DESIGN AND IMPLEMENTATION OF SOFTWARE DEFINED RADIO SYSTEM ON FPGA FOR MEDICAL SIGNALS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRICAL ENGINEERING
(With Specialization in Measurement and Instrumentation)

By

PALLE SHAILAJA

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JUNE, 2008

I.D. No.-M.T 444 /2008-09/ VK

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Student declaration

I hereby declare that the work that is being presented in this dissertation report entitled **"Design and implementation of Software Defined Radio system for medical signals"** submitted in fulfillment of the requirements for the award of the degree of **Master of Technology in Electrical Engineering** with specialization in **Measurement & Instrumentation,** submitted in the **Department of Electrical Engineering, Indian Institute of Technology Roorkee,** Roorkee, is an authentic record of my own work carried out, under the guidance of **Dr.Vinod Kumar,** Professor, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee.

I have not submitted the matter embodied in this project report any other degree.

Date: 30-06-08

shailaja

Place: Roorkee **(PALLE SHAILAJA)**

CERTIFICATE

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

3/6/08

Prof. Dr. Vinod Kumar
Professor,
Department of Electrical Engineering,
Indian Institute of Technology Roorkee,
Roorkee – 247667.
India

Acknowledgements

First and foremost, my sincere regards to my supervisor Dr. Vinod Kumar, Professor, Department of Electrical Engineering, for his encouragement, valuable suggestions, amiable, amicable and caring nature. His guidance was helpful throughout this dissertation work.

Thanks are due to Prof. H. K. Verma, Group Leader (M&I) and Prof. Vinod Kumar (Chairman, DRC), for their moral support and advice.

I thank Prof. S. P. Gupta, Head of the Department, Dr. R.S.Anand, Assoc. Professor, Department of Electrical Engineering for their support.

I would also like to thank the Department of Electrical Engineering at the Indian Institute of Technology, Roorkee for giving me the opportunity to do a Master's degree in this honorable institution.

I am extremely grateful to my parents, friends, and well-wishers for their candid help, suggestions and persistent encouragement given to me at different stages of my work.

Shailaja

(PALLE SHAILAJA)

Abstract

The fourth generation of wireless communications will demand the use of software radio technology as the basic architecture to support multi-standard and multi-mode radio designs. The aim of this thesis is to design and implement a baseband section of software defined radio based wireless communication system for transmission of medical signals, which requires multi communication links. Software defined radio is a feasible solution for reconfigurable radios, which can perform different functions at different times on the same hardware. The baseband section of a wireless communication system is first simulated and then implemented in hardware. Baseband processing of medical signals involves functions such as forward error correction, modulation and digital up conversion on transmitter side and digital down conversion, demodulation and viterbi decoding on receiver side. System involving all these functions is developed. Digital Up/Down conversion is present in intermediate frequency section of Software defined radio. The performance of the baseband transmitter is analyzed using constellation and eye diagrams for different modulation techniques i.e QPSK, QAM techniques and different signal-to-noise ratios, while considering an additive white Gaussian noise channel. The performance of the receiver is analyzed by comparing the input and output waveforms by transmitting EEG and EMG signals. The performance of the system in is also analyzed by implementing the system in hardware using Xilinx Spartan 3E field programmable gate array. A comparison of the simulation results with the results obtained from implementing the system on Spartan 3E hardware is presented and discussed. It is shown that the simulation results and experimental results are similar.

Table of Contents

Table of Contents

List of Figures

List of Figures

List of Tables

Abbreviations

AWGN	Additive White Gaussian Noise
CDMA	Code Division Multiple Access
DA	Distributed Arithmetic
DSP	Digital Signal Processing
ECG	Electro Cardio Gram
EEG	ElectroEncephaloGram
EMG	ElectroMyeloGram
EDK	Embedded Design Kit
FIR	Finite duration Impulse Response
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
GSM	Global System for Mobile
GUI	Graphical User Interface
HDL	Hardware Descriptive Language
HR	Hardware Radio
IC	Integrated Circuit
IF	Intermediate Frequency
IP	Intellectual Property
IS	Interim Standard
ISE	Integrated Software Environment
ISDR	Ideal Software Defined Radio
JHDL	Java Hardware Descriptive Language

JTAG	Joint Test Action Group
LCM	Least Common Multiple
LO	Local Oscillator
MAC	Multiply and Accumulate
NCO	Numerically controlled oscillator
PC	Personal Computer
PCS	Personal Communication Services
PN	Pseudorandom Noise
PSD	Power Spectral Density
QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
RTL	Register Transfer Logic
SCR	Software Controlled Radio
SDR	Software Defined Radio
SNR	Signal-to-Noise Ratio
TACS	Total Access Communication System
USR	Ultimate Software Radio
VHSIC	Very High Speed Integrated Circuit
VHDL	VHSIC Hardware Description Language
VLSI	Very Large Scale Integration
WI-FI	Wireless-Fidelity
WLAN	Wireless Local Area Network
XST	Xilinx Synthesis Technology
XUP	Xilinx University Programme

CHAPTER I
INTRODUCTION

1.1 Introduction

Wireless communication networks have become more popular in the past two decades since the advent of cellular communications. The rapid growth in cellular communications has proved that wireless communication is viable for voice and data services. Traditional wireless devices are designed to deliver a single communication service using a particular standard [1]. With the steady increase of new wireless services and standards, single purpose devices with dedicated hardware resources can no longer meet the user's needs. It is also expensive to upgrade and maintain a wireless system each time a new standard comes into existence.

A feasible solution to make communication systems more flexible and user friendly can be achieved through the software defined radio (SDR) concept. Software defined radio refers to the class of reprogrammable or reconfigurable radios in which the same piece of hardware can perform different functions at different times [4]. Software defined radio is an emerging technology, for multi-service, multi-standard, multi-band, reconfigurable radio systems, which are reprogrammable by software.

A working definition of a software defined radio is a radio that is considerably defined in software and whose physical layer behavior can be significantly altered through changes to its software. Thus, the same piece of hardware can be used to realize different applications by modifying the software.

Software defined radio has generated tremendous interest in the wireless communication industry because of the wide-ranging economic and deployment benefits it offers [5]. Programmable hardware modules are increasingly being used in communication systems design at different functional levels.

Software defined radio (SDR) technology can be used to take advantage of programmable hardware modules to build open system architecture based on software. In this case, a variety of transceiver functions such as automatic gain control, frequency translation, filtering, modulation and demodulation can be integrated on a single hardware platform.

This could result in maximizing the number of radio functions for a particular application. Software defined radio offers the flexibility and upgradeability necessary to satisfy these requirements [5].

1.2 Communications in medical field

Transmission of medical data i.e. telemedicine has become increasingly possible due to influence of ongoing technical advances in multimedia, imaging, computers and information systems, as well as in telecommunications. A more advanced approach emerged in the 1960s, when the National Aeronautics and Space Administration first used radio signals to beam orbiting astronaut's physiological data back to Earth. Today, with the advent of high-speed broadband communications, telemedicine is becoming a part of everyday medical practice. Some people suffering from congestive heart failure equip their homes with remote monitoring systems that periodically transmit vital data, such as heart rate and weight, for expert evaluation and some hospitals teleconference on same high-speed networks.

Ambulances, Rural Health Centers (RHC) or other remote health location such as Ships navigating in wide seas and Continuous transmission of physiologic monitoring data from a passenger on a commercial airliner are common examples where transmission of medical data is required. In order to support the above different growing application fields, the integrated system had developed, which consists of a base unit and a telemedicine (mobile) unit. It can be used when handling emergency cases in ambulances, RHC or ships by using a mobile telemedicine unit at the emergency site and a base unit at the hospital-expert's site, and enables home telemonitoring, by installing the telemedicine unit at the patient's home while the base unit remains at the physician's office or hospital. The system allows the transmission of vital biosignals (3–12 lead ECG, SPO2, NIBP, IBP and Temp) and still images of the patient [6]. The transmission is performed through GSM mobile telecommunication network, through satellite links (where GSM is not available) or through Plain Old Telephony Systems (POTS) where available.

1.2.1 Networks used in transmission of medical data

(a) Local Area Network (LAN)

A LAN is used to connect digital devices such as personal computers and mainframe computers over a localized area such as a building or campus of a hospital, university or factory. LANs are normally installed and maintained by the organization and are essentially a small private computer network. Distances are small, 1-2 kilometers at the most, and this allows high data transmission rates. LANs are used to share information throughout an organization. In a hospital they are often used to access a Patient Master Index, medical record tracking, appointment booking systems and pathology test results. Any organization with two or more computers will generally have those networked [2].

(b) Wide Area Networks (WAN)

A wide area network is a network which covers a greater geographic area than a LAN. Generally, the dispersed sites are linked by lines leased from the telephone companies. Because of the distance involved, WANs have lower transmission rates than a LAN. In health a typical WAN would connect the LANs from all the hospitals in a city or region.

(c) Public Switched Telephone Network (PSTN)

This is the analogue telephone network which is the largest network in existence. It can be used to carry voice and, by using a modem, data as well. It consists of a large number of carriers whose networks are interconnected. Telecommunications companies are continually developing and offering an expanding range of value-added communication services beyond the basic telephone service. For district hospitals ISDN (integrated services digital network) services are the most relevant additional service. An ISDN connection is required for video-conferencing and also for high bandwidth access to the Internet.

(d) Private Automatic Branch Exchange (PABX)

Many organizations have a PABX to automatically switch calls between telephone extensions in an organization and to and from the public telephone network.

1.2.2 Transmission technologies for medical signals

The technologies involved in the transmission and receiving of the medical information (i.e. running the network) [2],

(a) Wires and cabling

Digital signals are sent as streams of electrical impulses. The existing telephone networks are generally built around copper wires and co-axial cables. The amount of information which can be carried (bandwidth) depends on the type of cable or wire. Most computer networks within LAN are connected with co-axial cable. Here, the quality of the connection and the potential bandwidth of the connection to the exchange fall rapidly with distance.

(b) Fibre optic cable

Fibre optic cable is very fine glass fibres which carry digital signals as pulses of light. Fibre optic cable has revolutionized telecommunications because it is able to carry enormous volumes of information. It is also not subject to electrical interference. The initial roll-out and maintenance is expensive.

(c) Satellite technology

Satellites are able to receive radio signals from earth and then retransmit them back. The device which does this is a satellite transponder. Most communications satellites are in geostationary orbit above the equator. The equipment required for transmitting a signal to a satellite in geostationary orbit and the equipment required to receive the signal from the satellite is expensive. Numbers of repeaters are required for covering large distances. For this reason, satellite technology is often used to broadcast signals.

(d) Radio-based technologies

This uses radio waves to send information through the air. The fixed equipment required is a transmitter and a receiver. Radiotelephony has been used in health care since it was first introduced commercially in the 1950s, but it was not until the introduction of hand-held transceivers and nation-wide cellular network coverage in the last decade. Radio frequencies are regulated in each country with frequency bands being reserved for various purposes. It may be used for radio broadcast or point-to-point telephony as in the mobile phone network. Its range varies with frequency and the power of the transmitter. Their advantages are mobility, reliability and not being reliant on wires and cabling.

1.3 Software defined radio for medical signals

The communication of medical data can occur among patients, specialists, clinics and hospitals. In medical application, there are two communication modes. One is point to point mode. It means the communication is going on between two users; the other is multipoint to multipoint mode. It means the communication is going on among many users' i.e. patients, specialists and medical centers.

The point to point mode is not enough to meet the actual requirements for applications. Because an application is usually asked to provide the following functions:

(1) The patient's data need to be sent to one or more specialists, a database, and the display terminal at the same time.

(2) A specialist needs to send information to the others in the same working group as well as to receive information from other specialists in a medical conference.

Therefore, a multipoint to multipoint communication mode is required for the communication of data. They have different communication mode and rates.

Internet can support various physical connections and provide multiple logical channels [3]. The communication may be difficult to manage. Firstly, each patient or specialist need to setup a connection with other specialists related to the same work. Secondly, each patient need to send the same data packages to many specialist and respond to the information sending backward. It is a complex process. The communication rate could be very low, for example, the telephone line has the rate of 9600-14400 bps.

Multipoint communication can be provided by using Wireless communications which provides higher bandwidth and data rates. for example, protocols such as IEEE 802.11 and its siblings. IEEE 802.11 allows for use of either the RF or IR mediums. When using RF, IEEE 802.11 uses the Frequency Hopping Spread Spectrum (FHSS) wideband system and has bandwidth limitation of 3 Mbps. IEEE 802.11b uses the Direct Sequence Spread Spectrum (DSSS) wideband system and can reach speeds up to 11Mbps.Bluetooth is a low cost, low power short range radio link for wireless connectivity between sensors and mobile devices and access points.

To meet these requirements communication system, software defined radio is required which provides different rates and modes at different times, and able to receive data packages and deliver them according to the request asked by users.

System with multi communication links is to provide patient monitoring during the prehospital transport and to offer health services, for people who live in underserved areas [6]. Therefore, medical information transmission becomes very crucial, since there is no transmission link stability guarantee. Selection of the communication links which includes VHF radio, internet, GSM/CDMA mobile phones, and GPRS depends on the availability of the local communication infrastructure.

The SDR system will exploit the advantage of wireless technology and combine it with other communication technologies such as VHF radio to meet different locals and geographic requirements.SDR system is reprogrammed to provide required transmission link at any time.

As mentioned above, the system for medical data communication consists of two main units, namely a Mobile Telemedicine unit that is placed in an ambulance, and a Base unit or Hospital unit. The Mobile Telemedicine unit is responsible for collecting medical information that includes biosignals and image from the patient and display the critical signals, e.g. ECG signal, blood pressure (BP) and fetal heart rate (FHR). The unit must also be able to write and to record the data, and support the data transaction with variety of communication links. Furthermore, the unit should be able to transmit the patient's biosignals to the base unit automatically. To support the functions, the mobile telemedicine unit is provided with processing unit and communication manager module.

Communication manager module is makes use of Software defined radio. It is a modem array that comprises of mixture a number of GSM, CDMA, radio, GPRS, and satellite (optional) modems i.e. system is bandwidth independent. The SDR system is reconfigured to select a most suitable communication link to transmit the data.

While advances in implantable RF transceiver chips is facilitating in-body medical communications, rapid developments in ultralow-power wireless body sensors is resulting in on-body communications[9]. Thus, creating a platform for body area network or BAN to wirelessly connect in/on-body medical sensors with monitoring tools and provide patient health data in real time.

To implement such a medical healthcare

service, advanced medical info-communication technology uses mobile communication systems based on advances in ultra wideband (UWB), software-defined radio (SDR), and multiple-input, multiple-output (MIMO) technologies. Software defined radio system provides multichannel system,can continuously monitor multiple health signs, such as heart rate, body temperature, pulse rate and respiration and transfer that data to a base station where a medical record is kept.

From above applications, it implied that SDR is used in telemedicine to support multi standard, multi data rates and wider bandwidth.

1.4 Motivation

Consider a typical communication system scenario where the user would like to have access to information through different wireless networks (e.g., wireless local area network (WLAN), Bluetooth, etc.), or a mobile phone user may be traveling between two regions around the globe, where the wireless technologies or standards are different. To utilize the services offered by the broad range of technology alternatives around the world, the user has to carry different devices due to incompatibility of systems and standards.

The practical solution to overcome this problem is to use a single device that can adapt to different technologies [5]. This could be possible using software defined radio, since it represents a radio that uses a reprogrammable hardware to create a generic hardware base. On top of the generic hardware platform, flexible software architecture is embedded.

The software allows for multiple protocols, fast upgrades, and complete reconfigurations of radio features and functions. Some of the attractive features of SDR are as

(1)Performance: The functionality of conventional radio architectures is usually determined primarily by hardware with minimal configurability through software. The hardware consists of the amplifiers, filters, mixers and oscillators dedicated to a particular mode of transmission. The software is confined to functions such as controlling the interface with the network, and error correction. Since the hardware dominates the design, upgrading a conventional radio design essentially means completely abandoning the old design and starting over again, resulting in a waste of time and resources. Software defined radio solves this problem by implementing radio functionalities as software modules running on generic hardware platforms. Since the radio functionalities are defined in software, when a new technology is introduced, it can be easily implemented by dynamic selection of parameters for its functional modules, i.e., reprogramming the software. Software defined radio provides a greater advantage to normal radio systems, since such radio systems can provide only fixed parameters with limited performance [5].

(2) **Flexibility**: The inflexibility of conventional radio systems limits the ability to get the right information to the right users at the right time. Conventional radio systems do not provide the wave form agility necessary to achieve this objective. With software defined radio, modulation waveforms and multiple air interface standards are possible. Thus; SDR platforms can serve a range of applications including analog cellular , digital cellular , personal communications services (PCS), wideband systems, spread spectrum, navigation waveforms (e.g., global positioning system) , emergency radio , public safety , and other radio systems[10]. Depending on the waveform, architecture, and implementation, a single software radio platform has the flexibility or potential to support a broad range of communication service.

(3) **Compatibility**: The concept of seamless global coverage requires that the radio support two distinct features. (a) Global roaming or seamless coverage across geographical regions (b) -interfacing with different systems and standards to provide seamless services at a fixed location. Existing technologies for voice ,video, and data use different packet structures, data types, and signal processing techniques. Integrated services can be obtained with either a single device capable of delivering various services or with a radio that can communicate with devices providing

complementary services. The supporting technologies and networks that the radio might have to use can vary with the physical location of the user. To successfully communicate with different systems, the radio has to communicate and decode the signals from devices using different air interfaces [5].

(4) **Cost**: Every time a new technology evolves, it results in the migration of functions from an older design to the new design. Implementing a new design involves manufacturing and testing. The cost of this process increases since upgrading to a newer design is not always possible in conventional systems. Software based radio can reduce the cost of manufacturing and testing, while providing a quick and easy way to upgrade the product to take advantage of newer signal processing techniques and new service applications [11].

1.5 Statement of Problem

The analysis presented in this thesis has many attractive features and several contributions to the current state of knowledge. The general and specific contributions of this research include the following:

(1) The development of a framework for the design and implementation of baseband section of software defined radio based communication systems.

(2) The analysis of a sample implementation of software defined radio based wireless communication system with coding (i.e., convolution encoding), viterbi decoding, puncturing and depuncturing, modulation and demodulation, Digital Up Conversion and Digital down conversion.

(3) The comparison of the performance of different modulation and demodulation techniques in a SDR implementation environment. The modulation techniques considered are quadrature phase shift keying (QPSK) and Quadrature Amplitude Modulation.

(4) Initiation of the development of a testbed for the design and implementation of SDR based wireless communication system.

(5) Presentation of results based on the simulation and actual experimentation.

(6) Evaluation of the performance of the SDR system in terms of signal-to-noise ratio in an additive white Guassian noise (AWGN) channel.

(7) Evaluation of the performance for the simulation of an SDR system and its implementation on the Xilinx Spartan 3E field programmable gate array (FPGA) platform [8]. The results obtained during simulation and experiments are compared.

1.6 Outline of Thesis

In this thesis, the fundamentals of software defined radio are first presented in Chapter II, which includes the general background information and various definitions for SDR. The presentation includes the difference between the SDR and conventional radio, characteristics and advantages of SDR, possible design issues, and the platform choices for implementing SDR based wireless communication systems.

In Chapter III, Various softwares and hardware kits utilized for this dissertation are explained.. The framework for the implementation of a wireless communication system in SDR is presented. This includes a brief introduction of wireless communication systems, with a block diagram of the end-to-end communication system architecture, and the methodology of implementation.

SDR system model is explained in Chapter IV. Different baseband functions like encoding/decoding techniques, modulation/demodulation methods and frequency translation techniques are explained in this chapter.

An illustrative baseband communication system implementation, simulation and results are presented and discussed in Chapter V. Results of the simulation including constellation diagrams, eye diagrams, output waveforms, for different modulation techniques are presented and analyzed. The system is implemented on the Xilinx Spartan 2E FPGA platform [8]. The results of the implementation are compared with the simulation results.

Chapter VI summarizes the content of the thesis. Also, possible extensions of the thesis are discussed.

1.7 Summary

This chapter gives brief description about Software defined radio based wireless communication system. It explains the statement of problem and how whole thesis organized.

CHAPTER II
FUNDAMENTALS OF SOFTWARE DEFINED RADIO

2.1 Introduction

Cellular communication systems have undergone tremendous growth since the early 1980's. As a result, mobile communication has become a major worldwide business. Because of this rapid growth, many analog and digital communication standards such as total access communication system (TACS), global system for mobile (GSM), digital cellular system-1800 (DCS-1800), interim standard-95 (IS-95), code division multiple access 2000 (CDMA2000), have been developed [7]. In fact, many competing standards have been introduced. The proliferation of standards is not only difficult for manufacturers but also for consumers. Manufacturers have to develop a new device for each technology or standard. This results in extra development costs and divided markets. It is also bad for consumers because users cannot use their mobile communication systems everywhere [10].

New generation wireless systems [1, 4] are being designed to provide a wide variety of multimedia services and to seamlessly switch between different wireless standards, such as wireless LAN and wideband CDMA. Each of these standards require different physical layer algorithms to be implemented. Also, algorithmic parameters such as the coding rate and constraint length for decoding need to be configured based on the channel environment. The wide range of configuration parameters and flexibility in the choice of algorithms to be implemented motivates the need for a software defined radio (SDR) solution.

Efforts to define a unique worldwide standard to overcome the above problems often results in a new standard [13]. A unique common worldwide standard has its own advantages, but the industrial competition between different manufacturers introduces many difficulties. Therefore, software defined radio (SDR) concept is considered by many as an emerging technology that offers potential pragmatic solutions. For example, a software implementation of the user terminal will be able to dynamically adapt to the radio environment in which it is located [14]. Software defined radio concepts can be viewed as a means to make users, service providers, and equipment manufacturers more independent of standards.

Software radio also describes radio functionalities defined by software. The possibility to define the typical functionalities of a radio interface by software will be an excellent opportunity to improve system performance. Currently the radio functionalities in communication systems are usually implemented by dedicated hardware. The presence of software defining the radio interface implies the use of digital signal processors (DSPs) replacing dedicated hardware to execute in real time, the necessary radio functions [15]. To completely realize a digital programmable transceiver, it is necessary for the digital signal processors and the programmable logic such as field programmable gate arrays (FPGAs) to have a high processing power. Although advances have been made in digital signal processing since the 1980's, the processing power of DSPs and FPGAs is still not enough to realize fully functional software defined radios. The required processing power is expected to become available in the near future.

It is expected that multiple radio access standards and systems will coexist in the same environment beyond 3G. The superior reconfigurability and reprogrammability of software defined radio (SDR) had made itself become the most promising technology to realize such a flexible radio system The future wireless environment is expected to consist of multiple radio access standards that provide users different level of mobility and bandwidth.

In this chapter, the definitions and meanings of the software defined radio are presented. The difference between SDR and conventional radio is highlighted, as well as the characteristics, advantages and disadvantages. Then, different hardware platforms available to implement SDR are discussed. The design issue in implementing SDR is highlighted.

2.2 Definition of Software Defined Radio

Because of the many features of SDR, there are many definitions available. The level of reconfigurability required to define a radio function in software is still not clear. A radio that includes a microprocessor or digital signal processor does not necessarily qualify as a software radio. However, a radio that defines in software its functions such as modulation, error correction, and encryption processes, exhibits some control over the RF hardware, and can be reprogrammed, qualifies as software defined radio [5]. The degree of configurability is largely determined by a complex interaction between a number of common issues in radio design, including system engineering, antenna form factors, RF electronics, baseband processing, speed and reconfigurability of the hardware and power supply management [5].The FCC has proposed to define SDR as a radio that includes a transmitter in which the operating parameters of the transmitter, including the frequency range, modulation type, and maximum radiated or conducted output power can be altered by making a change in software without any hardware.

One definition of SDR is provided by the SDR forum [16], is that SDR is the radio that accepts fully programmable traffic and control information and supports a broad range of frequencies, air interfaces, and application software. The SDR forum discriminates between different levels of flexibility in a radio. These are:

Hardware Radio (HR): In a HR, system attributes cannot be changed since the functionality of the hardware radio is fixed. However, this radio can use internal software as long as it cannot be changed externally.

Software Controlled Radio (SCR): This is the radio in which only the control functions are implemented in software. For example, the transmitted power level of a radio can be controlled by software, while all other functions are fixed in hardware. Current radio designs often fall under this category.

Software Defined Radio (SDR): These are radios that provide software control of almost every radio function, including modulation, multiplexing, amplification, superheterodyne mixers, multiple access and other transmitter and receiver processes. The software should have the capability to add new air interfaces without reloading the entire set of software.

Ideal Software Defined Radio (ISDR): This radio has the same functionality as the SDR, but it does not have an analog front-end (amplification, mixers, etc.), thereby unable to eliminate analog noise and distortions. The analog front-end contains an antenna, analog-to-digital converters (ADCs) and digital-to-analog converters (DACs), directly attached to it.

Ultimate Software Radio (USR): The USR is an ideal, flexible, small, lightweight, low-power radio which is fully programmable. Please note that, software radios use digital techniques, but software controlled digital radios are generally not software radios [2]. The difference between software controlled digital radios and software radios is the total programmability of software defined radio. This programmability includes programmable radio frequency bands, channel access modes, and modulation.

It is obvious that unique definition for the software radio concept may not be possible. The most common definitions are summed up below and quoted from [4], [16]]:

(1) "Flexible transceiver architecture, controlled and programmable by software."

(2) "Signal processing able to replace, as much as possible, radio functionalities."

(3) "A system with air interface downloadability. That is, it is possible to Dynamically reconfigure radio equipment by downloadable software, at every level of the protocol stack."

(4) "Software realization of terminals."
(5) "A transceiver with frequency band and radio channel bandwidth, modulation and coding scheme, radio resource and mobility management protocols, and user applications."

It appears that in SDR, the parameters of interest can be adapted and changed by the network operator, service provider, and end users. A software defined radio system can operate in multi-service environments. This means that the system is able to offer services of any already standardized systems or future ones, on any radio frequency band. The system is not constrained to a particular standard. For that reason the software radio system is very flexible. The compatibility of a software radio system with any defined mobile radio standard is guaranteed by its reconfigurability, which is achieved by DSP processors. These processors implement in real time radio interface and upper layer protocols.

A software defined radio not only transmits and receives signals but it does more in an advanced application [17]. Before transmission, SDR can distinguish the available transmission channel, select suitable channel modulation, direct the transmit beam in the direction of interest, check for proper power level and then transmit the signal. Similarly, on the receive path, apart from just receiving the signal, SDR can characterize the energy distribution in the desired channel and adjacent channels, provide adaptive equalization, null interference, approximate the dynamic properties of the desired signal, decode the channel modulation using appropriate schemes, correct errors through forward error correction (FEC), and hence help in obtaining the desired signal with less bit error rate (BER).

2.3 Software Defined Radio Concepts

In this section, the difference between conventional radio and software defined radio is presented. Also, the characteristics of SDR, its advantages and disadvantages are discussed.

2.3.1 Conventional Versus Software Defined Radio

To compare the functionalities of the conventional radio and SDR, we provide a tabulation of their functions in Table 2.1.

Conventional Radios	Software Defined Radios
Radio functionalities are primarily defined in hardware with minimum configurability in software.	Radio functionalities are defined in software.
Since the design is dominated by hardware upgrading the design is not possible.	Software based architecture allows for easy upgrade of the design without abandoning the older design
The user has to use different mobile devices due to incompatibility of standards	Global mobility can be achieved by downloading the appropriate air interface thus overcoming the incompatibility of standards.
Multi-function radios design including separate silicon for each system decreases the efficiency and becomes bulky.	Reprogrammability makes SDR to be efficient and compact
Results in waste of silicon area since each system has to be implemented separately	Silicon area is conserved by using the same chip to perform a function and changing the configurations during runtime to perform another function

Table 2.1 Difference between conventional and software defined radios [17]

Finally, software radio supports incremental service enhancements through a wide range of software tools. These tools assist in analyzing the radio environment, defining the required enhancements, prototyping incremental enhancements via the software, testing the enhancements in the radio environment, and finally delivering service enhancements via software and/or hardware [17].

2.3.2 Characteristics of Software Defined Radio

Consider a system model of a software defined radio shown in Figure 2.1. The receiver implemented with a smart antenna that provides a gain versus direction characteristic to minimize interference, multipath and noise. The smart antenna provides similar benefits for the transmitter. Digitization of the signal is carried out as close as possible to the antenna in the receiver using the analog-to-digital converter. Similarly, the signal is converted to the analog domain as late as possible in the transmitter using digital-to-analog converter. The digitization of the signal is mostly done in the intermediate frequency (IF) range. Software defined radio uses analog-to-digital converter to digitize the signal in the IF range, thus overcoming the problems like carrier offset and imaging involved in digitizing the signal using superheterodyne method commonly used in conventional radios.

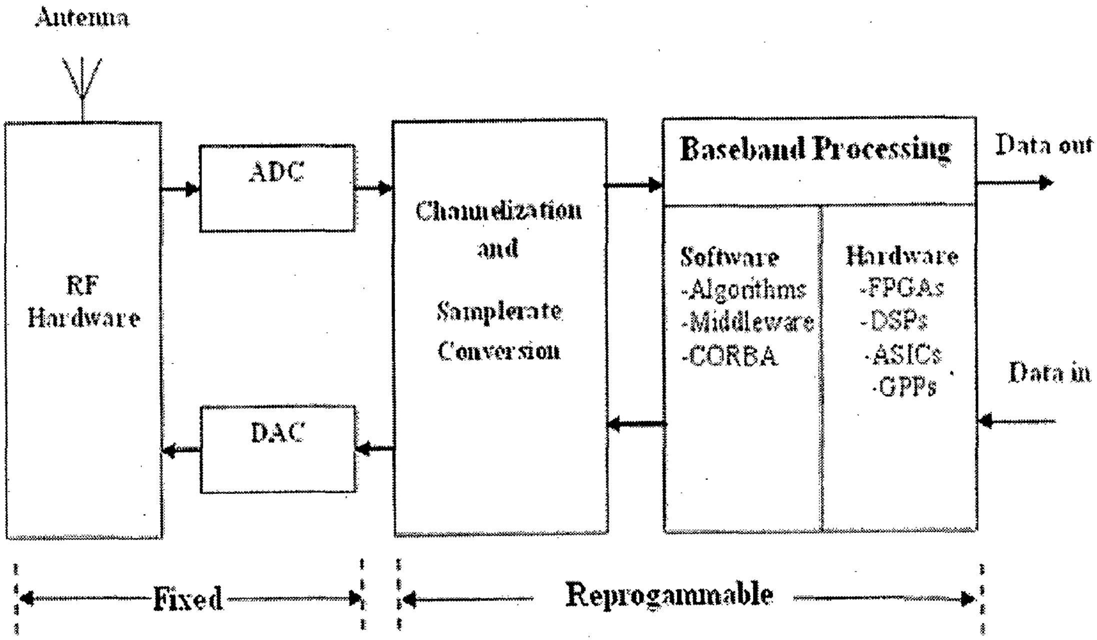


Figure 2.1 Model of software defined radio processes [5].

Channelization and sample rate conversion on the transmit path is used to interface the digital hardware to the digital-to-analog converter and to interface ADC to the processing hardware on the receive path. Baseband processing is performed in software using digital signal processors (DSPs), field programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), or general purpose processors (GPPs). The algorithm used to modulate or demodulate signal may use a variety of software methodologies, such as middleware, e.g., common object request broker architecture (CORBA) [5].

2.3.3 Advantages and Disadvantages of Software Defined Radio

According to its operational area an SDR can be

(i) **A multiband system** which is supporting more than one frequency band used by a wireless standard (e.g., GSM 900, GSM 1800, GSM 1900),

(ii) **A multistandard system** that is supporting more than one standard. Multistandard systems can work within one standard family (e.g., UTRA-FDD, UTRA-TDD for UMTS) or across different networks (e.g., DECT, GSM, UMTS, WLAN),

(iii) **A multiservice system** which provides different services (e.g., telephony, data, video streaming),

(iv) **A multichannel system** that supports two or more independent transmission and reception channels at the same time.

Following are the advantages of SDR technology:

Reconfigurability: SDR allows co-existence of multiple software Module implementing different standards on the same system allowing dynamic configuration of the system by just selecting the appropriate software module to run. This 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.

Ubiquitous Connectivity : SDR enables implementation of air interface 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.

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.

Easy upgradeability to new communication standards. Signal generation and analysis are largely performed by routines programmed into the Hardware. When new standards emerge, it's easy to create new programs for the new functions and distribute them to the owners of existing instruments via firmware upgrades.

Improved throughput due to faster frequency switching and signal analysis Wide bandwidth A/D converters and fast DSP devices can process large FFTs very efficiently. For example, a DSP-based analyzer can provide measurement times several orders of magnitude faster than traditional spectrum analyzers, under conditions of wide spans and narrow resolution bandwidths. Direct digital synthesis provides significantly faster frequency switching than traditional approaches allow. Fast frequency switching will improve the throughput of both signal generators and signal analyzers.

Faster time to market for test instruments. Test equipment manufacturers can leverage the capability of leading-edge, commercially available signal processing devices and achieve instrument-level performance from them. This reduces the amount of development required for test instruments dramatically. Also, the basic digital design can be shared across a range of instruments, further reducing development costs

With SDR, the same piece of hardware will be configured to perform different functions. The reconfigurability of the platform will ensure hardware reusability. System reprogrammability allows hardware reuse until a new generation of hardware platforms is available. This will provide cost and time savings. Manufacturers will not be limited to reduced hardware platform set. As a consequence, mass production will allow lowered costs [5].

Another advantage of SDR would be the possibility to improve the software in successive steps, and the correction of software errors and bugs discovered during the operation.

In addition, SDR can enhance the interoperability of different systems in many applications such as the military, law enforcement, or search and rescue teams. Incompatibility of radio systems that has always hindered the seamless operation of the military, the law enforcement agencies and many rescue teams, will be eliminated.

With the increase of channel data rates through multiplexing and spectrum spreading, SDR could be used in cellular networks, GSM based PCS network, and future generation systems network. A new approach to wireless base station design using SDR has the potential of offering significant benefits such as reduced size, complexity, and power consumption. More importantly, SDR can support a variety of air interface standards, modulation schemes and protocols, simultaneously. Some commercial telephone service providers have begun expressing interest in the SDR economic benefits in long term [10]. More highlights on the benefits of SDR are given in Section 1.2.

While SDRs offer benefits as outlined above, there are drawbacks in the design and implementation of SDR. Those include:

(1) The difficulty of designing software for various target systems or standards.

(2) The difficulty of designing air interfaces to digital signals and algorithms for different standards.

(2) The problem of poor dynamic range in some communication systems design

SDR technology may not be suitable for all kinds of radio equipment due to higher power consumption, higher processing power (MIPS) requirement and higher initial cost.

Interoperability	Supports of multiple standards through multimode, multiband radio capabilites
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 ,easier deployment

Table 2.2 Advantages of Software Defined Radio

2.4 Software Defined Radio Implementation Platforms

As indicated above, the global trend in the communication industry is to replace hardware by software, because of software flexibility. Real time software defined radio design can be implemented using a variety of digital hardware namely (a) field programmable gate arrays, (b) digital signal processors, (c) application specific integrated circuits and (d) general purpose processors. The different implementation platforms are shown in Figure 2.2. All the four platforms shown in Figure 2.2 possess a level of reprogrammability or reconfigurability (i.e., the ability to modify the hardware or software) [25]. The DSP platform is essentially a microprocessor based system optimized for digital signal processing applications [5], [17]. DSPs can be programmed repeatedly with a high level language such as C, MATLAB [5]. Modifications and upgrades to the design are made through these high level languages, thus reducing the design times for each iteration. The flexibility offered by the digital signal processor comes at the cost of efficiency. When there are several computations to be performed, parallel executions of these computations will slow down the rate at which data is processed and this leads to the use of more than one DSP. This solution is limited since synchronizing several DSPs is difficult.

A field programmable gate array is a general purpose integrated circuit that is programmed by the designer rather than the device manufacturer. A unique feature of FPGA is that it can be reprogrammed, even after it has been deployed into a system. Field programmable gate array is programmed by downloading a configuration program (bitstream) into the static on-chip random access memory [11]. This is similar to the object code of a microprocessor, in which the bitstream is the product of compilation tools that translate the high level abstractions produced by a designer into equivalent but low level executable code [18].

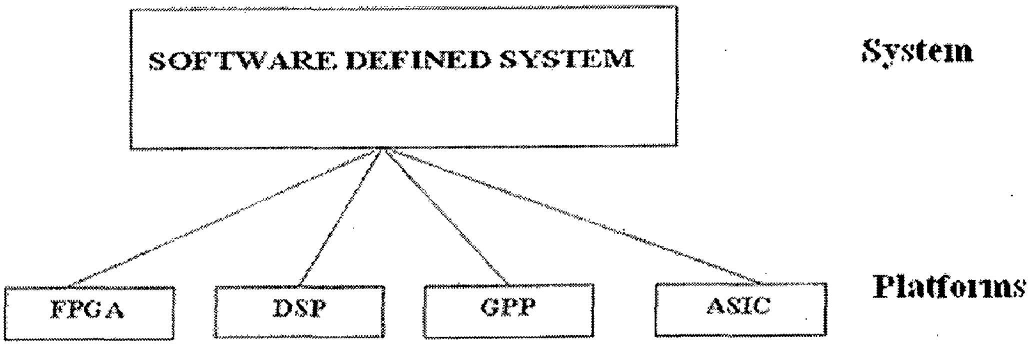


Figure 2.2: Hardware implementation platforms for SDR

Field programmable gate arrays were designed for multilevel circuits to handle complex circuits on a single chip. Since they are reprogrammable, their configurations can be easily changed to upgrade systems or correct system bugs, making it ideal for prototyping. Field programmable gate arrays are now used in various configurations, as in multimode systems, and are very useful in meeting the needs of a software defined radio implementation [18].

Application specific integrated circuits (ASICs) implement the system circuitry in fixed hardware, resulting in the most optimized implementation of the circuit in terms of speed and power consumption. However, ASIC design requires sophisticated circuit design and layout software tools [5]. Also, as the name implies, their use are for specific application and not subject to modification at a later date.A general purpose processor is similar to DSP as a hardware platform in the design of software defined radio. Like DSP, it offers flexibility and ease of design. Radio functionalities can be implemented in high level languages such as C and C++. Designers can use the familiar approaches of object oriented programming anddebugging to develop real time software radio systems. This increases productivity significantly and reduces system development time [19].

Digital signal processor is the most generalized type of hardware that can be programmed to perform various functions, while ASIC is the most specialized and can be used only in specific application. Field programmable gate arrays offers a compromise in flexibility between ASIC and DSP platforms. In general, these hardware components constitute design spaces that trade flexibility, processing speed, and power consumption among other things. There should be a tradeoff between the maximum flexibility and high power consumption of DSP platforms to minimum flexibility and less power consumption of ASICs compared to FPGAs, which have good hardware optimization. Recently, FPGAs have become increasingly popular due to their ability to reduce design and development cycle time. Furthermore, latest FPGAs come with intellectual property (IP) cores, which are used for specific applications [11].

There are other advantages of using FPGAs instead of DSPs for signal processing in commercial telecommunication systems. The power consumption is lower; the size is smaller, quicker to use and the costs are much lower in comparison to DSPs. Since the chip can be reused after fixing the bugs or upgrading a system, they are ideal for prototyping and testing the circuit design. Since FPGAs are reprogrammable, one chip can be configured to perform more than one function and the configurations can be changed during run time [11].

2.5 Technical Challenges

This section discusses the technical issues, which have to be solved before software radio can be commercially available. The important technical issues involved in the development of a software radio system are as follows:

(1)In transceivers, the border between analog and digital domain should be moved closer, as much as possible, towards the RF. This requires ADC and DAC wideband converters placed as near as possible to the antenna. The border between the analog and digital domain is not reduced to a extent software defined radio. Much research has been carried out in the wideband transceiver realization [16]. The primary goal of this transceiver was to extend the digital domain at the IF stage and keeping the RF stage analog [14].

(2) The process of replacement of dedicated hardware in communication systems with DSPs or FPGAs should be further developed. In other words, we need to define the radio functionalities as much as possible in software. This opens the way to two possible horizons: software implementation of baseband functions, such as coding, modulation, equalization and pulse shaping; and reprogrammability of the system to guarantee multi-standard operation. Though DSP technology has been used in implementing the baseband processing in base stations, it is not possible to categorize it as SDR since not all baseband functionalities are implemented in DSPs. Also, the software is limited and preloaded; therefore the system is constrained to a specific radio interface and cannot be reconfigured [14]. Hence, implementing communication functions in software presents a major challenge in practical systems.

(3) Analog-to-digital and digital-to-analog conversions for the ideal software defined radio are difficult to achieve. In practice, the selection requires trading power consumption, dynamic range and bandwidth. Current conversion technology is limited and is often the weak link in the overall system design. There are post digitization techniques based on multirate digital signal processing that can be used to improve the flexibility of the digitization process [5], [20].

(4) Power management is also a major challenge. For example, sleep modes of DSPs or other hardware save power but introduce a probability that the radio will be asleep during a paging message. A possible solution is a structured timing of paging messages, which reduces the probability of a miss, and further conserves battery life [21], [22].

(5)The clock generation and distribution is another challenge in SDR design. Every standard such as GSM or IS-95 has its own clock rate. Using one reference oscillator per standard may increase parts count, increase complexity, and therefore cost. A single master clock may use the least common multiple (LCM) of the required clocks, but this leads to a high clock rate, which is power inefficient. A possible solution is to use normalize standards to avoid clock rates with large LCMs [17].

(6) Receiver complexity is typically four or more times the transmitter complexity [5]. Thus, the receiver architecture has a first order impact on handset cost. The challenge is to develop a simple receiver. With the current technology, the support of many standards leads to complex and power inefficient solutions. Application specific integrated circuits are power efficient but inflexible. Field programmable gate arrays could be a possible solution. Hybrids of platform implementation could be utilized. The ideal radio frequency stage for SDR should incorporate flexibility in selection of power gain, bandwidth, dynamic range, etc. Achieving strict flexibility is impractical and trade-offs must be made [5].

2.6 Summary

The fundamentals of SDR are presented in this chapter. It also dealt with definition and concepts of SDR. The difference between the conventional radio and SDR, characteristics, advantages and disadvantages were also presented. The choices of hardware available for real time implementation and technical challenges involved in implementation were discussed.

CHAPTER III

FRAMEWORK FOR SDR BASED DESIGN

3.1 Introduction

Signal processing systems for transmission of medical signals will have to operate in rapidly changing environments. To suitably adapt to the varying requirements, control strategies targeted at selecting and tuning the signal processing algorithms need to be developed

In this chapter, a conventional wireless communication system model used in telemedicine is briefly reviewed. Raw medical data cann't be transmitted as in its natural form as there is provision for noise addition. The operations performed on data to transmit through the channel are explained. This is followed by explanation of softwares used. The softwares were utilized for the simulation, verification and for the validation purposes. The hardware here means the FPGA development kit, on which the developed design is implemented [16]. In this thesis, FPGA development kits are utilized which were provided as a part of the Xilinx University Program (XUP). The XUP Spartan 3E FPGA development kits are used. Finally, steps involved in implementing the design on a reconfigurable computing platform are presented. In this thesis, only the baseband section of a communication system is modeled and simulated.

3.2 Wireless Communication System Model for transmission of medical signals.

The generic wireless communication system consists of a transmitter, channel and a receiver. The functional block diagram of the digital transceiver is shown in Figure 3.1 [5].

3.2.1 Radio Frequency Section

The radio frequency (RF) section is responsible for transmitting and receiving the RF signal and converting the RF signal into an intermediate frequency (IF) signal. The RF section consists of antennas and analog hardware modules. The RF front-end is designed in such a way to reduce interference, multipath and noise. The RF front-end on the receive side performs RF amplification and down conversion from RF to IF. On the transmit side, the RF section performs analog up conversion and RF power amplification.

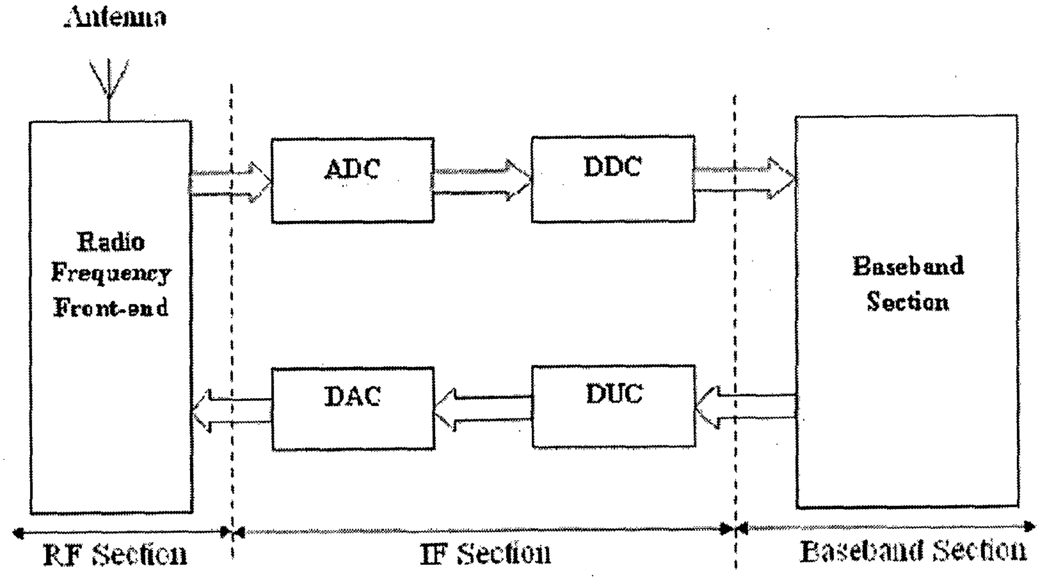


Figure 3.1: Functional block diagram of a wireless communication system [5].

3.2.2 Intermediate Frequency Section

The ADC/DAC performs analog-to-digital conversion on the receive path, and digital to analog conversion on the transmit path. These blocks interface between the analog and digital sections of the radio system. Usually, the above conversion takes place in the IF stage. Digitizing the signal with an ADC eliminates the last stage in the conventional model, where problems such as carrier offset and imaging are encountered.

As the names imply, the digital down converter (DDC) and digital up converter (DUC) perform digital down conversion on the receive path and digital up conversion on the transmit path, respectively[19]. Digital filtering and sample rate conversion are often needed to interface the output of the ADC to the processing hardware at the receiver. The same happens in the reverse direction in the transmitter, where digital filtering and sample rate conversion are necessary to interface the digital hardware to the DAC that converts the modulated waveform to an analog waveform.

3.2.3 Baseband Section

The baseband section performs operations, such as error correction, equalization, frequency hopping, modulation, demodulation, spreading, despreading and timing recovery. Forward error correction is a method of obtaining error control in data transmission in which the transmitter sends redundant data and the receiver recognizes only the portion of the data that contains no apparent errors. Equalization is done to counteract the inter symbol interference in the channel. Frequency hopping and spreading is used to minimize unauthorized interception or jamming of the communication system by repeated switching of frequencies during radio transmission using a specified algorithm. In a wireless communication system, many modulation techniques such as MPSK, QPSK, DPSK, etc., are used. The DDC, DUC and the baseband processing requires large computing power and these modules are generally implemented using DSPs, FPGAs, and ASICs Details of the specific blocks that were implemented in this thesis are provided in Chapter IV.

3.3 Softwares used

The following softwares were used in this thesis.

1. Hardware Descriptive Language (HDL)
2. Matlab 6000b

3. Xilinx System Generator 9.2i
4. Xilinx ISE 9.2i
5. Drivers for FPGA development kits

3.3.1 Hardware Descriptive Language

A Hardware Descriptive Language (HDL) is a computer language designed for formal description of electronic circuits. It can describe a circuit operation, its structure, and the stimuli input to verify the operation (using simulation). A HDL model is a text-based description of the temporal behaviour and the structure of an electronic system. In contrast to a software programming language, the HDL syntax and semantics include explicit notations for expressing time and concurrent execution, which are the primary attributes of hardware. Languages, whose only characteristics are to express circuit connectivity within a hierarchy of blocks, are properly classified as netlist languages. One of the most popular netlist formats and industry standards is EDIF, acronym for Electronic Data Interchange Format [17].

The system that is modeled using System Generator can be compiled into low level representations. That is, the algorithms used to model the system can be broken into processes and coded in VHDL, which gives the description of the hardware. More precisely, it describes the architecture of the system i.e., its components and interconnections. The VHDL description results in a collection of HDL files that implement the design and are later used for HDL and hardware co-simulation [38]. If required, test benches can also be created with other descriptions of the model. One of the most important applications of VHDL is to capture the performance specification for a circuit, in the form of a "test bench". Test benches are VHDL descriptions of circuit inputs and corresponding expected outputs that verify the behavior of a circuit over time. Test benches are an integral part of any VHDL project and should be created in tandem with other descriptions of the circuit.

Simulation may be defined as the process of verifying the functional characteristics of models at any level of abstraction. VHDL simulation verifies the functionality of the system i.e., given the expected inputs and test whether the outputs are as expected or not. A VHDL testbench and data vectors, which has been created by System Generator for DSP represents the inputs and expected outputs seen in the MATLAB SIMULINK simulation, and allow the designer to easily see any discrepancies between the SIMULINK and VHDL simulation results. ModelSim is equired, when HDL co-simulation is done. ModelSim provides a complete HDL simulation environment that enables to verify the functional and timing models of the design, and the VHDL source code.

3.3.2 MATLAB

MATLAB, The language of Technical Computing, is a high level language and interactive environment that enables to perform computationally intensive tasks faster than with traditional programming languages such as C, C++ and Fortran. MATLAB [19] can be utilized in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

The algorithms and concepts used to define the system is modeled using high level software languages like MATLAB, SIMULINK and C. The Xilinx's System Generator for DSP is a new tool, which comes with a predefined block set along with MATLAB SIMULINK software packet and can be used to implement the algorithms [12]. These high level languages can also be used to verify the accuracy of the algorithms. However they do not directly aid in the hardware implementation.MATLAB is widely used by many DSP algorithm developers. It is considered the best environment for algorithm development and debugging because of its built-in functions and toolbox extensions for communications, signal processing and wavelet processing.

In addition to the intellectual property functions provided in MATLAB, the software packet is uniquely adept with vector- and array-based waveform data at the core of algorithms, which is suitable for applications such as wireless communications and image processing. MATLAB SIMULINK is fully integrated with the MATLAB engine for visual data flow environment for modeling and simulation of dynamic systems. In addition to the graphical block editor, event-driven simulator, and extensive library of parameterizable functions, it has blocksets for Communications, DSP, wavelets and many more. MATLAB SIMULINK is used in this thesis as the high level development tool in the design process.

3.3.3 Xilinx System Generator

Xilinx System Generator [12], is a system-level modeling tool that aids in FPGA hardware design. It extends SIMULINK capabilities in many ways to provide a modeling environment well suited for hardware design. System Generator is a tool for developing and debugging high performance systems based on advanced Xilinx FPGAs. System Generator and MATLAB SIMULINK tool, provide the graphical design environment commonly used by FPGA designers [24].

Xilinx's System Generator tool was the first tool to bridge the gap between DSP and FPGA applications [9]. System Generator along with SIMULINK is a powerful visual data flow environment ideally suited for modeling and simulating algorithms, and allows the developer to automatically generate bit- and cycle-accurate hardware implementation from the system model [26]. System Generator automates the design process, debugs, and implements and verifies the Xilinx-based FPGAs. It comes with IP core libraries for high-level modeling and automatic validation code generation, and also provides a high-speed hardware description language (HDL) co-simulation interface, system-level resource estimation, and high-speed hardware co-simulation interfaces for design verification using FPGA hardware platforms [9].

System Generator provides high-level abstractions that are automatically characteristics of models at any level of abstraction. VHDL simulation verifies the functionality of the system i.e., given the expected inputs and test whether the outputs are as expected or not. A VHDL testbench and data vectors, which has been created by System Generator for DSP represents the inputs and expected outputs seen in the MATLAB SIMULINK simulation, and allow the designer to easily see any discrepancies between the SIMULINK and VHDL simulation results. ModelSim is equired, when HDL co-simulation is done. ModelSim provides a complete HDL simulation environment that enables to verify the functional and timing models of the design, and the VHDL source code.

3.3.2 MATLAB

MATLAB, The language of Technical Computing, is a high level language and interactive environment that enables to perform computationally intensive tasks faster than with traditional programming languages such as C, C++ and Fortran. MATLAB [19] can be utilized in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

The algorithms and concepts used to define the system is modeled using high level software languages like MATLAB, SIMULINK and C. The Xilinx's System Generator for DSP is a new tool, which comes with a predefined block set along with MATLAB SIMULINK software packet and can be used to implement the algorithms [12]. These high level languages can also be used to verify the accuracy of the algorithms. However they do not directly aid in the hardware implementation.MATLAB is widely used by many DSP algorithm developers. It is considered the best environment for algorithm development and debugging because of its built-in functions and toolbox extensions for communications, signal processing and wavelet processing.

In addition to the intellectual property functions provided in MATLAB, the software packet is uniquely adept with vector- and array-based waveform data at the core of algorithms, which is suitable for applications such as wireless communications and image processing. MATLAB SIMULINK is fully integrated with the MATLAB engine for visual data flow environment for modeling and simulation of dynamic systems. In addition to the graphical block editor, event-driven simulator, and extensive library of parameterizable functions, it has blocksets for Communications, DSP, wavelets and many more. MATLAB SIMULINK is used in this thesis as the high level development tool in the design process.

3.3.3 Xilinx System Generator

Xilinx System Generator [12], is a system-level modeling tool that aids in FPGA hardware design. It extends SIMULINK capabilities in many ways to provide a modeling environment well suited for hardware design. System Generator is a tool for developing and debugging high performance systems based on advanced Xilinx FPGAs. System Generator and MATLAB SIMULINK tool, provide the graphical design environment commonly used by FPGA designers [24].

Xilinx's System Generator tool was the first tool to bridge the gap between DSP and FPGA applications [9]. System Generator along with SIMULINK is a powerful visual data flow environment ideally suited for modeling and simulating algorithms, and allows the developer to automatically generate bit- and cycle-accurate hardware implementation from the system model [26]. System Generator automates the design process, debugs, and implements and verifies the Xilinx-based FPGAs. It comes with IP core libraries for high-level modeling and automatic validation code generation, and also provides a high-speed hardware description language (HDL) co-simulation interface, system-level resource estimation, and high-speed hardware co-simulation interfaces for design verification using FPGA hardware platforms [9].

System Generator provides high-level abstractions that are automatically compiled into FPGA bitstream. It is delivered both with a predefined Xilinx blockset library and other languages such as VHDL which are commonly used in FPGA platforms. Finally, it facilitates the design at the system level, and allows simulation, implementation, and verification within the same environment, usually without writing a single line of HDL code or even looking at the Xilinx integrated software environment (ISE) tools [24].

In spite of these advantages, System Generator fails to satisfy certain needs of the algorithms developing functions like handling matrix operations and vector based processing. Examples of such algorithms include linear algebra, which involves matrix inverse and factorization operations, and complex number operations such as calculating magnitude and angle, and normalizing complex numbers [26].

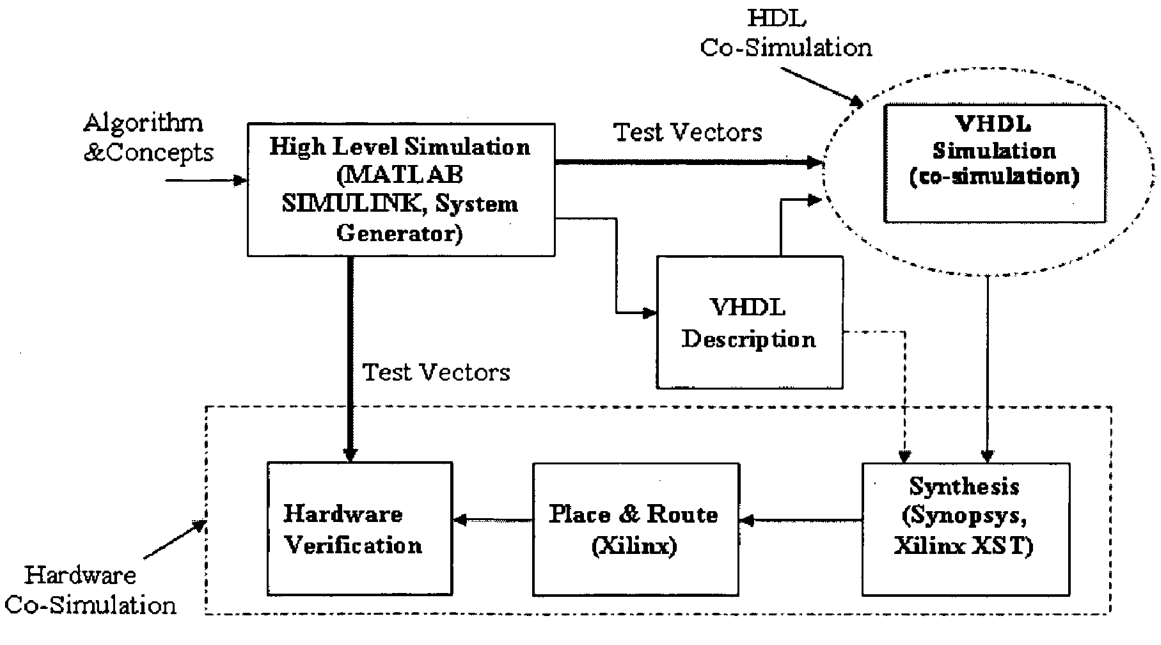


Figure 3.2: Design process for reconfigurable computing.

Test vectors can also be created using the System Generator. To construct test vectors, System Generator simulates the design in SIMULINK, and saves the values of the outputs [27]. These test vectors are later used to check the differences between the SIMULINK simulation, HDL simulation and hardware implementation of the model.

3.3.4 Xilinx ISE 9.2i

Xilinx ISE stands for Xilinx Integrated System Environment (ISE). ISE controls all aspects of the design flow. Through the Project Navigator interface, one can access all of the design entry and design implementation tools. One can also access the files and documents associated with your project. Project Navigator maintains a flat directory structure; therefore, the project should be updated through the use of snapshots.

The Xilinx ISE [22] system is an integrated design environment that consists of a set of programs to create (capture), simulate and implement digital designs in a FPGA or CPLD target device. All the tools use a graphical user interface (GUI) that allows all programs to be executed from toolbars, menus or icons.Xilnx System generator and ISE softwares provides Hardware Co-Simulation. Hardware co-simulation capability accelerates simulation and verification of design in hardware. System Generator's hardware-in-the-loop co-simulation [24] interfaces make a push-button flow and bring the full power of MATLAB and SIMULINK analysis functions to hardware verification.

Once VHDL has been generated by System Generator and before the design is implemented in FPGA, it is necessary that it is synthesized for optimal FPGA implementations. In synthesis, the conceptual HDL design definition is used to generate the logical or physical representation for the targeted silicon device. That is, synthesis maps the HDL to the gate level representation [24], [26]. The VHDL modules can be transferred to the hardware using Xilinx synthesis technology (XST) synthesis tool, which comes with Xilinx's ISE [27].

The next step is to place and route the design in order to verify it on the FPGA. This is achieved using the Xilinx's ISE implementation tools. The place and route function in FPGA design places the synthesized subsystems into FPGA locations and makes connections between these subsystems, enabling their operation as an integrated system [26].Placing and routing is followed by hardware verification. The design is implemented on the desired hardware. Hardware verification checks if the module created in high level simulation works well on the desired FPGA device. Test vectors are used to check any discrepancies between the simulation and the hardware implementation.

3.3.5 Drivers for FPGA development kits

Xilinx EDK [30] is the Embedded Development Kit, it provides the necessary drivers for the various FPGA development kits. EDK is a suite of tools and IP that enables to design a complete embedded processor system for implementation in a Xilinx FPGA device. To run EDK, ISE must be installed as well. Think of it as an Umbrella covering all things related to embedded processor systems and their design.

For any hardware to be interfaced with a Personal Computer, it should be loaded with drivers which enable communication between the hardware and the Personal Computer. These are again available from the vendor from whom the FPGA development kit is procured. Synthesis on the FPGA development kit cannot be done without the driver softwares installed.

3.4 XUP Spartan 3E FPGA development kit

The Spartan-3E [28] FPGA Starter Kit is a complete development board solution. It features in brief are as follows:-

- Device Family Support

1. Spartan-3E
2. CoolRunner-II

- Key Features

1. Xilinx Devices: Spartan-3E (XC3S500E-4FG320C)

- ♦ Up to 232 user-I/O pins
- ♦ 320-pin FBGA package
- ♦ Over 10,000 logic cells
- ♦ Parallel NOR Flash configuration
- ♦ MultiBoot FPGA configuration from Parallel NOR Flash PROM
- ♦ SPI serial Flash configuration

2. Embedded development
   - ♦ MicroBlaze™ 32-bit embedded RISC processor
   - ♦ PicoBlaze™ 8-bit embedded controller
   - ♦ DDR memory interfaces

2. CoolRunnerTM-II (XC2C64A-5VQ44C) and Platform Flash (XCF04S-VO20C)
3. Clocks: 50 MHz crystal clock oscillator
4. Memory: 128 Mbit Parallel Flash, 16 Mbit SPI Flash, 64 MByte DDR SDRAM

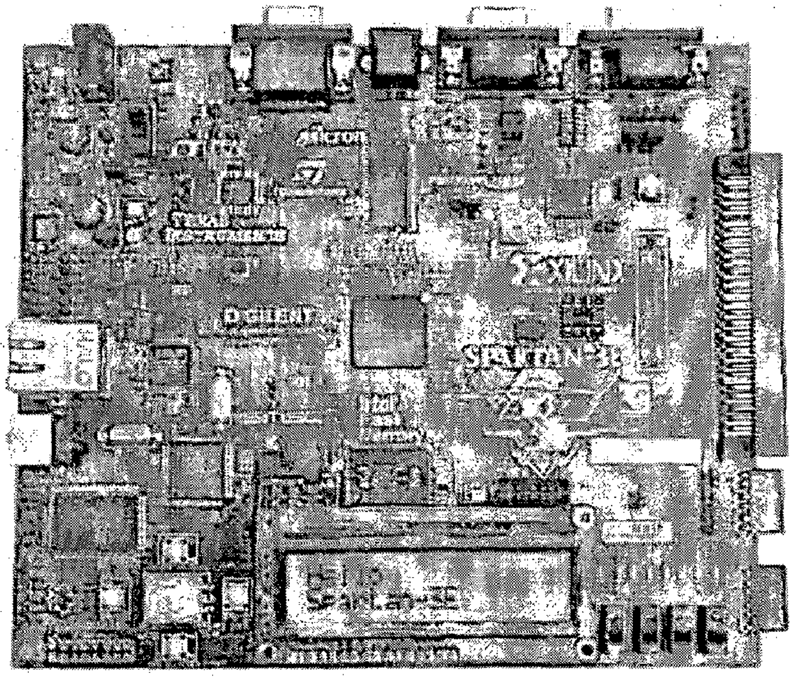


Figure 3.3: XUP Spartan 3E FPGA development kit [30]

5. Connectors and Interfaces: Ethernet 10/100 Phy, JTAG USB download,Two 9-

pin RS-232 Serial Port , PS/2- style mouse/keyboard port , rotary encoder with push button , Four Slide Switches , Eight Individual LED Outputs, Four Momentary{Contact Push Buttons and Three 6-pin expansion connectors.

3.5 Design Process for Reprogrammable Computing

A general overview of the design process for reconfigurable computing is given in Fig3.2. Using MATLAB SIMULINK along with Xilinx System Generator and the Xilinx ISE synthesis and implementation tool, it is possible to implement DSP designs in FPGA. As a plug-in to the MATLAB SIMULINK modeling software, the Xilinx System Generator provides a bit accurate model of FPGA circuits and automatically generates a synthesizable VHDL code including

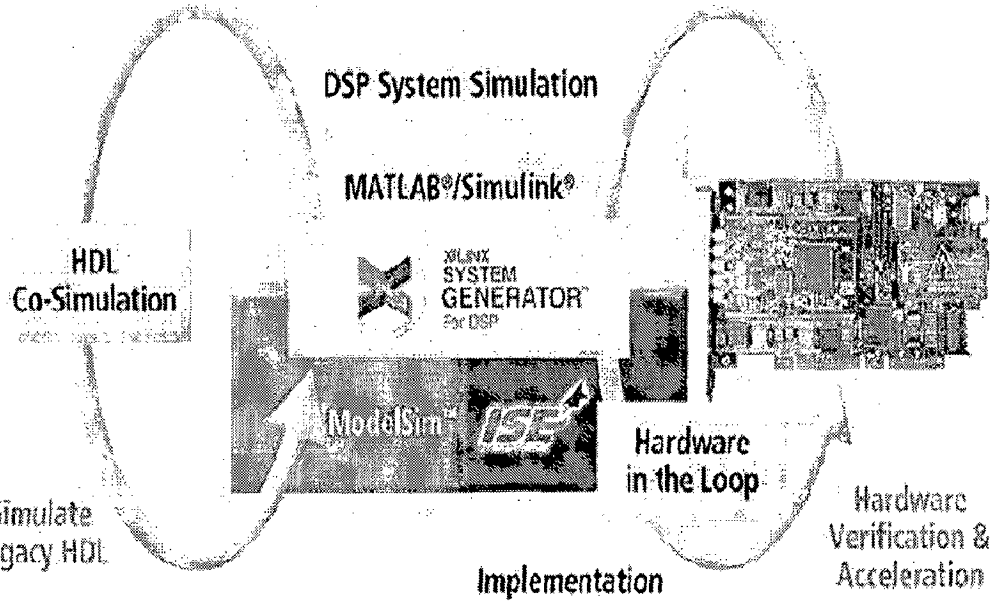


Figure 3.4 Illustration of test-bed implementation process [24].

Testbench. This synthesized VHDL design can be used for implementation in the Xilinx's FPGAs platform. Figure 3.3 illustrates the test-bed implementation of the design process for reconfigurable computing. The design implementation is described below.

3.5.1 Simulation with SIMULINK and System Generator

In this thesis, the algorithm is designed and simulated using Xilinx System Generator system level tool. For an exact representation of the FPGA implementation, the Xilinx blockset in MATLAB SIMULINK is used.

The Xilinx blockset enables bit-true and cycle-true modeling and includes common parameterizable blocks such as finite impulse response (FIR) filter, fast Fourier transform (FFT), direct digital synthesizer (DDS), multipliers, and much more. The following are the key steps in the design simulation process using MATLAB SIMULINK and System Generator [30]:

(1) Start the design by implementing the Xilinx blocks in the MATLAB SIMULINK model design.
(2) Select the Xilinx System Generator block and add it on the top of the design hierarchy.
(3) "Gateway In" and "Gateway Out" blocks are used to define the inputs and outputs to the Xilinx design. Xilinx gateway blocks automatically converts the double precision floating point numbers from the MATLAB SIMULINK environment into the fixed point numbers for the Xilinx environment.

All the system components inside the gateway blocks should be Xilinx blocks only.

However, any other MATLAB SIMULINK blocks such as scope, scatter plots and eye diagrams can be used to interface with Xilinx design and represented in system level. In this thesis, we used all three representations in the simulation. Then the design can be simulated and the outputs can be verified using visual output blocks like scopes or by writing the output to the MATLAB workspace.

3.5.2 HDL Co-Simulation

A system level design can be converted to the gate level representation using System Generator, which will automatically generate the VHDL code for all Xilinx blocks contained in the hierarchy. Additionally, automatic generation of testbench enables design verification upon implementation. The following are the key steps in the HDL co-simulation process [31]:

(1) Double click on the System Generator block and bring up the graphical interface

(2) User interface (GUI). This is illustrated in Figure 3.5, which shows the System Generator dialog box. Set different options, such as targeted FPGA family, testbench generation and IP core generation. Notice that, the compilation type can be selected for HDL simulation or hardware co-simulation. The tool used for synthesis can be chosen from a choice of synthesis tools Xilinx's XST, Synplify Pro and Mentor Graphics. Also, HDL can be chosen either as VHDL or Verilog.

(3) The next step is to run the System Generator by initiating the "Generate" button. This creates a top-level VHDL file, automatically generating IP cores using Xilinx Core Generator System or generates synthesizable VHDL. A VHDL testbench and data vectors are created if selected. These vectors represent the inputs and expected outputs stated in the MATLAB SIMULINK simulation step, and allow additional verification using a behavioral simulator.The design can be synthesized or a VHDL functional simulation can be run.

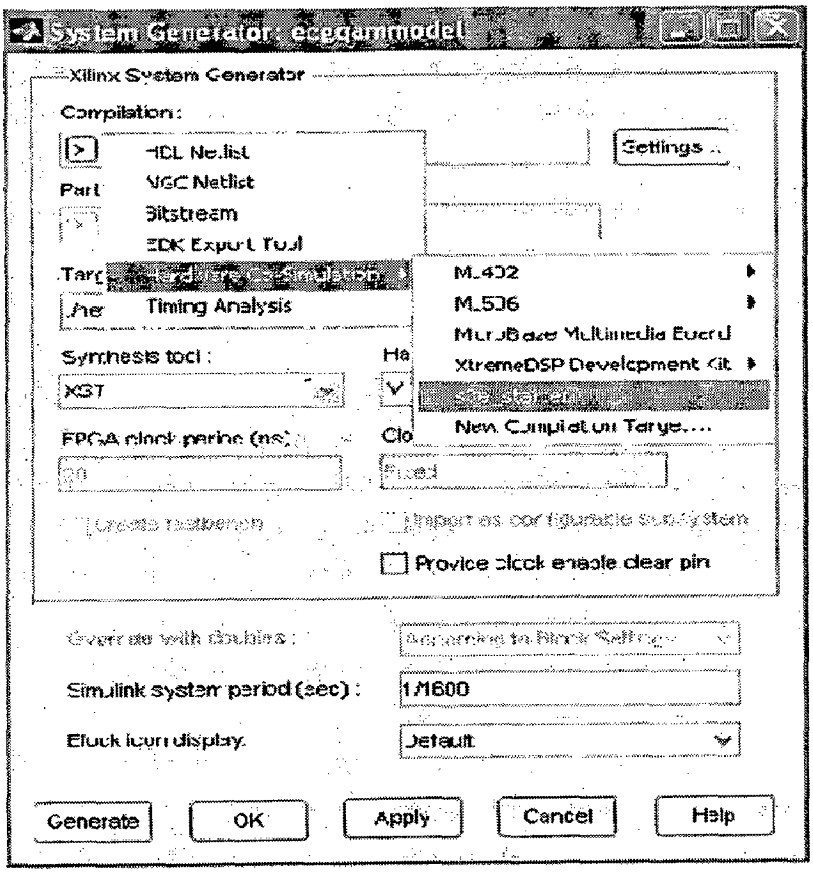
3.5.3 Hardware Co-Simulation

The key steps involved in hardware co-simulation are similar to that in

HDL cosimulation. The System Generator will automatically synthesize, and place and route the design on the target FPGA platform upon selecting the appropriate options, such as compilation type, target FPGA, synthesis tool, and so on. The key steps in the hardware co-simulation process can be summarized as follows [31]:

(1) The hardware co-simulation platform can be chosen from the System Generator dialog box. When the compilation target is selected, the fields on the System Generator dialog box are automatically configured with settings appropriate for the selected compilation target.

(2) After initiating the "Generate" button, the code generator is invoked and produces an FPGA configuration bitstream for the design that is suitable for hardware co-simulation. System Generator not only generates the HDL and netlist files for the



Synthesis tool: XST

FPGA clock period (ns): 20

Fixed

☐ Provide clock enable clear pin

Simulink system period (sec): 1/1600

Block icon display: Default

Figure 3.5 System Generator dialog box.

model during the compilation process, but it also runs the downstream tools necessary to produce an FPGA configuration file

(4)After the FPGA configuration bitstream is created, a new hardware co simulation block is created by the System Generator and stored in the MATLAB SIMULINK library. Hardware co-simulation blocks can be used in the design with other MATLAB SIMULINK blocks. When the hardware co-simulation block is simulated, it interacts with the underlying FPGA platform and facilitates the design implementation and verification of the desired FPGA.

In this thesis, hardware co-simulation and hardware software co-simulation is performed using Spartan 3E FPGA.

3.6 Summary

This chapter is a description of the general design and simulation processes for implementing a wireless communication system on a SDR. In this chapter, the software and hardwares used for SDR implementation was discussed. The design process involved in implementing the system on a reconfigurable platform has been presented. The steps involved in simulating and implementing the system in real time has been discussed. The actual implementation is presented in the next chapter.

Chapter 4
SYSTEM MODEL DESIGN

4.1 Introduction

This chapter describes the experimental setup, model for the baseband section of software defined radio based wireless communication system used in biodata transmission. At transmitter side, the operations such as forward error correction, modulation and up conversion are performed on medical data samples and the upconverted data is passed through channel. At receiver side, the operations such as down conversion, demodulation and decoding are performed on data taken from channel to regain the original signal.The system is modeled and simulated using MATLAB SIMULINK and Xilinx's System Generator. The simulated system is then targeted to the Xilinx's Spartan 3E FPGA for hardware implementation.

4.2 OSI seven layer model

Some years ago, a group got together in an attempt to define a model for the way communications should be structured, which was known as the Open Systems interconnection (OSI) seven-layer model. Transmission Control Protocol/Internet Protocol (TCP/IP) network communication comes the closest to this model. Seven layers are described below.

LAYER 1: PHYSICAL LAYER

The data layer is the lowest layer and defines the physical and electrical characteristics. It is the layer dealing with sending bits over the physical medium. All communications[7] have a physical layer of some sort. In some systems, it may be the only layer. Baseband communications, modulation, demodulation, and transmission through the channels are all topics that loosely belong in this layer.

LAYER 2: DATA LINK LAYER

This layer deals with blocks of data on the physical media. It controls the sharing of the communication path, frames, flow control, and some low-level error checking. This is the multiple access (MAC) layer in network communications. Many strategies exist for sharing access to a transmission channel. Access and error-checking techniques are topics we can cover that belong to this layer.

LAYER 3: NETWORK LAYER

This layer is responsible for routing, making, maintaining, and breaking connections. This is the IP layer in network communications.

LAYER 4: TRANSPORT LAYER

This layer is responsible for the error-free transmission of data from one machine to another. This is the TCP layer in network communications.

LAYER 5: SESSION LAYER

This layer handles the life of the current connection and keeps the data traffic moving.

LAYER 6: PRESENTATION LAYER

This layer handles the data from applications. It performs packing, encryption, decryption, compression, and so on.

LAYER 7: APPLICATION LAYER

This layer is where the application software resides

4.3 System Model

In this thesis, software definable baseband section of the wireless communication system is designed, simulated and implemented. MATLAB SIMULINK and Xilinx's System Generator are used as high level modeling tools in the design process.

Simulation of the system with these tools forms the first step of the design process for reconfigurable computing as discussed in Chapter III. The transmitter and Receiver section of the baseband is implemented in Spartan 3E hardware board using the hardware co-simulation process.

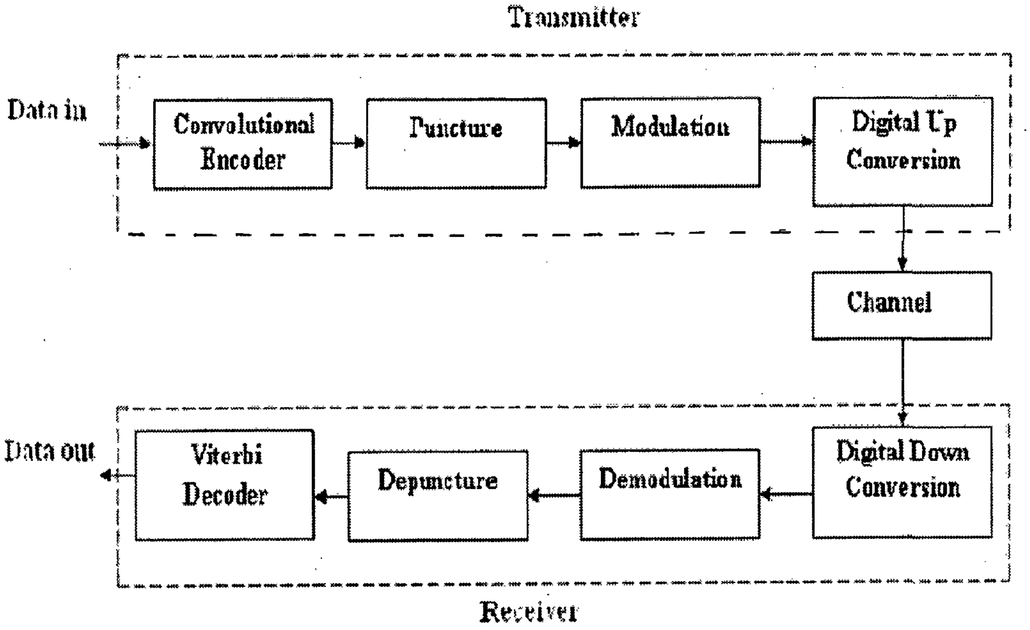


Figure 4.1 Software definable baseband communication system

Figure 4.1 shows the software definable baseband section of the communication system model implemented in this thesis

In the baseband section of the communication system, the transmitter consisting of the convolutional encoding, puncturing, modulation and Digital Up Conversion is simulated. The receiver side consists of Digital down Conversion, demodulation, depuncturing and Viterbi decoding.

4.4 Forward Error Correction

FEC adds redundant data to the data stream to allow the data receiver to detect and correct errors. An important aspect of this concept is that it does not require a return channel for the acknowledgment [32]. If a data receiver detects an error, it simply corrects it and accurately reproduces the original data without notifying the data sender that there was a problem. The FEC coding technique is most effective if errors occur randomly in a data stream.

4.4.1 Convolutional Encoder

Convolutional Codes were first introduced by Elias in 1955, which offer an alternative to block codes for transmission over a noisy channel. Convolutional coding can be applied to a continuous input stream (which cannot be done with block codes), as well as blocks of data. In fact, a convolutional encoder can be viewed as a finite state machine. It generates a coded output data stream from an input data stream. It is usually composed of shift registers and a network of XOR (Exclusive-OR) gates. The stream of information bits flows in to the shift register from one end and is shifted out at the other end. XOR gates are connected to some stages of the shift registers as well as to the current input to generate the output. The location of stages is determined by the generator sequence, which also determines the minimum Hamming distance.

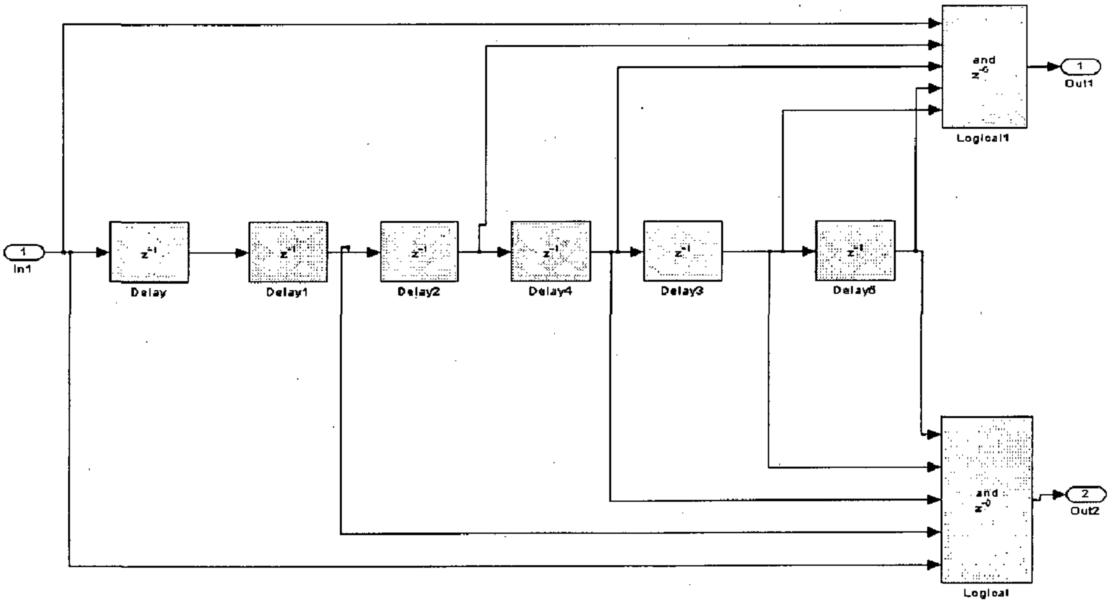


Figure 4.2 Conventional encoder

Minimum Hamming distance determines the maximal number of correctable bits also the decoding performance. Usually with a lower coding rate and a longer constrain length; there must be a larger minimum Hamming distance.

The convolutional encoder, which corresponds to Viterbi decoder considered in this thesis, is shown in the Figure 4.2. This encoder in the system fulfills IEEE the 802.11 and the 802.16 standards. The parameters are shown in the following table.

Definition	**Symbol**	**Value**
Input number	K	1
Output number	N	2
Encoder rate	K/N	1/2
Constrain length	L	7
Generator sequence	G	[1111001] [1011011]

Table 4.1 Parameters of conventional encoder

The constraint length denotes the number of shift registers over which the modulo-2 sum of the input data is performed. The rate 1/2 signifies that for every1 bit input, the encoder will output 2 encoded bits [32].

4.4.2 Puncturing Coding

By using the puncturing coding block, our goal is to achieve a higher rate in the whole system. This block deletes some predetermined convolutional codes bits in the encoded sequence. These discarded bits are neither transmitted nor considered in the decoding procedure. Puncturing is a method of constructing new codes by removing the user-defined bits from the encoded data. The use of puncturing significantly reduces the number of bits to be transmitted over the channel [32]. The puncture codes are a bit pattern that identifies the bits from the encoder to be transmitted and the codes used in this thesis are 10 and 11. Based on the puncture code parameter, the binary vector decides the bits that are to be removed. In a puncturing sequence, 0 and 1 means that the corresponding code symbol is not transmitted or is transmitted, respectively.

Figure 4.3 shows how the rate 1/2 convolutionally encoded output can be punctured with puncture code 10 and 11 to give a rate 2/3 punctured output. In Figure 4.3, the puncture block 0 and 1 uses 10 and 11 as puncture codes respectively. Consider bits A, B, C and D as input to the convolutional encoder. Bit A input to the encoder is encoded as A0 and A1. Similarly bits B, C and D are encoded as B0, B1, C0, C1, D0 and D1.

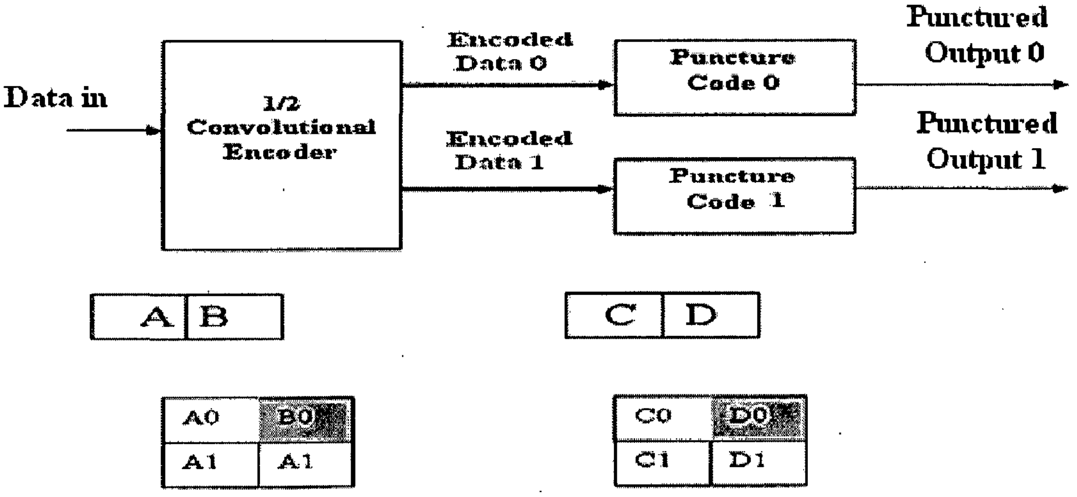


Figure 4.3: Example of puncturing rate 1/2 encoder

The encoded bits are now input to the puncture blocks. Since the puncture codes are 10 and 11, two bits in parallel should be input to the puncture blocks. Thus, when A0 and B0 are input to puncture block 0, bit B0 is not transmitted because the puncture code 10 will delete every second bit input to the puncture block 0. However, all the bits input to the puncture block1 will be transmitted because the puncture code 11 will not delete any bits.

Similarly, bit D0 will be deleted when C0 and D0 are input to the puncture block 0 and bits C1 and D1 will be transmitted from puncture block 1. Thus, the output rate of the forward error correction block is 2/3, i.e., for every 2 input bits, 3 out of 4 encoded bits are transmitted.

4.4.3 Depuncturing coding

Data is depunctured prior to decoding, by inserting the null-symbols are inserted according to the puncture code patterns. Figure 4.4 shows an example of depuncturing rate 2/3 punctured data.

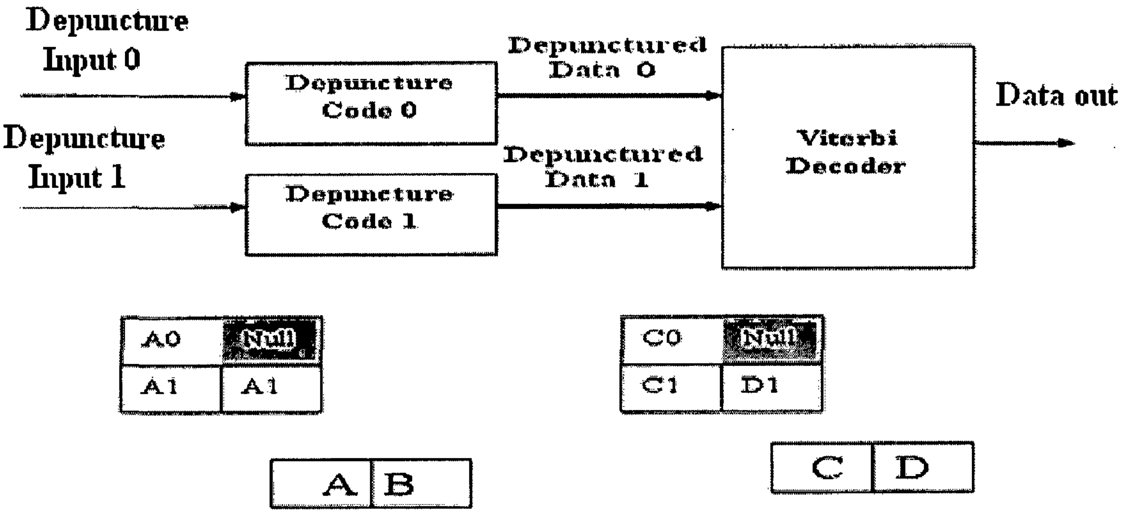


Figure 4.4: Example of depuncturing rate 2/3 encoded data.

The depuncture blocks have the same puncture codes as in the puncture blocks. Hence depuncture block 0 has a code of 10 and block 1 has a code of 11. Therefore, the null symbol is inserted after every other bit coming out of depuncture block 0. No null symbols are inserted for block 1 as no bits were punctured. Since the bit B0 and D0 were punctured before transmission, null symbol is inserted in those locations and input to the Viterbi decoder along with the other bits.

4.4.4 Viterbi decoding

A. J. Viterbi proposed an algorithm as an 'asymptotically optimum' approach to the decoding of convolutional codes in memory-less noise. The Viterbi algorithm (VA) had already been known as a maximum likelihood decoding algorithm for convolutional codes.

As the figure 4.5 shown, the Viterbi decoding block takes 5 steps to finish a package data decoding. The branch metric computes 4 possible branch metrics Euclidean distance, which are 00, 01, 10, and 11.

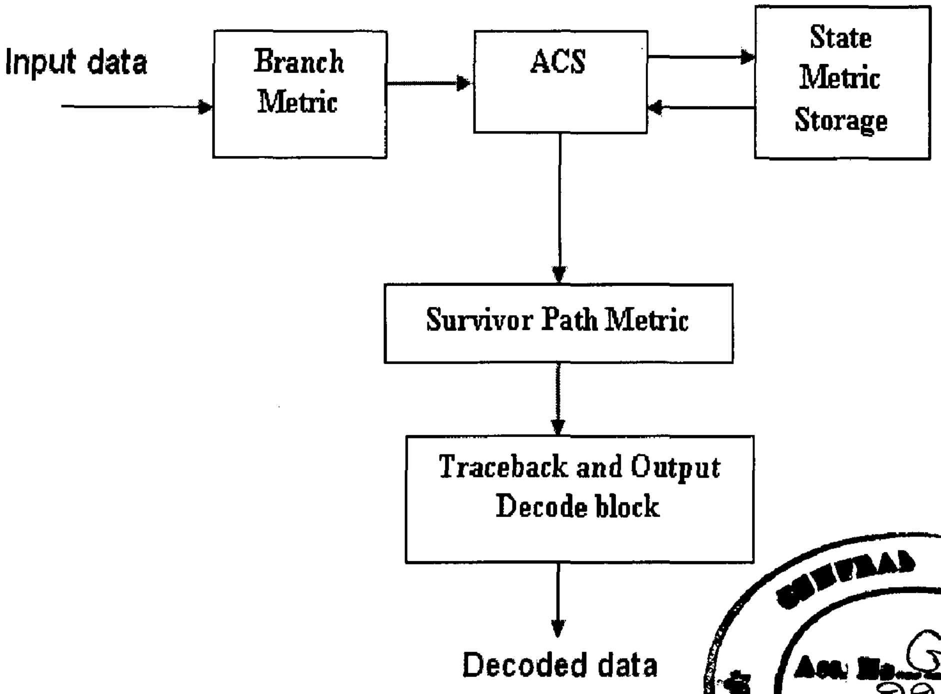


Figure 4.5 Viterbi decoder algorithms [33]

The Add-Compare-Select (ACS) receives the possible branch metrics and the state metrics storage's value. An ACS module adds each incoming branch metric ofthe state to the corresponding state metric and compares the two results to select a smaller one. Then it will update the state metric storage with selected value[33].

State metric storage: this block stores partial path metric's Euclidean distance Survivor path storage: this block records the survivor path of each state selected by the ACS module.

Trace-back and output decode block: this block first begin with trace-back to find the most likelihood path from the last state to the first state in the survivor path metric. Then it generates the decoded output sequence.

The constraint length of 7 and the code array 127 and 117 used for decoding are the same as in convolutional encoder. The traceback length parameter, that is, the number of trellis states processed before the decoder makes a decision on a bit, is set to 96. The decoder outputs the data bits which are later grouped accordingly.

4.5. Modulation/Demodulation

Digital data cannot be sent through a band-limited channel in its original format. Hence the data is encoded into a base-band carrier (usually sinusoidal wave). This process, encoding of digital data onto a sinusoidal carrier is termed as a modulation. The reverse process, decoding of digital data from the carrier is called as demodulation.Modulation can be used to transform digital symbols into waveforms that are compatible with the characteristics of the channel. And minimize the effects of interference. Modulation can also be used to place a signal in a frequency band where design requirements, such as filtering and amplification can be easily met. Size of antenna depends on Wavelength; hence higher Frequencies are used for transmission.

Performance of any radio system is dependent on the efficiency of the modulation scheme used. The two most important factors that decide the overall efficiency of the modulation scheme are power efficiency and bandwidth efficiency. Power efficiency is the ability of a modulation technique to preserve the quality (e.g. BER) of the signal with minimal signal power.

It is defined as the ratio of signal energy per bit to noise spectral density

(*Eb/No*) required to achieve a particular BER. Bandwidth efficiency is the ability of a modulation technique to transfer more data at the given bandwidth, which decides the system capacity. It is defined as the ratio of data rate in bits per second to allocated bandwidth in Hertz (*R/B*). There exists a fundamental tradeoff in any communication system between the power efficiency and bandwidth efficiency as one can be achieved only at the expense of the other. The data can be encoded onto sinusoidal carrier's Amplitude, Frequency or Phase. Encoding data by changing (*modulating*) the amplitude of two carrier waves. These two waves, usually sinusoids, are out of phase with each other by 90° and are thus called quadrature carriers is nothing but Quadrature Amplitude Modulation. phase (digital PSK) can be regarded as a special case of QAM, where the magnitude of the modulating signal is constant, with only the phase.varying[31].In this thesis QPSK and 16-QAM modulation techniques are used for modulation the encoded data.

4.5.1 QPSK modulation

QPSK (4-ary PSK) involves changing the phase of the transmitted waveform. Each finite phase change represents unique digital data. A phase-modulated waveform can be generated by using the digital data to change the phase of a signal while its frequency and amplitude stay constant. A QPSK modulated carrier undergoes four distinct changes in phase that are represented as symbols and can take on the values of $\pi/4$, $3\pi/4$, $5\pi/4$, and $7\pi/4$. Each symbol represents two binary bits of data.

Figure 4.6 represents the process of a QPSK modulator There are a baseband modulation signal processing block (Symbol mapper) and a carrier generation block (DDS synthesizer) with multipliers.

First, the input binary bit stream is split into two bit streams which are the even and odd bit streams (in-phase and quadrature streams) by the serial to parallel converter Then, send alternating bits to I, Q even bits to Q channel, odd bits to I channel

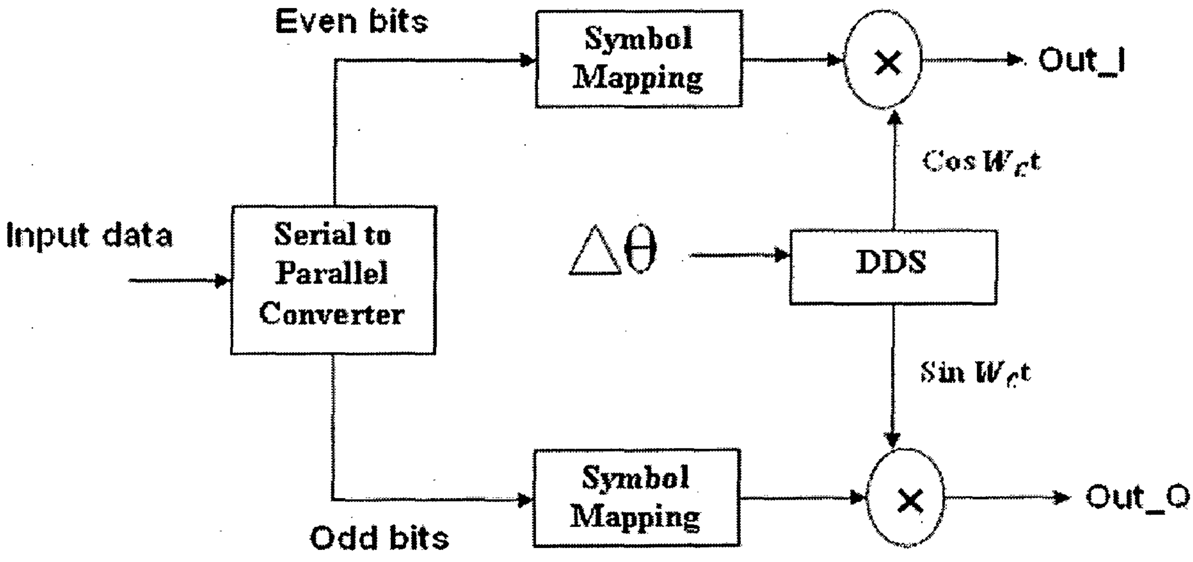


Figure 4.6 QPSK modulator schematic diagram [35]

Second the even and odd bits are converted to symbols in symbol mapping. Here Grey encoder is used to map the data in such a way as to help reduce bir errors. Bit pairs that are used to generate the symbols are only one bit different from each adjacent symbol.QPSK constellation diagram shows symbols, each represented by two data bits that were first grey encoded.

Digital sinusoidal signal sources of programmable frequency and phase commonly referred to as a digital signal synthesizer (DSS) or numerically controlled

Symbol	Bits	Phase	I	Q
S1	00	п/4	2^(-1/2)	2^(-1/2)
S2	10	3п/4	-2^(-1/2)	2^(-1/2)
S3	11	5п/4	-2^(-1/2)	-2^(-1/2)
S4	01	7п/4	2^(-1/2)	-2^(-1/2)

Table 4.2 Four symbols mapping definitions for QPSK

oscillators (NCOs) are commonly used to create the quadrature signals. The resulting output signal is a spectrally translated complex signal that is separated into in-phase and quadrature-phase components. NCOs are typically implemented using programmable phase accumulators followed by sine wave and cosine wave look-up tables, as described in figure 4.7. The carrier frequency of each sinusoid can be set to any precision by defining the phase increment input to the NCO.

This is the LUT-based method in which, the carrier is formed by addressing the memory with the phase value and mapping the current phase to the value of the sinusoidal signal. Then the generated carrier is multiplied with the modulating data by using digital multipliers [42]. Advantages of this method are the good frequency resolution and relative simplicity. Drawbacks are that the multipliers and memories may require a large area, especially with high resolutions, and memories may become the speed bottleneck at high sampling rates. With a look-up table method, it is possible to realize every function $y = f(x)$ of variable x by mapping the value x to the output y with some mapping element, usually a memory block. The accuracy of the mapping depends on the resolution of the input value x and the area (accuracy) of the mapping element. When this is applied to the frequency synthesis, the variable to be mapped is usually the phase of the sinusoid and is mapped to the corresponding amplitude value [42]. In Fig. 4.7.$\Delta\theta$ is the phase

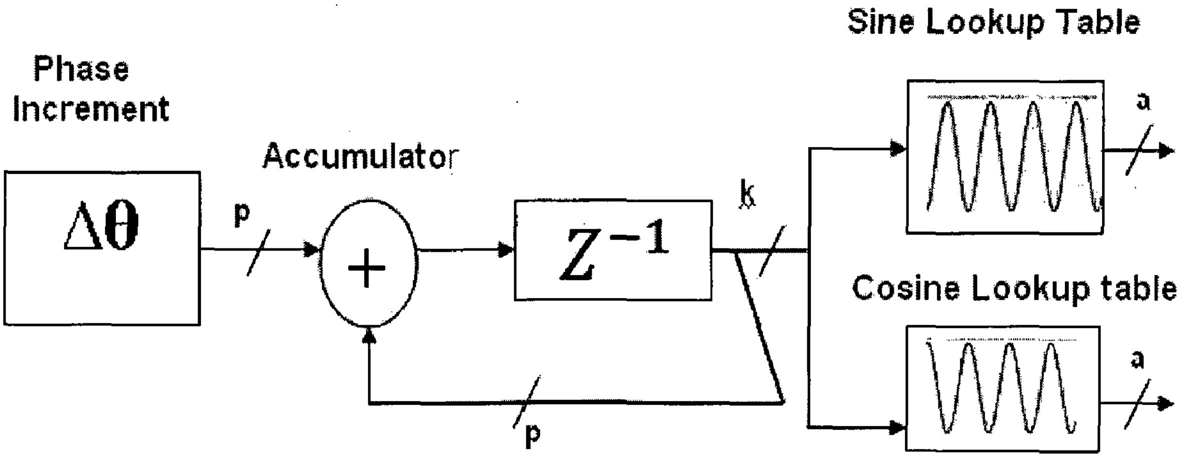


Figure 4.7 Direct Digital Synthesizer [42]

increment, which is integrated over time with a digital integrator also called a phase accumulator. The phase value obtained by integration is then used as the address to the ROM memory, which maps the phase value to the desired amplitude value. The accuracy of the synthesis is controlled by the values of *p*, *k* and *a* (Fig.4.7) [39], [40].

The enhancement of the performance of the synthesizer is usually achieved by increasing the values of *k* and *a*. The value *p* affects mostly the frequency resolution of the synthesizer that is not the speed bottleneck. The phase accumulator can be made almost arbitrarily fast with pipelining techniques, for example. The increase of the values *a* and *k* results the increase of the area of the ROM memory. The increase of this area may slow down the operation of the memory block and limit the achievable frequency band.

Next, multiply Q channel with a sine wave of frequency fc and multiply I channel with by cosine of frequency fc.

Modulating signal

$$A = A_I(t) + jA_Q(t) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Carrier Signal

$$B = \cos(2\pi f_c t + \theta) + j \sin(2\pi f_c t + \theta) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Modulated signal

$$\text{Re}\,[A\times B] = A_I(t)\ \cos(2\pi f_c t + \theta) - A_Q(t) \sin(2\pi f_c t + \theta)\ldots.(3)$$

4.5.2 16-QAM modulation

Quadrature amplitude modulation (QAM) is a modulation scheme in which two sinusoidal carriers, one exactly 90 degrees out of phase with respect to the other, are used to transmit data over a given physical channel. Because the orthogonal carriers occupy the same frequency band and differ by a 90 degree phase shift, each can be modulated independently, transmitted over the same frequency band, and separated by demodulation at the receiver. For a given available bandwidth, QAM enables data transmission at twice the rate of standard pulse amplitude modulation (PAM) without any degradation in the bit error rate (BER). QAM and its derivatives are used in both mobile radio and satellite communication systems [39].

Here 16-QAM modulation is used. 16-QAM is achieved by modulating two 4-level PAM signals onto orthogonal carriers, 4-bits are mapped into a symbol .same operations are applied the input signal as that of QPSK modulation.

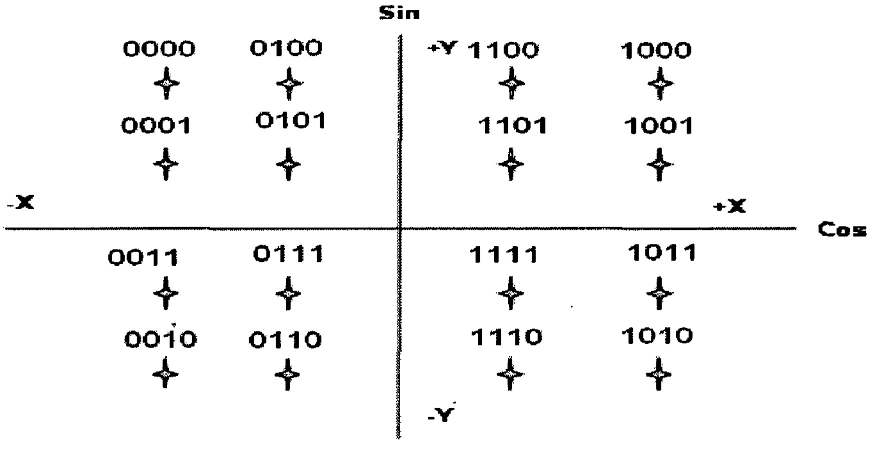
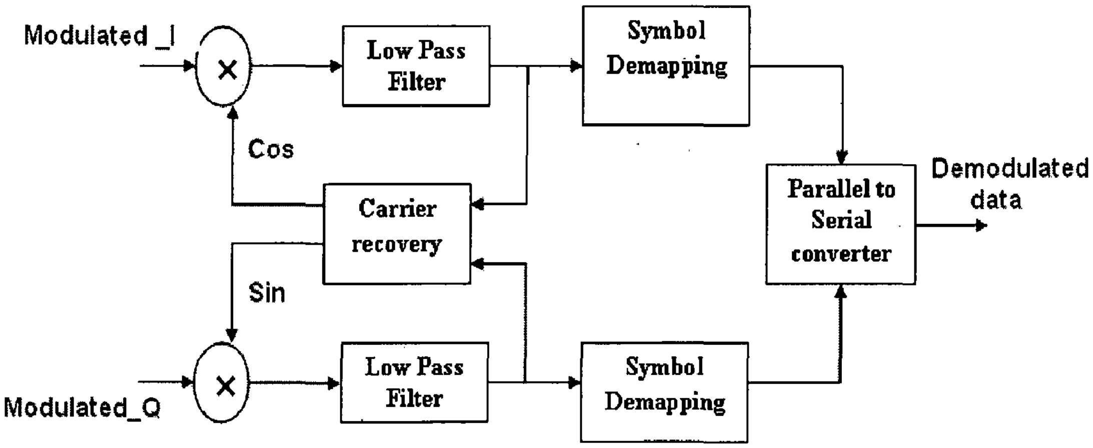


Figure 4.8 Symbol mapping for QAM modulator

4.5.3 Demodulation

The modulated carrier is synchronously demodulated in order to recover the message signal, as shown in Figure 4.9. The incoming signal is multiplied by two locally generated sinusoidal signals cos ($2\pi f_c t$) and sin ($2\pi f_c t$) *and* low-pass filtered. This results in the recovering of the original $A_I(t)$ and $A_Q(t)$ signals, fading can occur if the local oscillator phase is different from the received carrier phase. The signal is attenuated by a term proportional to the cosine of the phase difference. As in the synchronous AM demodulation, a frequency drift of the local oscillator can shift the demodulated signal frequency and disturb the reception.

First, since the incoming waveform is suppressed carrier in nature, coherent detection is required. The methods by which a phase-coherent carrier is derived from the incoming signal are termed, carrier recovery, and will be covered first. Next, the raw data are obtained by coherent multiplication, and used to derive clock-synchronization information [39]. The raw data are then passed through the low pass filter, to remove higher frequencies. This filtered data is then routed, along with the derived clock, to the symbol demapper which outputs the demodulated data. The Costas loop performs both phase-coherent suppressed carrier reconstruction and synchronous data detection within the loop.



4.9 Demodulator schematic diagram

Modulated signal

$$MI_c(t) = A_I(t)\cos(2\pi f_c t) - A_Q(t)\sin(2\pi f_c t)\ldots\ldots.(4)$$

To suppress carrier signal, modulated signal is multiplied by exponentional term

$$E(t) = \cos(2\pi f_c t) + j\sin(2\pi f_c t)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(5)$$

$$M(t) = MI_c(t) \times E(t)$$

$$M(t) = MI_c(t) \times (\cos(2\pi f_c t) + j\sin(2\pi f_c t))\ \ldots\ldots.(6)$$

Real part of M(t)

$$Re[M(t)] = A_I(t)\cos(2\pi f_c t)\cos(2\pi f_c t) - A_Q(t)\sin(2\pi f_c t)\cos(2\pi f_c t)$$

$$= A_I(t)\,cos^2(2\pi f_c t) - A_Q(t)\sin(2\pi f_c t)\cos(2\pi f_c t)$$

$$\text{Re}\,[M\,(t)] = A_I(t) + A_I(t)\cos(4\pi f_c t) - A_Q(t)\sin(2\pi f_c t) \ldots\ldots(7)$$

$$\text{Im}\,[M\,(t)] = A_I(t)\cos(2\pi f_c t)\sin(2\pi f_c t) - A_Q(t)\sin(2\pi f_c t)\sin(2\pi f_c t)$$

$$= A_I(t)\cos(2\pi f_c t)\sin(2\pi f_c t) - A_Q(t)\,sin^2(2\pi f_c t)$$

$$\text{Im}\,[M\,(t)] = A_Q(t) - A_Q(t)\cos(4\pi f_c t) + A_I(t)\sin(2\pi f_c t) \ldots.(8)$$

$$\begin{array}{l} \text{Re}\,[M\,(t)] = \\ \text{Im}\,[M\,(t)] = \end{array} \underbrace{\boxed{\begin{array}{c} A_I(t) \\ A_Q(t) \end{array}}}_{\textbf{Terms want}} \begin{array}{c} + \\ - \end{array} \underbrace{\boxed{\begin{array}{c} A_I(t)\cos(4\pi f_c t) - A_Q(t)\,\text{Sin}\,(2\pi f_c t) \\ A_Q(t)\cos(4\pi f_c t) + A_I(t)\,\text{Sin}\,(2\pi f_c t) \end{array}}}_{\textbf{Terms don't want}}$$

......(9)

From Eqn 9 , it shown that data terms are reserved and extra terms are modulated into frequency of 2*fc . So Low pass filter is required to sarate them.

4.5.3.1 Costas loop

Costas loop is a phase-locked loop used for carrier phase recovery from suppressed-carrier modulation signals

The Costas loop consists of three basic functional blocks:

1 A voltage-controlled oscillator (VCO)

2 A phase detector (PD)

1 A Loop filter (LF)

There are few complications while demodulating the signal, like carrier frequency mismatch between Tx and Rx due to delay and parameters changes due to time and temperature. So costas loop is required.

In costas loop DDS provides quadrature outputs to complex multiplier The same phase input signals is also applied to complex multiplier .outputs from multiplier is passed to phase detector, which detects phase error [37]. Phase error is passed through loop filter. The loop filter is a second-order IIR filter which determines how to rotate the signal around the constellation.output of Loop filter used to control the DDS.

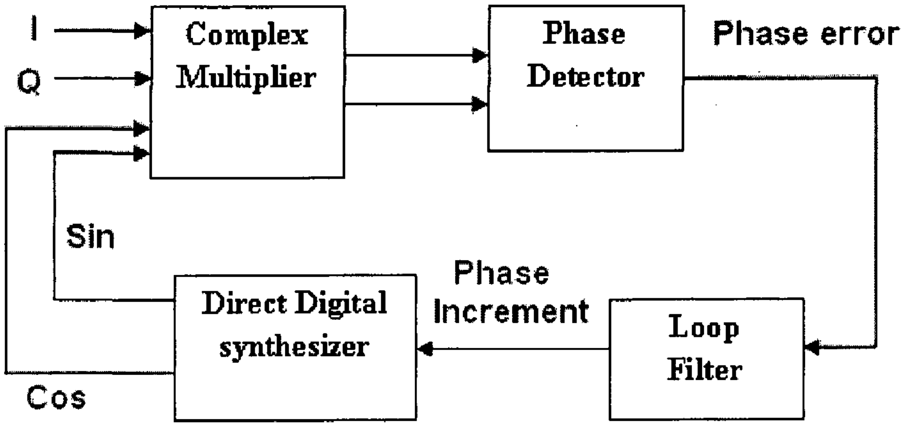


Figure 4.10 Carrier recovery schematic diagram [41].

4.6 Up/Down Conversion

After up-converting the baseband signal, the signal spectrum is shifted from centered at 0 Hz to an intermediate frequency in the range of [-Fs/2, Fs/2], where Fs are sampling frequency. This process requires a Direct Digital Synthesizer (DDS) and a mixer. The mixer is basically a complex multiplier which multiplies the up-sampled signals (interpolation filter output) with the complex exponential generated from the DDS [36].

4.6.1 Frequency Converter Fundamentals

The 2G and 3G multichannel digital frequency upconversion has the following major requirements [37]:

- Translate one or more narrowband signal sources (usually modulated carriers) up in frequency, usually from baseband to an IF.
- Combine the baseband sources to create one wideband result

- Increase the data rate to a digital intermediate frequency rate.

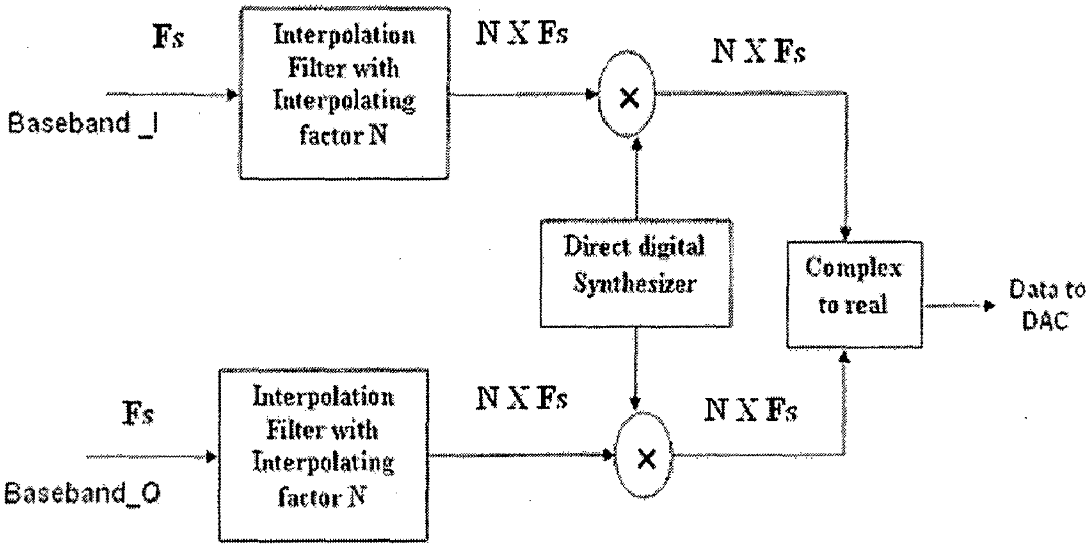

Fs
Baseband _I
Interpolation Filter with Interpolating factor N
N X Fs
N X Fs
Direct digital Synthesizer
Complex to real
Data to DAC
Fs
Baseband_Q
Interpolation Filter with Interpolating factor N
N X Fs
N X Fs

Figure 4.11 Digital up conversion [37]

The 2G and 3G multichannel digital frequency downconversion has the following major requirements

- . Filter or isolate a narrow band of frequencies (usually a modulated carrier) from the wideband source and reject the remainder of the band.
- Translate that isolated carrier down in frequency, usually from an IF to baseband.
- Reduce the data rate to some integer multiple of the information rate.

Subfunctions needed to achieve these requirements are explained in below sections.

As the incoming (downconvert) or outgoing (upconvert) information signal is multiplied by the NCO-generated waveform, the NCO performance must be better than the specification for the multiplied result. Spurious responses in the NCO will mix with unwanted out-of-band signals; some of the resultant components will fall back in band and corrupt the required information signal. The SFDR of the NCO must be considerably better than the SFDR of the band-shifted and filtered output of the up- or downconverter.

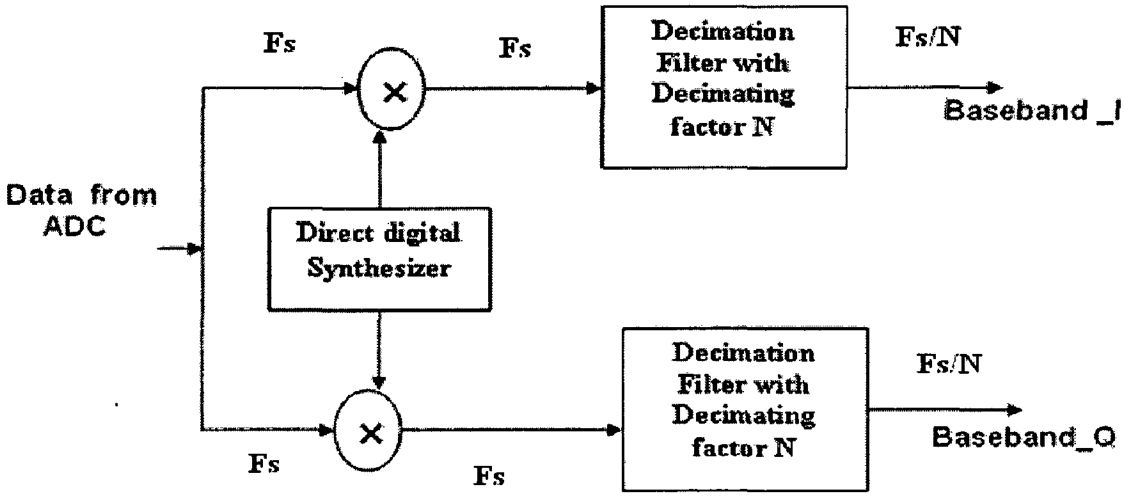

Data from ADC
Fs
Fs
Decimation Filter with Decimating factor N
Fs/N
Baseband _I
Direct digital Synthesizer
Decimation Filter with Decimating factor N
Fs/N
Baseband_Q
Fs
Fs

Figure 4.12 Digital down conversion [38]

Digital mixers perform the same function as their analog counterparts. Sum and difference frequencies are generated by multiplying the incoming digitized source signal with a digital NCO [38]. In most cases, for an upconverter, the input to the mixer will be quadrature (as provided by the baseband processing), and the output will also be quadrature, since the mixer is followed by several stages of quadrature filtering. For a downconverter the input will not be quadrature, since it is usually fed by a nonquadrature ADC; however, the output will be quadrature, since the mixer is again followed by several stages of quadrature filtering. The required output signal will either be the difference signal for downconversion or the sum signal for upconversion; the other component is not needed and will be removed by the following stages of digital filtering.

4.6.2 Sample rate conversion filters

The requirement to process data at more than one sample rate in a system has long been recognized and is now referred to as multirate processing [7]. The two key multirate DSP operations of decimation and interpolation.

When sampling rate conversion takes place, digital filters are needed. Filters used in sampling rate conversion must fulfill two criteria [36]. The first and most important is that they have to filter out the image band in the case of interpolation, or the aliasing band in the case of decimation. The second is that they should be as simple as possible. When we have to interpolate or decimate signals with a reasonably wide bandwidth compared to sampling frequency.multi rate filters are used for sample rate conversion.

Interpolation is achieved in two steps: zero stuffing and lowpass filtering.The zero stuffing function inserts N – 1 zero-valued sample between each input sample; this increases the data rate but produces aliased images of input signal. The images are removed by lowpass filtering [36]. To decrease the sampling rate by a factor N, a decimation operation is required and is implemented by keeping every Nth sample of the input sequence and discarding the other N-1 in-between samples. In the frequency domain, this leads to aliasing if there are frequency components in the input sequence that are greater than half of the target sampling frequency. A low pass filter is therefore required before the decimation operation to ensure that out-of-band

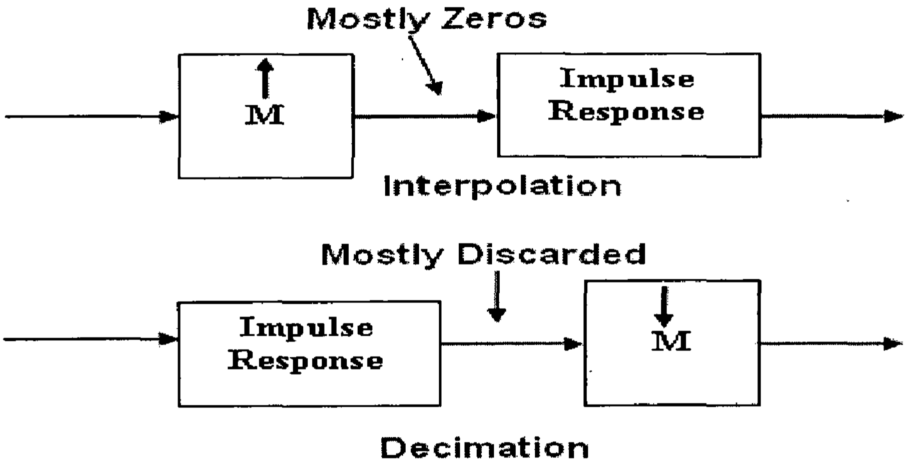


Figure 4.13 sample rate conversion filter structure [36]

frequencies are attenuated. This low pass filter has the same specifications as the filter required for a interpolate by *N*.

4.6.3 Polyphase filter

Basic multirate filters are very inefficient due to

- Inputs are mostly zeros after upsampling.
- Outputs are mostly discarded after downsampling.

By using polyphase filters

Break impulse response into subfilters

h [0, .1......,N-1] is broken down as

h0=h [0, M, 2M, 3M,........]

hl==h [1, M+1, 2M+1, 3M+1,........]

h2=h [2, M+2, 2M+2, 3M+2,....,]

and so on.

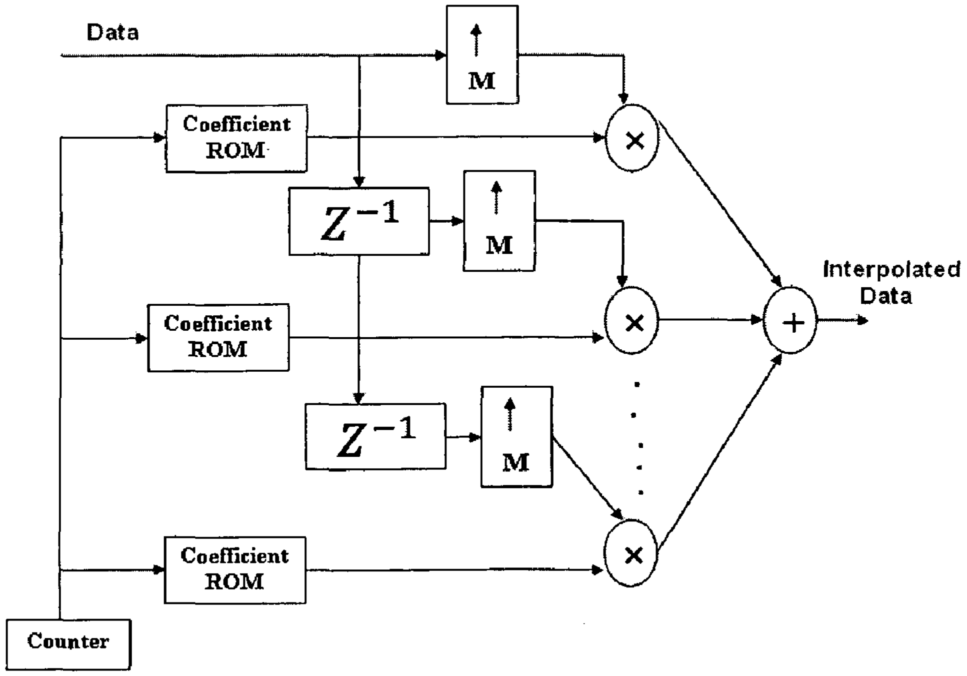


Figure 4.14 Polyphase interpolation filter [25]

The advantage of the polyphase decompositions is that the computation can always be performed with the lower clock frequency, resulting in either the possibility of reducing supply voltage in order to minimize power dissipation or the use of the pipelining/interleaving (*P/I*) technique in order to minimize the area [25]. In the case of FIR filters with symmetrical or anti symmetrical coefficients (linear phase FIR filters), the symmetry of the coefficients can be exploited in order to further reduce the area.

4.7 Summary

In this Chapter, the model for the baseband section of a wireless communication system is presented. The functions of different block like forward error correction, modulation and sample rate conversion are discussed. The parameters set to carry out the simulation were also presented.

Chapter 5
SIMULATION AND IMPLEMENTATION RESULTS

5.1 Introduction

This chapter describes the experimental setup, simulation and results for baseband processing the software defined radio based wireless communication system. The system is modeled and simulated using MATLAB SIMULINK and Xilinx's Sytem Generator. The simulated system is then targeted to the Xilinx's Spartan 3E FPGA for hardware implementation. The designed system is used for transmission of ECG and EMG signals.

Here system is tested by passing medical signals. As explained in chapter 1, SDR provides different communication links and multi point communication. SDR can be reconfigured to provide required communication parameters. Upgrading of standard is done by developing corresponding software and downloading its configured data to hardware platform. Therefore Software defined radio is used for transmission of medical data. ECG signal samples with sampling frequency of 750Hz are used for QAM and EMG signal with sampling frequency of 24 KHz for QPSK system [43].

5.2 System Model for transmission of medical signals

System model as described in chapter IV is consists of conventional encoding, modulation and digitalupconversion on transmitter side and on receiver side digitaldowconversion, demodulation and viterbi decoder are simulated and implemented on FPGA Here two types of modulation techniques QPSK and 16-QAM are considered. System for each modulation is described in below sections.

5.3 16-QAM Transmitter Model

At the transmitter, the ECG samples with sampling frequency 750Hz are taken as input from the ECG measuring instrument after analog to digital conversion.

Initially, The ECG sample is input to the"Gateway In" block from matlab workspace. This block converts the data in double precision to the Xilinx fixed point representation. Then three bytes are added to each input sample for synchronization in Attached Synchronization marker (ASM) for framing. the data is parallel-to-serial converted and given to the data input port of the forward error correction block, which comprises of the convolutional encoder.

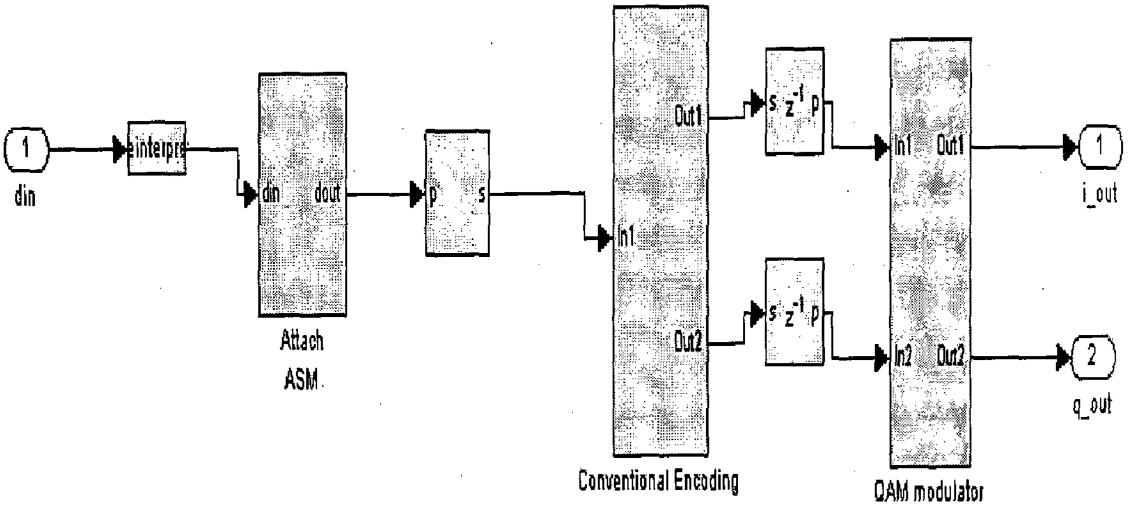


Figure 5.1 Simulation model of QAM transmitter

Here, a rate 1/2 convolutional encoder with constraint length 7 and code array 127 and 117 is used. A constant value of 1 is used as input to the input port (vin) to specify to the encoder that the data on its input port is valid and is ready to be encoded The outputs from the data output ports 1 and 2 of the encoder are modulated using 16-

QAM modulation. In modulator each four bits are mapped into symbol according to gray mapping method and interpolated with interpolation factor N=4. The data from the modulator passes through a"Gateway out" block so that the Xilinx's fixed point representation is converted to SIMULINK double precision. complex to real operation is performed on in- and quadrature-phase signals to get real signal .The output from transmitter is passed through AWGN channel, where noise is added to signal by varying signal to noise ratio(SNR).

5.4 QAM Receiver Model

At the receiver, the signal is first decimated and demodulated. Then error correction is applied to the demodulated data.

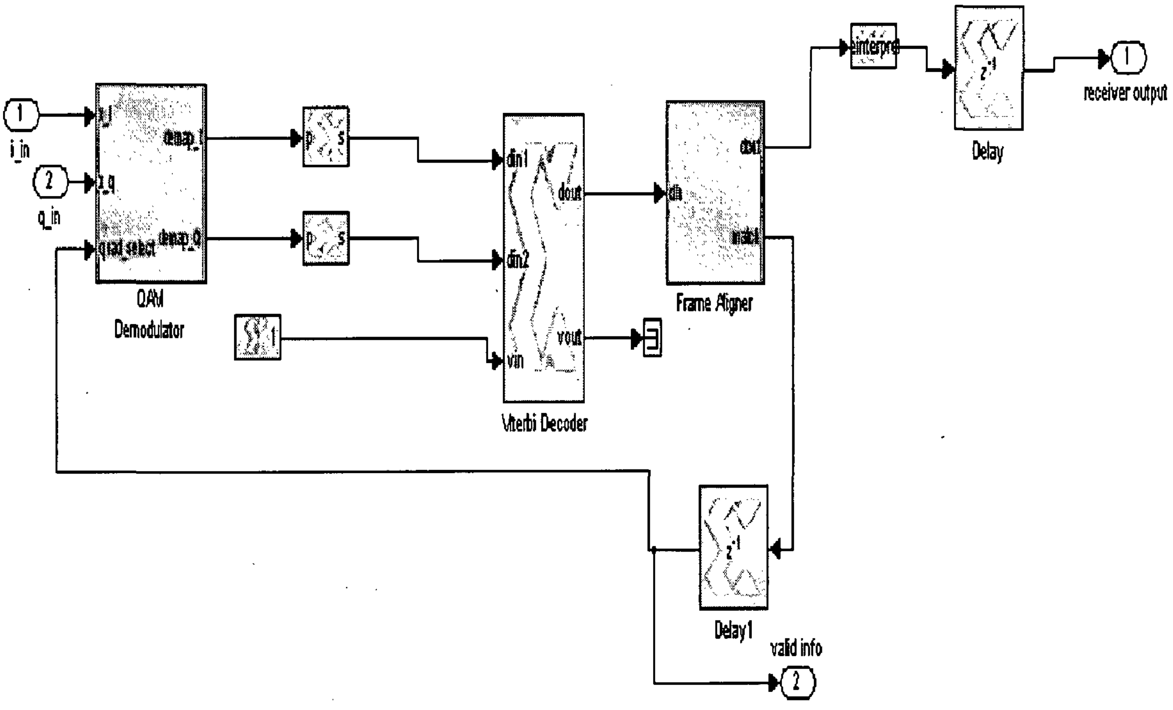


Figure 5.2 QAM receiver system

Real signal for channel is converted into Rectangular complex signal by using real-complex simulink block. Complex signal is first passed through the "Gateway In" blocks converting it to Xilinx's fixed point representation. The receiver has a 16-QAM demodulator that performs adaptive channel equalization and carrier recovery on a receiver input data. Demodulated data is parallel-to-serial converted and is given as input to the Viterbi decoder for forward error correction. The constraint length of 7 and the code array 127 and 117 used for decoding are the same as in convolutional encoder. The traceback length parameter, that is, the number of trellis states processed before the decoder makes a decision on a bit, is set to 96.Decoded data passed trougth frame aligner. The 'Frame Aligner' subsystem converts the serial Viterbi decoder output into bytes. The serial data is aligned so that the first bit of the received ASM becomes the MSB of the corresponding byte output.Aligning the bytes this way ensures the bit ordering of the original 16-bit transmitter input source is preserved when the data is recovered in the receiver.output from frame aligner is passes through a"Gateway out" block to converted to SIMULINK double precision. Output from receiver is compared with input to transmitter.

5.5 QPSK transmitter

In transmitter model, The EMG samples are given to the "Gateway In" block from workspace to convert the data in double precision to the Xilinx 16-bit fixed point representation.

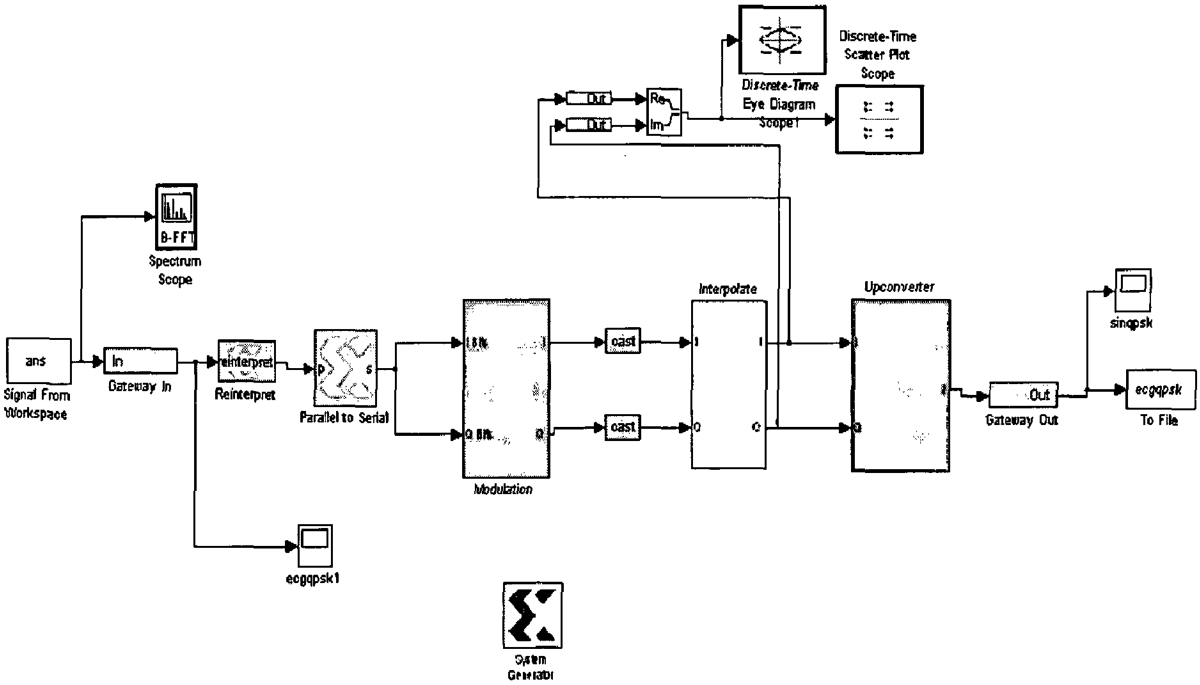


Figure 5.3 QPSK transmitter model

Then data is send to reinterpret block to convert signed to unsigned data. Data parallel to serial converted and returns two bits i.e. 1-even and 1-odd bits at a time. These even and odd bits are modulated using QPSK modulator. Two bits are mapped into symbols by using grey code mapping. Modulated data is interpolated with interpolation factor 8 with help of polyphase interpolation filter. Here sampling rate is converted from 24 Ksamples/sec to 192 Ksamples/sec.

Output data from interpolation filter is given to upconverter for frequency conversion.signal is shifted from DC to upper frequency by using direct digital synthesizer and complex multiplier.Upconverted data is passes through a" Gateway out" block so that the Xilinx's fixed point representation is converted to SIMULINK double precision. The output data from transmitter is passed through channel and output data is stored in a file.

5.6 QPSK receiver

Input data is taken from file and is applied to the "Gateway In" block to convert the data in double precision to the Xilinx 16-bit fixed point representation. Data is passed to downconverter to convert bandpass to baseband i.e. higher frequency 30MHz signal down to DC.Downcoverter consists of direct digital synthesizer and complex multiplier and it gives in and quadrature phase signals as output.Downconverted data is given as input to decimation filter with decimation factor 8 for sample rate conversion.

Carrier recovery is used for recovering data from carrier signal. Then data is passed through demodulation and symbol demapping to demodulate the signal. Symbols are demapped into bits with the help of grey coding. Output from symbol demapping block is serial to parallel converted with 16-bits.16-bit floating point data is is passes through a" Gateway out" block so that the Xilinx's fixed point representation is converted to SIMULINK double precision. Output from receiver is compared with input to transmitter.

5.7 Simulation Results

The results obtained from the simulation is presented and discussed in this section. Parameters are used are shown in table 5.1. The constellation diagram and eye diagram of the modulated signal from which conclusions about the modulated signal can be drawn is observed at the output of the channel.

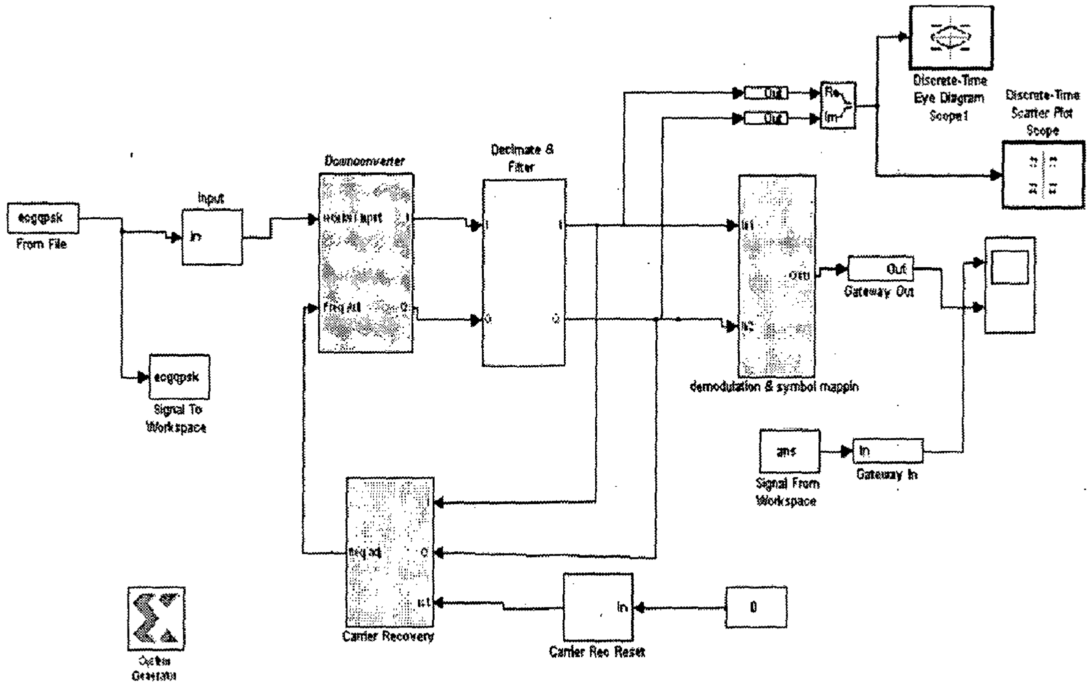


Figure 5.4 QPSK receiver system

Parameter	Value
Data rate	384 Kbps for QPSK, 12 Kbps for QAM.
Modulation	QPSK, 16-QAM
Error correction algorithm	Conventional encoding with Viterbi -decoding
Interpolation factor	64
Channel	Rayleigh fading with AWGN white noise

Table 5.1 Parameters used for implementation

These diagrams reveal the modulation characteristics of the signal and help to depict the impact of impairments, such as pulse shaping or channel distortions. They are commonly used to evaluate the overall performance of the digital communication systems. Since the channel used in this thesis is AWGN, the extent to which the noise has affected the modulated signal can be seen from constellation and eye diagrams.

5.7.1 Constellation Diagrams

Figure 5.5 shows the constellation diagram of the QPSK modulated signal with signal-to -noise ratio (SNR) of 15 dB. The figure shows that each constellation point becomes a cloud around the central point. When the noise is more in the channel, the constellation points spreads around the central point. While demodulating in the receiver, the chances of misinterpretation of one point as other is more and this leads to incorrect demodulation and error.

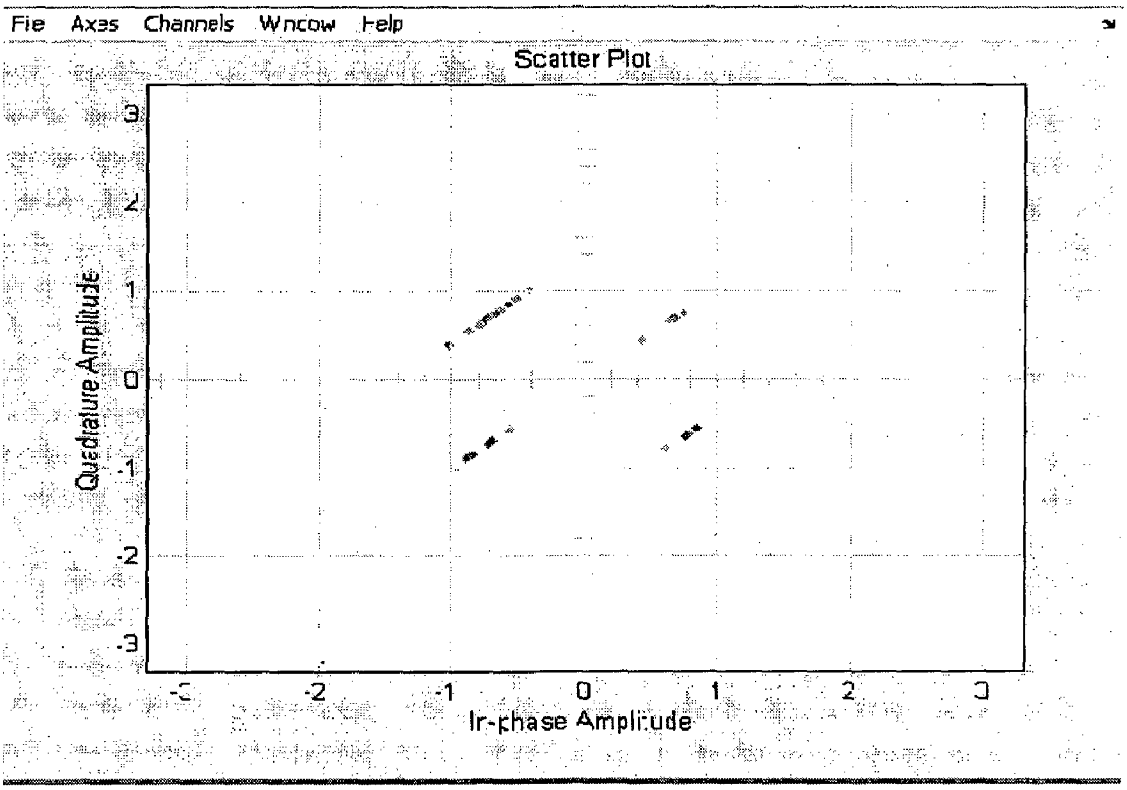


Figure 5.5: Constellation diagram for QPSK modulated signal with SNR = 15 dB.

Increasing the SNR of the AWGN channel will increase the performance of the system. The constellation diagram of modulated signal with SNR of 25 dB is shown in Figure 5.6. Since the SNR is high, the constellation points form a more dense cloud around the central point thus reducing the transmission error.

Comparing Figures 5.5 and 5.6, it can be seen that the constellation points of Figure 5.6 are denser than in Figure 5.5. If the noise in the channel is smaller, then the constellation points will be denser, and thus the transmission error is less and the receiver output is more accurate

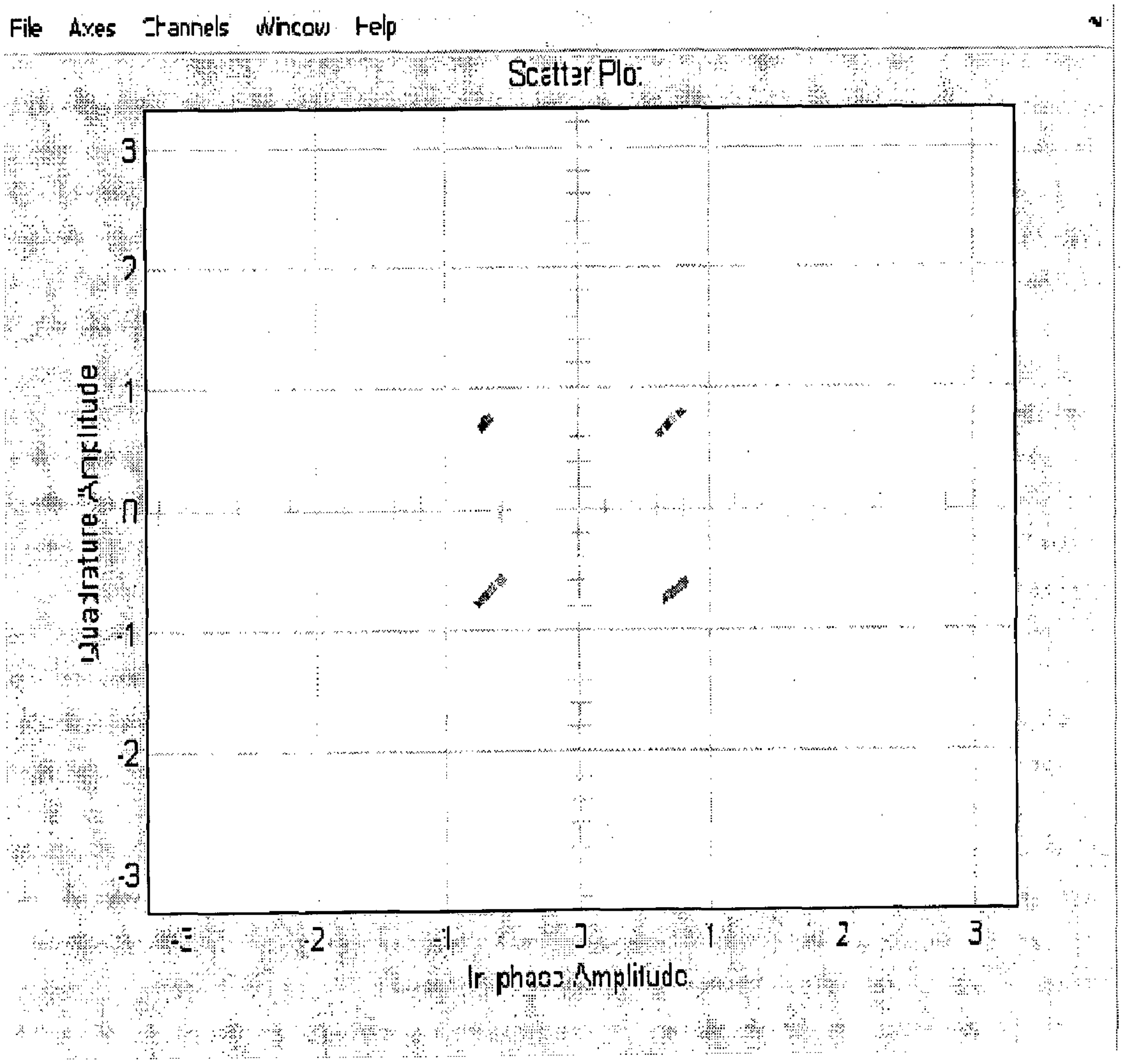


Figure 5.6: Constellation diagram for QPSK modulated signal with SNR = 25 dB.

5.7.2 Eye Diagrams

The measure of distortion, timing jitters and noise margin can be found from the eye diagrams. Figure 5.7 shows the eye diagram of the QPSK modulated signal with SNR of 15 dB. From the figure, A shows the distortion which is equal to 0.4; B and C show the timing jitter and the noise margin which are equal to 0.02 and 0.5. Due to the noise in the channel, when the noise margin in the eye diagram decreases, the eye starts to close in, thus making the errors to increase. Since the SNR is less, it can be seen from the figure that the eye has more distortions and is not properly open due to the presence of noise in the channel.

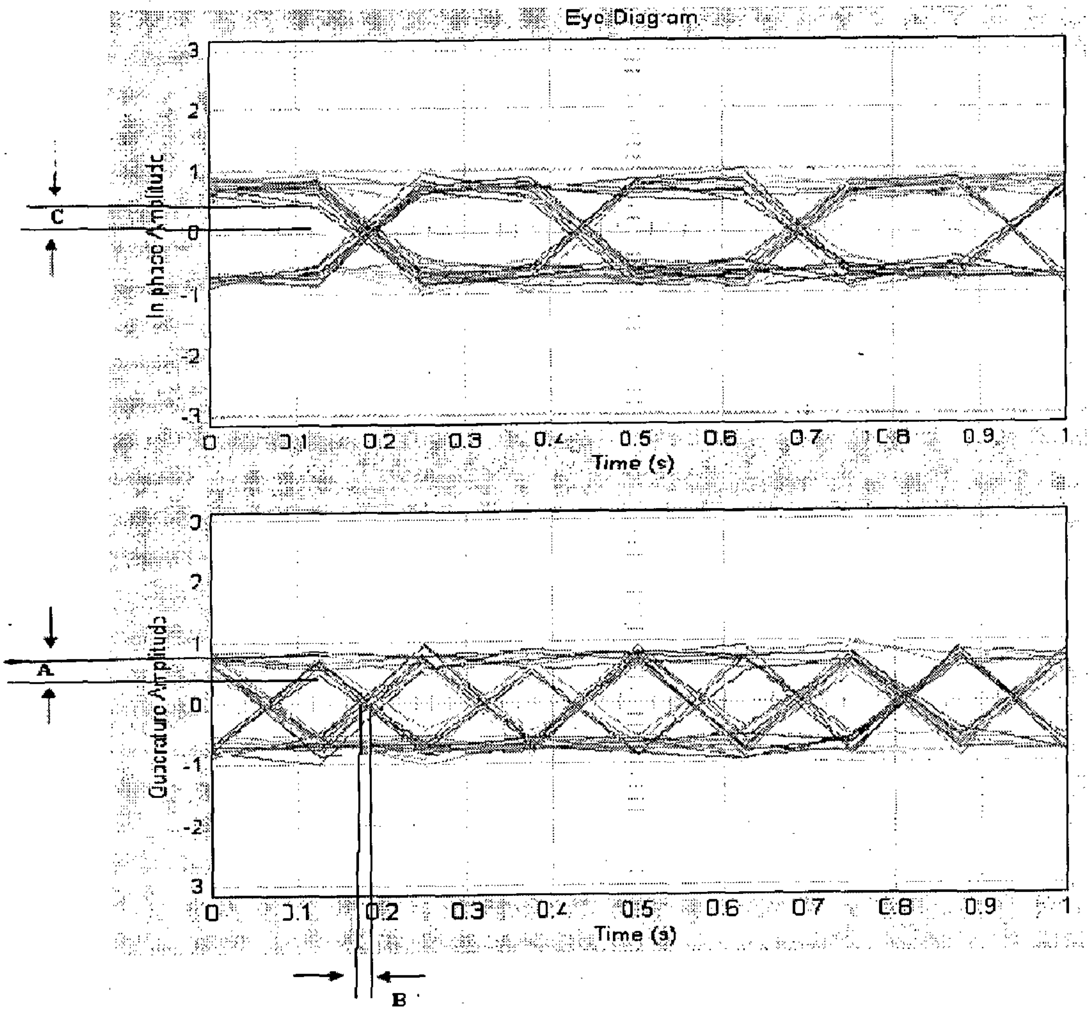


Figure 5.7: Eye diagram for QPSK modulated signal with SNR = 15 dB.

Figure 5.8 shows the eye diagram for SNR of 25 dB. The eye diagram has less distortion given that the eye opening is more defined. The proper eye opening defines less bit errors and hence less transmission error. Comparing Figures 5.7 and 5.8, it can be seen that the distortion in Figure 5.8 is 0.25 when compared with the distortion in Figure 5.7 which is 0.4. Similarly, the timing jitter is 0.01 for SNR = 25 dB and 0.02 for SNR = 15 dB.

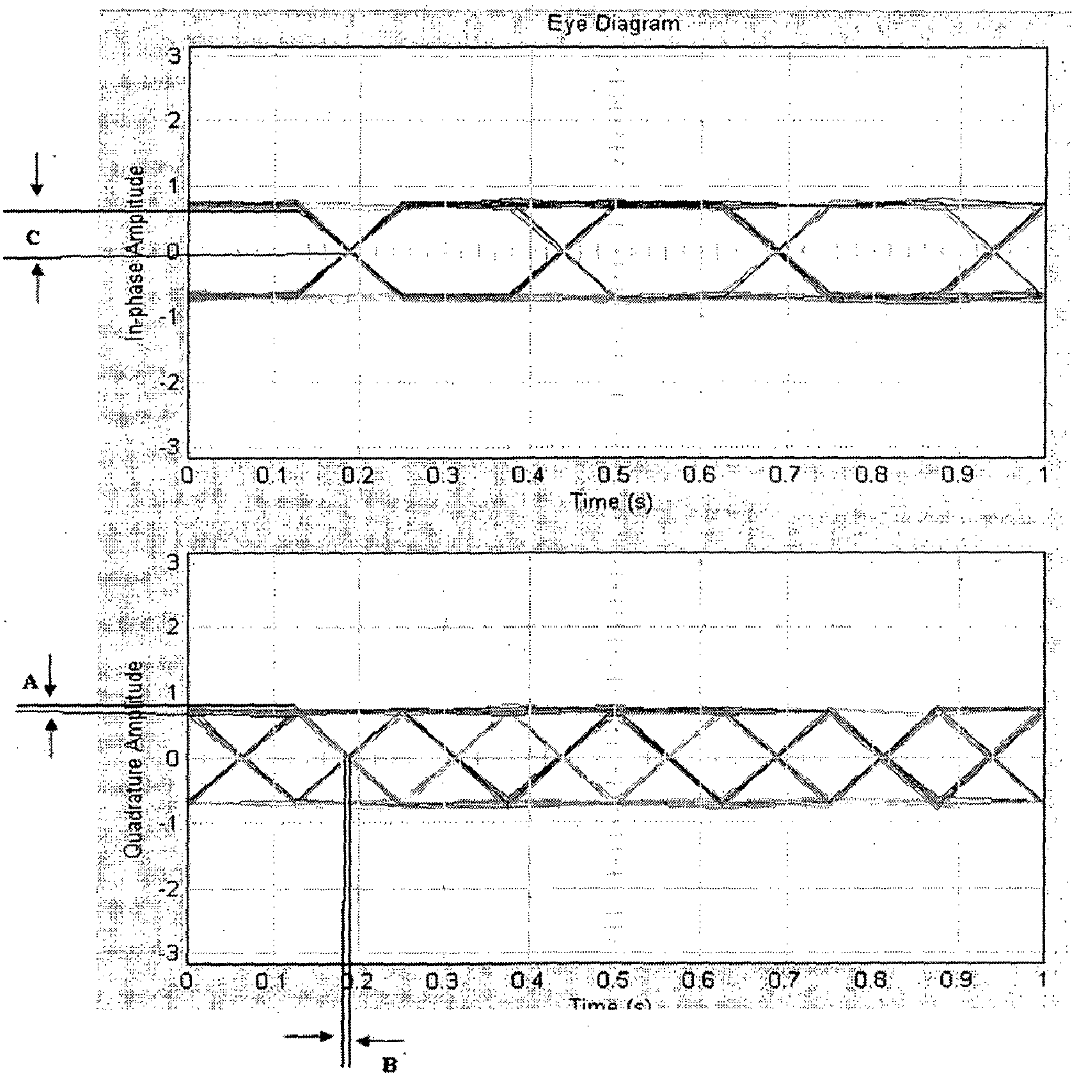


Figure 5.8: Eye diagram for QPSK modulated signal with SNR = 25 dB

The eye diagram for QAM transceiver is shown in figure 5.9. Recall that the power spectral density (PSD) of a QAM signal has a null-to-null bandwidth that is equal to the bit rate, which is half that of a QPSK signal. Therefore, QAM has twice the bandwidth efficiency of QPSK, since 4 bits are transmitted in a single modulation symbol instead of 2 bit for QPSK. Further, the bit error probability of QAM is identical to QPSK, while twice as much data can be sent in the same bandwidth. Thus, when compared to QPSK, QAM provides twice the spectral efficiency with exactly the same energy efficiency [7].

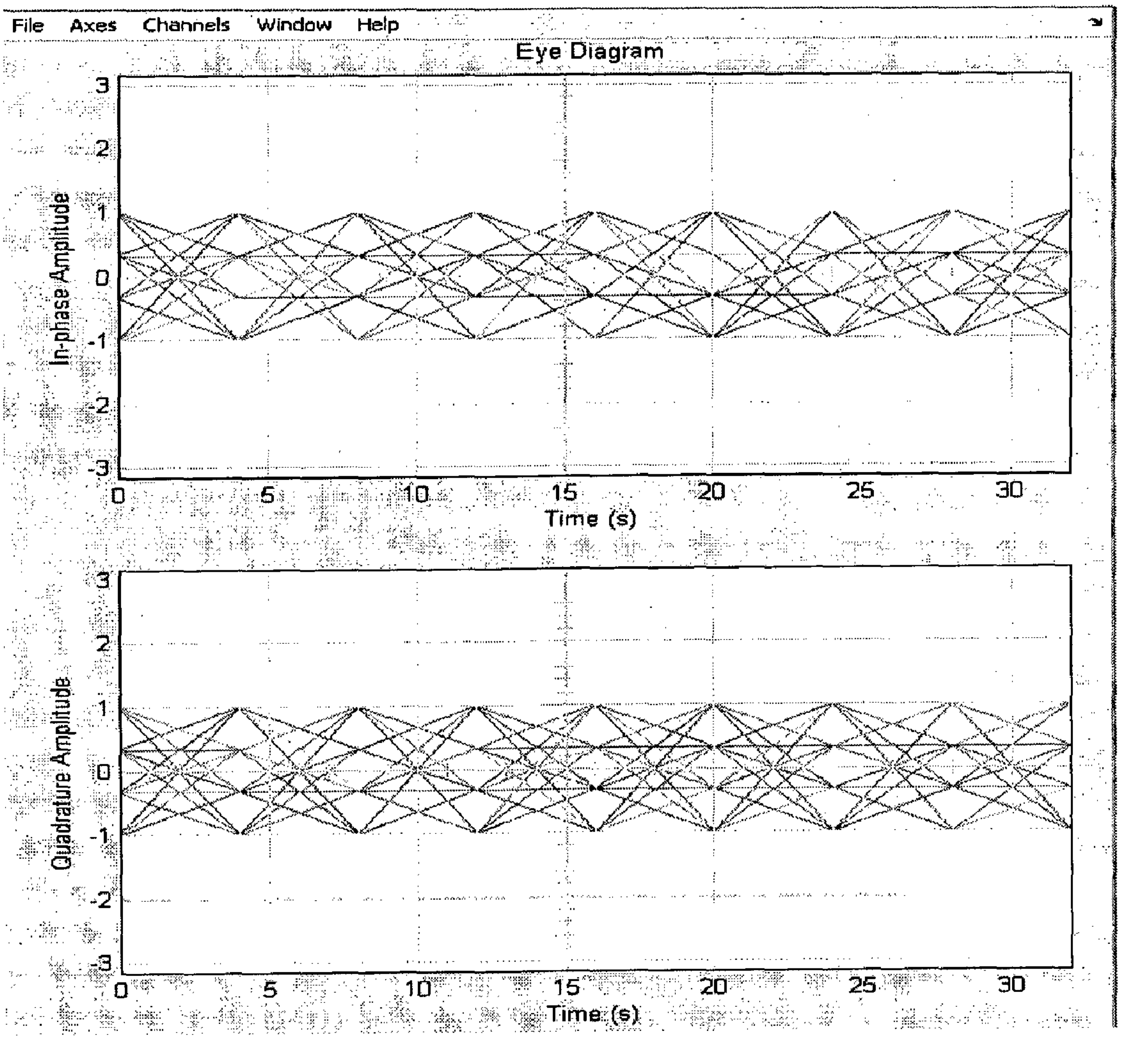


Figure 5.9: Eye diagram for QAM modulated signal with SNR = 100 dB

5.7.3 Results for sample rate conversion of EMG signal

Figures 5.11 to 5.19 shows functions of interpolation, up conversion on transmitter side and decimation, downconversion functions on receiver side.

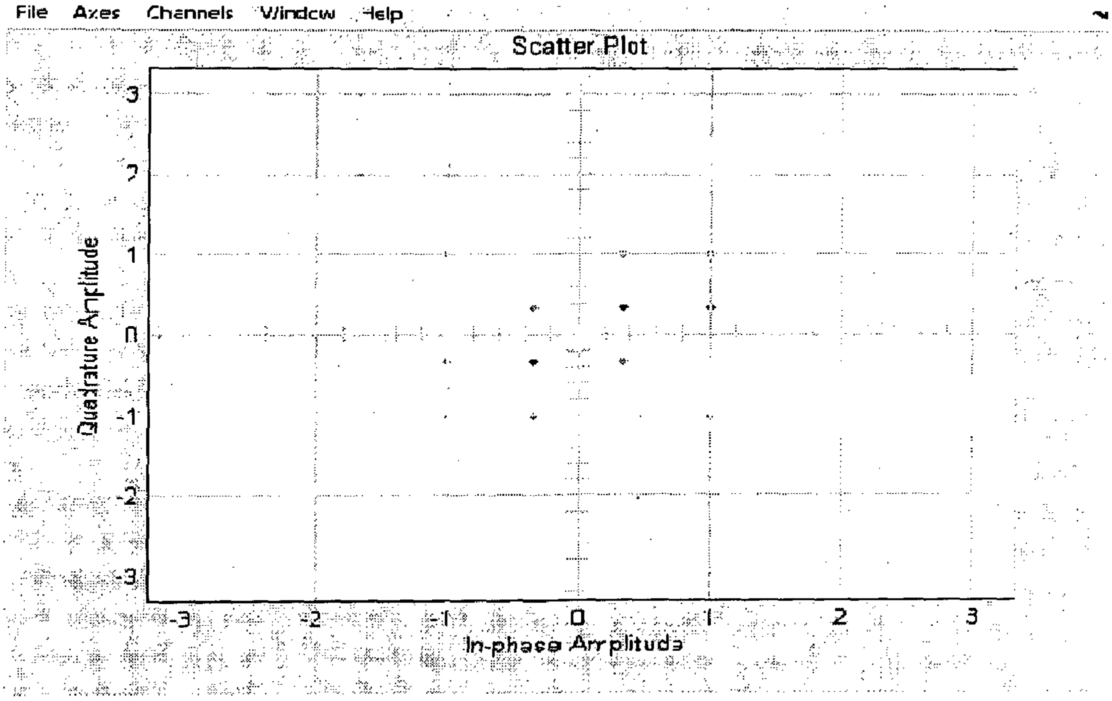
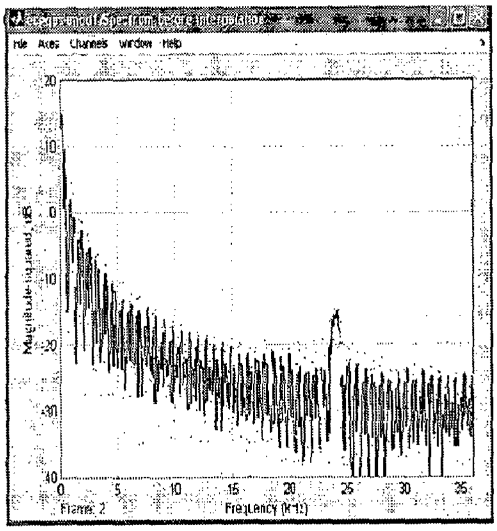


Figure 5.10: Eye diagram for QAM modulated signal with SNR = 100 dB

At transmitter end, EMG signal with sampling frequency of 24KHz centered at DC is converted into signal with sampling frequency 4.608 MHz shifted to frequency of 1.152 MHz. input to interpolation is having sampling frequency of 72 KHz.



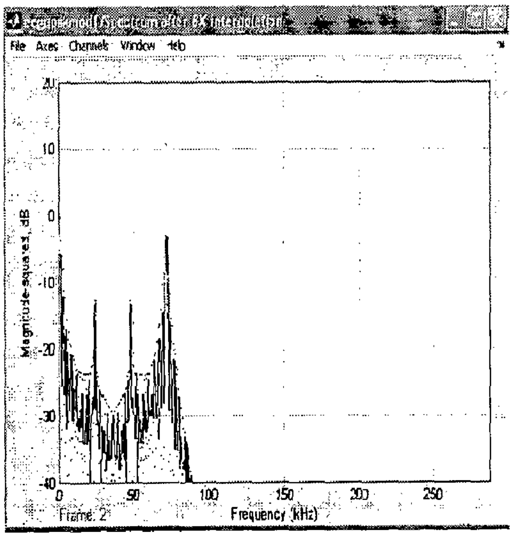
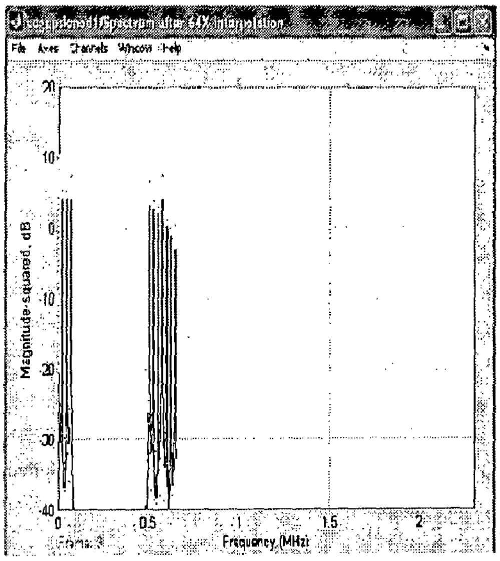


Figure 5.11 Spectrum before interpolation Figure 5.12 Spectrum after 8Xinterpolation



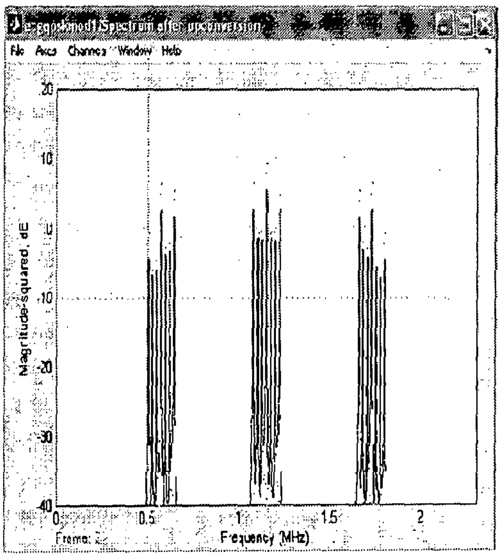
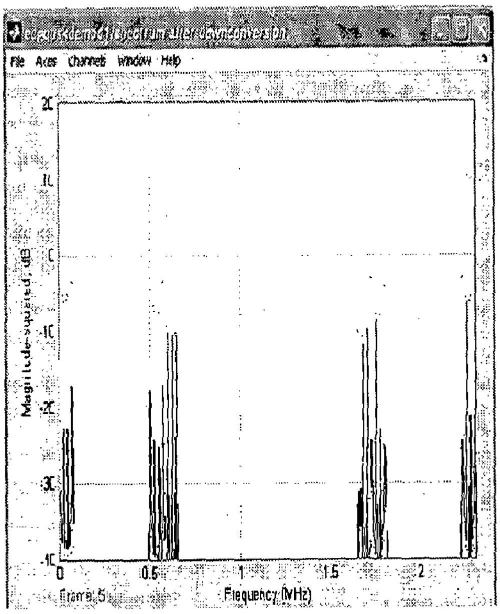


Figure 5.13 Spectrum after 64Xinterploation Figure 5.14 Spectrum after upconversion

At receiver side, Signal at sampling frequency of 4.608 MHz is received and downconverted into required frequency here it is DC.Then decimated into signal of sampling frequency of 24 KHz is shown in Figures 5.15 - 5.17.



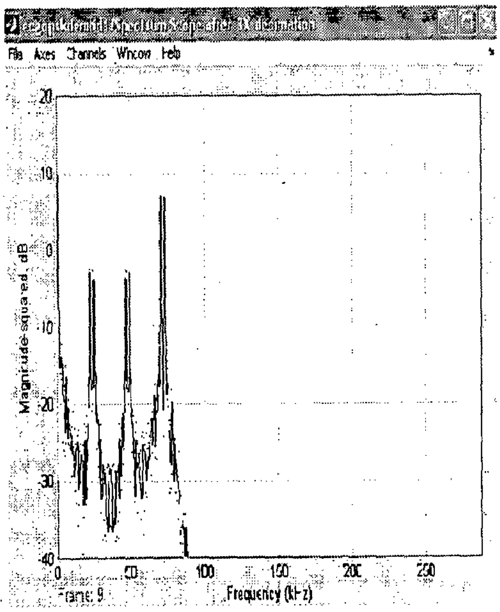


Figure 5.15 Spectrum after downconversion

Figure 5.16 Spectrum after 8X Decimation

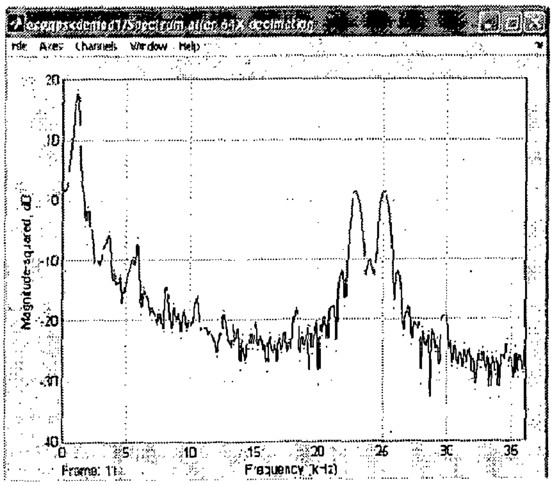
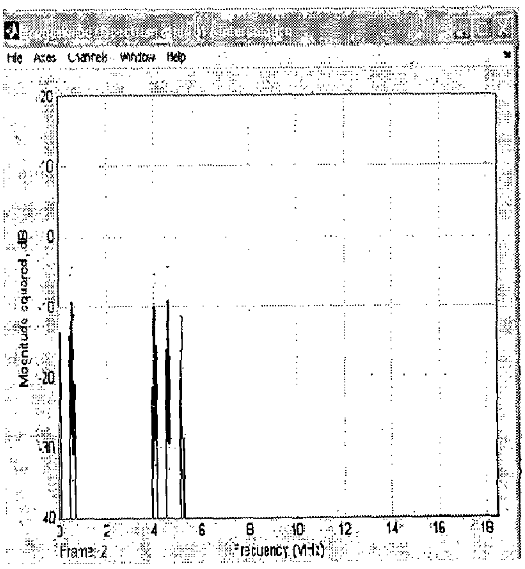


Figure 5.17 Spectrum after 64X Decimation

Figures 5.18 and 5.19 shows spectrum after interpolation with interpolation factor 512. and up conversion for sampling frequency of 36.864 MHz.



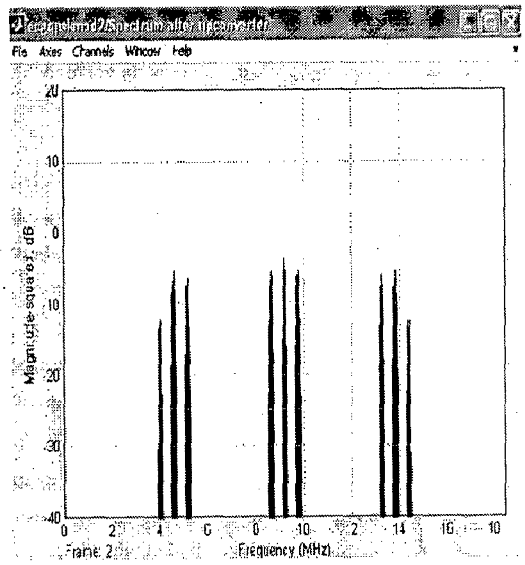
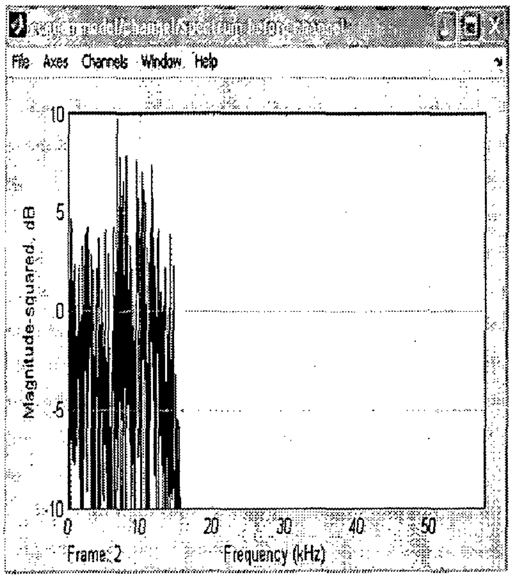


Figure 5.18 Spectrum after 512 X interpolation

Figure 5.19 Spectrum after Upconversion

5.7.4 Results from QAM system for ECG signal

ECG samples with sampling frequency 750Hz is given as input to QAM System. Figure 5.20 and 5.21 shows spectrum of signal before and after channel for QAM system



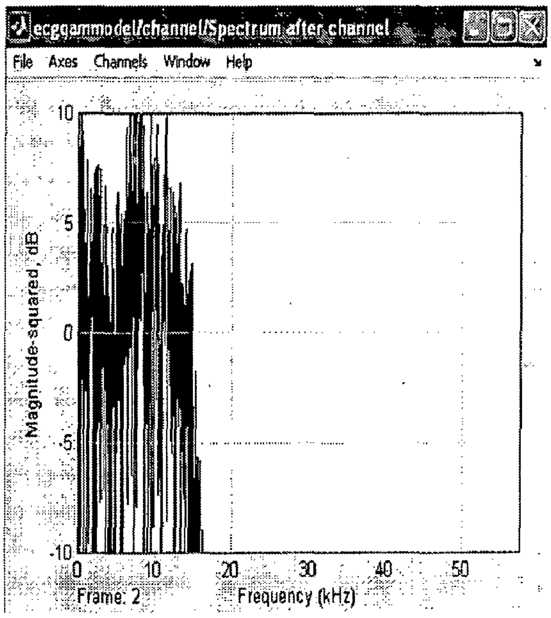

ecgqammodel/channel/Spectrum after channel
File Axes Channels Window Help
10
5
0
-5
-10
Magnitude-squared, dB
0
10
20
30
40
50
Frame: 2
Frequency (kHz)

Figure 5.20 Spectrum before channel

Figure 5.21 Spectrum after channel for QAM

Figure 5.22 square root raised cosine filter response with roll-factor 0.3 used for pulse shaping.

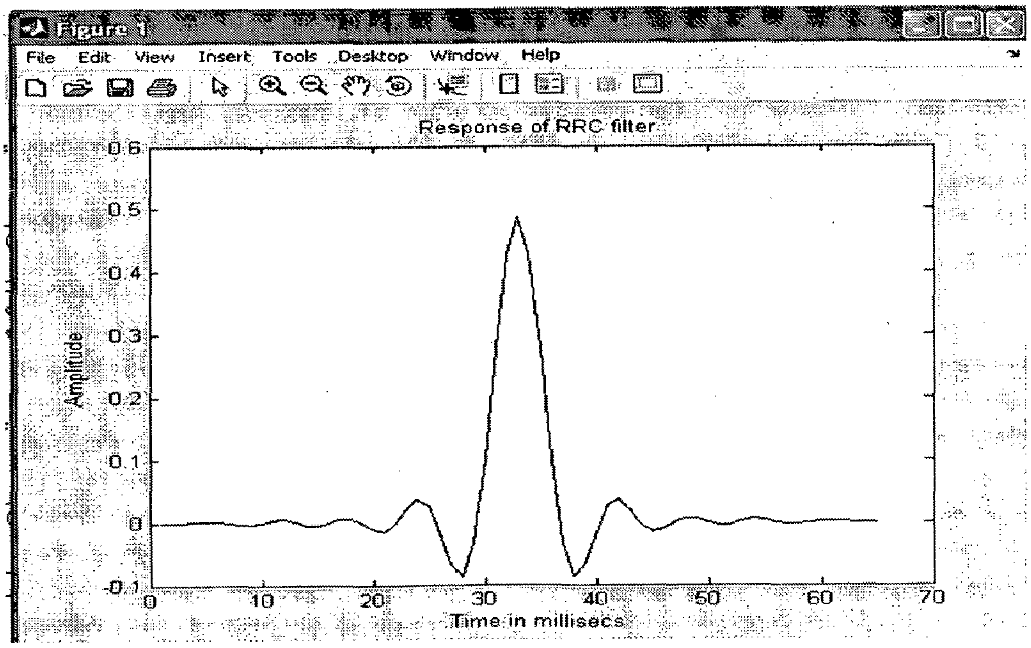


Figure 5.22 Response of RRC filter with roll of factor 0.3

5.7.4 Output Waveform of medical signals

Another indicator of performance is the observation of the output waveform compared to the input waveform. For compassion Electrocardiogram signal is transmitted through the whole system. For 16- QAM transceiver, when ECG signal samples with sampling frequency 750Hz is given as input. Figure 5.23 shows the input signal to the transmitter and the output signal of the receiver when both are synchronized. Observe that the input and output signals are similar for the system specifications discussed above, showing that the received signal is demodulated and decoded. Here it shows that there is a delay in output compared to input, as it takes time to download the configuration data into hardware i.e FPGA .After complete configuration, it provides the output in each cycle.

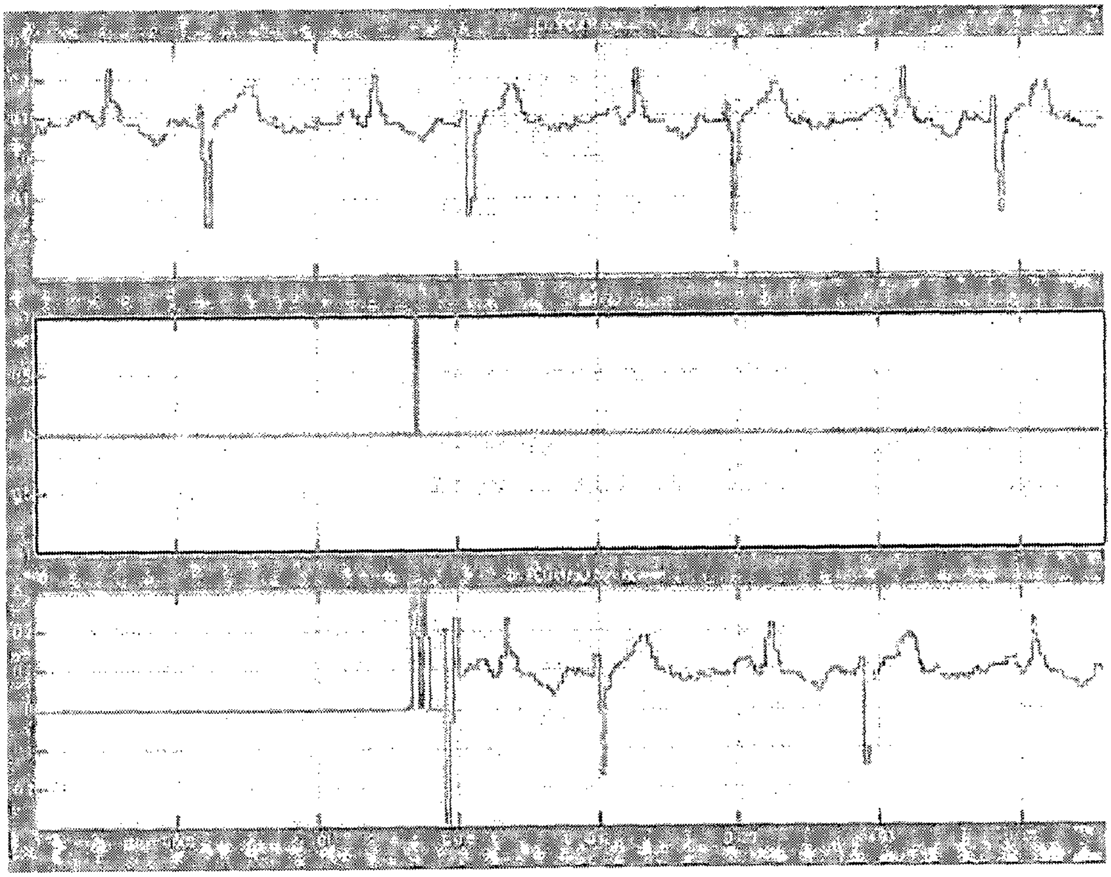


Figure 5.23: ECG Input and output waveforms of the QAM communication system.

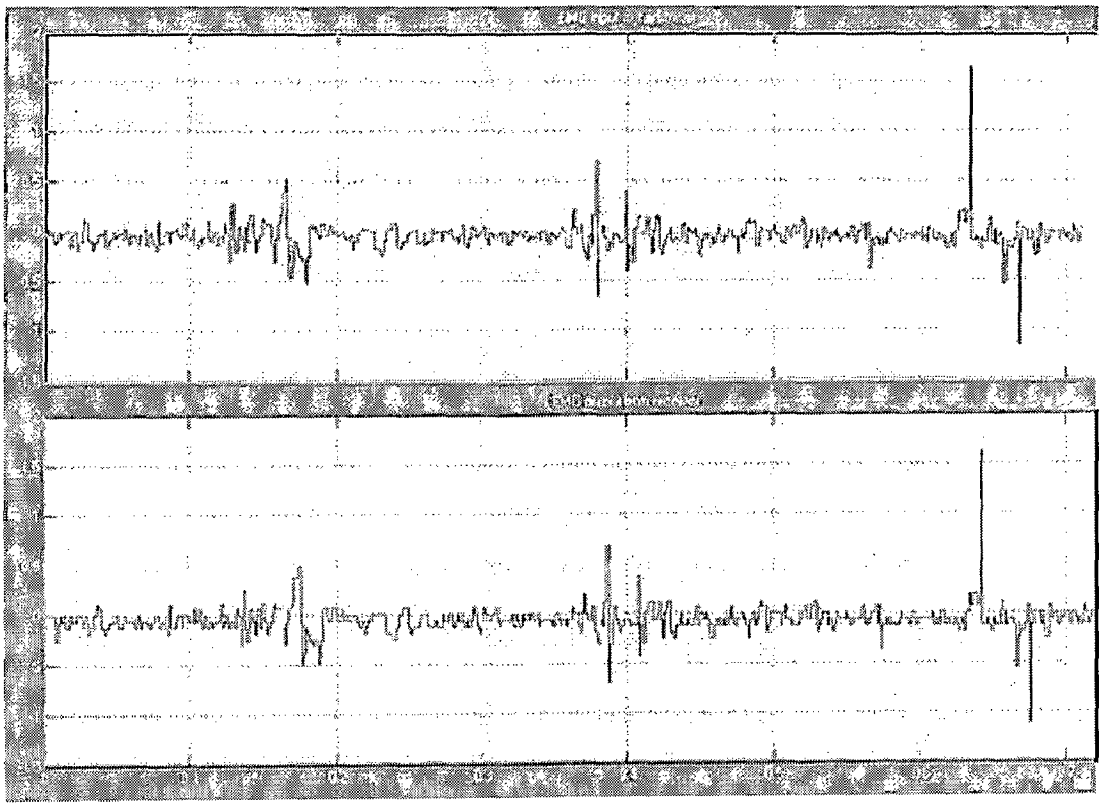


Figure 5.24: EMG Input and output waveforms of the QPSK communication system.

5.8 FPGA Implementation

After simulation and analysis of the results, the system is implemented on a Spartan 3E FPGA via hardware co-simulation [8]. The Digilent hardware board, which has a Spartan 3E FPGA chip on it is used to implement the system in hardware. The Spartan 3E is programmed through the joint test action group (JTAG) programming cable. The hardware co-simulation is performed by powering on the diligent board and connecting the JTAG cable from the board to the USB port of the personal computer (PC).

The Spartan 3E FPGA , 4Mbit Platform Flash configuration PROM and the 64-macrocell XC2C64A CoolRunner CPLD on the Diligent board, and any programmable devices on peripheral boards attached to the Diligent board can be programmed via the JTAG port. In this thesis, only the Spartan 3E FPGA was used.

FPGA is interfaced to PC through USB interface with data transfer frequency of 12MHz.

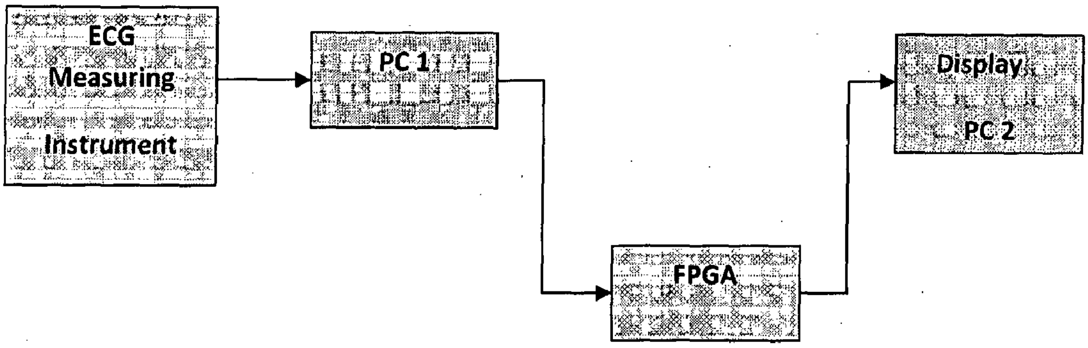


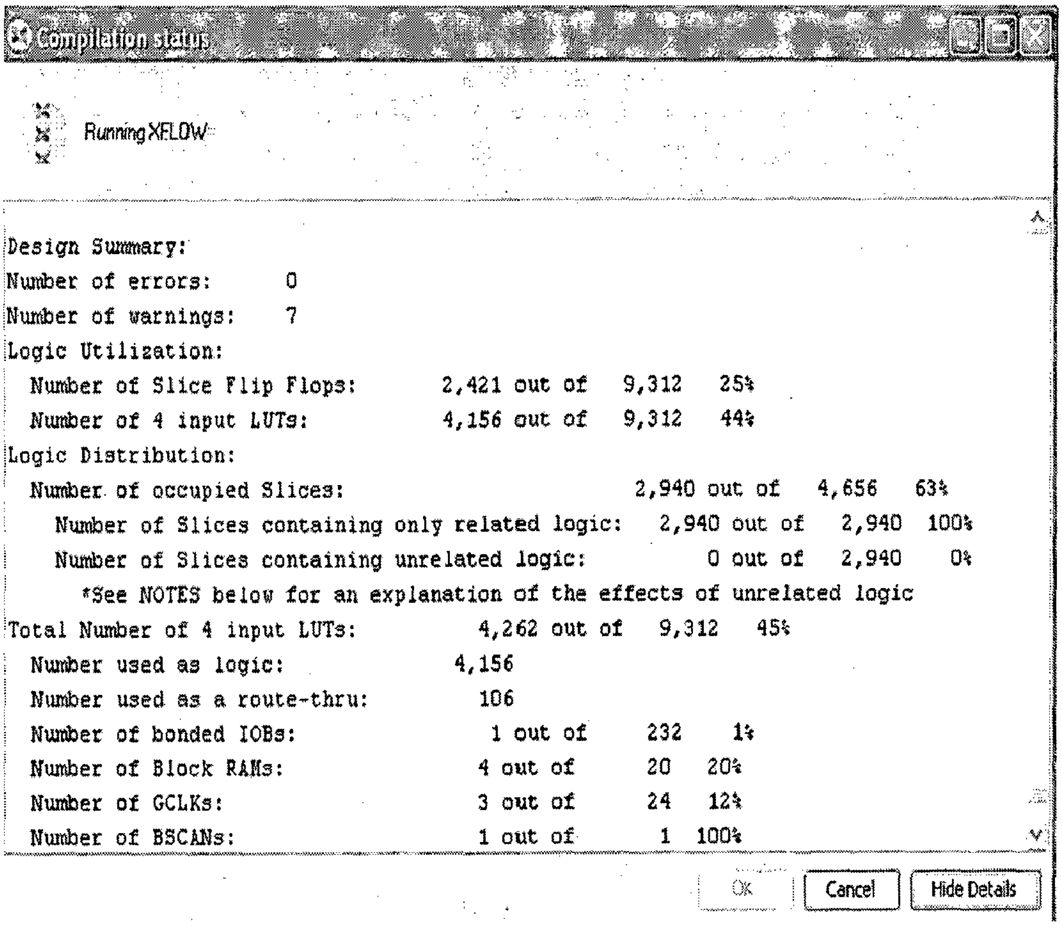
Figure 5.25 Experimental set up of system

Hardware co-simulation makes it possible to incorporate SIMULINK simulation into FPGA hardware directly. Hardware co-simulation is invoked by activating the System Generator, which should be present in all models containing System Generator blocks. Hardware co-simulation targets are organized under the hardware co-simulation sub-menu in the compilation dialog box field. When the compilation target, Spartan 3E already installed, is selected, the fields on the System Generator block dialog box are automatically configured with settings appropriate for the selected compilation target. System Generator remembers the dialog box settings for each compilation target.

Once the compilation target is selected, the System Generator code generator is invoked to compile the model for hardware co-simulation. The code generator produces a FPGA configuration bit stream for the design, suitable for hardware co-simulation. System Generator not only generates the HDL and netlist files for the model during the compilation process, but it also runs the downstream tools necessary to produce an FPGA configuration file.

A new window is opened when the implementation tools are running to produce the configuration bitstream file as shown in Figure 5.26. This window shows the progress and output of each tool as it is runs. It can be seen from Figure 5.26 that the information about the completion of mapping of the system in hardware is also displayed. The figure shows design summary The configuration bitstream contains the hardware associated with the model.

Hardware co-simulation compilation targets automatically create bitstreams and associate them with implementation blocks. When the design is simulated in SIMULINK, results for the complied portion are calculated in hardware. This allows the compiled portion to be tested in actual hardware, and can speed up simulation dramatically.



```
Design Summary:
Number of errors:       0
Number of warnings:     7
Logic Utilization:
  Number of Slice Flip Flops:       2,421 out of   9,312   25%
  Number of 4 input LUTs:           4,156 out of   9,312   44%
Logic Distribution:
  Number of occupied Slices:                      2,940 out of   4,656   63%
    Number of Slices containing only related logic:  2,940 out of   2,940  100%
    Number of Slices containing unrelated logic:         0 out of   2,940    0%
      *See NOTES below for an explanation of the effects of unrelated logic
Total Number of 4 input LUTs:         4,262 out of   9,312   45%
  Number used as logic:             4,156
  Number used as a route-thru:        106
  Number of bonded IOBs:              1 out of     232    1%
  Number of Block RAMs:               4 out of      20   20%
  Number of GCLKs:                    3 out of      24   12%
  Number of BSCANs:                   1 out of       1  100%
```

Figure 5.26: Command window showing the progress of implementation tools.

System Generator creates a new hardware co-simulation block for the design once it has finished creating the FPGA configuration bitstream. A SIMULINK library is also created in order to store the hardware co-simulation block information

Figure 5.27 shows the hardware co-simulation block generated for the QPSK transmitter of the model. At this point, the hardware co-simulation block can be copied out of the library and used in the SIMULINK simulations instead of the Xilinx blocks. In this thesis, Figure 5.13 is used in the design and is simulated with other SIMULINK blocks. When simulation is complete, the hardware co-simulation block interacts with the underlying hardware and produces the output.

A hardware co-simulation block consumes and produces the same types of signals as that of other System Generator blocks. When a value is written to one of the hardware co-simulation block's input ports, the block sends the corresponding data to the appropriate location in hardware. Similarly, the block retrieves data from hardware when there is an event on an output port.

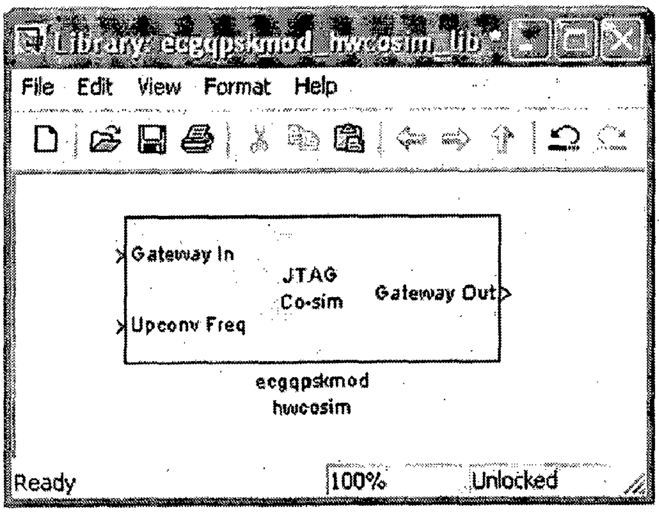


Figure 5.27: Hardware co-simulation library.

Figure 5.28 shows the constellation diagram output obtained by implementing the QAM transceiver model in hardware co-simulation and in simulink simulation

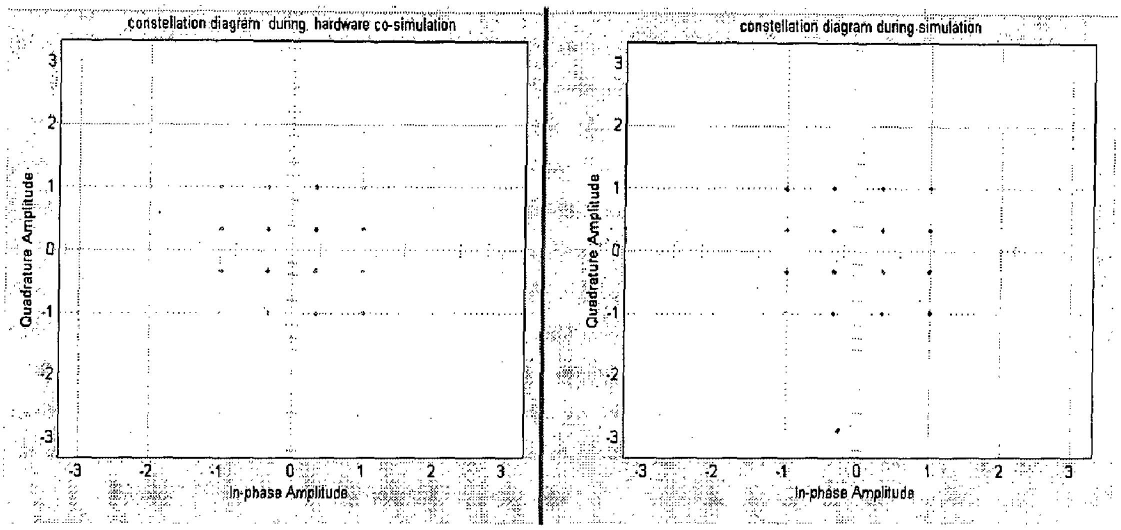


Figure 5.28: Constellation diagram for hardware co-simulation and simulation

Figure 5.29 shows output signal when, QPSK system is implemetated on FPGA with hardware co-simulation with Electrocardiogram signal as input.

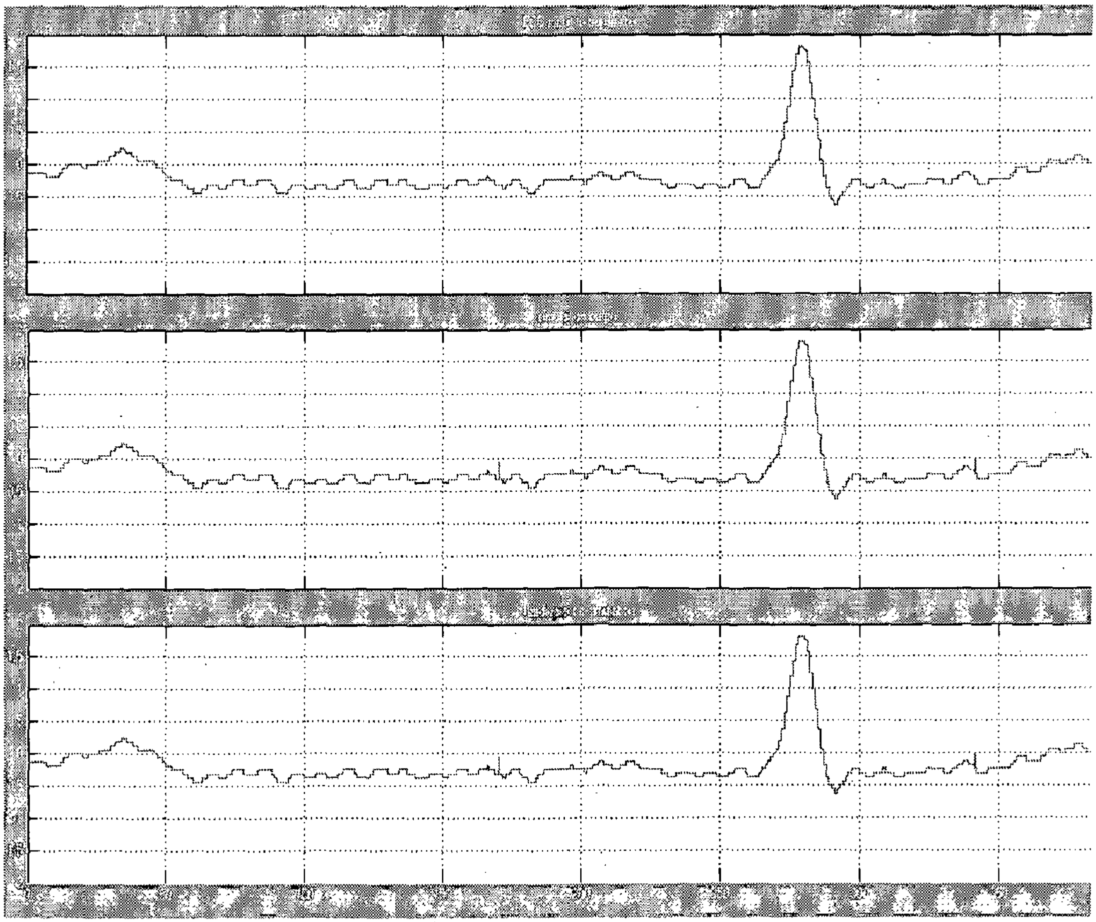


Figure 5.29 ECG input and output waveforms of system in hardware co-simulation

From figures 5.28 and 5.29, it is evident that the two results are similar, with little or no difference. Thus we have shown that the hardware implementation result is the same as the simulated results.

5.9 Summary

In this Chapter, the experimental setup, simulation, FPGA implementation and corresponding results for the baseband section of software defined radio is presented. Also hardware implementation of the system is presented. The simulation results were compared with the results from FPGA implementation.

Chapter VI
Conclusions and Future work

6.1 Conclusions

This thesis consists of two major tasks. First, a study of the software defined radio concept and, second design and implementation of software radio transmitter and receiver.

The first task of this thesis was to investigate the current state of the art in software defined radio. This is a very large subject area with many promising applications. A global overview of the software radio is given, and some of the most important aspects of software radio such as the definition, characteristics, advantages, technological challenges and the hardware choices available for implementation are described and analyzed.

The second task is to build a baseband section of software defined radio based wireless communication system. In this thesis, the baseband functions like modulation, forward error correction and frequency translation were simulated and then targeted to hardware implementation.

The design methodology, softwares and hardwares that are used were discussed. The results such as the constellation diagrams and eye diagrams at the output of the transmitter are used to quantify the performance of the system. Furthermore, the system is tested by comparing the input waveform to the transmitter and the output waveform from the receiver. The results obtained indicate that the system produces similar results from simulation and hardware implementation.

Though the automation seems to be helpful, there are few constraints encountered in this thesis while using System Generator. Not all the communication functions are available in the predefined library. But the functionalities can be written as code in MATLAB, C or VHDL and can be imported into the system using the black box features of the System Generator. The next constraint is in the hardware implementation. There are chances that when the design is targeted in hardware it would not be placed and routed optimally. In spite of these constraints, the automation facility of the System Generator proves to be useful for system implementation.

6.2 Future work

In this thesis the baseband and intermediate frequency sections of the wireless communication system has been implemented. The radio frequency section have not been implemented. Therefore, further research could focus on an end-to-end implementation of a wireless communication system in software defined radio.

The channel is considered to have additive white Guassian noise and Doppler shift. Inter-symbol interference, phase error, multipath fading, etc., can be added to the channel, in order to closely simulate real life systems.The single channel radio design that was implemented as a part of this research could be extended to a multi-channel design to boast the data rates of the system.

A complete communication system for any particular air standard (e.g., GSM, IS-95, or CDMA2000), can be defined in software. The partial reconfiguration capabilities of the FPGAs could be further exploited by moving more parts of the radio from static to reconfigurable sections, thus making the architecture more flexible and user controllable.

REFERENCES

[1] T. Rappaport, “Wireless Communications - Principles & Practice, ” 2nd edition. Prentice-Hall, Upper Saddle River, NJ, 1996.

[2] Kohno , “Telemedicine , communications and health information,” Springer-Verilag 4th edition 1997, pp. 267-281.

[3] Bingyi Hu and Jing Bai , “ An internet based communication server for telemedicine,” IEEE proceedings, 19th International Conference,oct 1997.

[4] J. Mitola, III, “The software radio architecture,” IEEE Commun. Mag., vol33, no.5, May 1995, pp. 26-38.

[5] Walter Tuttlebee, “Software Defined Radio Enabling Technologies ,” Jonwelly & Sons, Ltd England, 2002.

[6] Ediana Sutjiredjeki and Soegi Soegijoko “ Development of a Mobile Tele medicine System with Multi Communication Links for urban & Rural Areas in Indonesia,” Biomed 06 , IFMBE Proceedings 15 , pp. 660-663 , 2007.

[7] Bernard sklar,”Digital communicationsFundamentals and Applications,”2ndedition, Prentice Hall P T R, New Jersey, 2003.

[8] U. Meyer Baese , “ Digital Signal Processing with Field Programmable Gate Arrays," Springer-Verilag, New Delhi, 2003.

[9] Medical info-communications signals an era of body area networking from http:/rfdesign.com/next_generation_wireless/short_range_wireless/radio_medical_infocommunications_signals/index1.html.

[10] John Bard,”Software Defined Radio the Software Communications Architect ure,” John Wiley & Sons Ltd,, USA,2007.

[11] Pallavi Mannar Mannan ,” Framework for the design and implementation of software defined radio based wireless communication system , ” Ph.D Dissertation, Virginia University, Dec, 2002.

[12] Push-button performance using System Generator for DSP. from http://www.xilinx.com/products/software/sysgen/sysgen _sellsheet.pdf.

[13] M.Cummings and S.Haruyama,“FPGA in the software radio,” IEEE Commun. Mag., vol. 37, no. 2, Feb. 1999, pp. 108-112.

[14] Paul Burns," Software Defined Radio for 3G ,"Arctech house Inc , 2003.

[15] G. J. Minden , J. B. Evans ," KUAR: A Flexible Software - Defined Radio Development Platform," Prentice-Hall, Upper , 2000.

[16] Software Defined Radio definitions from http://www.sdrforum.org/faq.html.

[17] J. Mitola III , " Technical challenges in the globalization of software radio," IEEE Commun. Mag., vol. 37, no. 2, Feb. 1999, pp. 84-89.

[18] A. Wiesler and F. Jondral ," A software radio for second- and third-generation mobile systems, " IEEE Trans. On Vehicular Technology, vol. 51, no. 4, July. 2002,pp. 738-748 .

[19] Matlab , The Mathworks Inc ., [online] , [cited in Jan. 2007] , available from http://www.mathworks.com/products/matlab/description1.html.

[20] Marko Kosunen,"Digital signal processing and digital-to-analog converters for wide-band transmitters ,"November, 2006.

[21] Xilinx System Generator for DSP , Xilinx Inc , [online], [cited in Jan. 2007], available from http://www.xilinx.com/ise/optional prod/system generator.htm.

[22] Xilinx ISE 9.2 Overview,Digital Design Laboratory [online], cited in Jan.2007 available from http://www.seas.upenn.edu/ ese201/ ise/ISEIntroduction.pdf.

[23] Gerald Youngblood," A Software-Defined Radio for the Masses ,Part 1, " Jul/ Aug 2002

[24] Design Flow using System Generator for DSP.Retrieved on Feb 13, 2008 from http://www.xilinx.com/products/software/sysgen/design_flow.htm.

[25] Alexander Vießmann ,Tobias Scholand, " Falcon , a software defined radio transceiver concept ," The 17th Annual IEEE Symposium on Mobile Radio Communications , PIMRC 2006.

[26] K. Compton and S. Hauck, "Reconfigurable computing : A survey of systems software," ACM Computing Surveys, vol. 34, issue 2, June 2002, pp. 171-210.

[27] XilinxUser Guide http://www.xilinx.com/product/software/ap/user_guide.htm

[28] Spartan-3E development Kit Board User Guide UG230 (v1.0)March 9, 2006.

[29] Nikhil S. Bhatia," A Physical Layer Implementation of ReconfigurableRadio," Ph.D dissertations,December 3rd, 2004.

[30] Adam S. Harrington," Software Defined Radio The Revolution of Wireless Communication," Ball State University Fall Semester 2004.

[31] D.Chin and S.Lam. "Implementing DSP designs with Xilinx System Generator implementation tools." from http://www.synplicity.com/literature/sys/DSP.pdf

[32] C.Wang,"Forward error correction coding,"Crosslink-space Commu,Vol.No:1, Winter 2002 ,http://www.aero.org/publications/crosslink/winter2002/04.html.

[33] Abdul-Rafeeq and Valek Szwarc ," A High performance soft decision viterbi decoder for WLAN,"IEEE CCECE/CCGEI, Ottawa, May 2006.

[34] Digital Modulation in Communication System,Application Note 1298, Hewlett Packard, 1997.

[35] Chris Dick, Michael Rice , " Configurable logic for wireless communications: carrier and symbol synchronization , " International IC , Korea & Conference Proceedings,2000.

[36] Lingwen Zhang, Zhenhui Tan," Design of Optimum Filters for WCDMA Base Station Based on SDR," IEEE proceedings, 2006.

[37] Helen Tarn, Kevin Neilson , " Designing Efficient Wireless Digital Up and Down Converters Leveraging CORE Generator and System Generator ," application Note , October 22, 2007

[38] Xiaoxin Cui, Dunshan Yu," Design and Implementation of Digital Up Conve--rter for Homenet," IEEE Proceedings ,2005.

[39] Zhuan Ye, John Grosspietsch ,"An FPGA Based All-Digital Transmitter with Radio Frequency for Software Defined Radio,"IEEE proceedings, 2007.

[40] Gerard K. Rauwerda , Paul M . Heysters ," Towards Software Defined Radios Using Coarse-Grained Reconfigurable Hardware," IEEE transactions on large scale integration (VLSI) systems, Vol .16,No.1,January 2008

[41] Yanyang Zhao , Ligen Wang ," A software defined radio receiver architecture for UWB communications and positioning," IEEE CECE, Ottawa, May 2006.

[42] Asad, A. Abidi , Fellow, " The Path to the Software-Defined Radio Receiver ," IEEE journal of solid state circuits, vol.42, No.5, may 2007.

[43] Medical signals data samples from www.physionet.org.

APPENDIX A

RTL simulation

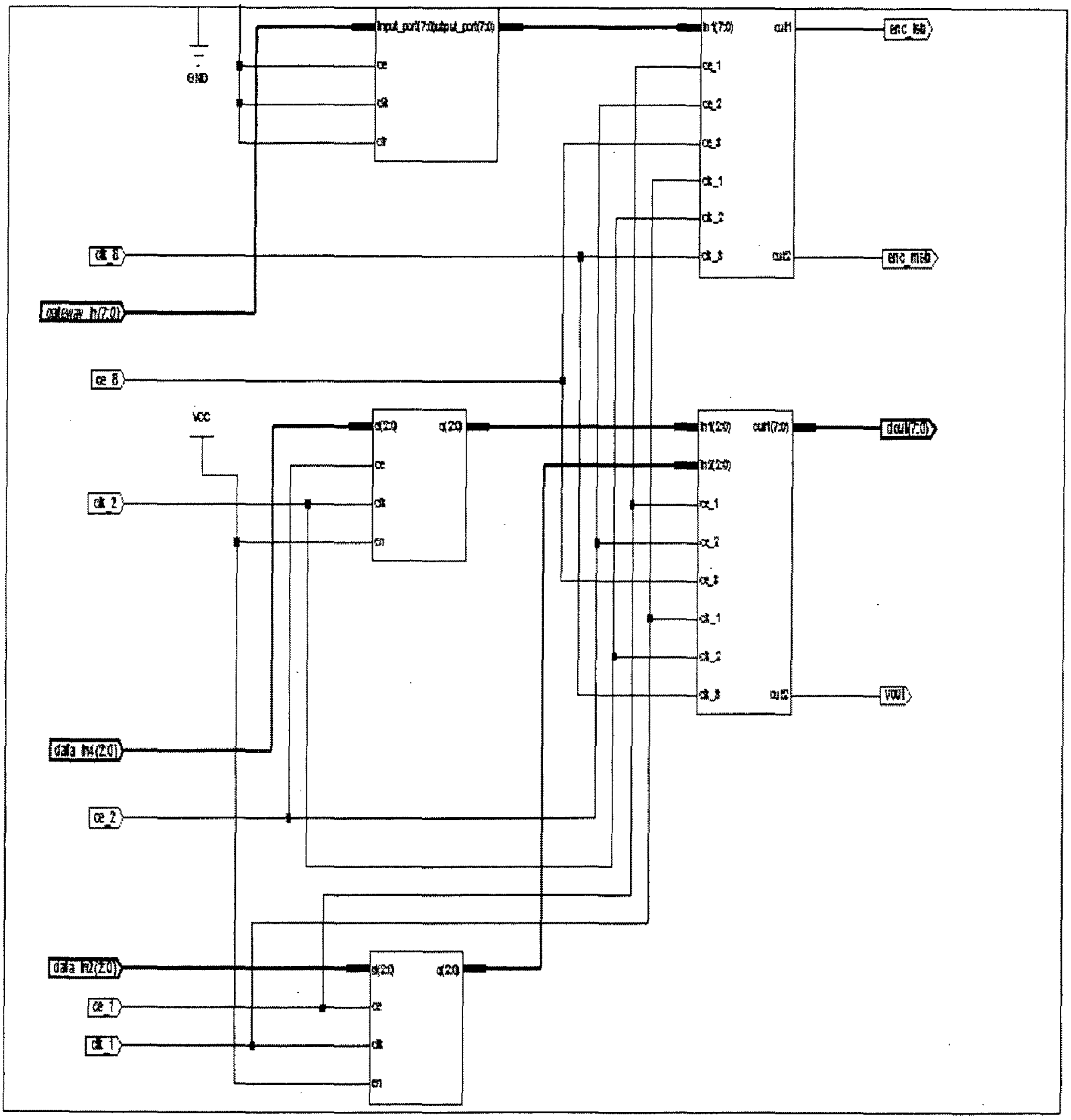


Figure A.1: RTL simulation of QAM transmitter

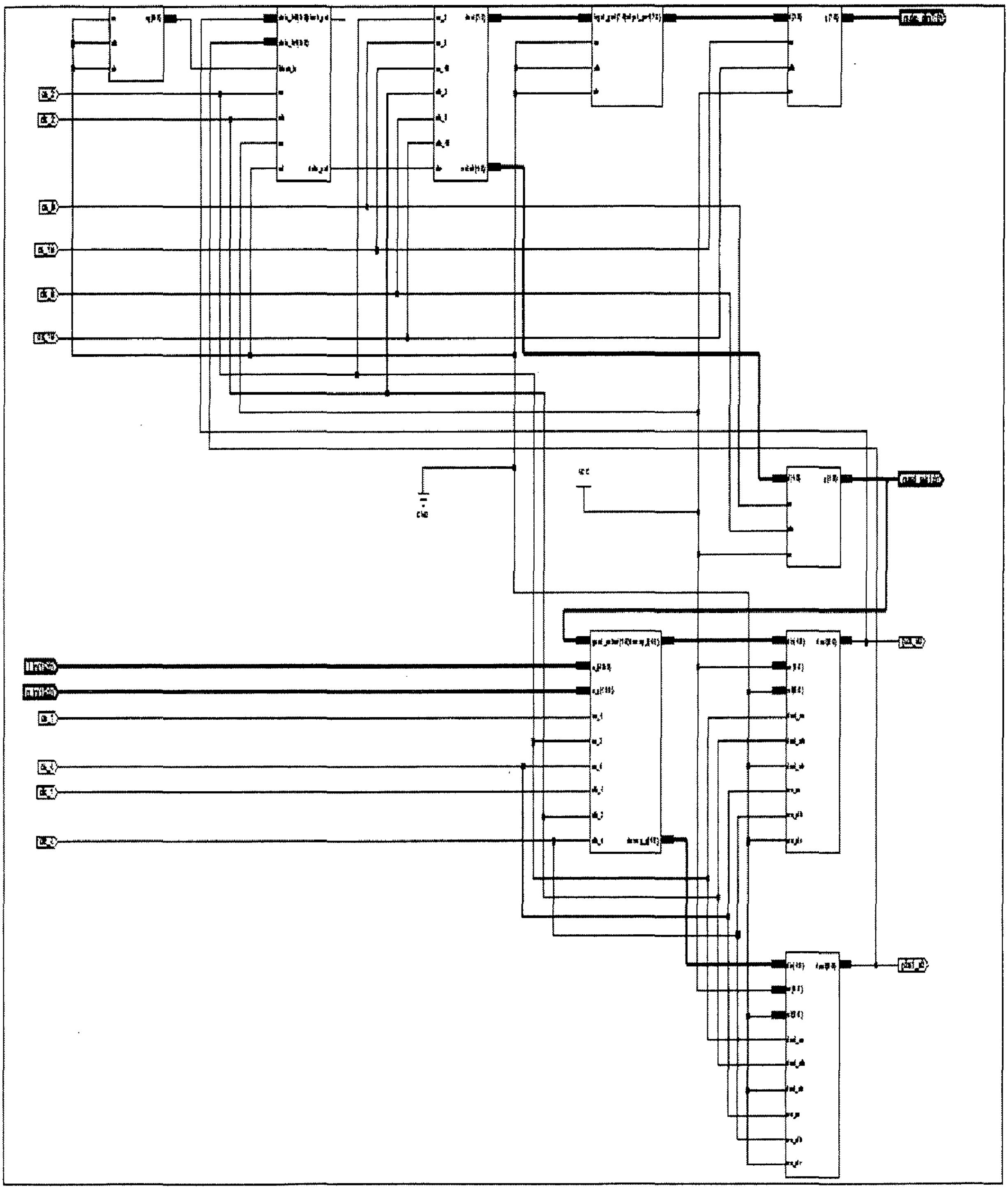


Figure A.2 RTL simulation of QAM receiver.

APPENDIX B

Synthesis details

Synthesis report for 16-QAM receiver

Translation report

```
Partition Implementation Status
-------------------------------
  No Partitions were found in this design.
-------------------------------
NGDBUILD Design Results Summary:
  Number of errors:    0
  Number of warnings:  6
Total memory usage is 76600 kilobytes
```

Map report

```
Logic Utilization:
  Number of Slice Flip Flops:                   1,503 out of  9,312   16%
  Number of 4 input LUTs:                       1,961 out of  9,312   21%
Logic Distribution:
  Number of occupied Slices:                    1,384 out of  4,656   29%
  Number of Slices containing only related logic: 1,384 out of  1,384 100%
  Number of Slices containing unrelated logic:  0     out of  1,384    0%
  Number used as Shift registers:               12
  Number of bonded IOBs:                        1   out of    232      1%
  Number of Block RAMs:                         6   out of     20     30%
  Number of GCLKs:                              3   out of     24     12%
   Number of BSCANs:                             1   out of      1  100%
Total equivalent gate count for design:         425,765
Additional JTAG gate count for IOBs:            48
```

```
Peak Memory Usage:                          222 MB
Total REAL time to MAP completion:          1 mins 46 secs
Total CPU time to MAP completion:           1 mins 32 secs
```

Place and route report

```
************************
Generating Clock Report
*************************
+-------------- - + ----------- +  ------  + ------ +  -----------   +  ------------  +
|  Clock Net   | Resource | Locked  | Fanout | Net Skew (ns)  |  Max Delay (ns) |
+-----------------+-------------+-  -----  +  -----  +  -----------   +  ------------ +
| clk_BUFGP  | BUFGMUX |   No    |  841   |     0.086      |     0.203      |
```

The Delay Summary Report

```
The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is:   0

The AVERAGE CONNECTION DELAY for this design is:                0.924
The MAXIMUM PIN DELAY IS:                                        4.461
The AVERAGE CONNECTION DELAY on the 10 WORST NETS is:           3.854

  Listing Pin Delays by value: (nsec)
   d < 1.00  <  d  <  2.00  <  d < 3.00  <   d <  4.00  < d  < 5.00  d >= 5.00
   ---------    ---------     ---------     ---------     ---------   ---------
     4478         1842          661           133            5           0

All constraints were met.
```

Static timing report

```
Timing summary:
Timing errors:  0        Score:     0
   Constraints cover  48817 paths   ,   0 nets ,   and     7095 connections
```

Design statistics:

Minimum period: 11.516ns (Maximum frequency: 86.836MHz)

Maximum path delay from/to any node: 11.516ns

Logic utilization summary for QAM transmitter

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	89	9,312	1%
Number of 4 input LUTs	28	9,312	1%
Logic Distribution			
Number of occupied Slices	86	4,656	1%
Number of Slices containing only related logic	86	86	100%
Number of Slices containing unrelated logic	0	86	0%
Total Number of 4 input LUTs	**54**	**9,312**	**1%**
Number used as a logic	28		
Number used as a route-thru	8		
Number used as shift registers	18		
Number of bonded IOBs	41	232	17%
Number of Block RAMs	2	20	10%
Number of GCLKs	1	24	4%
Total equivalent gate count for design	**133,185**		
Additional JTAG gate count for IOBs	1,968		

Table B.1 Device utilization for QAM transmitter

Logic utilization summary for QPSK transmitter

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	2,056	9,312	22%
Number of 4 input LUTs	3,755	9,312	40%
Logic Distribution			
Number of occupied Slices	2,528	4,656	54%
Number of Slices containing only related logic	2,528	2,528	100%
Number of Slices containing unrelated logic	0	2,528	0%
Total Number of 4 input LUTs	**3,777**	**9,312**	**40%**
Number used as a logic	3,755		
Number used as a route-thru	22		
Number of bonded IOBs	53	232	22%
Number of Block RAMs	2	20	10%
Number of GCLKs	1	24	4%
Total equivalent gate count for design	**200,466**		
Additional JTAG gate count for IOBs	2,544		

Table B.2 Device utilization for QPSK transmitter

Logic utilization summary for QPSK receiver

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	4,659	9,312	50%
Number of 4 input LUTs	6,510	9,312	69%
Logic Distribution			
Number of occupied Slices	3,805	4,656	81%
Number of Slices containing only related logic	3,805	3,805	100%
Number of Slices containing unrelated logic	0	3,805	0%
Total Number of 4 input LUTs	**6,880**	**9,312**	73%
Number used as a logic	6,510		
Number used as a route-thru	82		
Number used as shift registers	288		
Number of bonded IOBs	135	232	58%
Number of Block RAMs	2	20	10%
IOB Flip Flops	68		
Number of Mult 18X18IOs	15	20	75%
Number of GCLKs	1	24	4%
Number of macros	16		
Total equivalent gate count for design	**268,710**		

Table B.3 Device utilization for QPSK receiver