency**	**Stopband frequency**
FIR 1	f_c	$f_{in}/4$
FIR 2	f_c	$f_{in}/8$
FIR 3	f_c	$f_{in}/16$
FIR 4	f_c	$f_{in}/2^{k+1}$

Calculation of Filter Order using Multistage Filtering Technique

In the design of the filter for the AM Digital Receiver, the input sampling rate is 4000 kHz and the baseband message is of 4 kHz bandwidth, therefore we decided to reduce the output sampling rate to around 15 kHz to allow for some margin.

The number of stages of a N-stage filter would be N = $\log_2(4000/15) = 8$, wherein the sampling rate is reduced by a factor of 2 at each stage as discussed above.

$$\text{Filter Order} = 4000/(1000-4) \approx 4$$

Table 5 : A 9-stage filter for AM digital receiver

Stage	**Sampling Frequency(kHz)**	**Passband Frequency(kHz)**	**Stopband Frequency(kHz)**	**Filter Order**
1	4000	0-4	1000	4
2	2000	0-4	500	4
3	1000	0-4	250	4
4	500	0-4	125	4
5	250	0-4	62.5	5
6	125	0-4	31.25	6
7	62.5	0-4	15.625	9
8	31.25	0-4	7.8125	11
9	15.625	0-3	4.5	4
Total number of filter taps				51

3.2.2 Implementation of FM-LPF

Filters are used to eliminate unwanted harmonic frequencies. Table 6 shows the specifications for building an FM LPF.

Table 6 : Filter Specifications for FM-LPF

FM Filter
$F_{pass:}$ 90 kHz
F_{stop}: 100kHz
$A_{pass:}$ 3dB
$A_{stop:}$30dB

Table 7 : Filter Order for FM-LPF

Specification	**AM Filter**
Sampling frequency	44000kHz
F_{pass}	90kHz
F_{stop}	100kHz
Order	4400

Table 8 : Calculation of filter order for FM

Stage	**Sampling Frequency(kHz)**	**Passband Frequency(kHz)**	**Stopband Frequency(kHz)**	**Filter Order**
1	44000	0-90	11000	4
2	22000	0-90	5500	4
3	11000	0-90	2750	4
4	5500	0-90	1375	4
5	2750	0-90	687.5	5
6	1375	0-90	343.75	6

7	687.5	0-90	171.875	9
8	343.75	0-80	100	18
Total number of filter taps				54

3.3 AM Transmitter

The simulink model of an AM transmitter was realized using simulink blocks and the output was observed through spectrum scope and time scope.

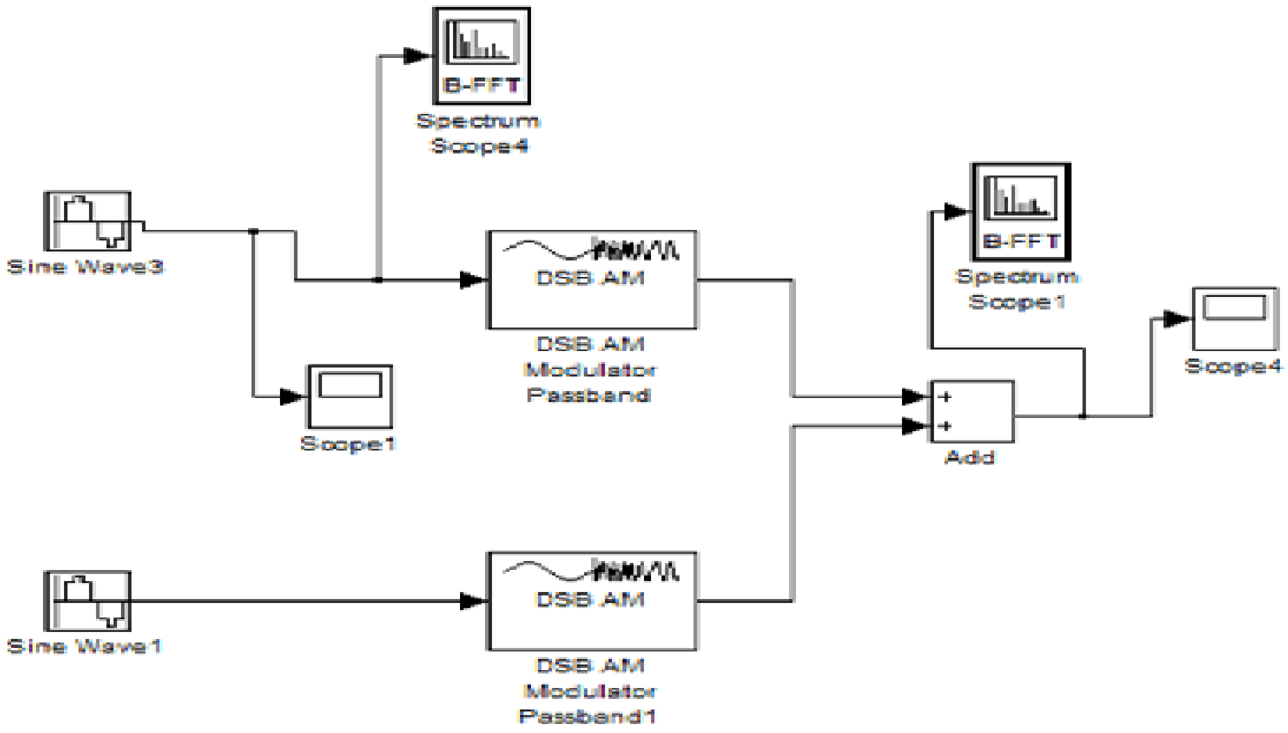


Figure 3.16 : Simulink Model for an AM Transmitter

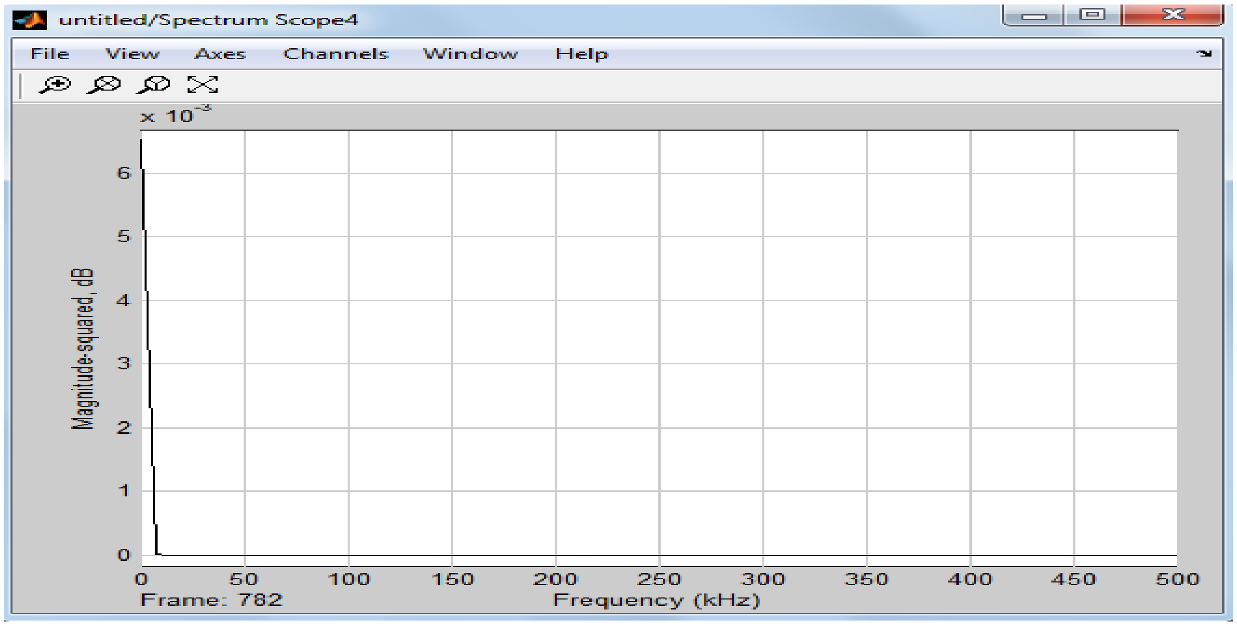


Figure 3.17 : Input frequency spectrum for an AM transmitter

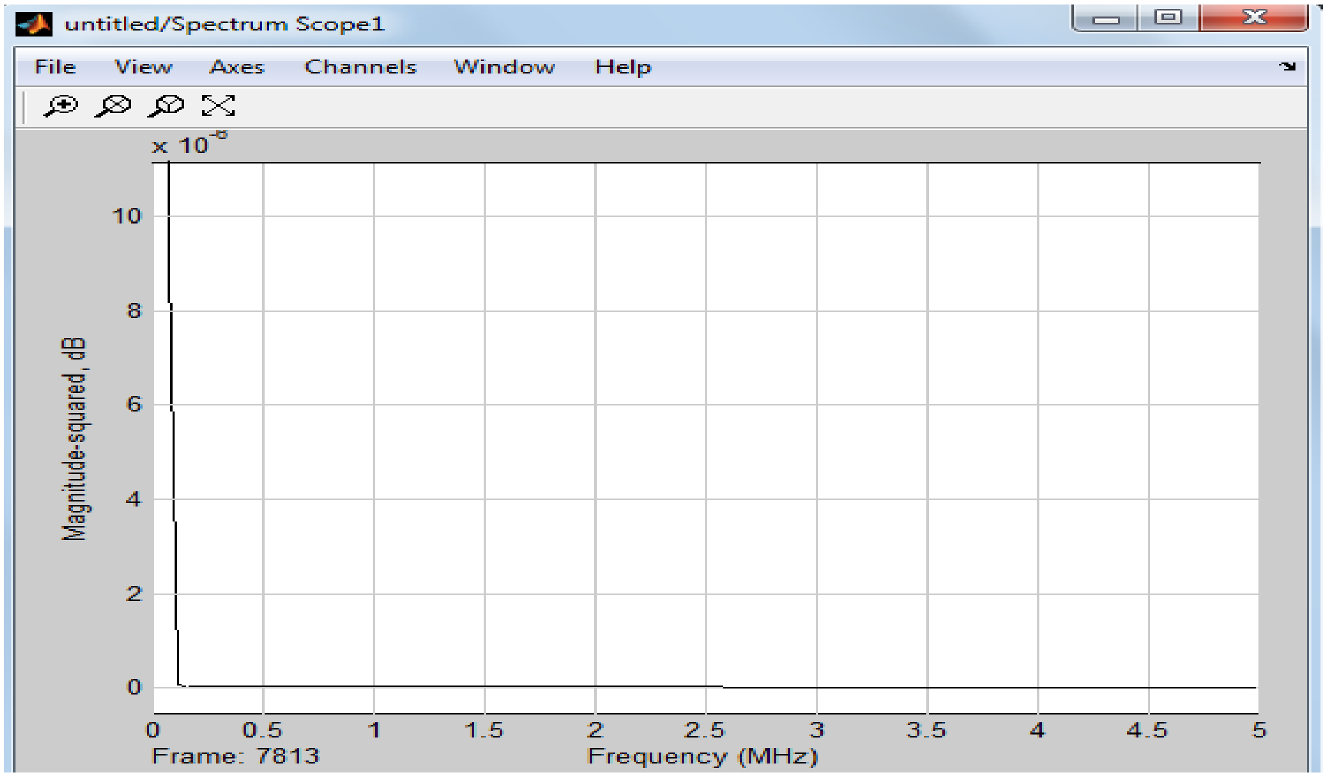


Figure 3.18 : Output frequency spectrum of AM transmitter

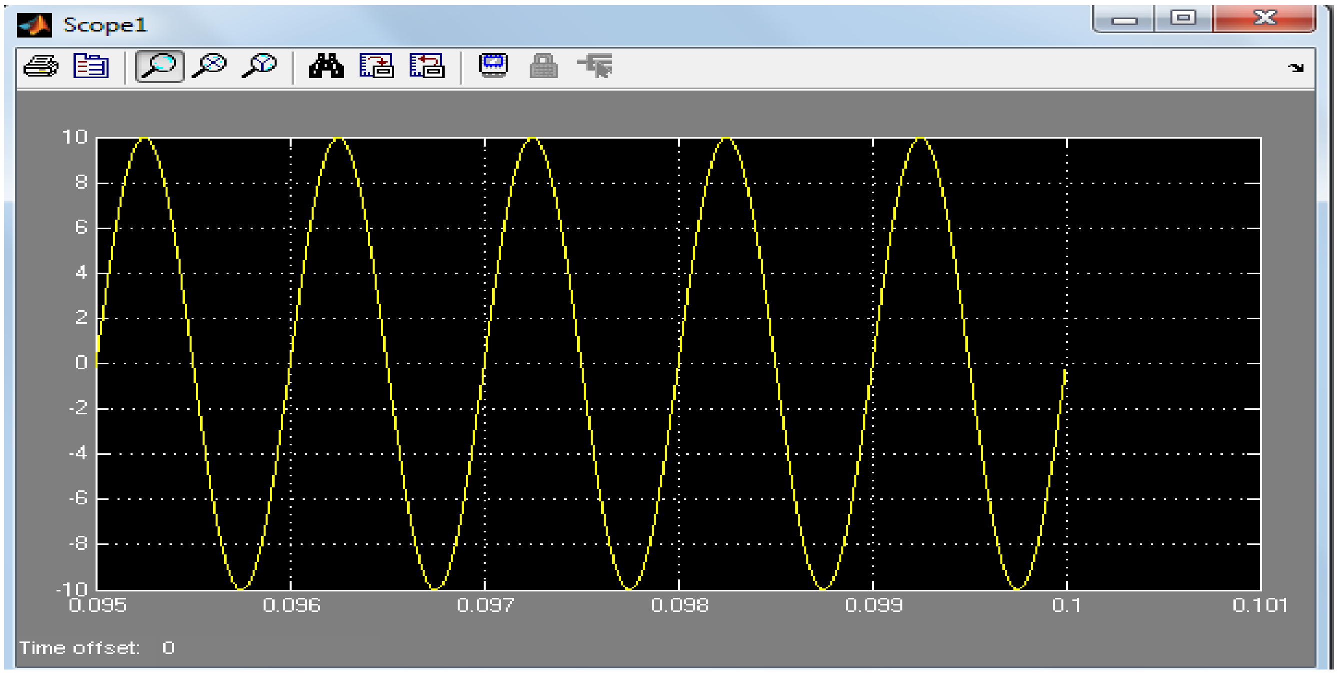


Figure 3.19 : Input waveform of AM transmitter

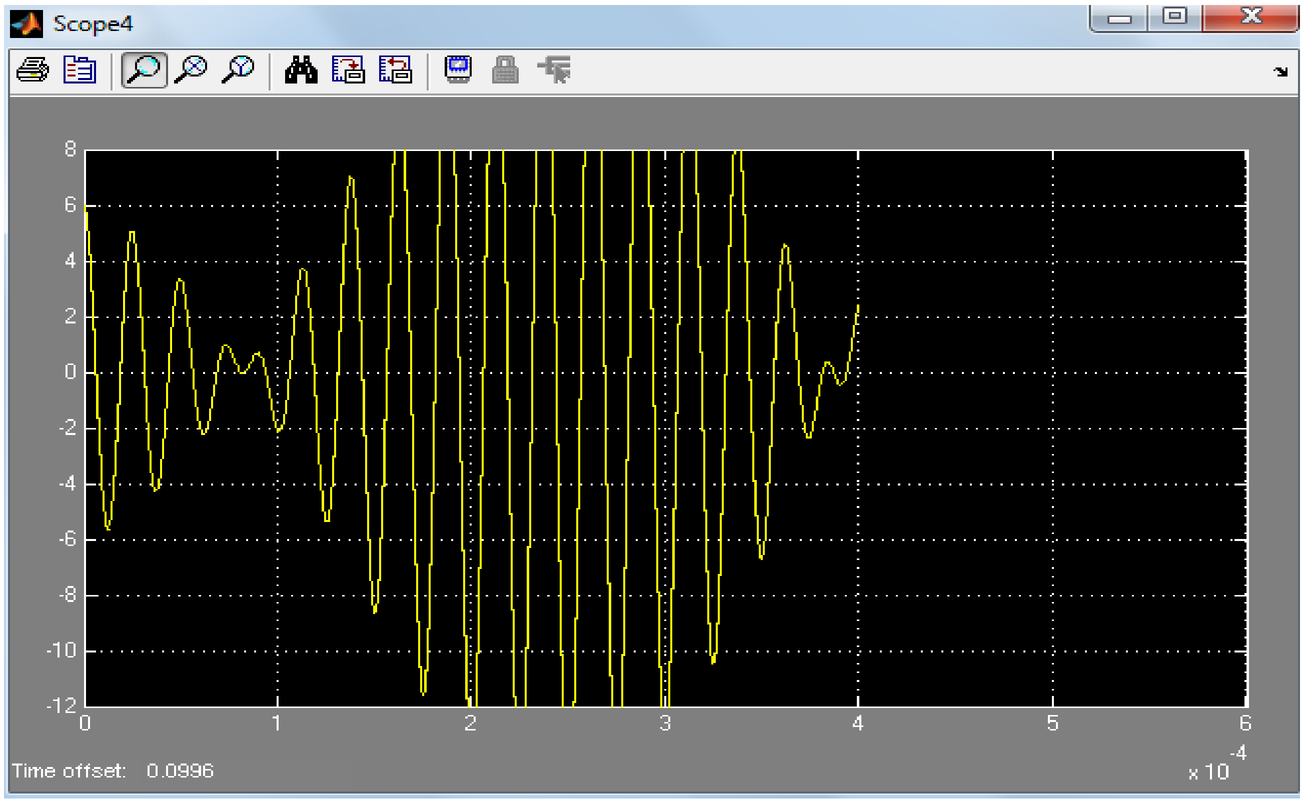


Figure 3.20 : Output waveform of AM transmitter

3.4 Implementation of AM receiver:

The AM receiver was implemented on MATLAB Simulink. An AM-DSB modulator block was used to modulate the input signal which was used as a signal transmitted from the radio station. The signal was then sampled using zero-order hold block. The mixer is used to translate the message from high frequency to DC level. The 9-stage low pass filter is used to remove unwanted high frequency components and the output is observed through spectrum scope.

The Simulink implementation is shown on the next page.[18]

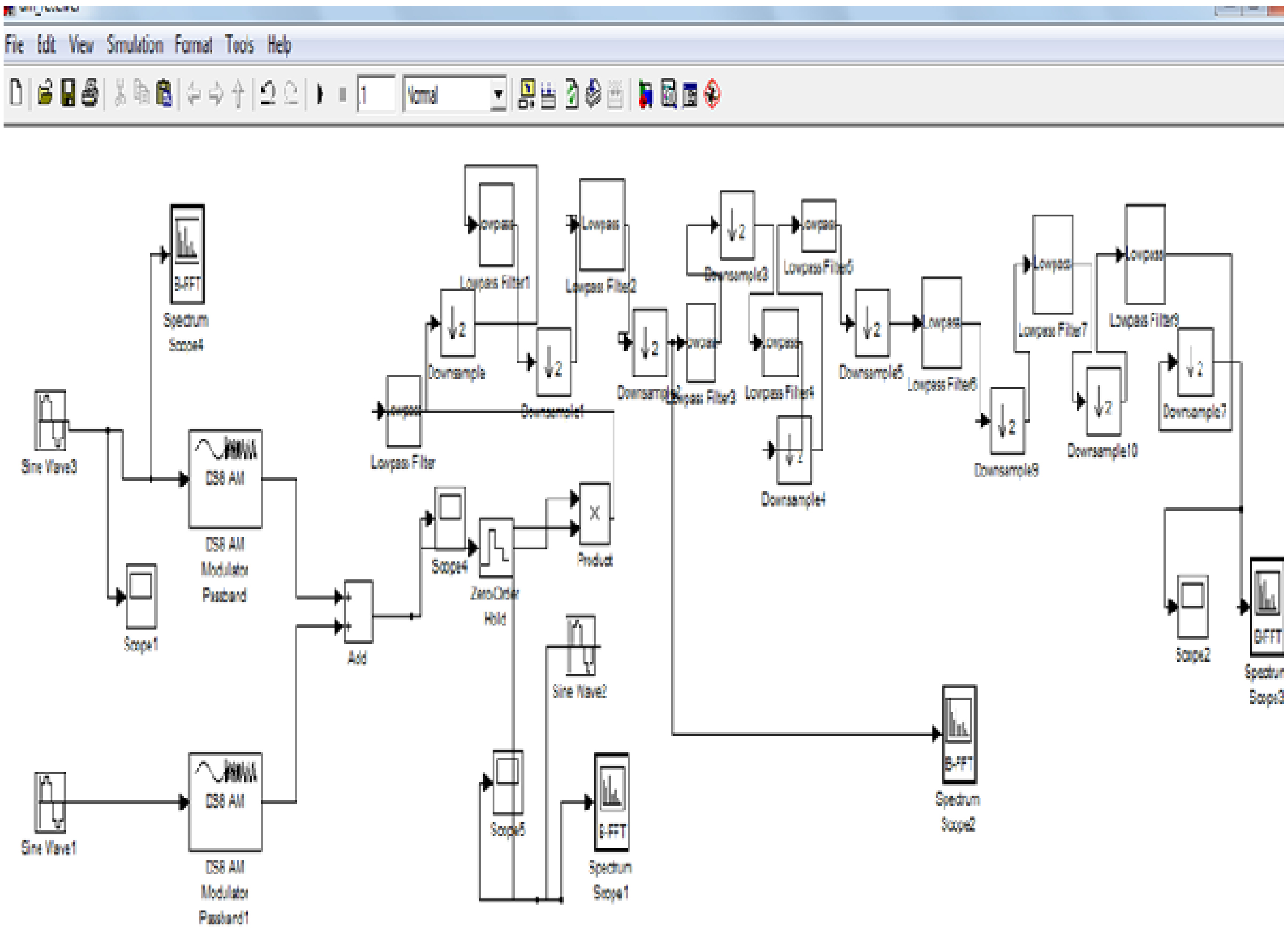


Figure 3.21 : Implementation of AM receiver on MATLAB Simulink

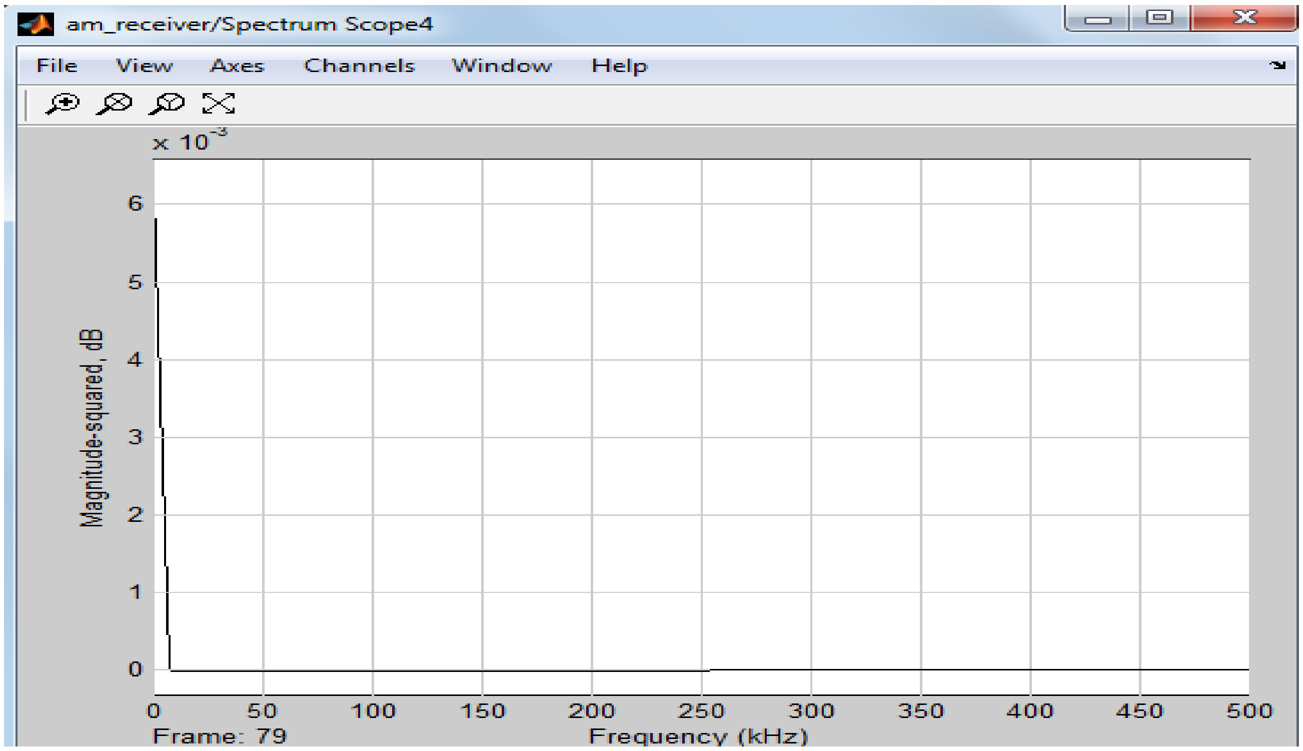


Figure 3.22 : Input frequency response

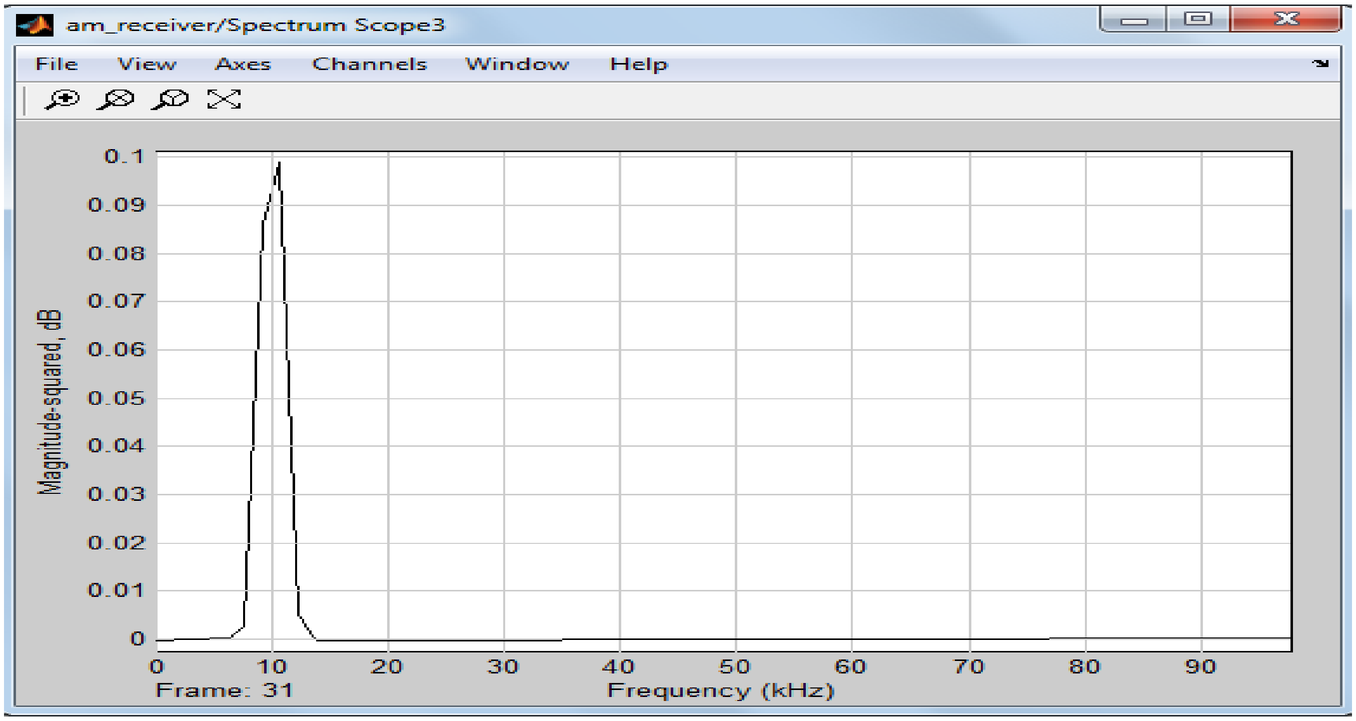


Figure 3.23 : Output frequency response of a sine wave of 10kHz

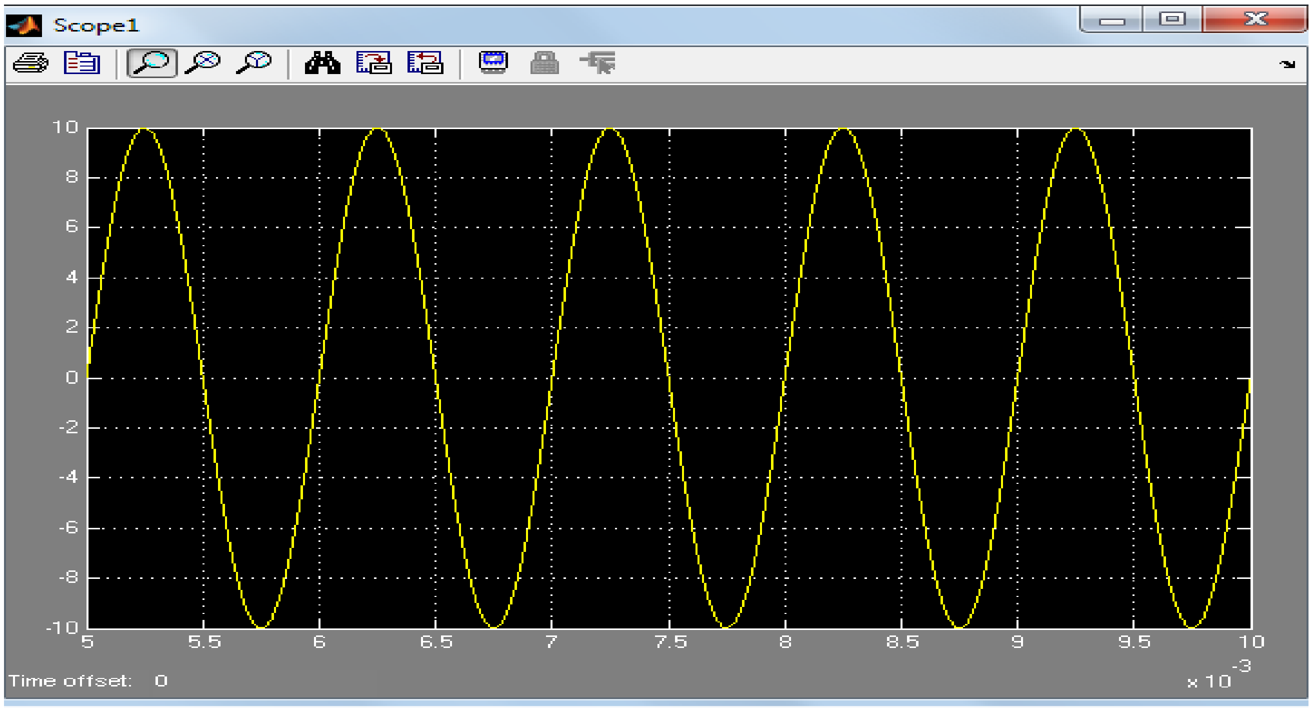


Figure 3.24 : Input time scope response of a sine wave of 10kHz

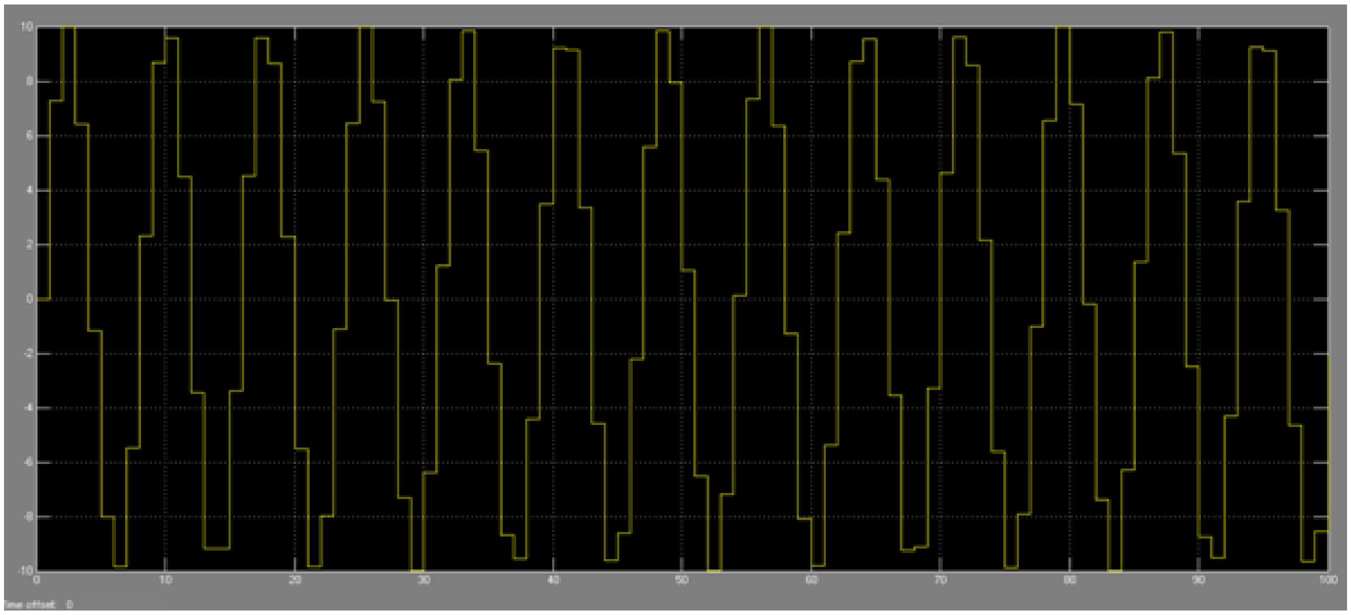


Figure 3.25 : Output time scope response of a sine wave of 10kHz

For a input wave of frequency 10kHz, the following input and output waves are observed on spectrum scope and time scope as shown in the above figures.

3.5 FM transmitter

The FM receiver was implemented on MATLAB Simulink with a simulation time of 0.1s.. An FM-DSB modulator block was used to modulate the input signal which was used as a signal transmitted from the radio station.

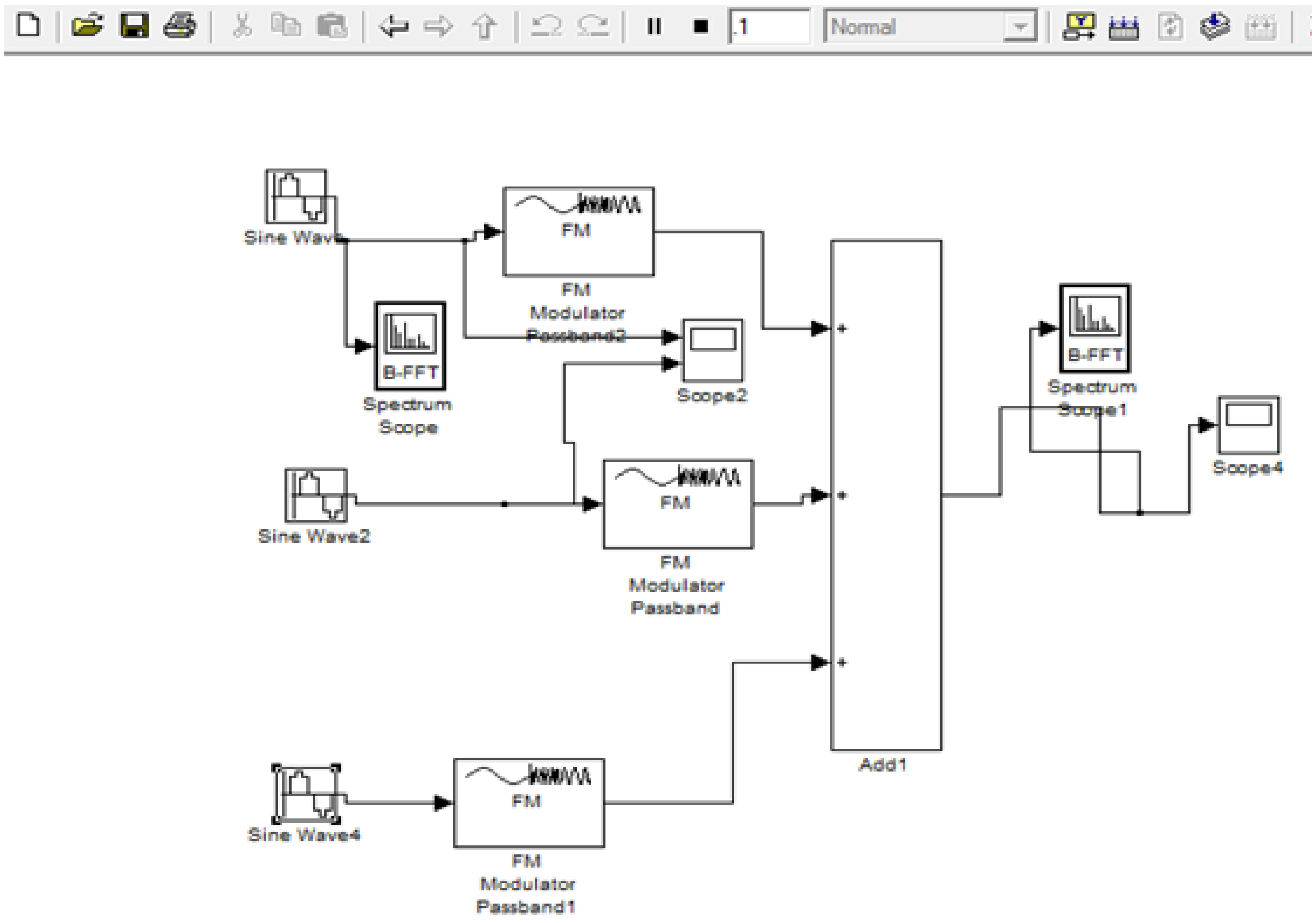


Figure 3.26 : Simulink model for an FM transmitter

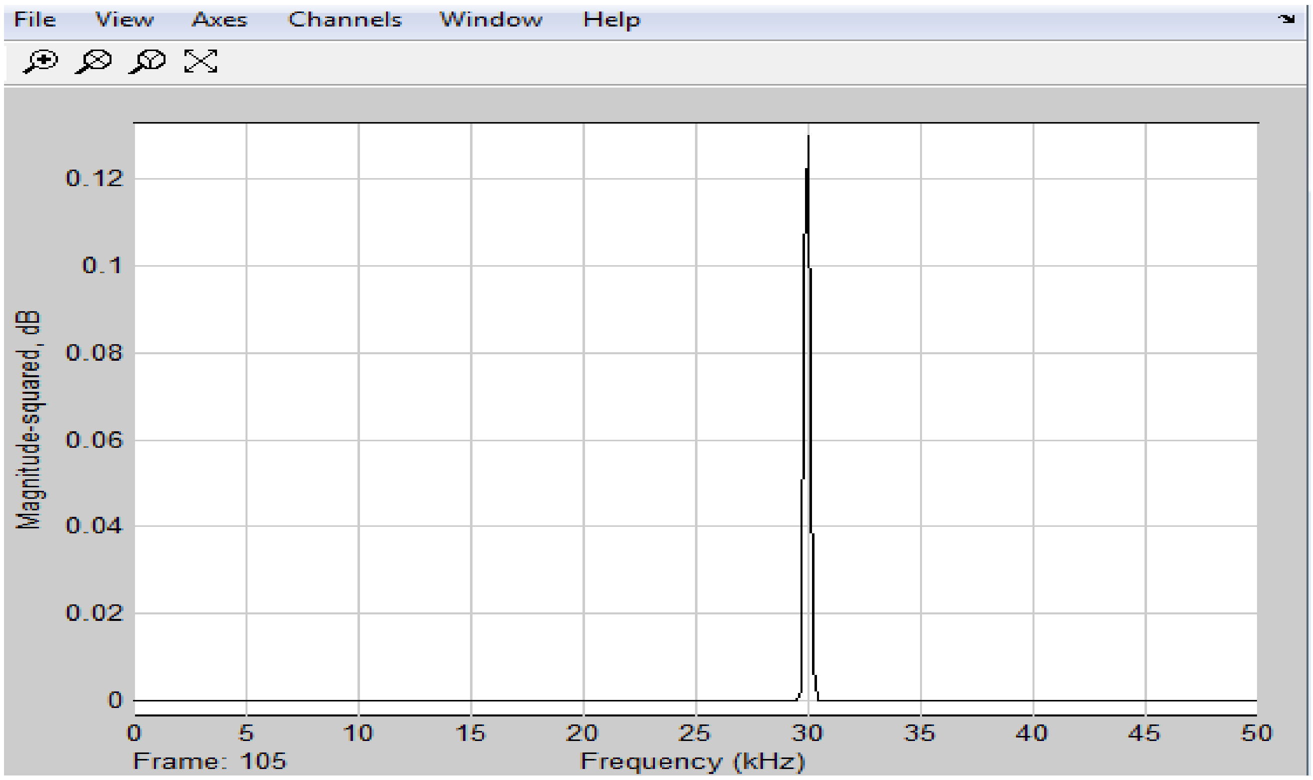


Figure 3.27 : Input frequency response for a sine wave of frequency 30kHz

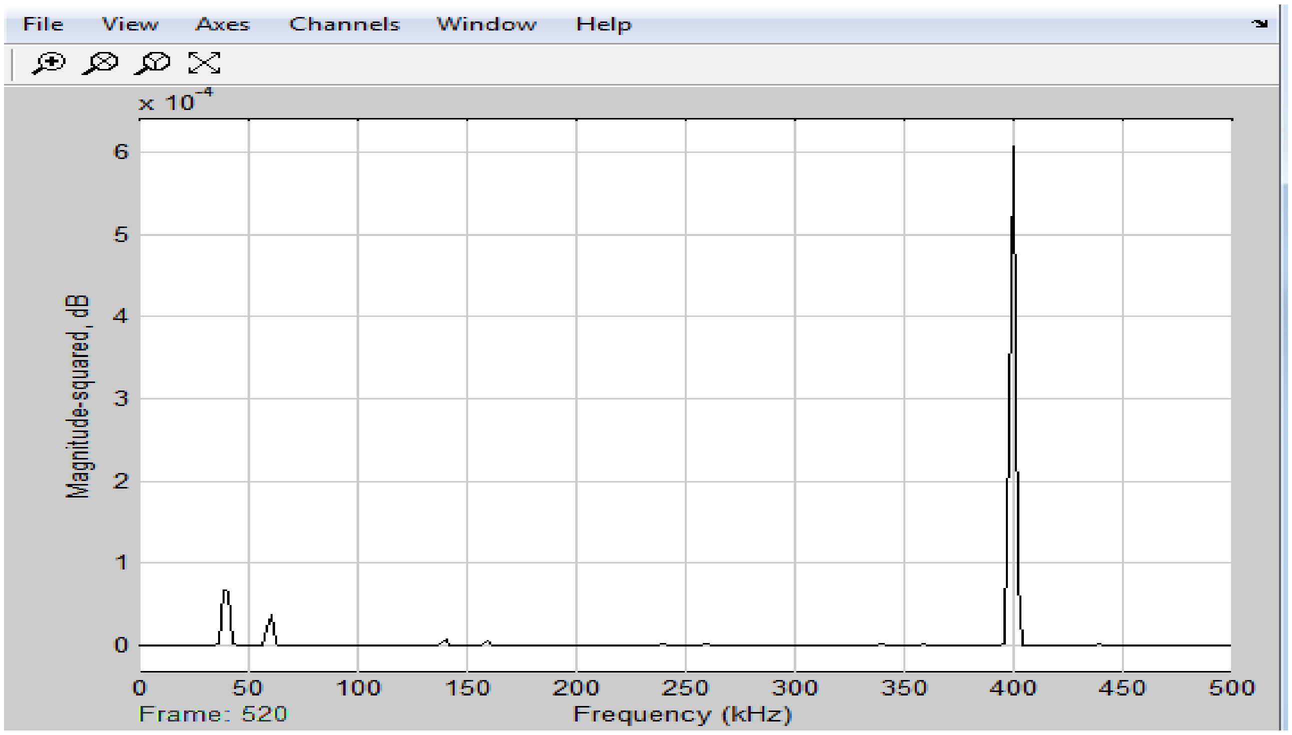


Figure 3.28 : Output frequency response of the transmitter

3.6 FM Receiver:

The FM receiver was implemented on MATLAB Simulink. An FM-DSB modulator block was used to modulate the input signal which was used as a signal transmitted from the radio station. The signal was then passed through a mixer. The mixer is used to translate the message from high frequency to DC level. The 8-stage low pass filter is used to remove unwanted high frequency components and the output is observed through spectrum scope.[19]

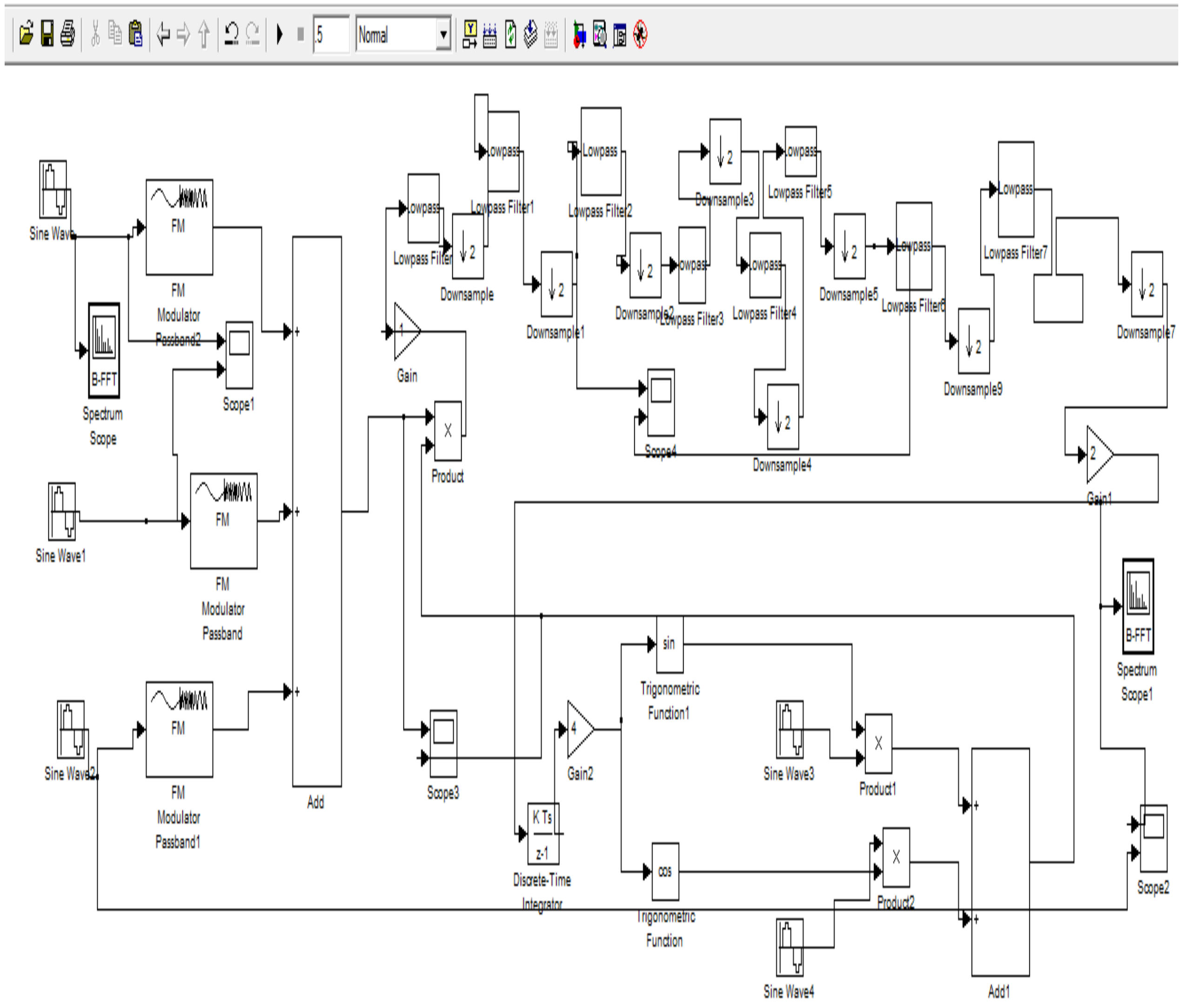


Figure 3.29 : Final Simulink model of the FM receiver

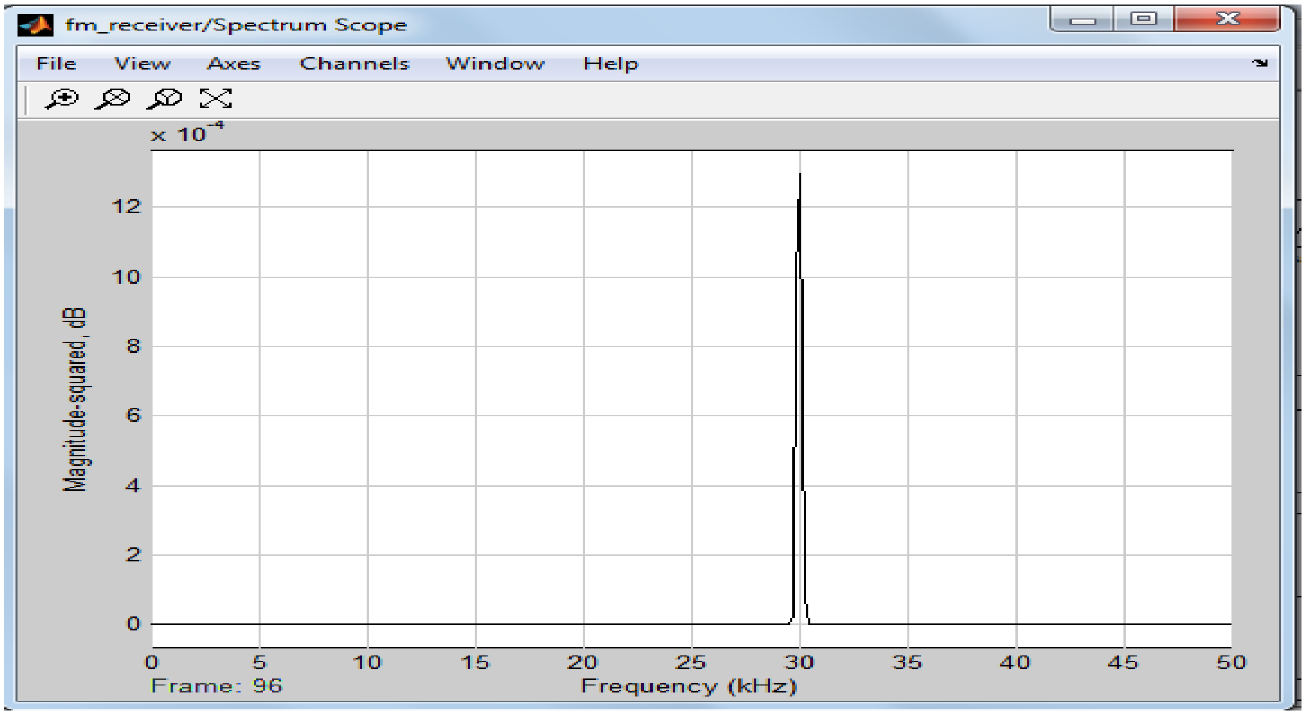


Figure 3.30 : Input frequency response for a sine wave of frequency 30kHz

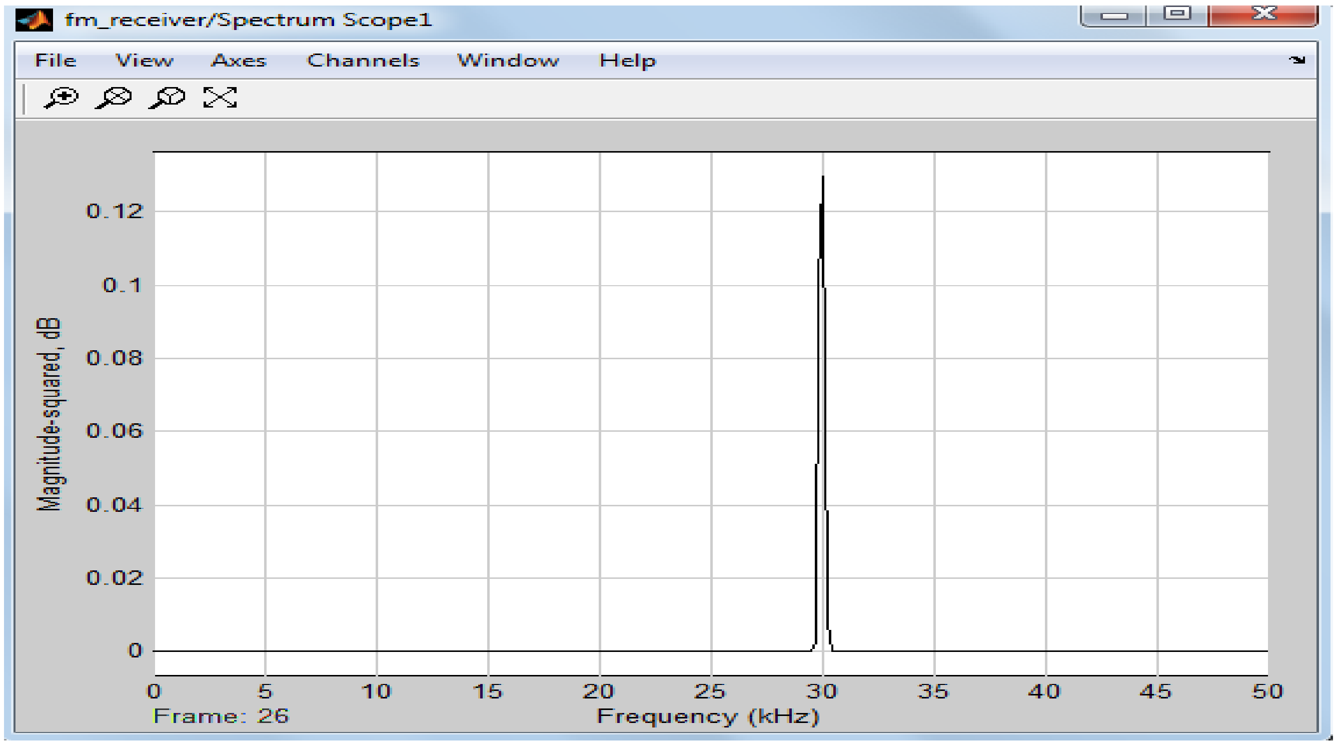


Figure 3.31 : Output of the FM receiver for a sine wave of frequency 30kHz

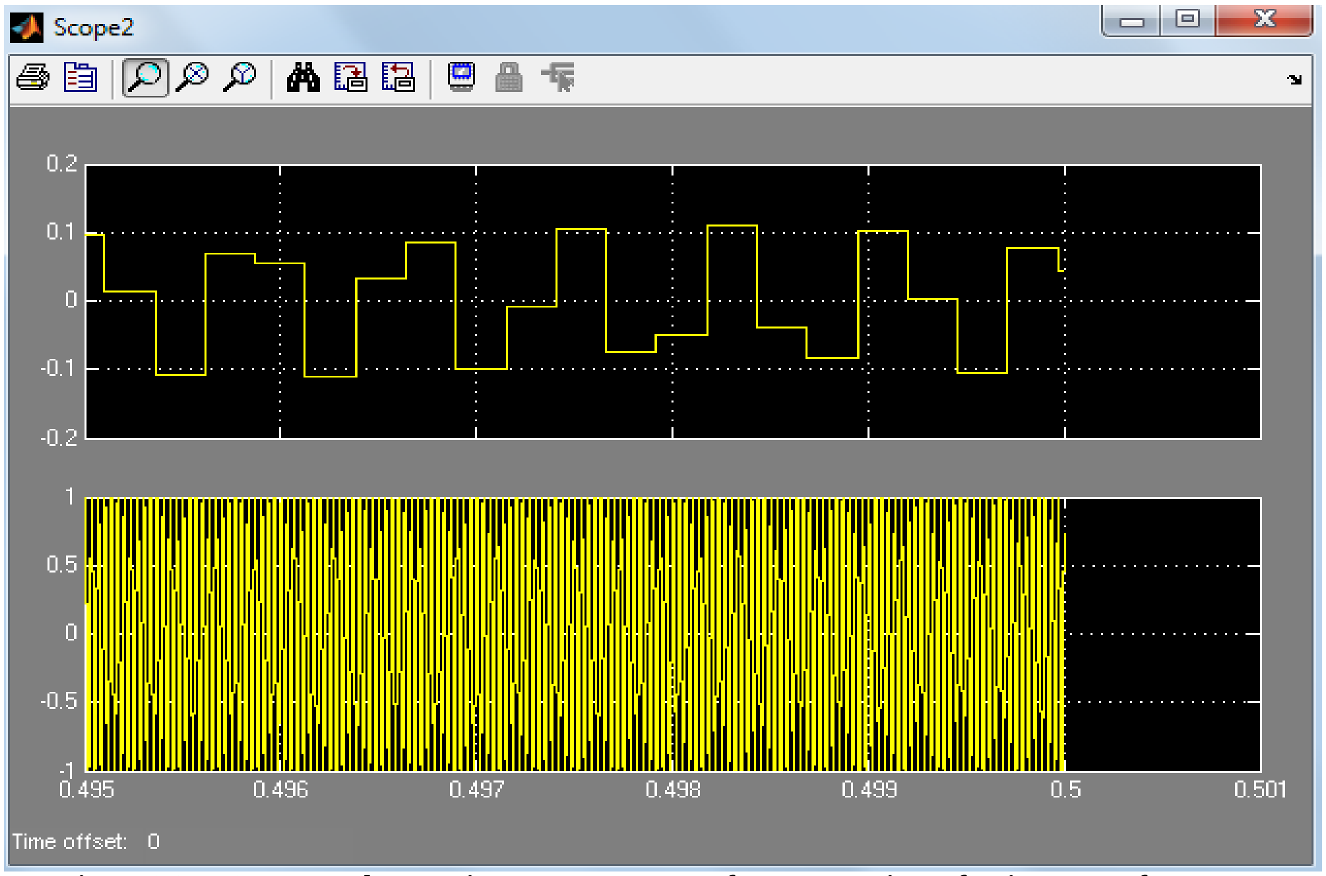


Figure 3.32 : Output and Input time scope response of an FM receiver of a sine wave of frequency 30kHz

The output and input for a sine wave of amplitude 1V and frequency 30KHz as observed through time scope are shown above.

Table 9 : Comparing message signal frequency with output frequency observed on spectrum scope

	Frequency of the message signal	Output Frequency observed
For AM receiver	10kHz	10kHz
For FM receiver	30kHz	33kHz

3.7 Analysis of a speech signal

This section represents the analysis of a speech signal. First the speech signal is recorded and is saved with a .wav extension. Next we write a code for adding zeros to speech signal so as to expand the signal accordingly and process it further. This is then passed through a 9-stage filter so as to remove noise and unwanted frequencies.

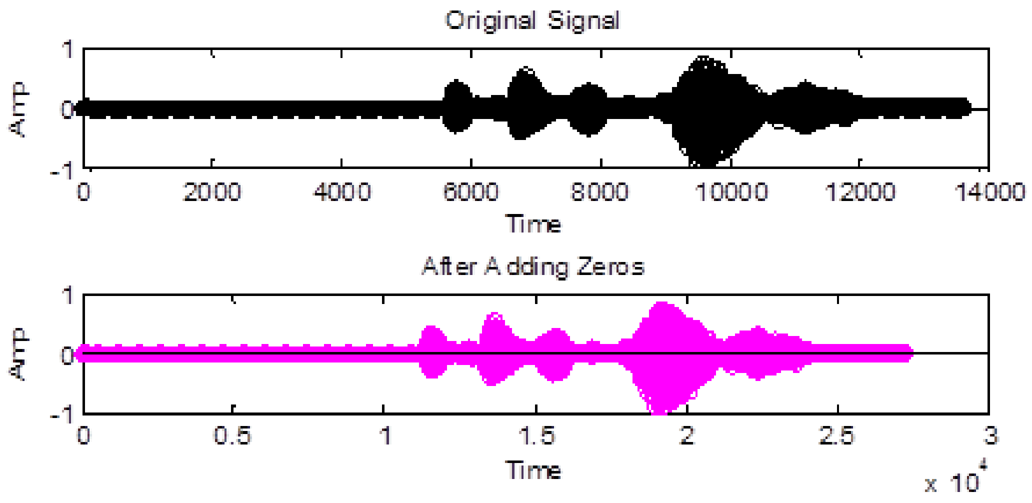


Figure 3.33: Speech signal plot

3.7.1 Simulink model for filtering of speech signal:

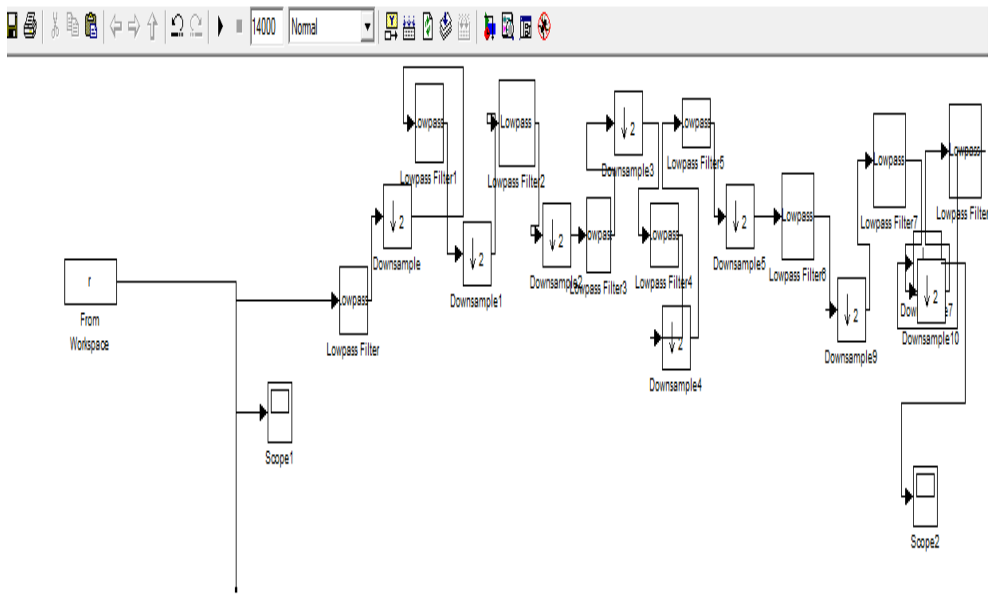


Figure 3.34: Analysis of speech signal

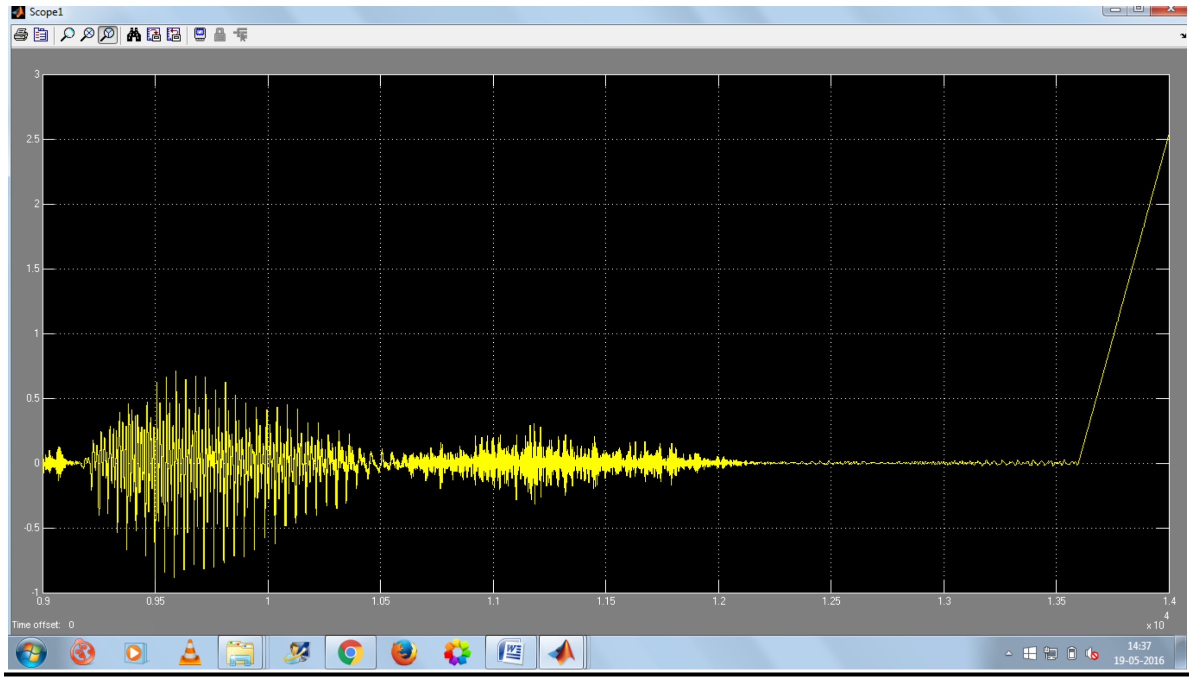


Figure 3.35 Input time scope response for a speech signal

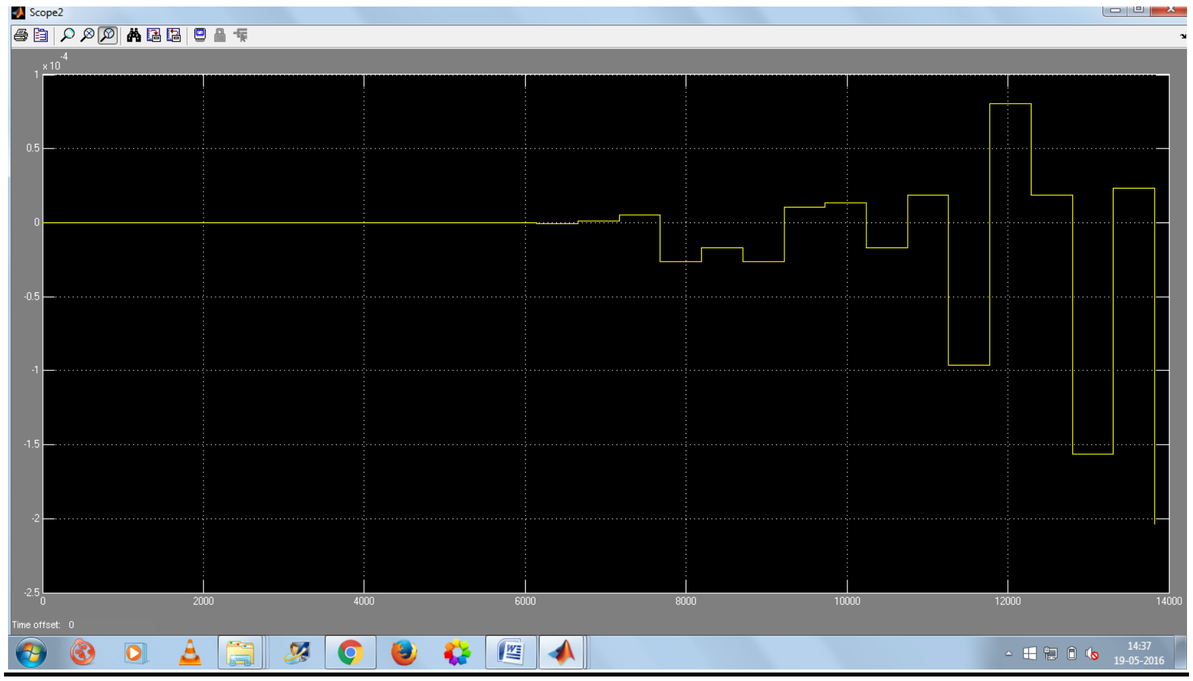


Figure 3.36 Output waveform of the speech signal

Figure 3.35 represents the output waveform on the time scope for a speech signal observed in figure 3.34.We have observed that the noise and other unwanted frequencies have been removed using multi-stage filtering technique.

<u>3.8 Implementation on FPGA</u>

This section presents the implementation of SDR on FPGA. The receivers are implemented on a XILINX Spartan XC 2S50 FPGA Development Board[15].
To implement the system in FPGA, it is necessary to describe our design models in Hardware Description Language(HDL). The software can then be used to synthesize the design onto the FPGA platform.

In this project, we chose not to describe our design directly in HDL, but rather used XILINX System Generator to speed up the development cycle.

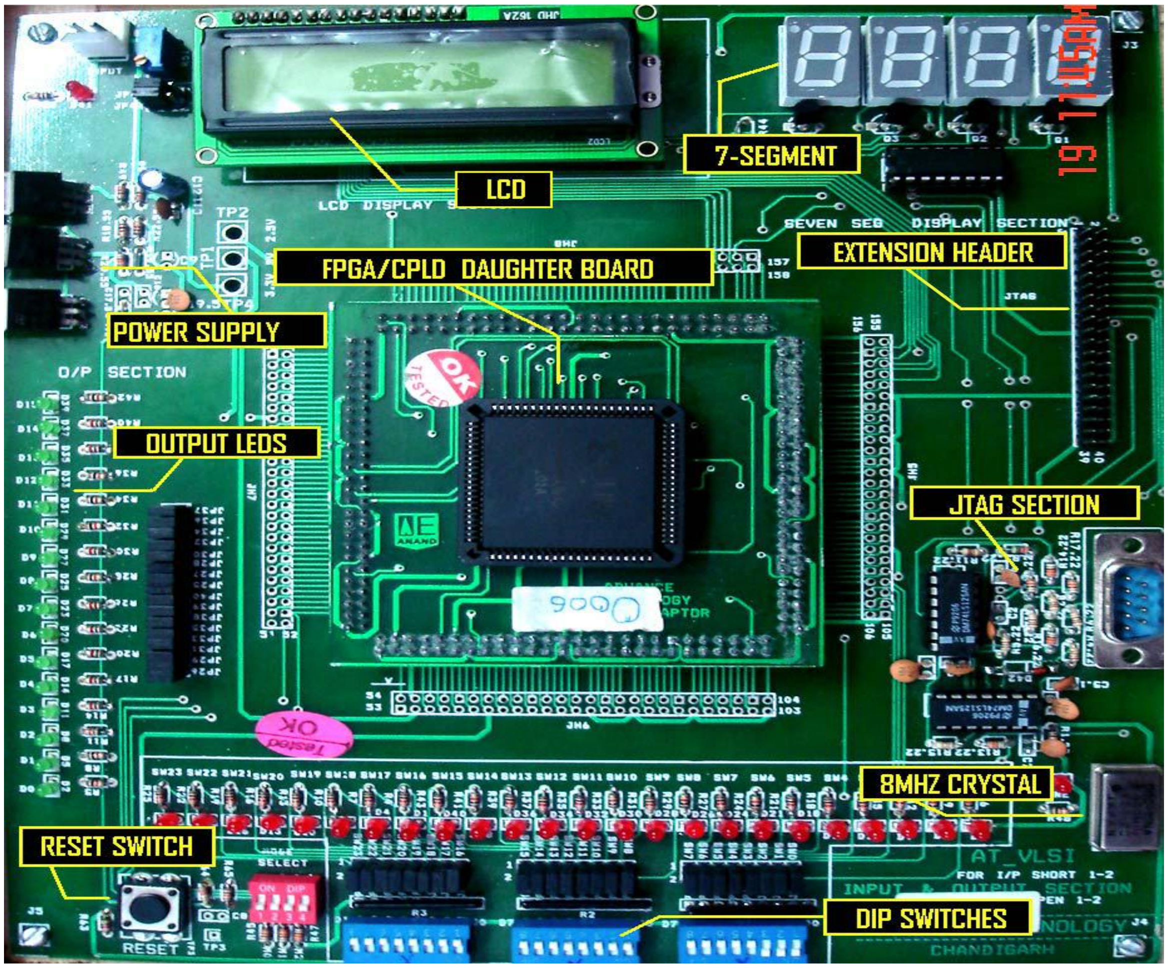


Figure 3.37 : VLSI Universal Board Viewer

The System Generator Design Flow

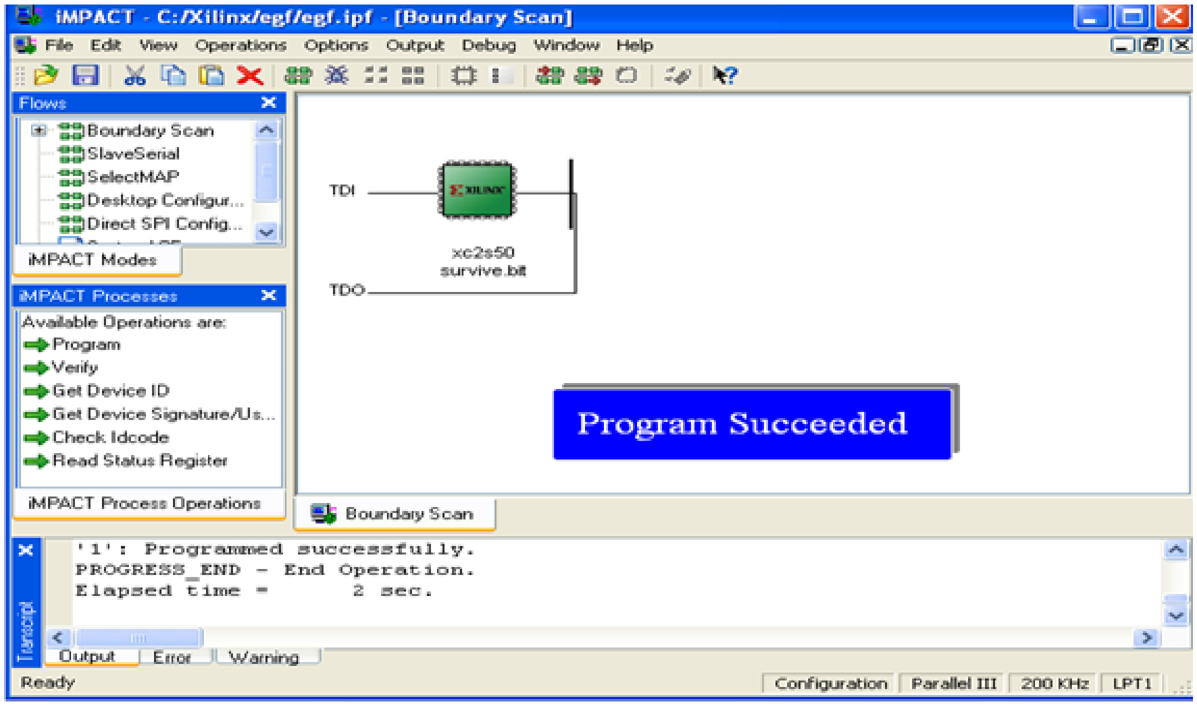
Simulink is a component of MATLAB software; it provides a graphical environment for creating and modelling dynamical systems. System Generator consists of a Simulink library called the Xilinx Blockset, and software to translate a Simulink model into a hardware realization of the model. System Generator maps system parameters defined in Simulink into entities and architechtures, ports, signals and attributes in the hardware realization.

System Testing in FPGA

1. The hardware model is configured to have its input and output ports connected properly to the appropriate pins of the FPGA chip.
2. VHDL software targetted into the FPGA platform is generated from the hardware model using Xilinx System Generator tool.
3. The input to the ADC comes from the designed transmitter.

Using XILINX System Generator Token tool

Once the design is completed, the hardware implementation files can be generated using the Generate button available on the System Generator token properties editor. If HDL Netlist is selected, it allows the FPGA implementation steps of the RTL synthesis ,place and route to be performed interactively using tool specific user interfaces. Alternatively, if Bitstream as the Compilation Target is selected, the System Generator automatically performs all the implementation steps. HDL Netlist > Select target part > select HDL language > Set the FPGA clock period > generate HDL Code.



```
'1': Programmed successfully.
PROGRESS_END - End Operation.
Elapsed time =       2 sec.
```

Figure 3.38 : Device Successfully programmed

CHAPTER 4

CONCLUSION

Software-Defined Radio (SDR) is a rapidly evolving technology that is receiving enormous recognition andgenerating widespread interest in the telecommunication industry. Over the last few years, analog radio systems are being replaced by digital radio systems for various radio applications in military, civilian and commercial spaces. In addition to this, programmable hardware modules are increasingly being used in digital radio systems at different functional levels. SDR technology aims to take advantage of these programmable hardware modules to build open-architecture based radio system software. SDR technology facilitates implementation of some of the functional modules in a radio system such as modulation / demodulation, signal generation, coding and link-layer protocols in software. This helps in building reconfigurable software radio systems where dynamic selection of parameters for each of the above-mentioned functional modules is possible. A complete hardware based radio system has limited utility since parameters for each of the functional modules are fixed. A radio system built using SDR technology extends the utility of the system for a wide range of applications that use different link-layer protocols and modulation/demodulation techniques. Commercial wireless communication industry is currently facing problems due to constant evolution of link-layer protocol standards (2.5G, 3G, and 4G), existence of incompatible wireless network technologies in different countries inhibiting deployment of global roaming facilities and problems in rolling-out new services/features due to wide-spread presence of legacy subscriber handsets. SDR technology promises to solve these problems by implementing the radio functionality as software modules running on a generic hardware platform. Further, multiple software modules implementing different standards can be present in the radio system. The system can take up different personalities depending on the software module being used. Also, the software modules that implement new services/features can be downloaded over-the-air onto the handsets. This kind of flexibility offered by SDR systems helps in dealing with problems due to differing standards and issues related to deployment of new services/features.[20,21]

In this project, we have looked into the concepts and design techniques of a new digital technology called Software Design Radio by implementing an AM/FM Digital Radio Transmitters and Receivers in software running on a FPGA platform. We have developed two models for demodulating AM and FM signals. Simulation of our models has shown that the general concepts and techniques of software radio are feasible in general and in particular to the implementation of an AM/FM radio receiver.

The two most important findings in this project are the use of multistage filtering technique to reduce the filter order, and the use of undersampling technique to lower the sampling rate requirement for the ADC.

- Firstly, in applying **multistage filtering technique**, we have mathematically formulated a formula for estimating the total filter order in an N-stage filter which reduces the number of filter taps approximately by a factor of Fs/4log2Fs) where Fs is the sampling rate of the filter input data.
- Secondly, in employing **undersampling technique** for our FM receiver, we have confirmed that it is a powerful technique for lowering the sampling rate requirement for the ADC and thus lowering the computational requirements for the entire system (e.g., lowering filter taps requirements).

Due to the scope of this project, only preliminary testing of the designed system in FPGA was carried out. Therefore, the most important future work extending on from this project would have to be comprehensive testing of the designed system. In this project we have designed **Low Pass Filter**, for our AM receiver system as well as for FM receiver system. These filters should ideally be high order filters but due to techniques specified above order is reduced which increases its efficiency.

Overall, this project has been a worthwhile effort in a sense that it has been a good learning process. This project posed an open-ended problem, for which we had to make our own choices and use our own initiative to find suitable solutions.[22,23]

Future Scope

The first generation of available SDR platforms occurred around 2004–2006. Technology has progressed since then and there have been significant improvements in signal processing performance, connectivity, and in the quality of RF components such as mixers and data converters. With current capabilities it has become possible to implement most narrowband communication schemes (e.g., GSM) though not without significant effort and expertise. However, in recent years there has been a movement toward wider band solutions such as wCDMA and OFDM technologies. The effect is that SDR platforms are challenged by increasing bandwidths, reducing minimum signal strengths, and reducing maximum allowable error vector magnitudes. Application specific SDR platforms can be constructed with a combination of available technologies[24]. General purpose experimental SDR platforms still face challenges and will be driven by three trends:

1. Increased capacity platform interfaces
2. An increasingly diverse range of processors.
3. Increased on-board processing capability.

The USRP2 from Mark Ettus is the first of the next generation of SDR platforms, and these trends are visible in the new design: significant on-board FPGA and a gigabit Ethernet connection.

References

[1]http://lr.ttu.ee/irm/sideseadmete_mudeldamine/PENTEK_Basics_of_software_radio.pdf

[2] http://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf

[3] http://www.ijareeie.com/upload/2013/july/14_THE%20SOFTWARE.pdf

[4] http://documents.mx/documents/amfm-software-radio-receiver-implementation-in-fpga.html

[5] http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1057&context=eesp

[6] http://scholar.utc.edu/cgi/viewcontent.cgi?article=1509&context=theses

[7] http://arstechnica.com/tech-policy/2012/07/how-software-defined-radio-could-revolutionize-wireless/

[8] http://www.cvarc.org/new-wp/download/technical/IntroToSDR.pdf

[9] http://www.dh1tw.de/understanding-the-sdr-concept

[10] http://in.mathworks.com/products/matlab/?requestedDomain=www.mathworks.com

[11] https://en.wikipedia.org/wiki/MATLAB

[12] http://in.mathworks.com/products/simulink/

[13] http://www.xilinx.com/products/design-tools/ise-design-suite.html

[14] http://in.mathworks.com/solutions/fpga-design/simulink-with-xilinx-system-generator-for-dsp.html

[15] http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/sysgen_gs.pdf

[16] P. B. Kenington "RF and Baseband Techniques for Software Defined Radio", 2005 :Artech House 5. Pentek, "software Defined Radio Hand book".

[17]Xinyu Xu, "Analysis and Implementation of Six Port Software defined Radio Receiver Platform" IEEE

[18] http://www.ti.com/solution/software-defined-radio-sdr-diagram

[19] http://www.dxzone.com/catalog/Technical_Reference/Software_Defined_Radio/

[20] http://www.mouser.com/pdfdocs/Software-Defined-Radio-Solutions-From-ADI.pdf

[21]https://books.google.co.in/books?id=sxAOBAAAQBAJ&pg=PA136&lpg=PA136&dq=sdr+references&source=bl&ots=VwkqJYndua&sig=Y5KH4Jz8EMPM6KPzrRN0h9c-5Ag&hl=en&sa=X&ved=0ahUKEwjOgePN2tHMAhUBVo4KHXcjA50Q6AEIPzAG#v=onepage&q=sdr%20references&f=false

[22]https://books.google.co.in/books?id=mpJJKoFYBL8C&pg=PA375&lpg=PA375&dq=sdr+references&source=bl&ots=xtDrnylFIL&sig=Cvr3nXsRFu1TAM_adL5NoKmIliQ&hl=en&sa=X&ved=0ahUKEwjOgePN2tHMAhUBVo4KHXcjA50Q6AEITDAJ#v=onepage&q=sdr%20references&f=false

[23] http://www.rtl-sdr.com/roundup-software-defined-radios/

[24] http://sdr-radio.com/