"SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique"

(Thesis submitted towards the partial fulfillment of requirement for the award of degree of)

Master of Engineering

In

Electronics and Communication

Submitted by

Sudhanshu Mehta

Roll no: 800961024(ECED)

Under the guidance of

Ms. Surbhi Sharma

Assistant Professor, ECED

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA-147004(PUNJAB)

June-2011

CERTIFICATE

I, hereby certify that the work which is being presented in this thesis entitled **"SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique."** in partial fulfillment of requirements for the award of degree of the Master of Engineering in Electronics and Communication from Thapar University, Patiala, is an authentic record of my own work carried under the supervision of **Ms. Surbhi Sharma**, Assistant Professor, ECED.

The matter presented in this thesis has not been submitted in any other University/Institute for the award of Master of Engineering.

(Sudhanshu Mehta)

Signature of the student

Dated: 17-6-11

This is certified that the above statement made by the candidate is correct to the best of my knowledge

(Ms. Surbhi Sharma)

Assistant Professor, ECED

Date: 17-6-11

Head of Department, ECED

Thapar University, Patiala

Date: 17/6/11

(Dr. S.K. Mohapatra)

Dean, Academic Affairs

Thapar University, Patiala

Date:

i

ACKNOWLEDGEMENT

A good job is never the outcome of the efforts of a single person. I feel myself lucky to express my profound sense of gratitude and respect to all those who helped me directly or indirectly throughout my thesis.

I would like to give special thanks to my guide **Ms. Surbhi Sharma** Assistant Professor **ECED**, Thapar University, Patiala, for his advice, kind assistance, and invaluable guidance. It has been a great honor to work under her.

I would like to express my deepest gratitude to **Dr. Rajesh Khanna**, Professor, Electronics and Communication Engineering Department, Thapar University, Patiala, for his advice, motivation, guidance, moral support, efforts and the attitude with which he solved all of my queries in making this report possible. It has been a great honor to work under him.

I am also thankful to **Dr. A.K. Chatterjee**, Professor & Head, Electronics and Communication Engineering Department, for providing us with adequate infrastructure in carrying the work.

I would also like to thank all the faculty members of ECED for their intellectual support and also special thanks to my family and my friends who constantly encouraged me to complete this work. I am also thankful to the authors whose work I have consulted and quoted in this work.

Sudhanshu Mehta

Roll no. 800961024

ABSTRACT

With the exponential growth in the ways and means by which people need to communicate data communications, voice communications, video communications, broadcast messaging, command and control communications, emergency response communications, etc. modifying radio devices easily and cost-effectively has become critical. The future information age is equipped with rich and affordable telecommunication services. So in future people will be more flexible while using the wireless equipments. Wireless communications systems and standards have been developed around the world without any global plan. Recently hardware technology evolved significantly. Some of the key milestones in this progress are transition from analog hardware to digital hardware and then introduction of sophisticated processors. This is followed by the development of Software Defined Radio (SDR) structures and virtual hardware that are under development currently. Software Defined Radio (SDR) system is a useful and adaptable future-proof solution to cover both existing and emerging standards, it provides elements with re-configurability, intelligence and software programmable hardware. Moreover, it has capability of providing global seamless connectivity and solves the interoperability issue.

In this thesis we have implemented different digital modulation techniques on SMT-8036 kit which has a DSP (TMS320C6416) and FPGA (VIRTEX I and VIRTEX II) processors. We are using multiple processors because FPGA give faster output than DSP processor. So if we implement Receiver which has more complexity than Transmitter on FPGA then it will give results faster than DSP. We are doing this by using concept of Threads and Task division for faster execution of program. We are also interested to design SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique in which transmitter changes its modulation scheme after 10 sec and receiver dynamically change the parameters of receiver chain and detects the transmitter’s modulation scheme and demodulate signal accordingly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

ASK	Amplitude Shift Keying
ADCs	Analog to Digital Converters
AM	Amplitude Modulation
AWGN	Additive White Gaussian Noise/
ASICs	Application-Specific Integrated Circuits
AGC	Automatic Gain Control
ASIC	Application Specific Integrated Circuit
ALU	Arithmetic and Logic Unit
BB	Base Band
BER	Bit Error Rate
BP	Band Pass
BASK	Binary Amplitude Shift Keying
BPSK	Binary Phase Shift Keying
CBP	Component Based Programming
CDMA	Code Division Multiple Access
CLBs	Configurable Logic Block
COMINT	Communications intelligence
CPU	Central Processing Unit
DACs	Digital to Analog Converters
DMR	Data Memory and Routing
DSPs	Digital Signal Processors
EDGE	Enhance Datarate for GSM Evolution
ELINT	Electronic Intelligence
ENoB	Effective Number of Bits
EW	Electronic Warfare
FDM	Frequency Division Multiplexing
FET	Field-Effect Transistor
FFT	Fast Fourier Transform
FH/MC DS-CDMA	Frequency Hopping/Multiple Carrier Direct Sequence-Code Division Multiple Access
FIR	Finite Impulse Response
FM	Frequency Modulation
FPGA	Field-Programmable Gate Arrays
FSK	Frequency Shift Keying
GPPs	General Purpose Processors
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HDL	Hardware Description Language
HF	High Frequency

IEEE	Institute of Electrical and Electronic Engineers
IF	Intermediate Frequency
IRM	Image Rejection Mixer
JPEO	Joint Program Executive Office
JTRS	Joint Tactical Radio System
LDPC	Low Density Parity Check
LNA	Low Noise Amplifiers
LO	Local Oscillator
LPF	Low-Pass Filter
LUT	Look-up tables
MAC	Multiply and Accumulate
MILS	Multiple Independent Levels of Security
MIMO	Multiple-Input and Multiple-Output System
MPSK	M-ary Phase Shift Keying
MPSoC	Multiprocessor System-on-Chip
NCO	Numerically Controlled Oscillators
OFDA	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division multiple Access
OOP	Object Oriented Programming
PA	Power Amplifier
PAPR	Peak-to- Average-Power-Ratio
PCI	Peripheral Communication Interface
PFD	Phase-Frequency Detector
QAM	Quardrature Amplitude
QoS	Quality of Service
QPSK	Quardrature Phase Shift keying
RAM	Random access Memory
RF	Radio Frequency
ROM	Read Only Memory
SBR	Software Based Radio
SCA	Software Communication Architecture
SDMA	Space Division Multiple Access
SDR	Software define radio
SHB	Sundance High-speed Bus
SIGINT	Signal Intelligence
SINAD	Signal-to-Noise-and-Distortion
SNR	Signal-to- Noise Ratio
SPI	Serial Peripheral Interface
SR	Software Radio
SSB	Single Side Band

STDC	Software Tunable-Down Converter
ST-PMC	Software Tunable-Power Management Circuitry
STUC	Software Tunable-Up Converter
TDM	Time Division Duplexing
TDMA	Time Division Multiple Access
UMTS	Universal Mobile Telecommunication System
VCO	Voltage Controlled Oscillator
VGA	Variable Gain Amplifier
VHF	Very High Frequency
VLF	Very Low Frequency
VHDL	Very high speed integrated circuit Hardware Descriptive Language
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
Wi-Fi	Wireless Fidelity
Wi-Max	Worldwide inter-operatebility for Microwave Access
8-PSK	8-Phase Shift Keying

CHAPTER 1
INTRODUCTION

Wireless communication devices are composed of three main entities; signaling, physical hardware, and its functionalities. These three main streams, which complement each other, have evolved since the invention of the radio transmission by Guglielmo Marconi. The primitive communications devices had very simple signaling, analog hardware, and limited functionality. In time, each of these entities evolved significantly. Different signaling methods have been invented and used around the world. Furthermore, numerous different Wireless communications systems and standards have been developed around the world without any global plan. Recently hardware technology evolved significantly. Some of the key milestones in this progress are transition from analog hardware to digital hardware and then introduction of sophisticated processors. This is followed by the development of Software Defined Radio (SDR) structures and virtual hardware that are under development currently. SDR is envisioned initially to be a promising solution for interoperability, global seamless connectivity, multi-standard, and multi-mode issues.

With the exponential growth in the ways and means by which people need to communicate data communications, voice communications, video communications, broadcast messaging, command and control communications, emergency response communications, etc. modifying radio devices easily and cost-effectively has become critical. Software defined radio (SDR) technology brings the flexibility, cost efficiency and power to drive communications forward, with wide-reaching benefits realized by service providers and product developers through to end users.

Software Defined Radio (SDR) system is a useful and adaptable future-proof solution to cover both existing and emerging standards, it provides elements with re-configurability, intelligence and software programmable hardware. In addition, the emerging user requirements on reconfigurable mobile systems and networks are paving the way for the introduction of re-configurability in future mobile systems.

Also in parallel, the functionality of wireless devices is increased by SDR and they become more and more sophisticated. For instance, the cellular technology was developed to provide voice communications for mobile users initially. However, current cellular phones have multi-functionalities such as internet access, digital camera, Global Positioning System (GPS), games, personal assistance, and music player. Ever increasing demands from the users and service providers result in continuously increasing Quality of Service (QoS) requirements. This trend requires adding intelligent functionalities to the wireless devices, which introduced cognitive radio technology. Nowadays, these three streams start to merge under the umbrella of cognitive radio technology. It is an emerging technology to realize wireless devices with cognition capabilities such as learning, sensing, awareness, and reasoning. Moreover, it has capability of providing global seamless connectivity and solves the interoperability issue. SDR is a key enabling technology to realize cognitive radios [1].

RADIO

A radio is any kind of device that wirelessly transmits or receives signals in the radio frequency (RF) part of the electromagnetic spectrum to facilitate the transfer of information.

1.1 SOFTWARE DEFINED RADIO (SDR)

A number of definitions can be found to describe Software Defined Radio, also known as Software Radio or SDR. The SDR Forum, working in collaboration with the Institute of Electrical and Electronic Engineers (IEEE) P1900.1 group, has worked to establish a definition of SDR that provides consistency and a clear overview of the technology and its associated benefits.

Simply **Software Defined Radio is defined as**

"**Radio in which some or the entire physical layer functions are software defined**" [2].

Traditional hardware based radio devices limit cross-functionality and can only be modified through physical intervention. This results in higher production costs and minimal flexibility in supporting multiple waveform standards. This problem is solved by software defined radios.

A **software-defined radio** system, or SDR, is a radio communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded computing devices [3]. Basically Software-defined radio (SDR) is based on software defined wireless communication protocols instead of hardwired implementations. In other words, frequency band, air interface protocol and functionality can be upgraded with software download and update instead of a complete hardware replacement. The main advantages of SDR over traditional radio communications are:

- A single SDR device can perform multiple functions simply by changing software modules.
- System updates can be implemented in software to be downloaded via the transmission network. These include updates to both the software application and to any soft configurable hardware.
- Highly configurable signal processing systems can be developed with modifications and upgrades made far simpler to implement than more traditional DSP systems
- More flexible communications protocols can be developed that adapt to their environment transparently to the system user (e.g. searching for and operating in locally available bands) This project aims to create a hardware platform that can facilitate simple narrow-band SDR signal reception applications.

1.2 SDR AND COGNITIVE RADIO RELATIONSHIP

One of the main characteristics of cognitive radio is the adaptability where the radio parameters (including frequency, power, modulation and bandwidth) can be changed depending on the radio environment, user's situation, network condition, and so on. SDR can provide very flexible radio functionality by avoiding the use of application specific fixed analog circuits and components. Therefore, cognitive radio needs to be designed around SDR. In other words, SDR is the core enabling technology for cognitive radio. One of the most popular definitions of cognitive radio supports the above argument clearly:

"A cognitive radio is an SDR that is aware of its environment, internal state, and location, and autonomously adjusts its operations to achieve designated objectives."

Even though many different models are possible, one of the simplest conceptual model that describes the relation between cognitive radio and SDR can be described as shown in Figure 1.1. In this simple model, cognitive radio is wrapped around SDR. This model fits well to the aforementioned definition of cognitive radio, where the combination of cognitive engine, SDR, and the other supporting functionalities (e.g. sensing) results in cognitive radio. Cognitive engine is responsible for optimizing or controlling the SDR based on some input parameters such as that are sensed or learned from the radio environment, user's context, and network condition. Cognitive engine is aware of the radio's hardware resources and capabilities. Hence, it tries to satisfy the radio link requirements of a higher layer application or user with the available resources such as spectrum and power. Compared to hardware radio where the radio can perform only single or very limited amount of radio functionality, SDR is built around software based digital signal processing along with software tunable Radio Frequency (RF) components. Hence, SDR represents a very flexible and generic radio platform which is capable of operating with many different bandwidths over a wide range of frequencies and using many different modulation and waveform formats. So we can say that Software radio provides an ideal platform for cognitive radio. Although cognitive radio algorithms can control a programmable digital radio, software radio and the emerging complexity of third-generation wireless (3G) systems motivate the development of cognitive radio. This includes assisting users in dealing with the exponentially increasing array of wireless access methods.

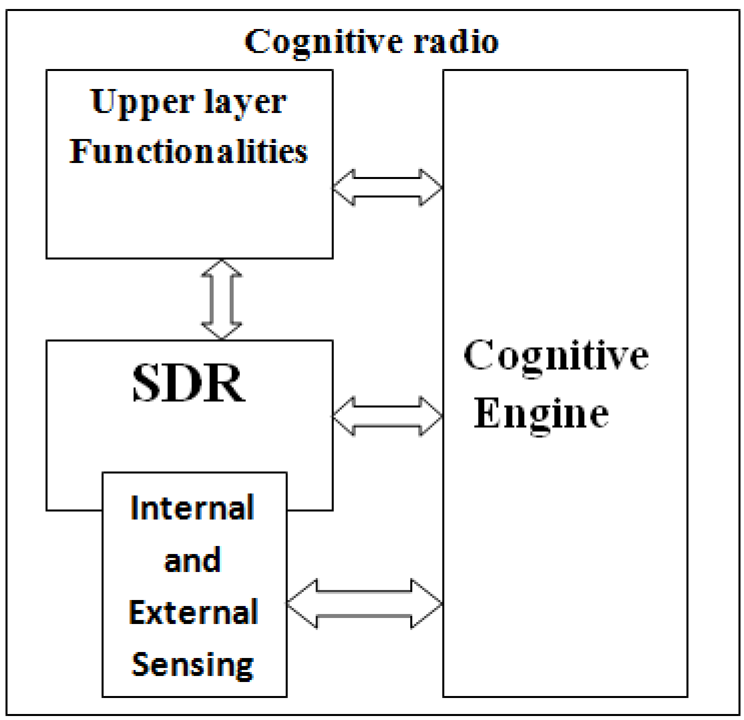


Fig. 1.1: Relationship between SDR and cognitive radio [1].

As a result, SDR can support multiple standards (i.e. GSM, EDGE, WCDMA, CDMA2000, Wi-Fi, and WiMAX) and multiple access technologies such as Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Orthogonal Frequency Division multiple Access (OFDMA), and Space Division Multiple Access (SDMA) in a single device.

1.3 BACKGROUND: A SHORT HISTORY OF SDR

The term "Software Defined Radio" was coined in 1991 by Joseph Mitola, who published the first paper on the topic in 1992 [4]. Though the concept was first proposed in 1991, software-defined radios have their origins in the defense sector since the late 1970s in both the U.S. and Europe (for example, Walter Tuttlebee described a VLF radio that used an ADC and an 8085 microprocessor).

SDR has a broad range of applications, both in civil and in military environments. This study, however, is focused on military electronic warfare (EW) applications such as electronic intelligence (ELINT), signal intelligence (SIGINT), and especially communications intelligence (COMINT) equipment. Many wireless (and wired) communication methods are based on frequency division multiplexing (FDM) encoding. In this method all communication channels of certain application are spread in a frequency band, which is allocated for this purpose by the local communication authority. The channels are allocated in equally, non-overlapping frequency spaces. In order to intercept and process such communication, the RF signal has to be channelized first. Channelization (in this context) is the process of separating a mixture of communication channels into distinct signals, each of single channel.

One of the first public software radio initiatives was a U.S. military project named SpeakEasy [5]. The primary goal of the SpeakEasy project was to use programmable processing to emulate more than 10 existing military radios, operating in frequency bands between 2 and 2000 MHz Further, another design goal was to be able to easily incorporate new coding and modulation standards in the future, so that military communications can keep pace with advances in coding and modulation techniques.

From 1992 to 1995, the goal was to produce a radio for the U.S. Army which could operate from 2 MHz to 2 GHz, and operate with ground force radios (frequency-agile VHF and FM), Air Force radios (VHF AM), Naval Radios (VHF AM and HF SSB teleprinters) and satellites (microwave QAM). Some particular goals were to provide a new signal format to make it compatible with different standard. So solution they made was software defined radio where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded computing devices and this hardware can be change or upgrade just by the use of software. The receiver performance of SDRs is directly related to the dynamic range of the analog-to-digital converters (ADCs) utilized [6]. Radio frequency signals are down converted to the audio frequency band, which is sampled by a high performance audio frequency ADC. First generation SDRs used a PC sound card to provide ADC functionality. The newer software defined radios use embedded high performance ADCs that provide higher dynamic range and are more resistant to noise and RF interference.

1.4 OBJECTIVE OF THESIS

- Study of Software Defined Radio architectures, its benefits in practical usage.
- Study of SMT-8036 kit.
- Study of Digital modulation Techniques.
- Implementation of Digital modulation Techniques on SMT-8036.
- SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique.

1.5 OUTLINE OF THESIS

The thesis is organized as follows:

In chapter 1, Introduction of Software Defined Radios, its benefits and comparison with cognitive radios are discussed.

In chapter 2, Literature survey of Software Defined Radios is presented.

In chapter 3, System architectures and design issues of Software Defined Radios are briefly presented.

In chapter 4, Digital modulation Schemes, proposed Transmitter and Receiver architectures are shown.

In chapter 5, Simulation details, Results and discussions are presented.

.CHAPTER 2

LITERATURE REVIEW

Mansour Ahmadian, Zhila (Jila) Nazari,Nory Nakhaee and Zoran Kostic [7] presents a paper focuses on the major phases present in the development of an SDR system: design; simulation; code generation and verification. The paper will illustrate the use of Model Based Design methodology and tools to integrate the major development phases into one continuous design cycle. Advanced system design concepts including, simulation, code generation, hardware in the loop testing is presented in it.

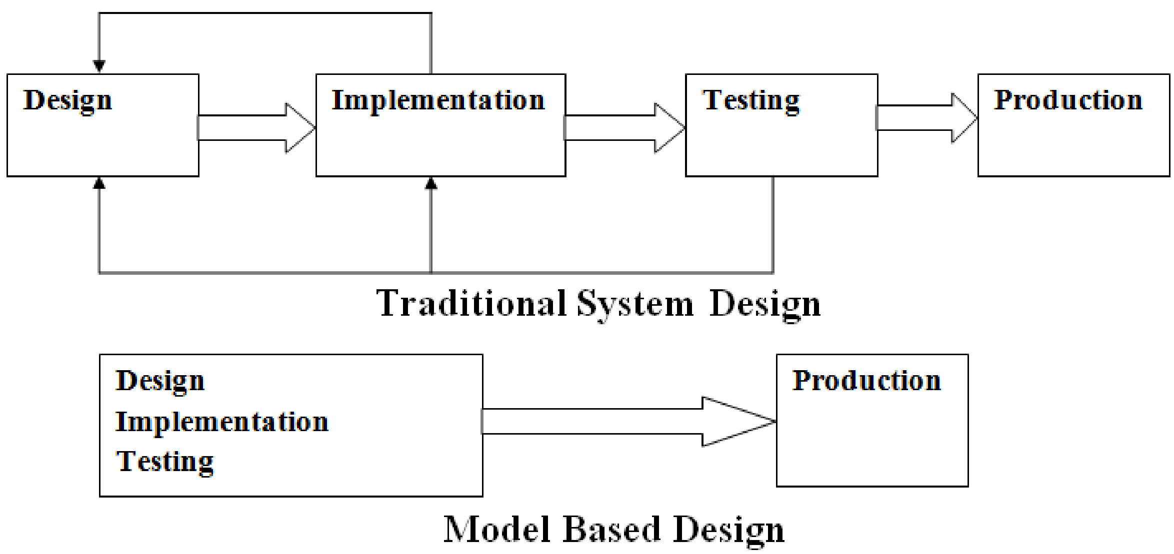


Fig. 2.1: Traditional System Design vs. Model Based Design [7]

Complicated systems can be created by using mathematical models, representing system components and their interactions with their surrounding environment on model based design tool as shown in figure 2.1 for example MATLAB Simulink. After that code is generated by using code-generation technology. Then we can implement that code on hardware provided**.**

A. P. Vinod & Edmund M-K. Lai & Amos Omondi [8] gives a special issue which gives the receiver architectures given in fig 2.2. This issue tells us that the ultimate objective of Software Defined Radio (SDR) is to replace the entire analog signal processing in the wireless transceivers with digital signal processing. This will provide the flexibility, through reconfiguration or reprogramming, to enable the transceivers to work with different air-interfaces standards using a single generic hardware platform. Most of the research in SDR to date has been focused on base stations, which do not have as tight constraints in size and power consumption as handsets. However, in order to realize the promise of integrated services and global roaming capabilities, the design and development of a multi-standard SDR handset with dynamic re-configurability is crucial. The size and power constraints imposed by handsets with limited resources pose a major design challenge. This special issue brings together 11 papers that address various aspects of this challenge from the perspectives of wireless systems, computing architectures, embedded systems and signal processing.

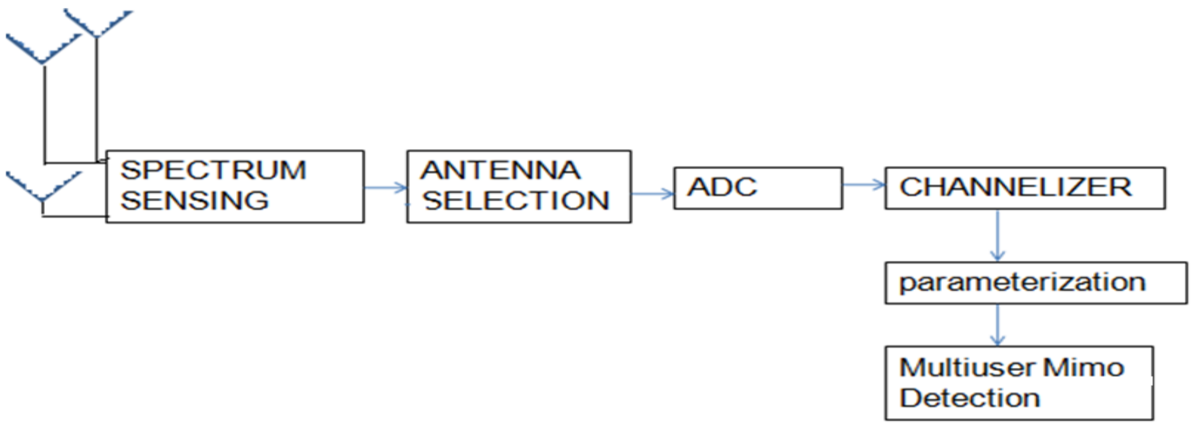


Fig. 2.2: summarized receiver structure [8]

Here in figure 2.2 After ADC, the digitized received signal will need to be separated into signals within the various bands specified by the communication standard. This is performed by the channelizer which has the highest computational requirements of the whole SDR system as shown in figure 2.3

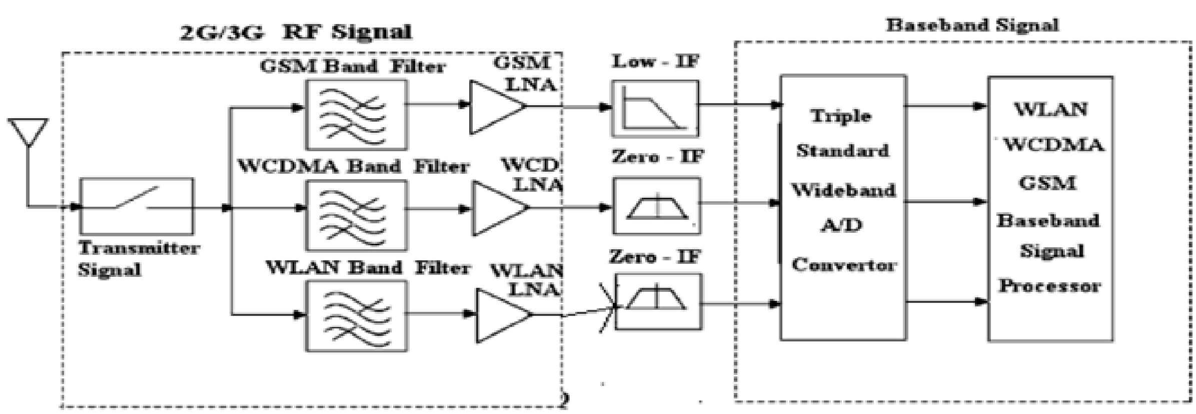


FIG. 2.3: Channelizer [8]

Adrian Tarniceriu, Bogdan lordache, Silvian Spiridon[9] This paper analyses the typical digital modulation techniques used in today's wireless communications. The paper presents the characteristics of the modulation techniques and determines the figure of merit for each particular modulation: Bit Error Rate (BER) vs. Signal-to- Noise Ration (SNR). The analysis emphasizes the importance of such figure of merit in the context of Software Defined Radios (SDR). This paper presented an analysis of the modern modulation techniques. These are used in the latest wireless standards, such as IEEE - 802.16, also known as WiMAX. The SDR systems that can support multiple modulation types can choose with the help of Reconfigurable digital radio part to the appropriate modulation type according to the expected channel quality.

Raúl Dopico-López , José M. Camas-Albar, ed. al. [10] this paper shows that demand in communication characteristics has caused the need of advancements in SDR terminals in order to efficiently deal with high data-rate processing and innovative capabilities. Consequently, different SDR architectures have been proposed by providers and developers in order to palliate communication performance problems, mainly providing lightweight software implementations or multi-core hardware replacements. However, those solutions are neither flexible enough to be integrated with conventional SDR platforms nor scalable enough to have their inner software adapted to forthcoming hardware advancements.

Some multi-core based solutions have been recently adopted to counteract SDR performance problems. However, current applications are not able to fully exploit such parallel architectures and hence obtain considerable improvements in the SDR field.

Therefore, an adaptable multi-core architecture is presented in this paper as shown in figure 2.4 for a suitable solution to cover nowadays SDR performance needs, characterized by supporting the heaviest SDR processing and being able to be fully exploited by means of advanced parallelism techniques.

Physical connections must exist among MPSoC coprocessor and all GPP and DSP type processors allocated in the SDR comprising “parallelization candidate” functions in order to reduce the number of hops to the MPSoC, and hence the slowdown produced by the communications time. The availability of a high number of processors is not a sufficient ingredient to get high parallelism level.

Applications should be adapted to this multiprocessing environment by using advanced parallel programming methods and techniques. So we use libraries for making reliable connections between different processors.

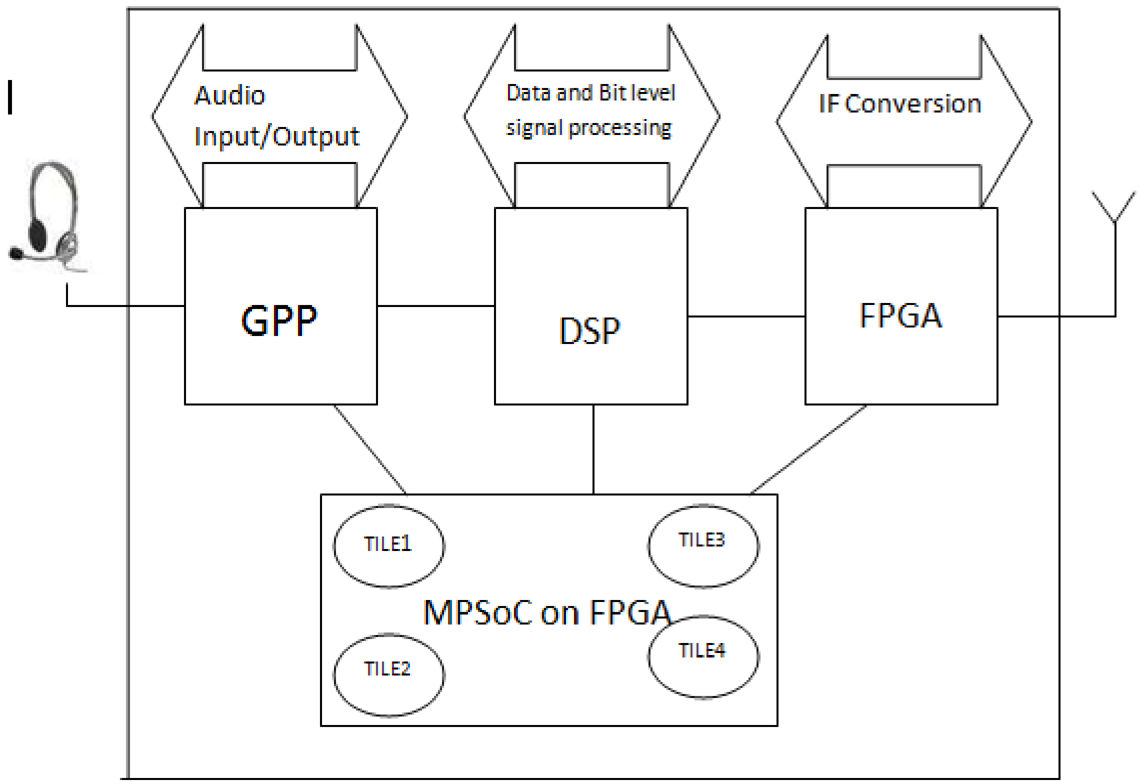


Fig. 2.4: Multi-core architecture for recent SDR [10]

M.Saravanan, Dr.S.Ravi [11] presents that Software defined radio development aims at wideband RF access and software partitioning for plug-and-play type of use. The development is facilitated by progress in silicon capabilities, silicon capabilities, signal processing power of new and future processors and reconfiguration methods (software download, smartcards, etc.). The future wireless and multimedia systems differ from old systems and from each other in terms of access methods, modulation, coding, etc. and they will also provide new type of services. This results in need for many product platforms unless we aren't able to design a common platform that could be programmed to different standards. The software radio is a good solution to this problem since it is reconfigurable to suit different standards at different continents. For this IS-95 CDMA technique is modified as a method of Spread Spectrum Multiple Access technique called Frequency Hopping/Multiple Carrier Direct Sequence-Code Division Multiple Access (FH/MC DS-CDMA). In this work, software defined radio (SDR) based FH/MC DS-CDMA transreceiver is proposed for future generation as well as existing generation Networks, which can carry multimedia applications at higher speed. Multiple carriers can be allotted as per user requirements with high secrecy and multiple encryptions.

Brian A. Dalio and Kevin A. Shelby [12] present that the agile development, implementation, and verification of an OFDM waveform PHY layer on a Software Defined Radio (SDR) development platform. In this paper author worked on comprises a hardware platform based on a massively parallel processing fabric, a unified toolset and design methodology, and a set of reusable, reconfigurable components i: e the HyperX processor architecture. The HyperX processor architecture is a scalable unit-cell-based hardware fabric—the HyperX fabric—which consists of an array of HyperSlic units. Each HyperSlice contains one to four Data Memory and Routing (DMR) units and one Processing Element (PE) unit. The complete hardware architecture is formed by replicating the HyperSlice core unit-cell to create a massively parallel processing system. Surrounding the core is a programmable I/O structure, the HyperIO. In this paper author has used Model based design tools for the implementation of OFDM on the hardware. To demonstrate the complete functionality simultaneously as well as provide a means for characterizing the receiver performance, a combined “loopback” configuration connected via a hardware Additive White Gaussian Noise (AWGN) channel impairment module was implemented.

Feng Ge and Charles W. Bostian [13] This paper addresses two practical issues governing Software Defined Radio (SDR) performance that are often overlooked: RF front end nonlinearity and dynamic computing resource allocation. Current SDR performance still depends heavily on analog radio frequency (RF) technologies. Inter-modulation and other nonlinear effects in these devices make it very challenging to create an RF front-end that is applicable to a variety of signals with widely differing center frequencies, modulation bandwidths, and power levels. Unanticipated inter-modulation products can seriously degrade receiver performance. Digital signal processing in wireless communication is fundamentally a real-time task. Achieving real-time performance in SDRs puts stringent requirements on dynamic computing resource allocation; these requirements may be much higher than those in conventional digital radios.

Digital communication systems are fundamentally real-time because data are transmitted as segments at the PHY layer; to receive all the data, the time to process each data segment is limited since the radio must accept and process each incoming data segment before the next one arrives. Such requirements are stringent in the digital domain when signal processing functions are complicated and there is limited battery life to support sufficient computing resources. It is possible to achieve both real-time performance and low power consumption for a single digital waveform because (1) the RF radio signals can be down-converted to the baseband with a necessary minimum data rate, thus minimizing required computing resources; (2) the required digital signal processing functions can be fully optimized and executed on Application-Specific Integrated Circuits (ASICs).

However, it is quite a different story for SDR. First, moving digital signal processing functions closer to the radio antenna requires a high data rate and more functional components in the software domain; this implies additional computing resources such as computing cycles in signal processing and memory space to hold more signal samples. Second, SDR usually needs to reconfigure itself to support more than one application; this makes it very challenging to highly optimize computing resource allocation for dynamically configured radio functions.

Among the three most popular computing systems for SDR development – field-programmable gate arrays (FPGAs), general purpose processors (GPPs), and digital signal processors (DSPs) – FPGAs are difficult to reconfigure and consume significant power, but they can support real-time performance. GPPs are capable of reconfiguration but are power demanding and also suffer from execution latency. DSPs consume less power and can reconfigure for simple real-time tasks but are not able to support computationally intensive tasks.

We concluded that SDR may require much more CPU resource and battery power than conventional digital radios. Very importantly, SDR performance may deteriorate abruptly if not enough computing resources are available because of the real-time constraint.

David Murotake, Ph.D. and Antonio Martin[14] the authors provided details of wireless network threats discovered during software defined radio (SDR) threat analysis study that exposed a potentially serious flaw in the security architecture of SDR. The reconfigurable radio terminal, and the host to which it is attached, is potentially vulnerable both to exploitation and malicious reconfiguration as a result of "proximity wireless" and Internet based network attacks. The best defensive approach to a blended attack is a "multi-layered" defense, or defense in depth. That is, a combination of methods, implemented in both hardware and software, is implemented throughout the end-to-end communication path if possible.

Author also suggests Virtex II/Pro embedded PowerPC processors employed prototypes of MILS certifiable separation kernel technology to run the security layer applications which included certified IPv4 and IPv6 compliant protocol stacks and firewall applications.

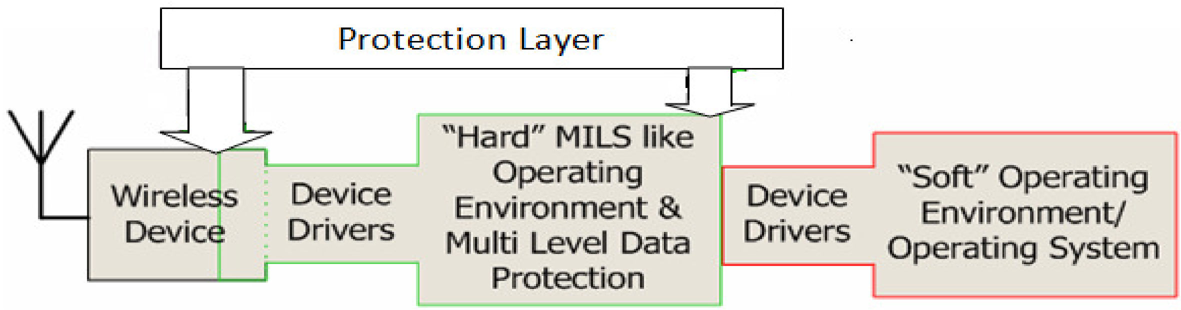


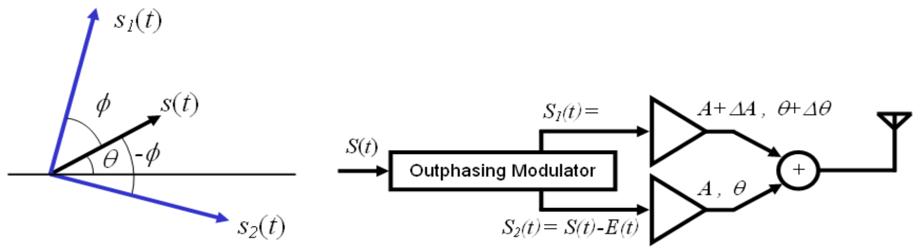
Fig. 2.5: Prototypes of MILS certifiable separation kernel technology [14]

For more secure SDR system we need The ability of the computing platform to be maliciously re-programmed or reconfigured, by-passing encryption, secure browsers and Private Networks through use of root kits/key logger Trojan horses and similar malicious software, may be easy to overlook.

Shervin Moloudi, Koji Takinami, Michael Youssef, Mohyee Mikhemar, Asad Abidi [15]

A software-defined radio (SDR) transmitter needs a power amplifier to support any modulation in any band. There is a simple solution, namely, a Cartesian I-Q up-converter followed by a linear power amplifier, but for complex modulations its power conversion efficiency is often under 10%.

Therefore, the search continues for a more efficient solution. In this paper Author present an SDR PA based on the out-phasing technique. An arbitrary signal, shown in Fig. 2.6 as phasor s(t), is decomposed into two constant amplitude phasors with amplitudes of max(|s(t)|)/2. The out-phasing angle Φ determines the instantaneous amplitude. The constant amplitude phase-modulated vectors are applied to two identical, saturated PAs, whose output powers sum together to reconstruct the intended modulation. Amplitude and phase mismatch in the two paths can cause distortion, but it is easier to match two nominally identical signal paths than to align two heterogenous paths as in polar modulation. The two outputs of the PAs cannot merely be shorted; otherwise each will experience a variable power envelope at its output



$$S_1(t) = S(t) + E(t) \quad S_2(t) = S(t) - E(t)$$

$$E(t) = A(t)\cos(\omega_0 t + \theta(t) + \frac{\pi}{2})\sqrt{\frac{A_m^2}{A^2(t)} - 1}$$

Fig. 2.6: the out phasing principle [15]

which degrades efficiency. PA outputs can be combined with lossless components such as a wideband balun transformer. The core PA is designed for +20dBm maximum power. For lower/higher power the same PA core can be downsized/replicated. The peak efficiency of the PA is 56%.

Sabbir A. Osmany, Frank Herzel and J. Christoph Scheytt [16] In this paper author present that an integrated fractional-N frequency synthesizer providing in-phase signal over 20-28 GHz for software defined radio applications. An integrated voltage controlled oscillator (VCO) with 34 % tuning range and a set of high speed dividers are used to accomplish all the frequencies. By employing a multi-bit, single-loop sigma delta modulator and an 8-bit reference divider, the synthesizer can achieve a frequency step size of less than 10 Hz. This frequency synthesizer is given in figure 2.7. The targeted output frequencies are generated by dividing the VCO frequency. The VCO with a tuning range of 20-28 GHz is fully integrated. The VCO provides an external output and also drives a cascade of divide-by-two circuits. The first divide-by-two gives an output at10-14 GHz and also drives an M-divider. The second divide-by-two gives an output at 5-7 GHz and also drives an 8-bit programmable divider. The M-divider consists of 8 divider ratios whose 8 output bands can cover 0.6-4.6 GHz with sufficient overlap. The output of the Programmable divider is compared with the divided reference signal in a phase-frequency detector (PFD). The PFD controls a charge pump which is loaded with an integrated low-pass filter (LPF).

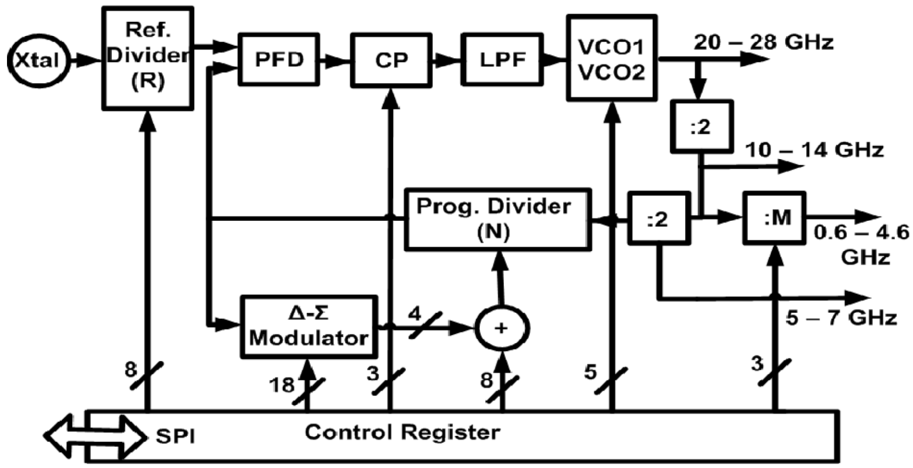


Fig. 2.7: Schematic view of the frequency synthesizer [16].

So we can get any frequency output in specified band.

Fabio Garzia, Waqar Hussain, Roberto Airoldi, Jari Nurmi [17] This paper presents the mapping of SDR applications on a reconfigurable SoC, based on a run-time reconfigurable coarse-grain accelerator called CREMA. CREMA is used to accelerate two kernels used in SDR applications: correlations for synchronization purposes and FFT for the OFDM modulation/demodulation. In both cases we show that the implementation on CREMA is 4X faster than a similar implementation on BUTTER accelerator, and that its resource occupation is reduced by 4.5X. CREMA is a Coarse-grain Reconfigurable array with Mapping Adaptiveness. Its architecture is based on BUTTER and is characterized by a matrix of 4x8 coarsegrain processing elements (PEs). Each PE can process two 32-bit inputs and generates two 32-bit outputs. In the work author describes the utilization of CREMACCINO as an enabling platform for Software Defined Radio (SDR) applications. SDR applications are a popular target for CGRA due to their requirements of high flexibility and computational power.

Antonietta Stango, Neeli R. Prasad [18] The aim of this paper is to address the problem of secure radio software download in SDR devices, The feature to be reconfigurable over the air interface is one of the main advantages of the SDR systems. Downloading software into terminals able to change the radio characteristics, introduces different security perspectives. Otherwise, downloads may be made to devices that could then broadcast on unauthorized bands, or to devices not compatible with the software. The main issue is to secure the protocol of download reconfiguration files. A security policy is a plan or course of action for tackling security issues made by an authority, thus a security policy becomes a set of regulations for state transitions”.

After the download and the installation, an encrypted copy of the radio software can be stored in the SDR device for future use, but before to be reactivated the Policy Decision. This means to verify the policy on the software, coming from the RS, and the policy applied to the device. People proposed a protocol to secure the download of radio reconfiguration files in SDR. They propose a Lightweight version of SSL (LSSL) to secure the connection between the server and the SDR device and a protocol for securing the download of reconfiguration files.

Chris H. Dick, San Jose [19] This paper will describe how many of the functions required in a software radio system can be realized in an FPGA. The FPGA resources of particular interest to the signal processing engineer are configurable dual-port block memories, distributed memory and the multiplier array. The multiplier array is composed of 18x18-bit precision multipliers that can operate in combinatorial mode (140 MHz) or they can be pipelined (1-stage) to support clock frequencies up to 250 MHz. FPGA based signal processors are being employed in a diverse range of signal processing applications for reasons of performance, economics, flexibility and power consumption. FPGAs offer great flexibility, which can, for instance, enable designers to service multiple standards. DSP microprocessors, even with advanced architectural extensions (very long instruction word (VLIW), super-scalar, etc.) do not satisfy the arithmetic or I/O requirements of a modern communication signal processing engine. Advanced field programmable gate array technology offers a solution.

George Nychis, Thibaud Hottelier, et. al. [20] In this paper, author presented a set of techniques that support the implementation of diverse, high-performance MAC protocols on software radios. The work is motivated by the observation that a single one-size fits all MAC protocol cannot meet the demands of increasingly diverse deployments and application loads. Software radios offer flexibility, but their architecture, specifically the delay between the host and the radio frontend, has traditionally been a problem for MAC protocols. We introduce a split-functionally approach, which addresses this problem, and show that it enables the implementation of a set of core MAC functions. An implementation for the USRP and GNU Radio, along with the implementation of an 802.11-like and Bluetooth-like protocol, shows the approach is effective. To our best knowledge, these protocol implementations are the first high-speed, bi-directional MAC implementations for the GNU software radio platform. For future work, we plan to implement a more diverse set of MAC protocols to further evaluate our design and implement the architecture on different SDR platforms to evaluate its generality.

Stéphane LECOMTE, Samuel GUILLOUARD, et. al. **[21]** This paper presents the MOPCOM methodology, primarily developed to enable the efficient design of SDR – Software Defined Radio – equipments. The MOPCOM methodology defines a set of rules to build UML models for embedded systems, from which HDL code is automatically generated by means of MDE – Model driven Engineering – techniques. The UML/MARTE profile is used to describe real-time properties and to perform platform modeling. Developments of RTES include modeling activities, using languages based on either grammars or metamodels, as well as analysis activities such as formal validation or simulation. The Gaspard Methodology based on MDA approach, is intended to provide a framework for developing parallel and distributed applications implemented on SoC MDA techniques coupled with UML are used to perform code generation (VHDL, C/C++ for HLS – High-Level Synthesis – and System C). This process emphasizes application and platform modeling at different levels of abstraction and the allocation of the application models to the platform models. For each level, the selected MARTE stereotypes and the related constraints have been presented. This process has been applied on a SDR case study, for which a C/C++ code and a synthesizable RTL code have been generated. Our future works will consist in integrating the modeling of the partial reconfiguration (PR) of FPGA in the process and to simulate the analysis model of EML with System C language.

Mehrbod Mohajer, Abbas Mohammadi, Abdolali Abdipour [22] In this paper, a new direct conversion receiver based on multi-port structures that are very suitable for software defined radio applications is presented. The proposed receiver is a five-port architecture which provides some advantages in compare to the previous architectures. The microwave simulations have been conducted to show its abilities as a direct demodulator. Meanwhile, the measured results indicate that the introduced five-port can be effectively used as a software defined radio receiver.

Its main advantages are the higher isolation between its two inputs and the less constant parameters in its demodulation equations than the previous multi-port structures. The digital modulation schemes are examined and BER of a specific digital modulation (QPSK) is simulated. These results can be generalized to the MPSK and MQAM modulations. Simulation studies are conducted to evaluate the performance of the proposed demodulator.

Mark Cummings, Todor Cooklev [23] Software Defined Radio (SDR) is one of the most important emerging disruptive technologies that will shape the future of the wireless communication and mobile computing industries. It is the result of a long process of technology evolution. In these early systems, each function was implemented with discrete analog technology. This resulted in relatively large, expensive, high power consuming systems which were difficult to design. The desire to lower cost size and power consumption while making devices easier to manage in the field has driven the technology evolution path we are still on today. As digital technology arrived and entered the beginning of its period of rapid evolution, a pattern developed. This combination of discrete digital logic into a single chip came to be called an Application Specific Integrated Circuit (ASIC). It achieved the cost, size and power consumption advantages inherent in integrated. From a different perspective, in a more general sense the term SDR includes all the enabling technologies that are required to realize the ideal architecture. These technologies include hardware technologies such as wideband antennas, smart antennas, wideband ADCs, powerful DSPs, ASICs, FPGAs, hardware components such as MEMS, etc. The full value of software-defined radios will be achieved when they can fully meet the goals and objectives of all vendors in the wireless value chain.

Sanjay Pithadia [24] A Software-Defined Radio (SDR) system is a radio communication system, a combination of hardware and software technologies, which allows reconfiguration of wireless network architectures. Located at the receiver and the transmitter of a SDR, the ADC and DAC set the performance for the entire radio. A SDR receiver uses an ADC to digitize the analog signal in the receiver at an IF. Once digitized, the signals are filtered, demodulated, and separated into individual channels. SDR transmitter performs coding, modulation, etc. in the digital domain. In the final output, IF stage, DAC is used to convert the signal back to an analog format for transmission.Key specification to consider when selecting the DAC is SNR (determined by quantization and thermal noise).

Lyle Johnson, Mt Vernon [25] In this paper it is presented that it is just a place where interested parties can come to learn and help. The Gnu Public License (GPL) offers a way for software authors to share their work and still protect their rights. The HPSDR project is a loose confederation of enthusiasts trying to create the blocks out of which various systems may be built. An analogy is a bazaar. You wander down the aisles and pick out those things that interest you, and use them to create something else. There is no formal organization but a collection of individuals doing what they love to do, and sharing with others for the benefit of the Amateur community.

S. Weiss, A. Shligersky, S. Abendroth, et. al. **[26]** In this paper, the digital-to-analog and analog-to-digital conversion are performed close to the radio frequency so as to implement modulation, demodulation, channel coding and other required processing tasks in software. SDR test-bed comprises a transmitter and receiver, which is implemented on digital signal processors (DSP) performing the baseband operations. The aim of our testbed is to provide a wireless link between two PCs, which are connected to the baseband DSPs via RS232 connections. We aim to achieve re-configurability of the testbed by using adaptive modulation with modulation levels dependent on channel SNR. If the DSPs can reconfigure the DTP and DRP, a multiband approach is possible on the transmitter side, where the DSP initializes and controls the DTP. The DRP is configured through external software and is not accessible to the baseband DSP in the present form of the SDR.

S. Glass, V. Muthukkumarasamy, M. Portmann [27] In this paper author analyze traffic from public safety communications and uses APCO Project 25 (P25) standard. (P25 is the digital communications standard. It is used as emergency first-responder). It provides low-level access to the actual message traffic using the Wire Shark packet sniffer. A P25 radio system consists of both fixed and mobile equipment. P25 systems encode all voice traffic using the IMBE vocoder. Using an SDR approach enables a single station to simultaneously process many analog and digital signals because processing is partitioned into blocks with well-defined interfaces. The demodulation stage can be easily replaced to allow for reception of different modulation schemes whilst sharing the common code for packet assembly and decoding. Higher bandwidth requirements are met by Cognitive radio by opportunistic use of bands that are underutilized by their primary users. SDR approaches can ensure interoperability and backward-compatibility with existing equipment.

Daughter boards provide for frequency translation, amplification and filtering to enable receive and transmit access to the VHF and UHF bands used for public safety communications. If a radio needs a signal-processing block that isn't present then it can be written (often using an existing block as a starting point) and added to the framework. The receiver produces digital audio as its output and sends the decoded P25 frames to the WireShark network protocol analyzer where they can be analyzed in detail. The WireShark network protocol analyzer is used to recognize, filter and dissect P25 network traffic.

Shinichiro Haruyama [28] defines SOPRANO (Software Programmable and Hardware Reconfigurable Architecture for Network) is a software defined radio platform with multiport-based direct conversion. The main features of SOPRANO are a high-level design methodology for digital circuits, a new mixer-less direct conversion method, and software algorithms for multi-band and multi-mode operation. It is able to receive PSK and QAM signals with two different carrier frequencies at 2.45GHz and 5.25GHz by changing signal processing software.

SDR receiver should be able to receive a wide range of frequencies and bandwidths. In SOPRANO direct conversion is utilized, i.e., RF signals are down converted to baseband signals in just one stage of mixing. Its advantages are wide band reception capability, low power consumption, immunity to image frequency and adjacent channel interferences, relaxing the requirements of the channel selection filter, tolerance to variations of the input power level, making the dynamic range to depend mainly on ADC resolution, wideband operation and digital calibration. Problems such as flicker noise and DC offset can be removed with a spectrum shaping method by line coding.

The algorithms implemented in SOPRANO can be broadly categorized as

(1) Multiport specific (IQ computation and digital calibration) and

(2) Direct-conversion specific (DC offset and IQ imbalance compensation).

Louis Belanger [29] defines the small form factor (SFF) SDR development platform is a hardware/software co-development environment that supplies the full signal chain for a multiprotocol SDR, including RF front-end module, analog-to- digital and digital-to-analog data conversion (ADC and DAC) module and digital-processing module. It can be used as a prototyping and test platform and to optimize the architecture for cost and power. The SDR development platform includes a unique power measurement API. This API measures the loading of the FPGA, DSP and ARM, and reports real-time power data, due to which we can know burst and peak power for a specific data rate, and hence accurately estimate battery life. Because cost and power consumption are two critical factors when designing for portable applications, developers need to have easy access to actual power consumption and processor utilization.

Ajay Kr. Singh, Ankita Taneja, et. al. [30] In this paper it is stated that the SDR allows multiple waveforms to be rapidly ported. It is a communications device whose functionality is defined in software. Defining the architecture also enhances scalability and provides plug-and-play behavior for the components of a radio. SDR controls Transmit and receive frequencies, selects the desired bandwidth, performs the transmit gain control (TGC) and automatic gain control (AGC) functions, sets and monitors the output power level, provides system fault handling and provides the man-machine interface.
The encryption software process configures and monitors security with a recent trend toward performing the entire encryption process in software. Networking aspects of the data traffic are either completely or partially implemented in software. Finally, the man machine interface is completely defined within a software process. The goal is to pass the signal entering the analog to digital converter (ADC) directly to the digital to analog converter (DAC), without actual processing being performed. This process is called talk-through.

Tajinder Manku, Christopher Snyder, et. al. **[31]** designed an architecture that eliminates the need for the image-reject and IF filters present in the heterodyne architecture, while achieving better LO leakage, I/f noise, and second-order intercept performance than the direct conversion architecture. SDR radios can be implemented using one of the following two architectures: a baseband processor, with no RF components, sampling the radio signals or programmable radio architecture with tunable RF components. The first method is not readily realizable. The second method is achievable because tunable RF components such as antennas, down conversion elements, frequency synthesizers, filters and automatic gain controllers are available.

In direct conversion architecture, the RF signal is multiplied by a sinusoid with a frequency equal to the RF carrier frequency. This translates the RF signal directly to baseband and there are no image frequency problems at baseband. To achieve sufficient gain and noise performance, the LO port needs to be driven by a large signal which causes the LO signal to distort and clip and the RF signal is multiplied by the LO harmonic frequencies. When the mixers are transitioning from the “on” and “off’ states, the I/f noise is down converted from the switches to the baseband output. Since the RF signal is also translated to baseband, the I/f noise cannot be filtered and will degrade the signal to noise ratio.

One of the advantages of square waveforms is an increase in conversion gain, an increase in linearity, and a decrease in the amount of l/f noise.

T. Nesimoglu, M. A. Beach, J. R. MacLeod and P. A. Warr [32] worked on receivers and mixers that are used for down-converting the received radio frequency (RF) signal to baseband or to an intermediate frequency (IF) for further processing and in transmitters, for up-converting. The non-linearity of mixers creates undesired effects like harmonic (HD) and inter-modulation distortion (IMD), spreading the spectrum to a wider bandwidth. The HD can be filtered out since it appears at one octave higher frequency than the fundamentals. IMD creates ACI to other nearby channels as well as co-channel interference within the same channel. The SDR receiver frontend receives the wanted channel and a number of nearby signals. Non-linear mixers will down-convert all of these received channels to IF. During this frequency translation process, in-band interference will be added to the wanted channel, making it more difficult for the receiver to correctly detect the information.

Mixer linearization schemes used are:

A. Feed-forward Mixer: The signal used as a reference is only an approximation to the required reference signal. The output of the main mixer, which includes IMD is coupled and added in anti-phase to the output of the secondary mixer, thus cancelling the fundamental signals.

B. Single-Loop Feed-forward Mixer: The secondary mixer is driven with a much higher RF signal than the main mixer to provide a high level of IMD which is an approximation to the required error signal, also providing a high SNR. This error signal is amplitude and phase adjusted before being added to the final IF output for suppressing the IMD.

The distorted output of the down-converting mixer at IF is coupled, amplified, frequency retranslated back to RF and filtered. The reference signal at the receiver frontend is also coupled and added in anti-phase to the frequency retranslated sample of the IF output with amplitude correction. This process cancels the fundamental signals and produces an error signal including only the IMD products. This error signal is then combined with the received RF input signal with correct amplitude and phase relation to pre distort the saturated down-converting mixer.

Mohamed Ratni, Dragan Krupezevic, et. al. **,[33]** worked on Direct conversion receiver (DCR) which appeals broadband because it requires only one major integrated circuit for the RF portion. DCR can be implemented on monolithic integration easier than heterodyne receiver and DCR suffers less from bulky expensive filters needed for image rejection. The Six-Port receiver is used in high frequency microwave analyzers as well as in Direct Conversion Receiver. Where a wide bandwidth is required, an accurate -90 degree phase shifter is needed at a specific frequency.

The output RF signal is fed to the transmit antenna after a RF switch. The purpose of the RF switch is to switch between different standards. This includes filtering, RF amplification, down conversion and base-band amplification. On the other hand, digital signal processing needs to be reconfigurable and reprogrammable by software downloading. A power divider constitutes with a resistive structure and a phase shifter. The DC signal is then proportional to the RF signal amplitude. In order to have a multi-band and multi-channel receiver, a channel selection can he achieved by simple tuning of the local oscillator signal. The direct down converter technique consists of down converting the signal from RF to baseband without any mixer and with no intermediate frequency.

Zhiyu Ru, et. al. **[34]** worked on CMOS radio receiver architectures, based on radio frequency (RF) sampling followed by discrete-time (D-T) signal processing via switched-capacitor circuits which are proposed for SDR receivers. Via symbolic analysis and simulations we analyze the properties of D-T receivers, and show that at least three challenges exist to make a D-T receiver work for SDR:

1) Sampling clock frequency is related to the RF, complicating baseband filter design.

2) A frequency-dependent phase shift is introduced by pseudo quadrature and pseudo-differential sampling.

3) The conversion gain of a charge sampling front-end is strongly frequency dependent.

All the mentioned effects render frequency-dependent radio receiver properties, which are functionally not desired. To correct these variations, extra costs have to be made. In the proposed D-T receivers with RF sampling, the RF signal is down converted to IF at the same time with sampling. The speed of the D-T analog baseband signal processing, e.g. finite-impulse-response (FIR) or infinite impulse response (IIR) filtering, and decimation, is thus connected to the RF sampling speed. This poses unwanted constraints on the filter. For example, for FIR and IIR filtering, the filter cut-off frequency scales with the sample rate, which is now related to the RF. Tuning to another channel thus changes the filter bandwidth! In a SDR application, the baseband-filter bandwidth should preferably be free to choose, either for the channel-select function or for the anti-aliasing function before A/D conversion. To alleviate this problem, more complex D-T analog baseband blocks can be used, e.g. programmable capacitor arrays to tune the bandwidth of the IIR filter.

Howard Patterson, Forrest Scarpitto, Brian Bielick [35] In this paper a technique for applying a series of narrowband load pull measurements of a Gallium Nitride (GaN) transistor to the design of a broadband power amplifier (PA) is presented with potential application to software defined radio (SDR). Load pull uses computer controlled tuners to apply a series of impedances at discrete frequencies to the device under test (DUT) and then calculates performance contours such as gain, efficiency, output power, etc.

A key enabler of these high-performance amplifiers is Gallium Nitride (GaN) transistors, a new generation of semiconductor device that works at high temperature, possesses high breakdown voltage, and is able to produce substantial RF power at high frequency due to small size and low parasitic. Due to the high breakdown and low parasitic, GaN has higher output impedance than competing technologies such as LDMOS. Load pull is a method for determining matching parameters for power devices. It uses tuners to apply a controlled series of impedances to the inputs and outputs of a transistor. The system “pulls” the device, measuring the increase and decrease of performance vs. load state.

CHAPTER 3
SYSTEM ARCHITECTURE AND DESIGN ISSUES

3.1 IDEAL CONCEPT OF COMMUNICATION SYSTEM

Any communication system consists of the three basic components [36] as shown in Fig 3.1

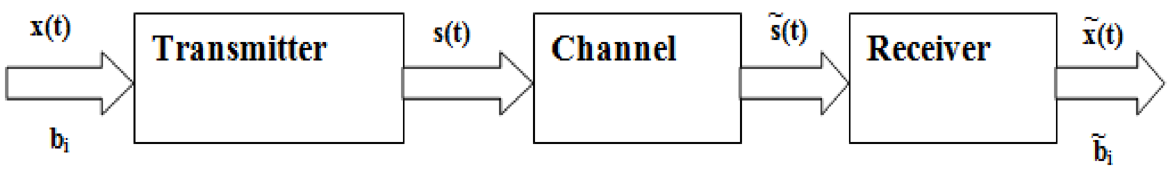


Fig. 3.1: Components of a communication system [36].

- The Transmitter transforms the analog information signal x(t) or the digital data sequence b_i into an analog signal waveform s(t), that is suited for transmission over the channel.
- The Channel is the physical medium that carries the transmit signal s(t) from the transmitter to the receiver. The channel i.e. distorts the transmit signal in various ways.
- The Receiver processes the corrupted signal waveform $\tilde{s}$(t) and transforms it into an analog signal $\tilde{x}$(t) or a digital data sequence $\tilde{b}_i$, which will be hopefully not too different from the transmitted one.

3.2 SDR ARCHITECTURES

During the last 10 years, the idea of Software Defined Radio (SDR) gained momentum pushed by the need of a wireless multi-standard radio terminal capable of operating according to a variety of different mobile communication standards. Starting from the ideal concept of Software Defined Radio (SDR), this chapter investigates a number of architectural issues and trade-offs involved in the design of a fully integrated multi-standard SDR front end. As discussed earlier Software Defined Radio (SDR) is a sophisticated radio that uses software to create high performance, flexible communication devices performing digitally most of the signal processing tasks that analog circuits traditionally handle. It offers the advantage of putting many traditionally inflexible features in modules whose characteristics can be changed while the radio is running. For example, rather than design a single radio to receive only a certain carrier frequency, bandwidth, and modulation as defined by the wireless standard, engineers could program a very flexible digital transceiver to provide receiving capabilities over a wide range of frequencies while the radio operates. Here is the brief explanation of above said.

3.2.1 Ideal SDR Architecture

Ideal SDR architecture consists of three main units, which are reconfigurable digital radio, software tunable radio along with embedded impedance synthesizer, and software tunable antenna systems. This structure is illustrated in Figure 3.2. The main responsibilities of reconfigurable digital radio are performing digital radio functionalities such as different waveform generation, optimization algorithms for software tunable radio and antenna units and controlling of these units.

Software tunable analog front-end system is limited to the components such as RF filters, combiners/ splitters, Power Amplifier (PA), Low Noise Amplifiers (LNA), and data converters. Moreover, this unit has impedance synthesizer subsystem, which is a crucial unit to optimize the performance of software tunable analog radio systems. For instance, impedance synthesizer is used to optimize the performance of software tunable antenna systems for an arbitrary frequency plan specified by cognitive engine. So the basic block diagram of ideal software defined radio Architecture can be constructed as Reconfigurable digital radio system monitors and controls the software tunable radio system continuously or periodically depending on system specifications. A basic relationship between the main units of SDR is described as follows. The cognitive engine sends radio configuration parameters to the reconfigurable digital radio so that it can reconfigure the entire radio according to the parameters. These parameters can be waveform type that needs to be generated (e.g. OFDM, CDMA, UWB), frequency plan (e.g. bandwidth, operating center frequency), and power spectrum specifications. Moreover, cognitive engine can request from reconfigurable digital radio to measure or calculate some parameters from environments such as location information of a particular user. Reconfigurable digital radio configures itself along with software tunable radio components and antenna systems. In order to optimize the performance of these two units, reconfigurable digital radio utilizes the feedback information from software tunable radio, especially from impedance synthesizer. Based on this information, it adjusts the parameters of software tunable radio and antenna units through radio and antenna control signals, respectively. Finally, reconfigurable digital radio acknowledges cognitive engine that the specified configuration is performed.

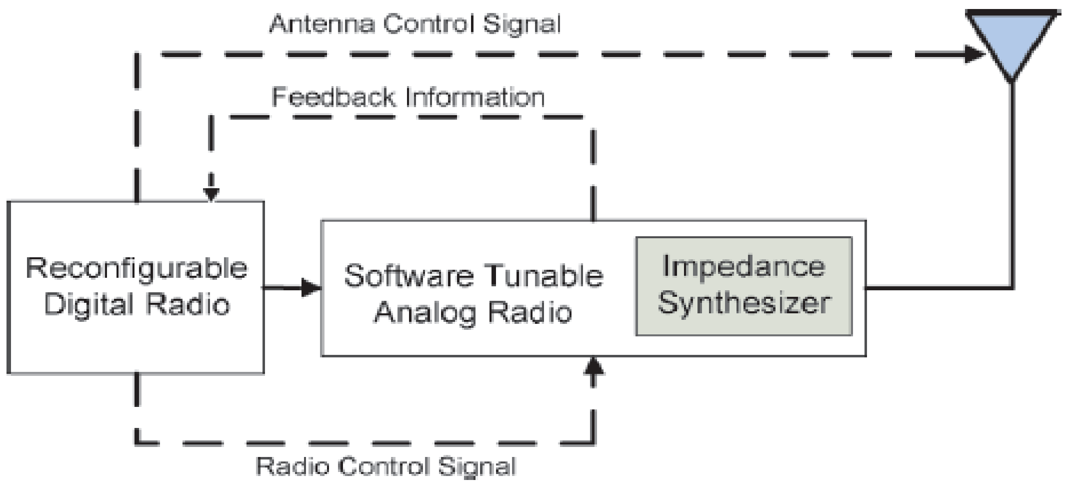


Fig. 3.2: An ideal SDR architecture [1].

3.2.2 Realistic SDR Architecture

Due to the current limitations (size, cost, power, performance, processing time, data converters), ideal SDR architectures are costly.

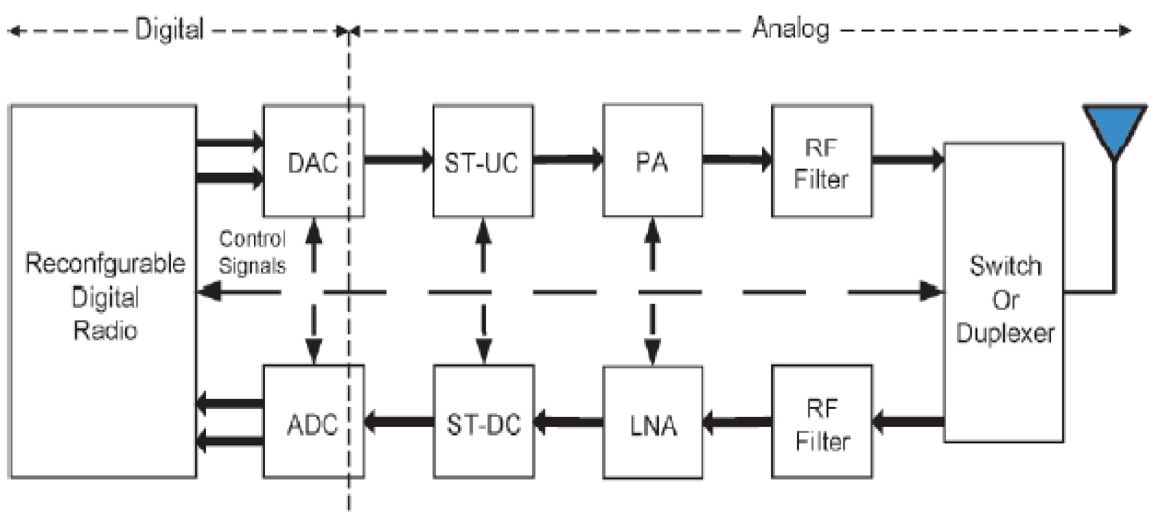


Fig. 3.3: A current practical SDR architecture [1].

There are various practical SDR platforms available in the literature. In the following, we will provide some current and practical SDR architectures. Figure 3.3 shows an example of practical SDR architecture for the Worldwide Interoperability Microwave Access (WiMAX) networks. Reconfigurable digital radio can be implemented using one of the SDR enabling technologies such as Digital Signal Processor (DSP) or Field-Programmable Gate Arrays (FPGAs).This unit mainly generates and demodulates OFDM waveform and controls the radio components. Generated OFDM signal is in the form of digital In-phase (I) and Quadrature (Q) samples. Interpolation, digital filtering, Peak-to- Average-Power-Ratio (PAPR), complexity reduction algorithms and digital Intermediate Frequency (IF) up-conversion are applied to the I/Q signals prior to Digital to- Analog Converter (DAC). Consequently, DAC converts the digital OFDM signal into the corresponding analog waveform. IF signal at the output of DAC is up-converted to a final RF stage using Software Tunable Up-converter (STUC).Transmit power level and local oscillator frequency is then adjusted by reconfigurable digital radio. Furthermore, IF-to- RF up-conversion can be performed in one or multiple stages depending on the performance criterion such as minimum image rejection level. For instance, the first IF signal is up-converted to the second IF and then it is up-converted to the final RF carrier in the case of two-stage up-conversion. It is important to note that the RF signal is amplified using PA according to the power spectral specifications that come from digital radio. A typical PA consists of software tunable attenuators for adaptive transmit power level control. The adaptive transmit power level is an important task for adjusting link quality. The amplified RF signal is filtered out and it is radiated to the air through antenna. Depending on the duplexing method, radio can be classified into two major groups, which are Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). In case of TDD radios, the transmit/receive switch is used, which is controlled by reconfigurable digital radio. On the other hand, for the FDD radios, duplexer filter is used to support simultaneous transmission and reception in different bands. On the receiver side, the received RF signal is filtered out to suppress unwanted out-of-band signals. The filtered RF signal is amplified using LNA. This unit can have digital attenuators to protect the receiver from the signals with high power. Furthermore, it can consist of Variable Gain Amplifier (VGA), which is controlled by the reconfigurable digital radio, for RF automatic gain control. The amplified RF signal is down-converted to a low IF stage using Software Tunable Down-converter (ST-DC), which can be performed either in one or multiple down-conversion stages as well. A typical ST-DC consists of software tunable internal or external AGC. The ST-DC as well as ST-UC can contain externally selectable IF filters to support different bandwidths. Consequently, ADC digitizes the analog IF signal and generates the corresponding digital I/Q samples. Decimation and digital filtering are applied to the samples. In the following steps, digital radio demodulates the received OFDM signal after time and frequency synchronization. Software tunable radio components in this example are implemented using programmable Application Specific Integrated Circuit (ASIC) technology. These components are configured by reconfigurable digital radio through Serial Peripheral Interface (SPI) and independent pins. Using SPI reconfigurable digital radio writes configuration parameters into the registers that are embedded in software tunable radio components. Moreover, software tunable radio components can be powered down by the digital radio to save power. For instance, the components in transmit chain can be switched to power down mode in the TDD radios when radios receive signal and vice versa.

Components of real SDR

- ***Software Tunable Filters***

Ideally, SDR requires IF and RF filters that can be optimized for given filter specifications such as arbitrary center frequency and bandwidth without compromising the performance. The desired performance parameter of such filters in SDR is low insertion loss, excellent out-of-band rejection, and high power handling. These filters have the passband and stopband region as specified by the Reconfigurable Digital radio block. It is important to note here that these filters are desigened by the use of software languages like VHDL etc. So if we want to change the properties we can change that by the use of software only. So it makes our system flexible as compared to traditional filters.

- ***Software Tunable Power Amplifiers***

PAs are one of the essential components of wireless communications systems. PAs are used to amplify the signal before transmission so that the signal can reach to distant receivers with a desired Signal-to-Noise Ratio (SNR). PAs can be divided into two major groups: linear and nonlinear PAs. Linear Pas have the advantage of high linearity which is very important for signals with a wide range of amplitude values. However, they suffer from poor power efficiency, limiting their applications in wireless communications systems. On the other hand, nonlinear PAs can achieve better efficiencies with poor linearity in their responses. The nonlinearity causes several problems like amplitude and phase distortion. Power back-off technique is widely used in current wireless communications systems (e.g. WLAN) to remedy the problems due to wide dynamic signal ranges. However, this technique sacrifices the efficiency and increases the power consumption. On the other hand, baseband linearization techniques are utilized to pre-distort the signal, and hence to compensate the nonlinear effects. Depending on the mode of operation and transmitted waveform characteristics, the linearization algorithms need to be adapted. Digital linearization techniques are often based on a feedback scheme and, therefore, able to react to drifts of the nonlinear PAs. Additionally, PAs generally have embedded digitally controlled VGA to support adaptive coverage systems in SDR. Furthermore, since PAs are the major source of power consumption and heat generation in SDR, PAs can operate in different power level modes such as high and low power modes to save power. By doing this, the temperature of SDR board can be controlled by reconfigurable digital radio. In order to achieve high power transmission, highly efficient and linear multiple PAs can be cascaded.

➢ ***Software Tunable Duplexing Devices***

Duplexing devices are used to manage transmission and reception process in transceivers. As previously mentioned, TDD and FDD are two major duplexing methods. To support each of these methods, different hardware components are required, which is costly. For example, transmit/receive switch is used for TDD radios whereas duplexer filter is employed for FDD radios. It is expected that SDR will support both these two radio structures and change its structure dynamically from one to another. Because, each duplexing method has some advantages, which cognitive engine can benefit from them? For instance, TDD radios are popularly known structure to be used for the unlicensed bands operation due to more relaxed regulations (e.g. output noise level) in these bands compared to the licensed bands. By considering spectrum usage capability of cognitive radio, it can reconfigure SDR as TDD when it operates in the unlicensed bands. When the licensed bands are utilized, cognitive engine can configure SDR as an FDD radio. Using transmit/receive switch and duplexer filter to support these two radio structures is not a practical solution. Furthermore, the current duplexer filters are designed for specific frequency plan. There is a need to develop a single hardware component or find a solution to support various duplexing methods in SDR. One way of achieving this goal is to develop a software tunable component that functions as a transmit/receive switch and duplexer filter.

➢ ***Software Tunable Antenna Systems***

Antennas are essential components of any radio system. An antenna radiation pattern (half power beamwidth) and gain are the two important characteristics that affect the system coverage and performance. Many of the current antennas are designed to operate for a specific frequency range and bandwidth. In SDR, it is important that the antenna would have uniform characteristics over a broad range of frequencies. In other words, ideal SDR requires software tunable antenna that its performance can be optimized for an arbitrary center frequency and bandwidth provided by cognitive engine. Also, the need for multi-band and UWB antennas is increasing. Reconfigurable antennas, smart antennas, and Multiple-Input and Multiple-Output antenna (MIMO) systems are already becoming an integral part of wireless communications systems and they will certainly take a major role in SDR.

The type of antenna that could satisfy the aforementioned SDR antenna requirements is a current research topic. However, shorted patch antenna is one of the approaches for SDR implementation due to its small size, low cost, and omni-directional patterns. In shorted patch antenna based electronically tunable antenna for SDR is proposed. The antenna is controlled by antenna control unit, which is composed of Field-Effect Transistor (FET) switches and FPGAs. These switches are employed to change the electrical length of the antenna. FPGAs control these switches based on the information that is received from the DSP, which can be thought as a cognitive engine in this context. With this structure the cognitive engine can tune the antenna to a specific frequency plan.

➢ ***Software Tunable Impedance Synthesizers***

Software tunable impedance synthesizer plays a key role in the optimization of entire software tunable analog radio for achieving a task given by cognitive engine. In cognitive radios, characteristics of the radio (e.g. RF carrier frequency or bandwidth) are dynamic. It is well known that if one of the fundamental characteristics of analog radio is changed, impedance matching network of the components (at least the affected ones) needs to be optimized. For instance, input impedance of antenna in mobile systems is one of the most frequently changeable radio parameters due to the mobility. A change in the input impedance of antenna results in impedance mismatch between power module and antenna. Such mismatches can deteriorate the performance of radio significantly, even it can be damaged.

For instance, the reflected power (by back radiations) due to the mismatch reduces the radiated power, which deteriorates the efficiency. In such case, radio needs to increase power level resulting in the reduction of battery life. In case of not pumping additional power, QoS is degraded. Note that the reflected power can damage radio components if there is no protection circuit. In order to eliminate the problems and degradation due to the impedance mismatching, software tunable impedance synthesizers are placed between software tunable radio components to perform impedance matching whenever the software tunable analog radio is reconfigured.

The underlying topology of the most software tunable impedance synthesizers are based on generic low-pass pi-network topology as shown in Figure 3.4. This topology consists of variable inductor (L) and capacitors (C1,1, . . . , C1,N1, C2,1, . . . , C2,N2). The more the number of capacitors (N1, N2) employed, the better the tunability is achieved. By varying

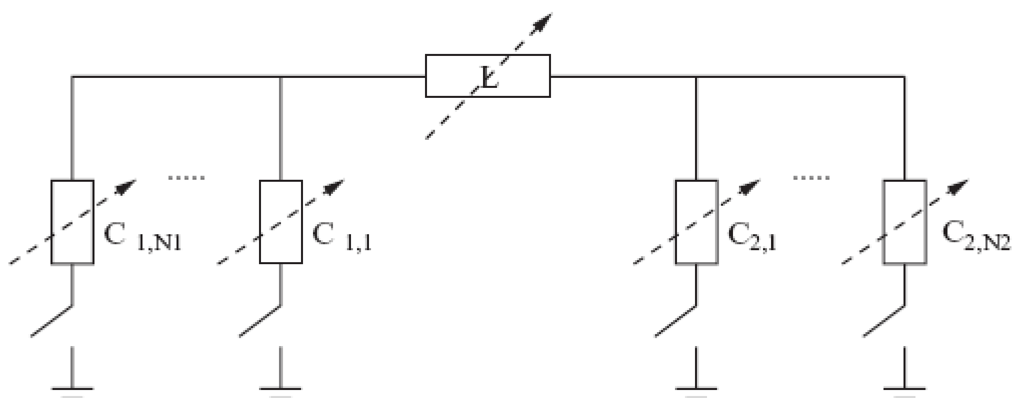


Fig. 3.4: software tuned Impedance synthesizer structure [3].

these values the impedance matching is achieved. Cognitive engine determines the values of these components. This is achieved by providing operating frequency plan and range of impedances to be matched information to reconfigurable digital radio.

- ***Software Tunable Power Management Circuitry***

Power management circuitry is an integral part of complex board design that consists of discrete components. These components can require different voltage supplies in a predefined sequence. Moreover, reconfiguring radio components may require resetting the entire software tunable analog radio platform. A Software Tunable-Power Management Circuitry (ST-PMC) can be incorporated into software tunable analog radio platform to supply an arbitrary voltage to radio components.

- ***Software Tunable Data Converters***

The principle feature of SDR is that the capabilities and operation of radio can be reconfigured efficiently on-the-air, rather than at the time of design. Reconfigurable blocks in SDR systems offer easy changes to radio characteristics such as channel coding/decoding methods, modulation types, multiple access schemes, frequency spreading/despreading algorithms, operating carrier frequencies, and bandwidths. Traditional hardware radio requires hardware changes to modify these fundamental characteristics of radios. Data converters (ADC and DAC) constitute the interface (boundary) between the analog and digital world.

A typical ADC consists of a sampler followed by a quantizer. Sampling rate is one of the important features of ADCs. One of the limitations towards ideal SDR is the sampling rate requirements as the bandwidth of the signal before ADC must be smaller than half of the sampling rate according to the Nyquist theorem. Quantizer converts the discrete samples (with continuous amplitude levels) into bits with a word length. The word length determines the resolution of ADC and also determines the quantization error. In addition to the quantization noise, distortions due to the static and dynamic nonlinearity features of ADCs also affect the performance of ADC. Signal-to-Noise-and-Distortion (SINAD) is the ratio of the root-mean-square (RMS) signal amplitude to the mean value of the root-sum-square of all other spectral components, including harmonics, but excluding DC. SINAD is a good indication of the overall dynamic performance of an ADC, because it includes all components which make up noise and distortion. Due to the distortions, the Effective Number of Bits (ENOB) that specifies the dynamic performance of an ADC at a specific frequency, amplitude, and sampling rate will be different than what is expected from an ideal ADC that only includes the quantization noise. Since cognitive radio requires receiving different waveforms with different operating center frequency and bandwidth, SDR needs to have a capability of reconfiguring ADC dynamically. In order to support such capability, the sampling rate, resolution, SINAD of the ADC needs to be optimized and reconfigured by reconfigurable digital radio. Reconfigurable digital radio requires to have capability of monitoring the above parameters of ADC in order to optimize its performance. This is the case as well for all the aforementioned software tunable analog radio components. For instance, sampling rate can be changed by reconfiguring clock circuit management block, which is responsible to generate and supply any type of clock signal to reconfigurable blocks. Note that the phase noise of clock signal plays an important role on the performance of SDR. Therefore, dynamically generated clock signal needs to be clean and satisfy minimum phase noise level dictated by reconfigurable digital radio.

A generalized DAC structure consists of DAC register, resistor string and followed by output buffer amplifier blocks. Similar to ADC, the main characteristics of DACs are resolution, maximum sampling rate, monotonicity, dynamic range, and phase distortion. Monitoring the parameters related to these characteristics is required to support adaptive transmission mandated by cognitive engine. Based on the feedback information from DAC, its performance can be optimized by reconfigurable digital radio.

➢ ***Software Tunable Up-converters and Down-converters***

Since the functional block diagram of ST-UC and ST-DC are identical, the discussion in this section is based on ST-UC. A basic ST-UC is composed of a mixer and software tunable frequency synthesizer, which is shown in Figure 3.5. SDR requires a mixer that support large input frequency range and it is desirable that mixer to have image rejection capability (Image Rejection Mixer (IRM)). Furthermore, the main features of software tunable frequency synthesizers are ultra-low phase noise and high precision tunability (small step size). The main source of the phase noise is the reference clock. This is generated from reconfigurable clock distributor and it is required to generate reference clock signals with ultra-low phase noise.

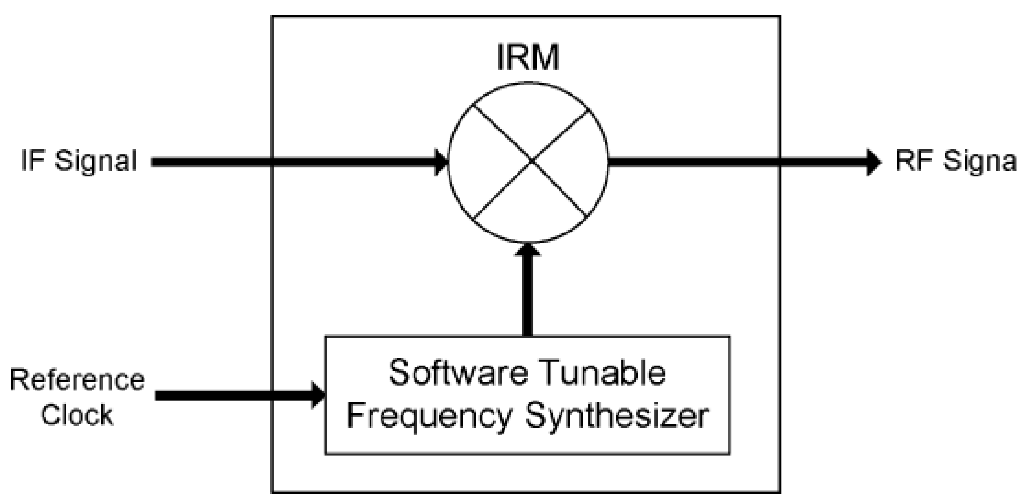


Fig. 3.5: Functional block diagram of ST-UC in SDR [5].

The synthesizer lock detection signal is usually monitored by reconfigurable digital radio to make sure the synthesizer is locked to the desired frequency. In case of employing single stage up-conversion, local oscillator in the synthesizer is required to support large frequency range. Alternatively, multiple-stage up-conversion can be employed in SDR. This approach provides some advantages over the single-stage case, but with the use of additional components. By employing multiple-stage up-conversion in SDR, the requirements of mixer and frequency synthesizers are relaxed.

3.3 ARCHITECTURES OF PROCESSORS USED

Software-defined radio technology is increasingly being used in a wide range of wireless applications. It has been gaining popularity in the wireless communications industry since the advent of powerful DSPs and FPGAs. Next-generation multiprotocol radios, whether re-programmable or not, pose new challenges to developers in term of increasing signal processing requirements, either in terms of raw processing power or algorithmic complexity. Hence, developers are considering the use of digital signal processors (DSP) combined with FPGAs in order to achieve sufficient processing headroom and programmability as well. The Architecture of DSP and FPGA systems is explained in this section.

3.3.1 Architecture of DSP Processor

In a DSP processor, the ALU is tailored towards DSP functions. The main DSP function is the “multiply and accumulate” operation and the important factor is memory bandwidth. The

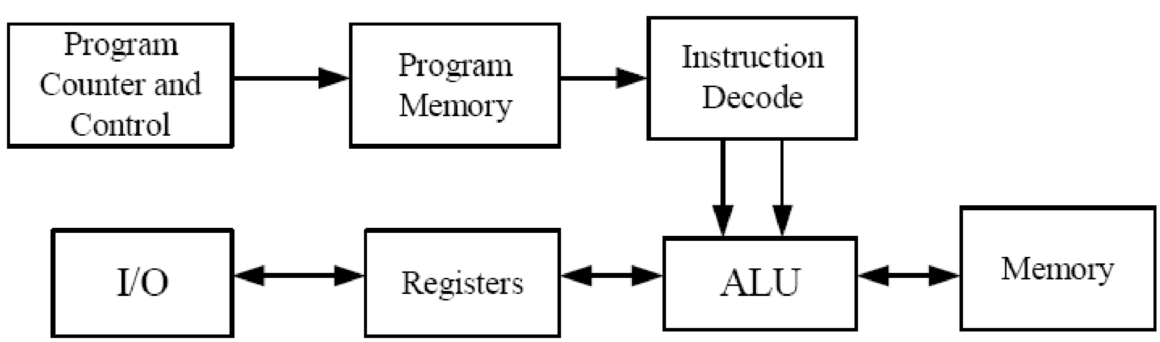


Fig. 3.6: Architecture of a DSP processor

program, which is stored in the program memory, determines the execution sequence and the program memory itself might be an overhead. The DSP structure is generally fixed with a fixed size data width, and some of the algorithms might not need the full data width. The ALU itself has instructions to support lots of different algorithms. Flexibility of a DSP processor lies in the fact that it executes instructions sequentially. But it would be disadvantageous when executing loops and for doing data input and output operations. The algorithm performance in a DSP processor (which in turn determines the maximum sample rate) is determined by the speed (maximum clock rate) supported by ALU, the number of parallel operations supported by the ALU and the number of operations required by the algorithm on a per symbol basis. Architecture of DSP is as shown in figure 3.6 The DSP processor has much flexibility for communication purposes. Some important features of the one of the state-of-art DSP processors, the TMS320C6416 from Texas Instruments (TI) are as given below

- Highest performance fixed point DSP with 2 ns instruction time
- Up to 600 MHz clock rate
- Six ALUs, which support Single 32-bit, Dual 16-bit, or Quad 8-bit arithmetic per clock cycle.
- Two multipliers supporting four 16*16-Bit Multiplies per clock cycle.
- 16 Mb of synchronous DRAM
- 512 Kb of non volatile flash memory.
- Single voltage power supply +5 volts
- Two external memory interfaces

3.3.2 Architecture of FPGA

There is currently a wide range of FPGA products being offered by many semiconductor vendors, including Xilinx, Altera and Atmel. The architectural approaches of different manufacturers are different. However, a generalization can be made in that most FPGAs are organized as an array of logic elements, and a set of Programmable interconnections between the logic elements, the I/O pins, and other resources such as on-chip memory.

The logic elements: Each logic element typically consists of 1 or more n-input RAM based look-up tables where n is between 3 and 6, and several flip-flops.

Configuration: Configuration of the FPGA and interconnection of logic elements, the I/O pins and other resources inside the FPGA is accomplished by downloading a bitstream into a static RAM configuration memory inside the FPGA. This bit stream defines the functionality of each of the logic elements and the internal routing between each of the logic elements. Different applications can be supported inside the FPGA by reconfiguring the FPGA with appropriate bit streams. Structure of FPGA is given in figure 3.7

The Xilinx Virtex-II series: As a specific example, the Xilinx Virtex-II family of FPGAs is considered. The basic logic element in this FPGA is a slice; it consists of

- Two 4-input look-up tables (LUTs)
- Two flip-flops
- Several multiplexers
- Additional silicon to support other applications

Four of these slices form a configurable logic block (CLBs).

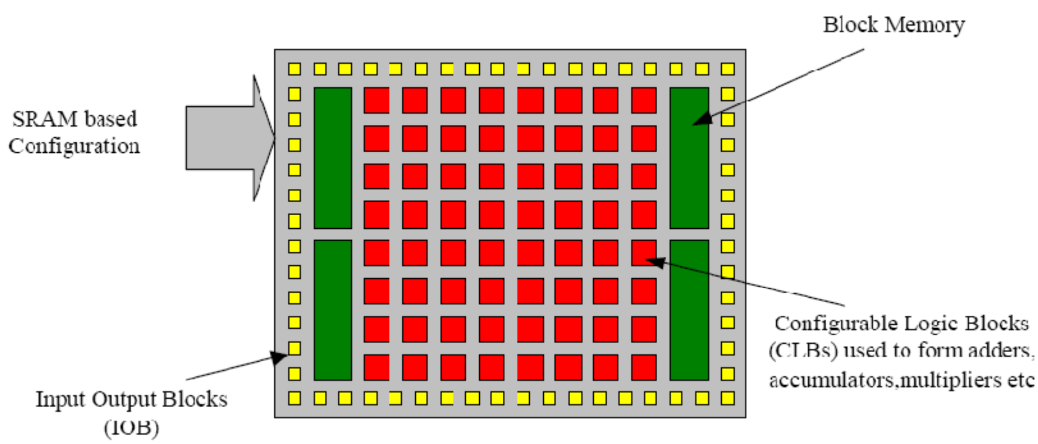


Fig. 3.7: Structure of a FPGA

Some important features of the Xilinx Virtex-II FPGA are

- Largest FPGA with up to 8 million equivalent system gates.
- Up to 3 Mbits of embedded Block RAM memory.
- Up to 168 18*18 multipliers.
- 16 global clocks.
- Up to 1108 user I/O pins.
- 64 to 16224 CLBs

3.4 DESIGN PHILOSOPHIES

Three basic design philosophies are used for programming today: linear programming (LP), OOPs and CBP.

3.4.1 Linear Programming (LP)

LP is a methodology in which the developer follows a linear thought process for the development of the code. The process follows a logical flow, so this type of programming is dominated by conditional flow control (such as "if-then" constructs) and loops. Compartmentalized functionality is maintained in functions, where execution of a function involves swapping out the stack, essentially changing the context of operation, performing the function's work, and returning results to the calling function, which requires an additional stack swap. An analogy of LP is creating a big box for all items on your desktop, such as the phone, keyboard, mouse, screen, headphone, can of soda, and picture of your attractive spouse, with no separation between these items. Accessing any one item's functionality, such as drinking a sip of soda, requires a process to identify the soda can, isolate the soda can from the other interfering items, remove it from the box, sip it, and then place it back into the box and put the other items back where they were. C is the most popular LP language today, with assembly development reserved for a few brave souls who require truly high speed without the overhead incurred by a compiler [37].

3.4.2 Object-Oriented Programming (OOPs)

OOPs is a striking shift from LP. Whereas LP has data structures, essentially variables that contain an arbitrary composition of native types such as float or integer. OOPs extends the data structure concept to describe a whole object. An object is a collection of member variables (such as in a data structure) and functions that can operate on those member variables. From a terminology standpoint, a class is an object's type, and an object is a specific instance of a particular class. There are several rules governing the semantics of classes, but they generally allow the developer to create arbitrary levels of openness (or visibility), different scopes, different contexts, and different implementations for function calls that have the same name. OOPs has several complex dimensions; additional information can be found elsewhere.

The differences inherent in OOPs have dramatic implications for the development of software. Extending the analogy from the previous example, it is now possible to break up every item on your desktop into a separate object. Each object has some properties, such as the temperature of your soda, and each object also has some functions that you can access to perform a task on that object, such as drinking some of your soda. There are several languages today that are OOPs languages. The two most popular ones are Java and C++, although several other languages today are also OOPs languages [37].

3.4.3 Component-Based Programming (CBP)

CBP is a subtle extension of the OOPs concept. In CBP, the concept of an object is constrained; instead of allowing any arbitrary structure for the object, under CBP the basic unit is now a component. This component comprises one or more classes, and is completely defined by its interfaces and its functionality. Again extending the previous example, the contents on the desktop can now be organized into components. A component could be a computer, where the computer component is defined as the collection of the keyboard, mouse, display, and the actual computer case. This particular computer component has two input interfaces, the keyboard and the mouse, and one output interface, the display. In future generations of this component, there could be additional interfaces, such as a set of headphones as an output interface, but the component's legacy interfaces are not affected by this new capability. Using CBP, the nature of the computer is irrelevant to the user as long as the interfaces and functionality remain the same. It is now possible to change individual objects within the component, such as the keyboard, or the whole component altogether, but the user is still able to use the computer component the same as always. The primary goal of CBP is to create stand-alone components that can be easily interchanged between implementations. Note that CBP is a coding style, and there are no mainstream languages that are designed explicitly for CBP. Even though CBP relies on a well-defined set of interfaces and functionality, these aspects are insufficient to guarantee that the code is reusable or portable from platform to platform. The problem arises not from the concept, but from the implementation of the code. To see the problem, it is important now to consider writing the code describing the different aspects of the desktop components that we described before, in this case a computer. Conceptually, we have a component that contains an instance of a display, keyboard, mouse, headphone, and computer. If one were to write software emulating each of these items, not only would the interfaces and actual functional specifications need to be written, but also a wide variety of housekeeping functions, including, for example, notification of failure. If any one piece of the component fails, it needs to inform the other pieces that it failed, and the other pieces need to take appropriate action to prevent further malfunctions. Such a notification is an inherent part of the whole component, and implementing changes in the messaging structure for this notification on any one piece requires the update of all other pieces that are informed of changes in state. These types of somewhat hidden relationships create a significant problem for code reuse and portability because relationships that are sometimes complex need to be verified every time that code is changed [37].

Chapter 4

DIGITAL MODULATION SCHEMES AND ADAPTIVE SDR Tx/Rx

4.1 MODULATION: AN INTRODUCTION

Modulation is a process by which a carrier signal is altered according to instantaneous amplitude of information signal and it is the modulated signal that is transmitted. The receiver then recovers the original signal through a process called demodulation. The carrier frequency, denoted F_c, is the frequency of the carrier signal. The sampling rate, F_s, is the rate at which the message signal is sampled during the simulation.

The frequency of the carrier signal is usually much greater than the highest frequency of the input message signal. The Nyquist sampling theorem requires that the simulation sampling rate F_s be greater than two times the sum of the carrier frequency and the highest frequency of the modulated signal, in order for the demodulator to recover the message correctly. Following are the basic Digital Modulation Techniques

a) Amplitude shift keying
b) Frequency shift keying
c) Phase shift keying

4.2 AMPLITUDE SHIFT KEY (ASK) MODULATION

In this method the amplitude of the carrier assumes one of the two amplitudes dependent on the logic states of the input bit stream. A typical output waveform of an ASK modulator is shown in Fig. 4.1.

A binary amplitude-shift keying (BASK) signal can be defined by

$s(t) = A\, m(t) \cos 2\pi f_c t,\ 0 < t < T$

Where A is a constant, $m(t) = 1$ or 0, f_c is the carrier frequency, and T is the bit duration. The effect of multiplication by the carrier signal $A\cos 2\pi f_c t$ is simply to shift the spectrum of the modulating signal $m(t)$ to f_c. As mentioned in the ASK equation there are two constellation points for the ASK. These constellation points are shown in figure 4.2

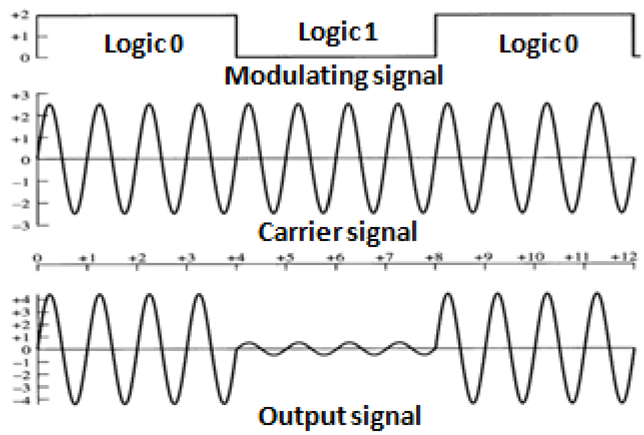


Fig. 4.1: Output waveform of an ASK modulator [38]

ASK has simple modulator circuit it simply multiplies the binary information signal with carrier signal and generates the output signal.

At the receiver, the received signal is multiplied by a local generated carrier and output is passed from a filter having center frequency same as we are using to operate the

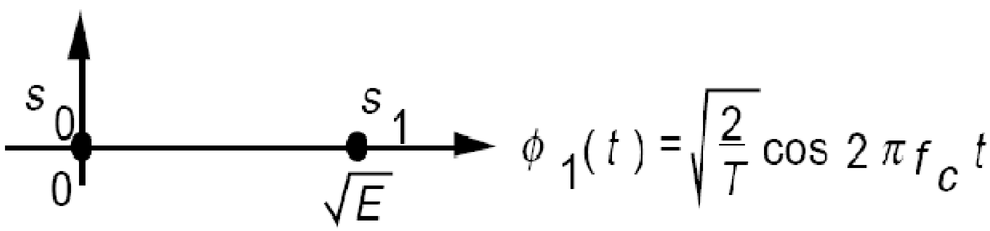


Fig. 4.2: Constellation points for ASK [38]

communication device. At the output we got a DC component based upon it we take decision weather received signal is 0 or 1 [38].

4.3 FREQUENCY SHIFT KEYING

As stated earlier, there are three basic forms of digital modulation techniques: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). FSK is probably the earliest type of digital modulation used in the communication industry. We describe binary FSK signals.

Binary FSK

In its most general form, the binary FSK scheme uses two signals with different Frequencies to represent binary 1 and 0.

$s_1(t) = A\cos(2\pi f_1 t+\Phi_1)$, $kT \leq t \leq (k+1)T$, for 1

$s_2(t) = A\cos(2\pi f_2 t+\Phi_2)$, $kT \leq t \leq (k+1)T$, for 0

where Φ_1 and Φ_2 are initial phases at t = 0, and T is the bit period of the binary data. These two signals are not coherent since Φ_1 and Φ_2 are not the same in general. The waveform is not continuous at bit transitions as shown in figure 4.3.

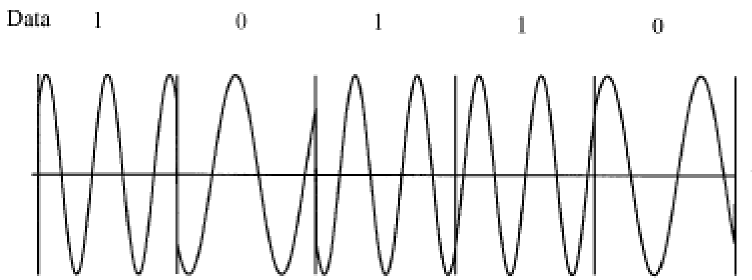


Fig. 4.3: Waveform of non coherent FSK[38]

This form of FSK is therefore called non-coherent or discontinuous FSK. It can be generated by switching the modulator output line between two different oscillators**.** It can be non-coherently demodulated.

The second type of FSK is the coherent one where two signals have the same initial phase Φ at t = 0:

$s_1(t) = A\cos(2\pi f_1 t+\Phi)$, $kT \leq t \leq (k+1)T$, for 1

$s_2(t) = A\cos(2\pi f_2 t+\Phi)$, $kT \leq t \leq (k+1)T$, for 0

The frequency synthesizer generates two frequencies, f_1 and f_2, which are synchronized. These frequencies are called mark and space frequencies. The binary input data controls the multiplexer. The bit timing must be synchronized with the carrier frequencies. If a 1 is present, $s_1(t)$ will pass and if a 0 is present, $s_2(t)$ will pass. Note that $s_1(t)$ and $s_2(t)$ are always there regardless of the input data. So when considering their phase in any bit interval $(kT) \leq t \leq (k+1)T$, the starting point of time is 0, not kT. For coherent demodulation of the coherent FSK signal, the two frequencies are chosen so that the two signals are orthogonal:

$$\int_{kT}^{(k+1)T} s_1(t)s_2(t)dt = 0$$

This requires that $2\pi(f_1 + f_2)T = n\pi$ and $2\pi(f_1 - f_2)T = m\pi$, where n and m are integers. This leads to

$f_1 = (n+m)/4*T$

$f_2 = (n-m)/4*T$

$2\Delta f = f_1-f_2=m/(2*T)$

Thus we conclude that for orthogonality f_1 and f_2 must be integer multiple of 1/(4*T) and their difference must be integer multiples of 1/(2*T). Using Δf we can rewrite f_1 & f_2 as

$f_1 = f_c+\Delta f$

$f_2 = f_c-\Delta f$

$f_c = (f_1+f_2)/2 = n/(2*T)$

Where f_c is the nominal (or apparent) carrier frequency which must be an integer multiple of 1/(2*T) for orthogonality. When the separation is chosen as 1/T, then the phase continuity will be maintained at bit transitions and the FSK is called Sunde 's FSK. It is an important form of FSK. As a matter of fact, if the separation is k/T, where k is an integer, the phase of the coherent FSK signal of equation given for non coherent FSK is always continuous. If the input bit is 1 then output will be $s_1(t)$, if the input bit switches from 1 to 0, the new signal $s_2(t)$ will start at exactly the same amplitude where $s_1(t)$ has ended.

The minimum separation for orthogonality between f_1 and f_2 is 1/(2*T). As we have just seen above, this separation cannot guarantee continuous phase.

Figure 4.4 is an example of Sunde's FSK waveform where bit 1 corresponds to a higher frequency f_1 and bit 0 a lower f_2. Since f_1 and f_2 are multiples of 1/T, the ending phase of the carrier is the same as the starting phase, therefore the waveform has continuous phase at the bit boundaries. Sunde's FSK is a continuous phase FSK.

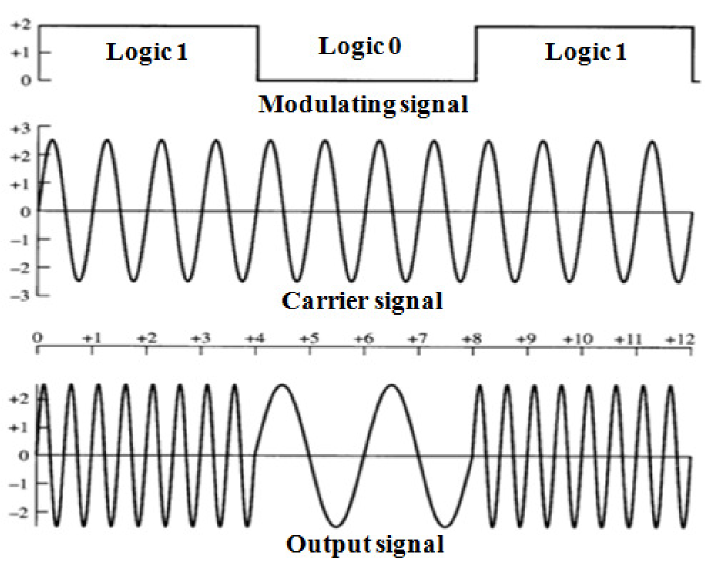


Fig. 4.4: output waveform for sunde's FSK or Binary FSK [38]

4.4. PHASE SHIFT KEYING AND M-ARY PSK

Phase shift keying (PSK) is a large class of digital modulation schemes. PSK is widely used in the communication industry. In this chapter we study each PSK modulation scheme where modulator/demodulator block diagrams are included. This chapter also involves M-ARY PSK. The motivation behind MPSK is to increase the bandwidth efficiency of the PSK modulation schemes. In BPSK, a data bit is represented by a symbol. In MPSK, $n = \log_2 M$ data bits are represented by a symbol, thus the bandwidth efficiency is increased to n times.

There are several different types of M-ARY Phase Shift Key (PSK) modulators. These are:

- Two-phase (2-PSK)
- Four-phase (4-PSK)
- Eight-phase (8-PSK)
- Sixteen-phase (16-PSK)

First we present binary PSK (BPSK). Then we discuss M-ary PSK (MPSK) i:e quadrature PSK (QPSK) and 8-PSK .

4.4.1 Binary Phase Shift Keying (BPSK)

Binary data are represented by two signals with different phases in BPSK. Typically these two phases are 0 and π, the signals are

$s_1(t) = A \cos 2\pi f_c t, \qquad 0 \le t \le T$, for 1

$s_2(t) = -A \cos 2\pi f_c t, \qquad 0 \le t \le T$, for 0

The reason that they are chosen is that they have a correlation coefficient of −1, which leads to the minimum error probability for the same E_b/N_o, as we will see shortly. These two signals have the same frequency and energy.

BPSK signals can be graphically represented by a signal constellation diagram as shown in fig 4.5.A two-dimensional coordinate system with

$\Phi_1(t) = \sqrt{2/T} \cos (2\pi f_c t) \quad , \quad 0 \le t \le T$

And

$\Phi_2(t) = - \sqrt{2/T} \sin (2\pi f_c t) \quad , \quad 0 \le t \le T$

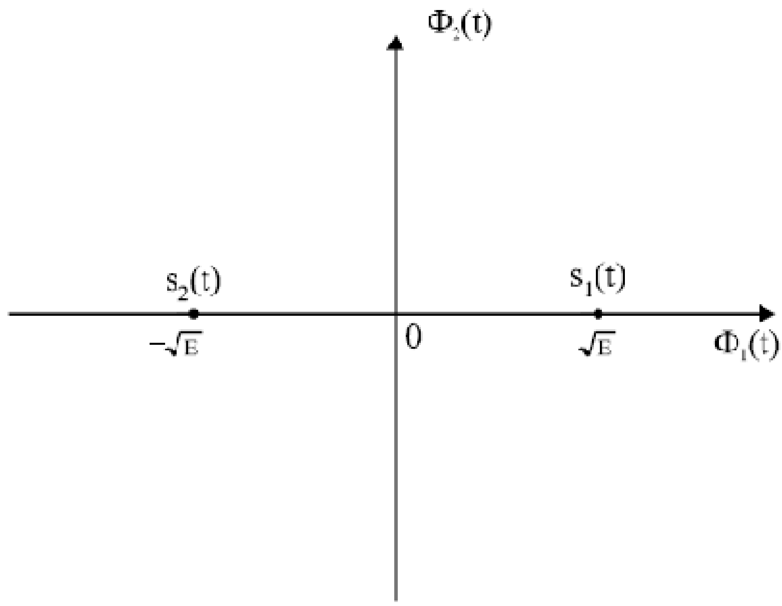


Fig. 4.5: BPSK signal constellation [38]

as its horizontal and vertical axis, respectively. The waveform of a BPSK signal is shown in Figure 4.6. and figure 4.7

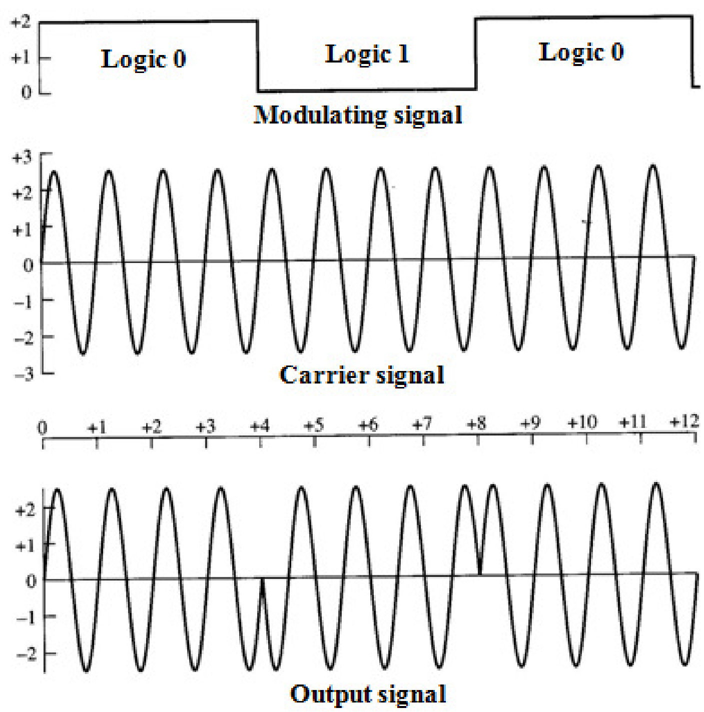


Fig. 4.6: BPSK output waveform if f_c = m R_b = m/T [38]

If the f_c = m R_b = m/T, where m is an integer and R_b is the data bit rate, and the bit timing is synchronous with the carrier, then the initial phase at a bit boundary is either 0 or π (Figure 4.6), corresponding to data bit 1 or 0.

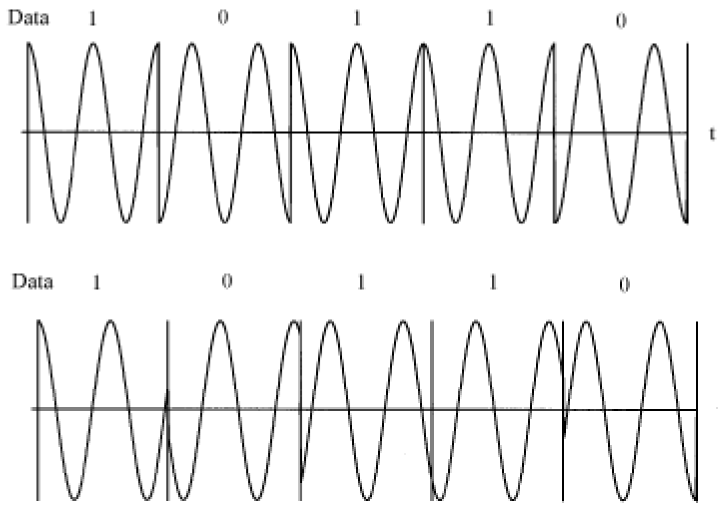


Fig. 4.7: BPSK output waveform if $f_c \neq m\ R_b \neq m/T$ [38]

However, if the f_c is not an integer multiple of R_b, the initial phase at a bit boundary is neither 0 nor π (Figure 4.7)**.** In other words, the modulated signals are not the ones given in BPSK equations. [38]

4.4.2 Quaternary Phase-Shift Keying (QPSK)

Among all MPSK schemes, QPSK is the most often used scheme since it does not suffer from BER degradation while the bandwidth efficiency is increased. Other MPSK schemes increase bandwidth efficiency at the expenses of BER performance.

Quaternary Phase-Shift Keying is sometime called another form of angle modulated, constant amplitude digital modulation .QPSK is an M-ary encoding scheme where n=2 and M=4 (hence the meaning Quaternary meaning 4). With QPSK four output phases are possible for a single carrier frequency. Because there are four output phases there must be four input conditions. Because the digital input to the QPSK modulator is binary signal, to produce four different input conditions the modulator requires more than a single bit to determine the output condition. With two input bits there are four possible combinations i:e 00,01,10,11. Therefore in QPSK the binary input data is combined in group of two bits called dibits. In modulator each dibit code generates one of the four possible output phases ($+45^o$, $+135^o$, -45^o, -135^o). Therefore, for each two bit dibit clocked into the modulator, a single output change occurs,and the rate of change of output is half the rate of change of bits at input.

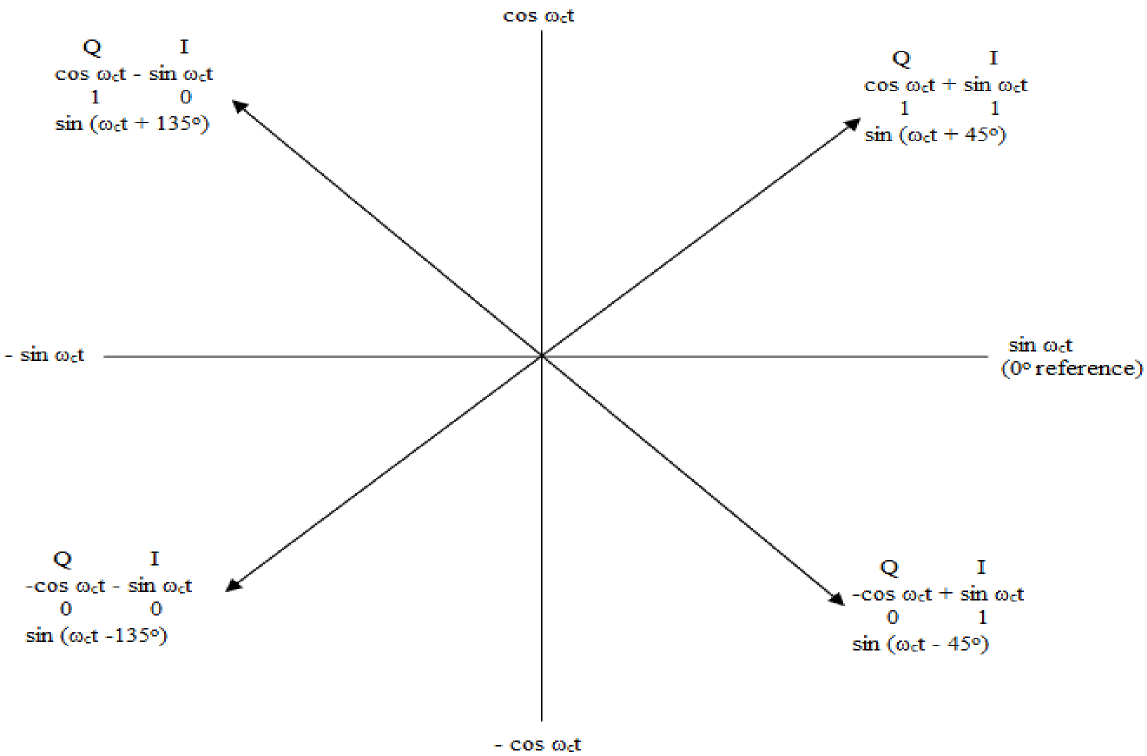


Fig. 4.8: Constellation diagram for QPSK [39]

In Figure 4.8 it can be seen that with QPSK each of the four possible output phasors has exactly the same amplitude. Therefore, the binary information must be encoded entirely in the phase of output signal. This constant amplitude characteristic is most important characteristic of PSK. The angular separation between any two adjacent phasor in QPSK is 90^o. The truth table 4.1 and waveforms supporting above is presented here in fig 4.9

Table 4.1 Binary Input vs QPSK Output phase

Binary Input		QPSK Output phase (degree)
Q	I	
0	0	-135^o
0	1	-45^o
1	0	135^o
1	1	45^o

Thus a QPSK signal can undergo almost a $+45^o$ or -45^o shift in phase during transmission and still retain the correct encoded information when demodulated at the receiver.

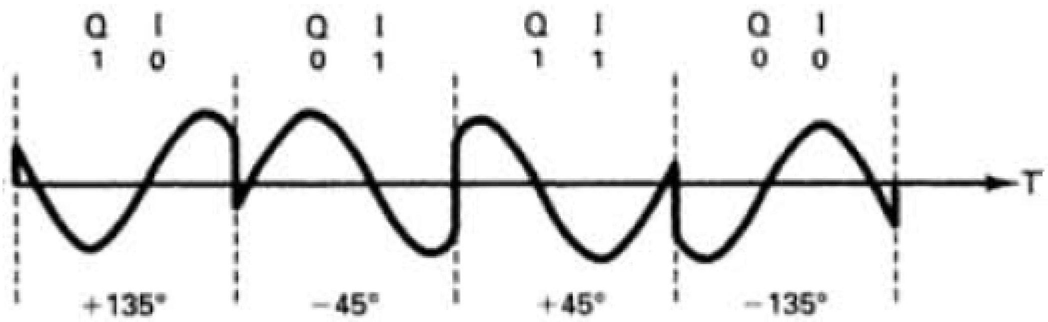


Fig. 4.9: QPSK output waveforms [39]

The received QPSK signal ($-\sin w_c t + \cos w_c t$) is one of the inputs to the product detector. The output is recovered carrier ($\sin w_c t$). The output of I product detector is

$I = (-\sin w_c t + \cos w_c t)(\sin w_c t)$

$= (-\sin w_c t)(\sin w_c t) + (\cos w_c t)(\sin w_c t)$

$= -(\sin w_c t)^2 + (\cos w_c t)(\sin w_c t)$

$= -1/2(1 - \cos 2w_c t) + 1/2 \sin(w_c - w_c)t + 1/2 \sin(w_c + w_c)t$

$= -1/2 + 1/2 \cos 2w_c t + 1/2 \sin 2w_c t + 1/2 \sin 0$

$= -1/2$ V(logic 0)

In Q product detector received signal is multiplied with $\cos w_c t$ so

$Q = (-\sin w_c t + \cos w_c t)(\sin w_c t)$

$= (\cos w_c t)^2 + (\sin w_c t)(\cos w_c t)$

$= 1/2(1 + \cos 2w_c t) - 1/2 \sin(w_c + w_c)t - 1/2 \sin(w_c - w_c)t$

$= 1/2 + 1/2 \cos 2w_c t - 1/2 \sin 2w_c t - 1/2 \sin(0)$

$= 1/2$ V(logic 1)

So demodulated I and Q bits were 0 and 1 respectively [39].

4.4.3 8-PSK

With 8-PSK, Three bits are encoded, forming tribit and producing eight different output phases. 8-PSK, n=3, M=8 and there are 8 different output phases. To encode 8 different phases incoming bits are encoded in group of three, called tribit ($2^3=8$).

From constellation diagram shown in figure 4.10 it is clear that angular separation between any two adjacent phasors is 45^o, half what is with QPSK.

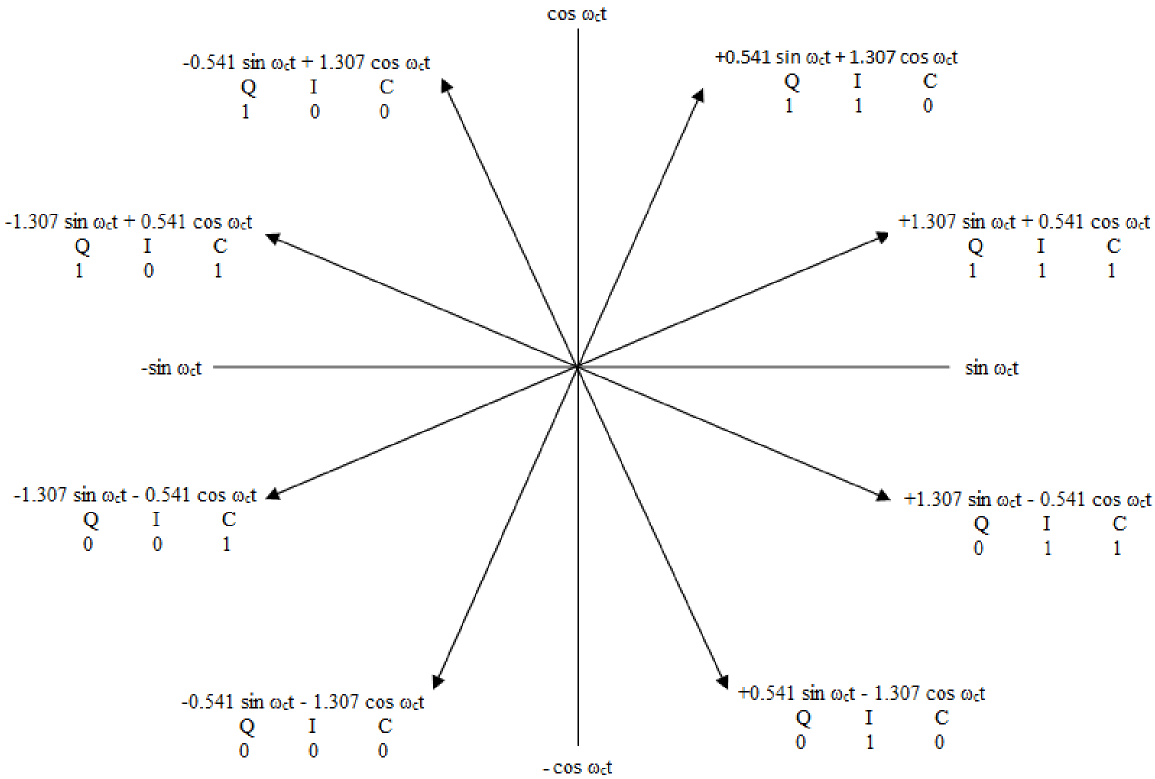


Fig. 4.10: Constellation diagram for 8-PSK [39]

Therefore, an 8-PSK signal can undergo almost 22.5^{o} phase shift in positive or negative direction during transmission and still retain its integrity. 8-PSK output phases for every Tribit input is shown in figure 4.11.

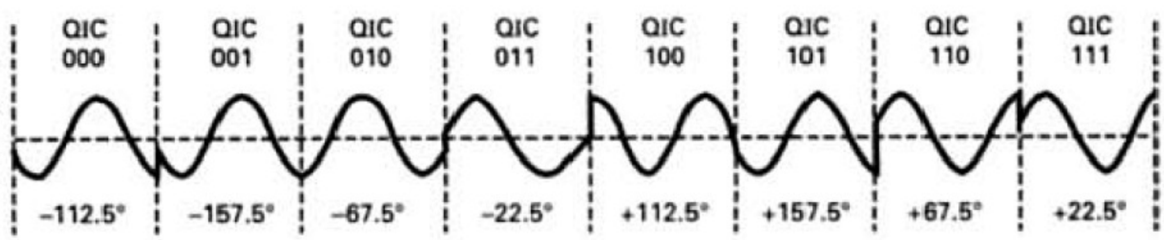


Fig. 4.11: Output phases for 8-PSK [39]

Also each phasor is of equal magnitude. The truth table showing values of output phases is showing in table 4.2. It should also be noted that tribit code between any two adjacent phases changes by only 1 bit. This type of code is gray code or sometimes called maximum distance code. This code is used to reduce the number of transmission errors. If a signal were to undergo a phase shift

Table 4.2 Binary Input vs 8-psk output phase

Binary Input			8-psk output phase
Q	I	C	
0	0	0	-112.5^{o}
0	0	1	-157.5^{o}
0	1	0	-67.5^{o}
0	1	1	-22.5^{o}
1	0	0	112.5^{o}
1	0	1	157.5^{o}
1	1	0	67.5^{o}
1	1	1	22.5^{o}

during transmission, it would most likely to be shifted to an adjacent phasor. Using the gray code results in only a single bit being received in error.

4.5 MULTI-MODULATION SDR TRANSMITTER AND RECEIVER

In this section, we present Software Defined Transmitter and Receiver architectures. In this we have a transmitter which can transmit signal with multiple modulation schemes and a receiver which detect the present modulation scheme and select demodulator accordingly.

4.5.1 Multi-Modulation Software Defined Radio Transmitter.

This section includes a includes a transmitter (which can transmit signal with modulation scheme like (FSK, BPSK, QPSK, 8PSK). Figure 4.12 shows the block diagram of transmitter which is implemented on the DSP processor. One of the input to transmitter is used to select the type of modulation and other is data input. Modulation input is given to the modulation modifier task which selects the modulation scheme and data is given to data processing task where data is channel coded. After processing data is further modulated [39] (FSK, BPSK, QPSK, 8PSK) converted into analog signal. We can change the modulation scheme dynamically by changing the value of modulation input with the help of threads in programming. The other output for the modulation modifier is given to the software control up-convertor which up-converts the modulated analog signal. Up-conversion is done for different modulation schemes in different RF frequency bands. These bands are specified in table 4.3. After baseband to RF conversion signal is passed through RF filter and then transmitted.

Table 4.3 Modulation schemes and their frequency bands.

s. no.	Modulation scheme	Frequency band MHz
1	FSK	<=2400 - 2425
2	BPSK	< 2425-2450
3	QPSK	< 2450-2475
4	8PSK	< 2475-2500

This up-conversion is completed basically in three steps.

1) Conversion of baseband signal to first IF signal in the range 70MHz + 8 MHz

2) Conversion of first IF signal to second IF signal in the range 374MHz + 43 MHz

3) Then finally 2nd IF signal to the RF signal in range 2.4 to 2.5 GHz.

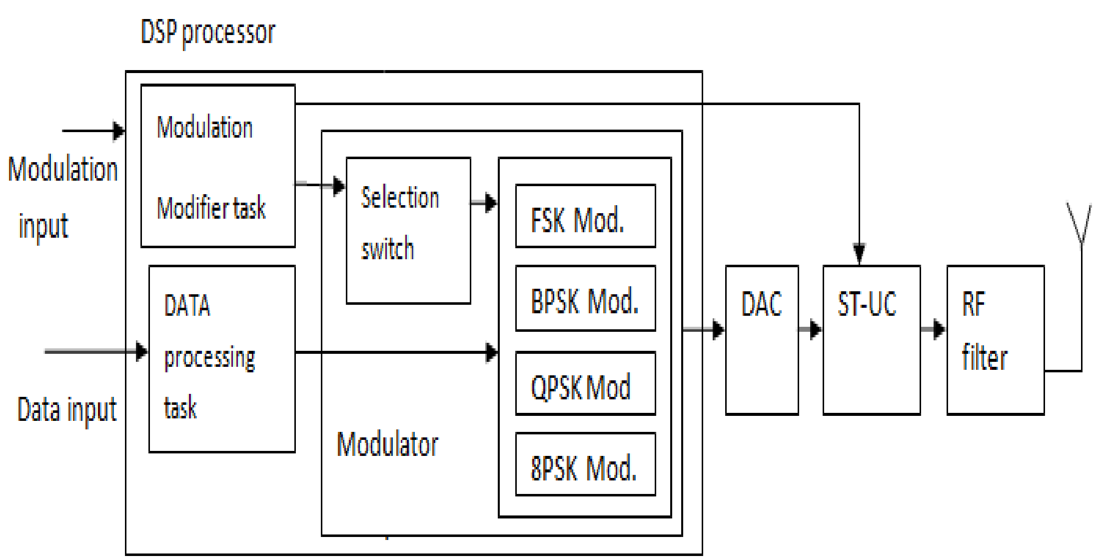


Fig. 4.12: Multi-modulation software defined radio transmitter.

4.5.2 Random number generator based Multi-Modulation SDR Transmitter.

Figure 4.13 shows the block diagram of transmitter which is implemented on the DSP processor. Data input to transmitter is random generated bits, which is fed to data processing task. A random number generation block generates four values (one value after 10 sec) based upon which modulation scheme is selected as shown in table 4.4. There are two output lines of random number generation block (which has same values on it). Both output lines are used at the same time by the use of Threads in programming.

TABLE 4.4. Value of random number generation block and corresponding selected modulation scheme

S. No.	Value at the output of Random number generator	Modulation
1	0	BPSK
2	1	QPSK
3	2	8-PSK
4	3	FSK

One of the output line of Random number generation block is given to selection switch, which selects the modulation scheme and data is given to data processing task where data is channel coded. After processing data is further modulated [39] (FSK, BPSK, QPSK, 8PSK) converted into analog signal.

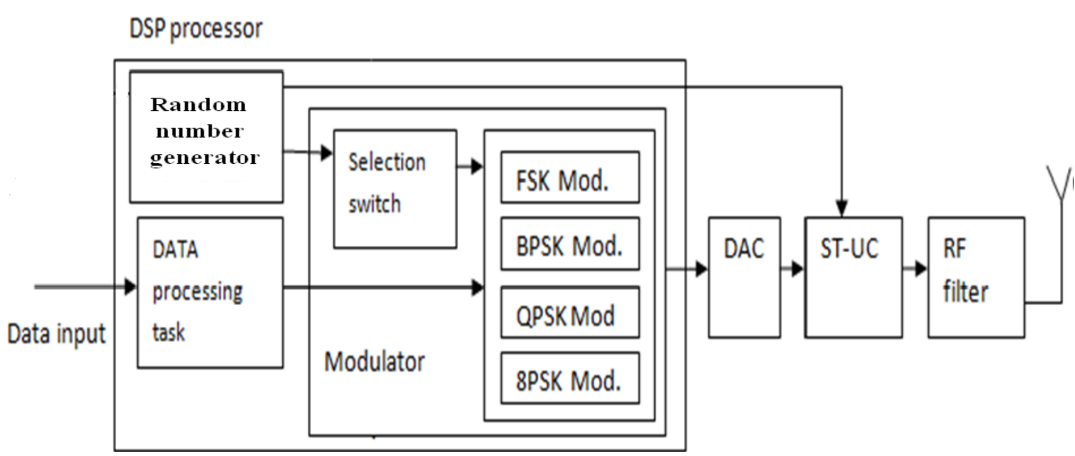


Fig. 4.13: Random number generator based Multi-modulation software defined radio transmitter.

Modulation scheme changes dynamically by the change in the value of random number generation block. The other output line is connected to the software control up-convertor which up-converts the modulated analog signal. Up-conversion is done for different modulation schemes in different RF frequency bands. These bands are specified in table 4.3. After baseband to RF conversion signal is passed through RF filter and then transmitted.

4.5.3 Adaptive SDR Receiver

Block diagram of adaptive software defined radio receiver is shown in figure 4.14. Adaptive SDR Receiver can be used to receive the signal transmitted from transmitters presented in section 4.5.1 and 4.5.2. The signal will be received from tunable receiver antenna, which is fed to the DSP processor through ADC-2 where the frequency of the received signal will be detected. Based on the received signal frequency, DSP processor modifies the parameters of software tunable (ST) band pass filter, ST down-converter, ST amplifier and sampling rate of analog to digital convertor (ADC-1). For the same received signal frequency, DSP processor activates the demodulator circuit (which is implemented on FPGA virtex (II)). The detection of received signal is done through selected demodulator circuit.

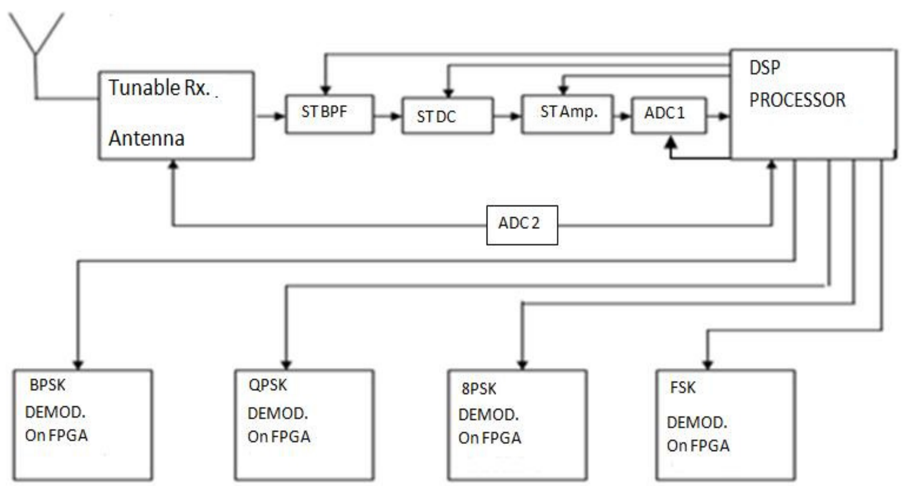


Fig. 4.14: Adaptive software defined radio receiver structure

CHAPTER 5

SIMULATION DETAILS, RESULTS AND DISCUSSIONS

5.1 SMT-8036 (PLATFORM USED)

The SMT-8036 is a development kit for SDR applications consisting of a DSP coupled with an FPGA connected to two ADCs and a DAC. It is a PCI system based on 3 main modules: C64xx-based module (SMT365-8-2) combined with a dual high-speed ADC/DAC module (SMT370), both plugged on a PCI carrier board (SMT310Q) as shown in figure 5.1.

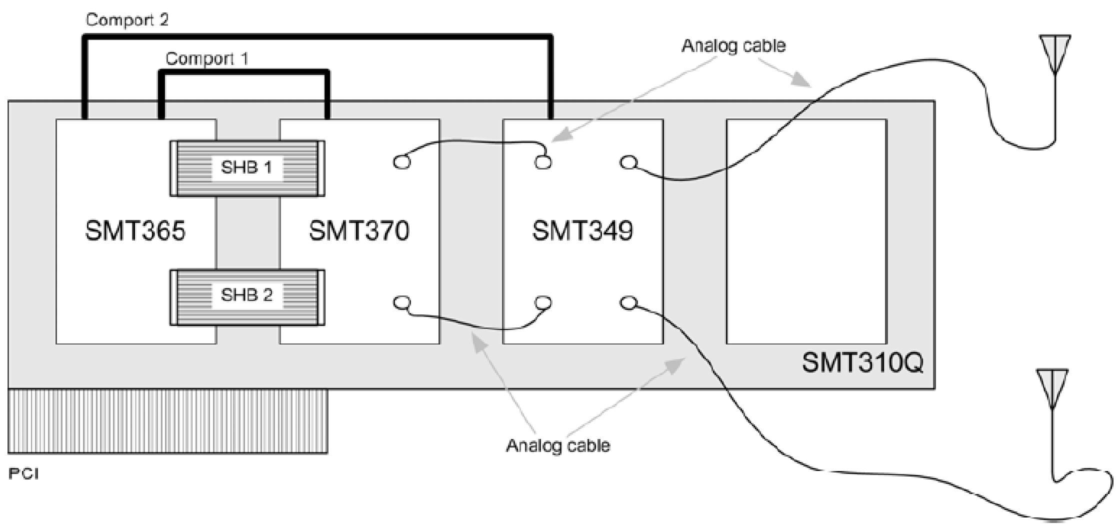


Fig. 5.1: SMT 8036 [40]

- ***SMT365-8-2 characteristics:***
  - ⇒ TMS320C6416 processor running at 600MHz.
  - ⇒A Xilinx Virtex II 2000 FPGA.
  - ⇒ Six 20MB/s communication ports (comm.-ports).
  - ⇒ 4MB of ZBTRAM (133MHz).
  - ⇒ 8MByte Flash ROM for FPGA programming.
  - ⇒ Global expansion connector.
  - ⇒ High bandwidth data I/O via 2 Sundance High-speed Buses (SHB).
- ***SMT370 characteristics:***
  - ⇒ Two 14-bit ADCs.

⇒ Dual 16-bit TxDAC.

⇒ Two Sundance High-speed Bus (SHB) connectors.

⇒ Two 20 Megabytes/s communication ports.

⇒ Low-jitter on-board system clock.

⇒ Xilinx Virtex-II 1000 FPGA.

⇒ User defined pins for external connections.

⇒ Compatible with a wide range of Sundance SHB modules.

⇒ TIM standard compatible.

The SMT8036 is demonstration software for the evaluation of the SMT365-8-2 and SMT370 modules. It can be used for prototyping 3G (3^{rd} generation) systems and high-speed data acquisition system with or without digital processor.

➢ ***SMT 349 characteristics:***

The SMT349 is an IF/RF module offering the following features:

⇒ Support two antennas transmit or receive in the same frequency.

⇒Two Sundance High-speed Bus (SHB) connectors.

⇒ Two comports.

⇒Low jitter system clock.

⇒Xilinx Virtex-II FPGA.

⇒50-Ohm analogue RF/IF inputs and outputs and external clock (option).

⇒User defined pins for external connections.

⇒Compatible with a wide range of Sundance modules via SHB connectors.

The technical specifications for the IF/RF part are:

⇒Input signals are filtered by 1st IF tuned to 70MHz ± 8MHz,

⇒RF output signal is in the 2.4–2.5 GHz ISM band,

⇒RF input signal is in the 2.4–2.5 GHz ISM band,

⇒IF output signal is 70MHz ± 8MHz,

⇒The 70MHz IF is converted to a 2nd IF of 374 MHz

The digital section of SMT349 is based on the SMT370 digital section design. The analogue section of SMT349 includes two RF transceiver modules, operating in the 2.4 GHz ISM band. The RF centre frequency of both transceivers is controlled by one synthesizer which can be set by the FPGA. The FPGA also controls the AGCs (attenuators) and TX/RX switch for each of the two IF/RF sections. The FPGA can communicate with other Sundance modules via the SHB, or via the Comports.

➢ ***Transmitter up-converter module***

Figure 5.2 shows the SDR-transmitter implements a heterodyne conversion concept [41]

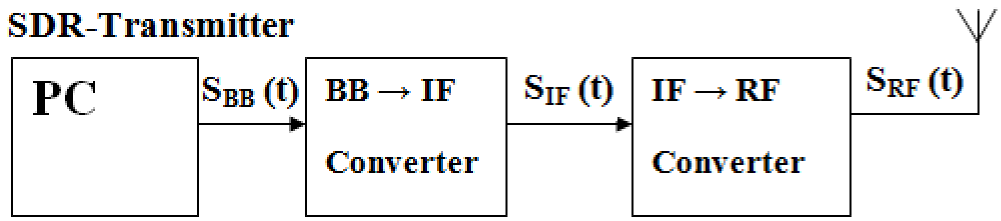


Fig. 5.2: SDR-transmitter, a heterodyne conversion concept [41]

A computer generates a complex BB-signal $S_{BB}(t)$ that will be fed into the first mixing stage. This stage converts the BB-signal up to a IF-signal $S_{IF}(t)$ with the center frequency f_{IF}. At the second mixing stage the IF-signal $S_{IF}(t)$ will be finally up-converted to the desired Radio Frequency (RF)-signal $S_{RF}(t)$ with the desired center frequency f_{RF}. At the end of the transmitter, the RF-signal will be fed to an antenna to be transmitted over the given channel. The up conversion from the 70MHz IF to the 2.4 GHz is done in two stages as shown in figure 5.3:

- The first stage converts the 1st IF to 374 MHz

- The second stage converts to the RF frequency

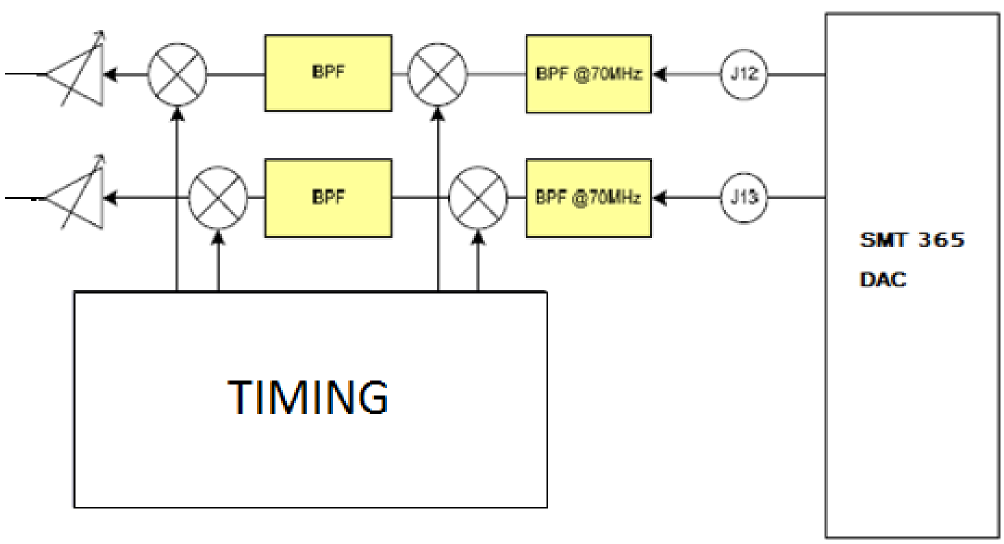


Fig. 5.3: Block diagram showing transmitter up-converter module [42]

➤ ***RX down-converter module***

The SDR-receiver also implements a heterodyne concept [41] as shown in figure 5.4. An antenna receives a BP-signal $S_{RF}(t)$. At the first mixing stage the RF signal will be down-converted to the IF-signal $S_{IF}(t)$ and at the second stage the IF-signal down to the complex BB-signal $S_{BB}(t)$. Further processing of the complex BB-signal will be done on a computer. This diploma thesis will focus only on PAM as BB computing and the BB_IF and IF_BB realization at the transmitter and receiver. The IF_RF and RF_IF realizations already exist.

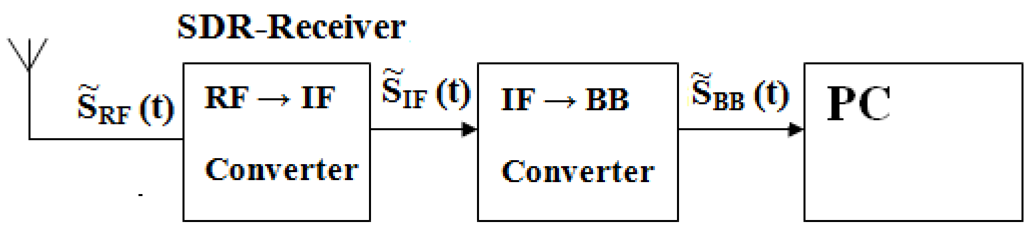


Fig. 5.4: SDR-receiver a heterodyne concept [41].

The down-conversion from the 2.4 GHz to the 70MHz IF is done in two stages as shown in figure 5.5:

- The first stage converts the RF to 374 MHz
- The second stage converts to the 70 MHz frequency

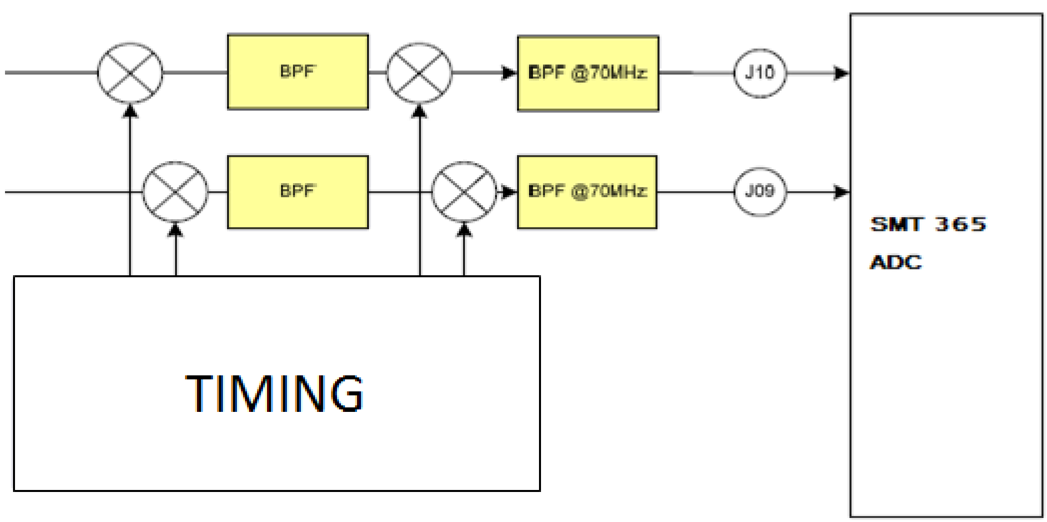


Fig. 5.5: Receiver down-converter [42]

RF sub module: The RF sub module provides the necessary pre-conditioning for amplifying and filtering the RF signal at 2.4-2.5GHz as shown in figure 5.6. The module contains the RF power amplifier and the low noise amplifier. The amplifier provides the necessary RF output power.

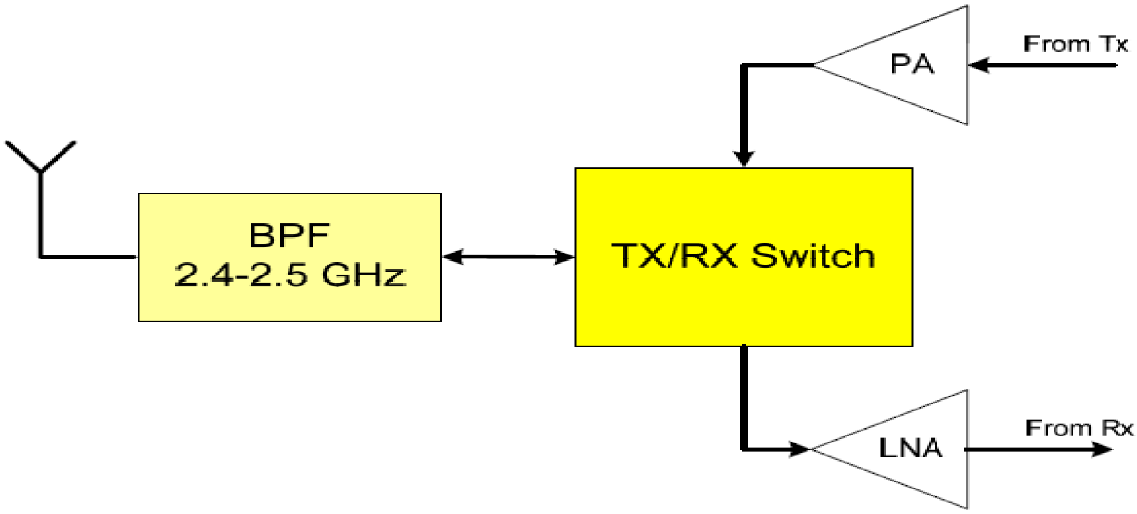


Fig. 5.6: 2.4 GHz RF sub module [42]

5.2 PROGRAMMING APPROACH

Basically if we want to design any hardware component like counters, multiplexers, de-multiplexers then we use VHDL language and implement it on FPGA processor, Where as data we want to use, we compute it in c language on DSP processor.

- ***Programming for parallel processing***

SMT 8036 is a very powerful platform. we can do many independent task by parallel processing on this kit.

Concept of threads

For parallel processing we use concept of threads. Diamond library provides a header file called thread.h. We include this file in our program if we want to include threads in our program.

Example:-

```
.//header files
.
.
.
#include<thread.h>
```

```
.// instructions
.
.

void update_thread(void *unused)// function with thread to calculated
{                           // output signal for bpsk modulation
int freq= 10,n,as;
while(1){
scanf("%d",&as);
if(as>=1)
{
for (n=0; n<WORDS; n++)
                                {
                            sine[n]= 100 * sin((2*n*3.14*freq)/WORDS);
                          printf("\n %f \n", sine[n]);
                                }
}
else
{
for (n=0; n<WORDS; n++)
                                {
                            sine[n]= -100 * sin((2*n*3.14*freq)/WORDS);
                           printf("\n %f \n", sine[n]);
                                }
}
}
}
.//instructions
.
..
Void main()
```

```
{
.//instructions
.
.
THREAD_HANDLE Updater;//object for thread
Updater = Startup(update_thread);//for sending data at output port
                    while(1)
            {
                     chan_out_message(BYTES, sine, &DATA1);
            }
}
```

so in this example we are doing 2 tasks one is to calculate data for input for bpsk and other is to send that information to output channel. we are doing this by the use of threads.

5.3 WORK DONE AND RESULTS

A software radio system is a software version of a typical hardware radio but in this case, many modes of transmission may be implemented without affecting the physical size of the radio. This is achieved by re-programming the software radio in order to implement the required mode of operation. A re-configurable radio is a software radio but the structure and parameters associated with each stage of software radio signal processing chain can be dynamically modified. A re-configurable radio views the software radio as a sequence of signal processing stages. The characteristics and objectives of each of these stages can be modified in response to a change in the wireless communications channel environment or the activity of the radio user.

5.3.1 Pipelining Implementation by Different Tasks

As described earlier also we are working on 3L diamond platform, which compiles our program and burns it on different processors. We divide our problem into a number of independent tasks that can communicate with each other as shown in figure 5.7

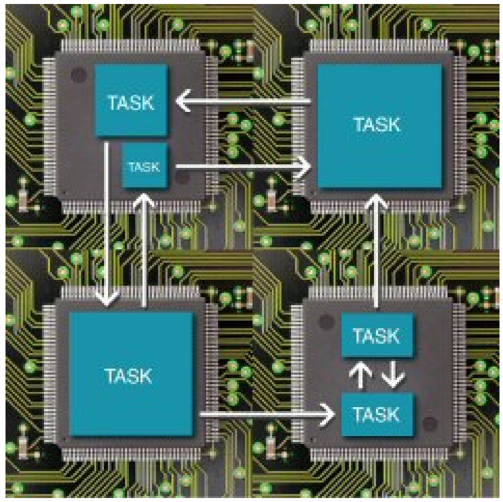


Fig. 5.7: Task division [43]

and then specify the channels that will be used to carry data between our tasks. This is shown here by a simple example which takes -ve and +ve number as input and returns -5 and 5 for +ve and –ve number respectively as described in figure 5.8.

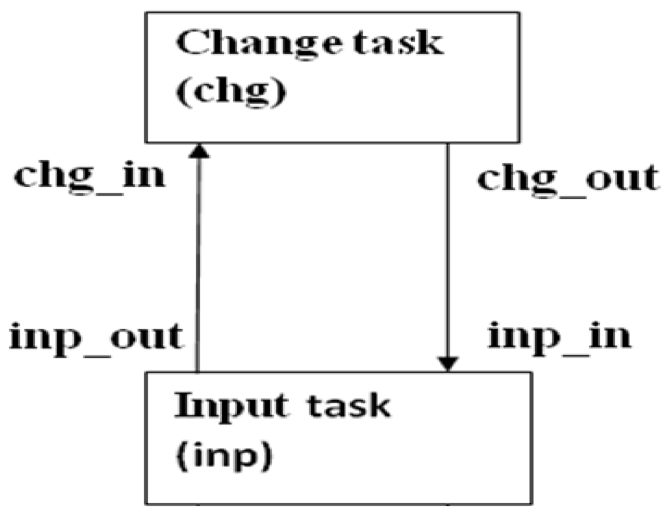


Fig. 5.8: Block diagram showing Communication between task

This is one way to achieve pipelining in DSP because the moment our input task will accept next data value (+ve or –ve number), previous value will be processed to change task this is made more clear as follows.

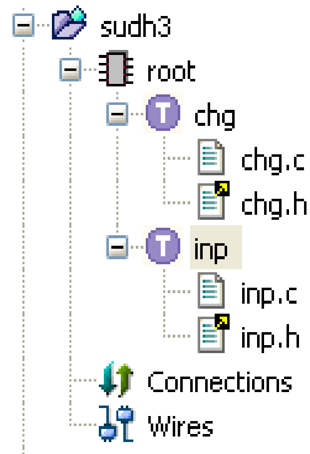


Fig. 5.9: Task names

Figure 5.9 is a screen shot which shows two tasks one is "inp" which takes input and gives us final output other task is "chg" i:e change which contains the logic to change –ve number to -5 and +ve number to 5.

Code for taking "inp" is as shown in figure 5.10.

```
 int a,i;


  printf("this program is for converting +ve value to 5 and -ve value to -5 \n");
  printf("enter 5 values  \n");
  for(i=0;i<5;i++)
  {
scanf("%d",&a);
  chan_out_word(a,&inp_out);
  chan_in_word(&a,&inp_in);


printf("%d   ",a);

  }
```

Fig. 5.10: Code for taking input and displaying output

Code for “chg” is given in figure 5.11**.**

```
 int a;
do
{
 chan_in_word(&a,&chg_in);
  if(a>0)
     {
     a=5;
     }
   else
   {
   a=-5;
   }
 chan_out_word(a,&chg_out);
}while(a != 10) ;
```

Fig. 5.11: Code for changing the values of variables

Now codes for the two tasks are ready. Next step is the connection of these tasks. For this firstly we define input and output ports of tasks and then make their connections as per our requirement. For the above described example connections will be as shown in figure 5.12**.**

Disconnect

Edit

Clear

Name	From	Source	To	Destination	Type
c1	chg	chg_out (0)	inp	inp_in (0)	<Default>
c0	inp	inp_out (0)	chg	chg_in (0)	<Default>

Fig. 5.12: Connection between tasks

Finally output for the code is given in figure 5.13

```
this program is for converting +ve value to 5 and -ve value to -5
enter 5 values
2
3
-5
6
-8
5   5   -5   5   -5
```

Fig. 5.13: Output of the pipelined program

5.3.2 Implementation of ASK on SMT-8036

In software defined radio we make baseband signals with coding (either in C or in VHDL). For ASK we send signal for logic 1 and we do not send signal for logic 0.

Logic for this is as given in Figure 5.14

```
for (i=0; i<PatternCount; i++)
{
    sineWave[i] = data == 1 ? sin(2 * PI * N * ( ( (float)i ) / PatternCount ) ) :0;

}
```

Fig. 5.14: Logic for ASK

For logic 1 output is shown in fig 5.15

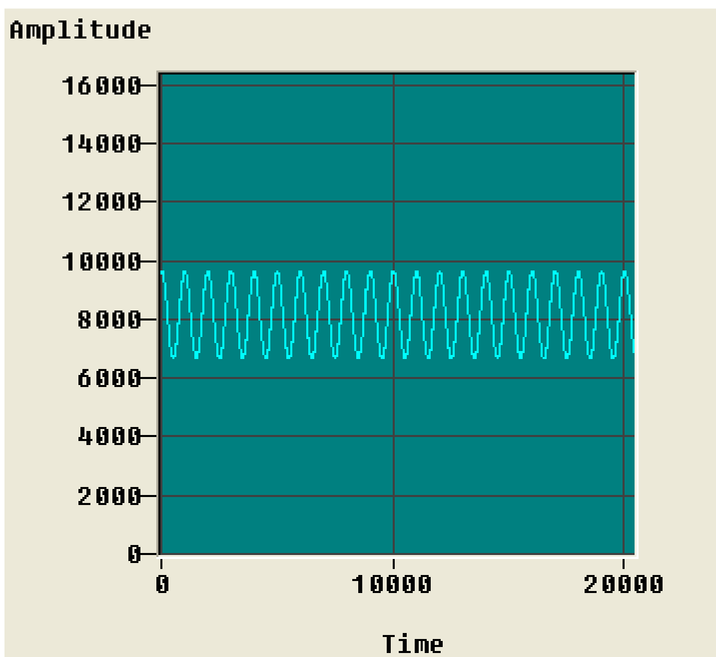


Fig. 5.15: Output waveform for logic 1

For logic 0 output is as shown in figure 5.16

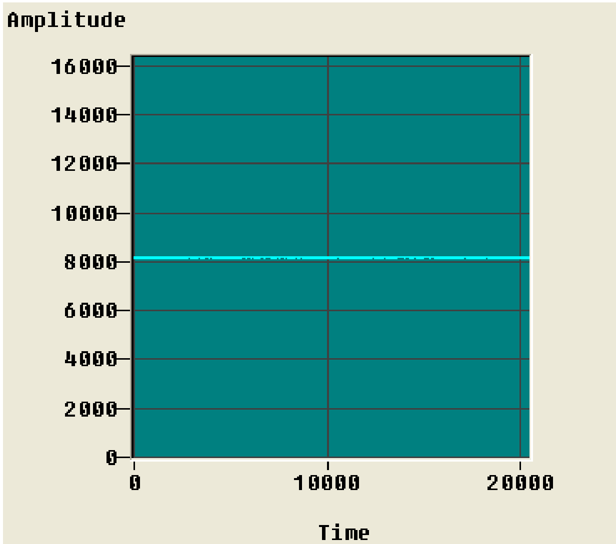


Fig. 5.16: Output waveform for logic 0

And at receiver logic used is as shown in fig 5.17

```
int DetectASK()
{
    int n;
    float max = 0;
    float min = 0;

    for (n=0; n<SAMPLES; n++)
    {
        if(max < stream2[n])
        {
            max = stream2[n];
        }

        if(stream2[n]  != 0)
        {
            if( min == 0 || min > stream2[n])
            {
                min = stream2[n];
            }
        }
    }

    if( (max-min) < 100)
    {
        return 0;
    }
    else
    {
        return 1;
    }
}
```

Fig. 5.17: Logic for detecting ASK

5.3.3 Implementation of BPSK on SMT-8036

In BPSK we send + sin(XXXX) for Logic1 and –sin(XXXX) for logic 0 this is shown in Fig 5.18

```
for (i=0; i<PatternCount; i++)
{

    sineWave[i] = data == 1 ?sin(2 * PI * N * ( ( (float)i ) / PatternCount ) ) :cos(2 * PI * N * ( ( (float)i ) / PatternCount ) );

}
```

Fig. 5.18: Logic for BPSK

For logic 1 output will be as shown in figure 5.19

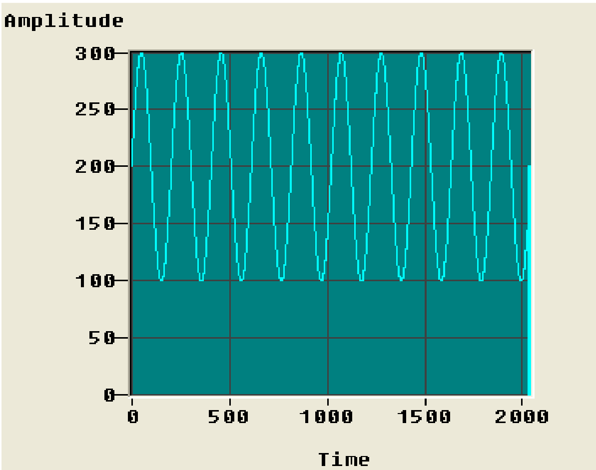


Fig. 5.19: Output waveform for logic 1

For logic 0 output will be figure 5.20

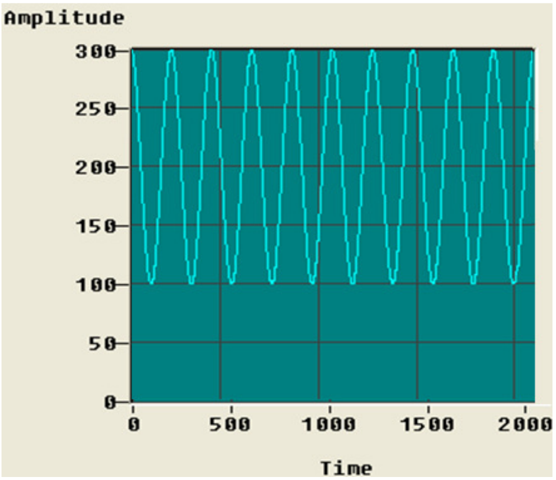


Fig. 5.20: Output waveform for logic 0

Values detected after digitalizing the signal is as shown in figure 5.21

```
Data = 1

  8190.000000  8191.000000  8193.000000  8197.000000  8195.000000  8189.000000  8191.000000  8194.000000  8197.000000  819

Data = 0

  8187.000000  8182.000000  8184.000000  8186.000000  8187.000000  8188.000000  8183.000000  8185.000000  8187.000000  818

Data = 1

  8192.000000  8195.000000  8195.000000  8194.000000  8194.000000  8191.000000  8195.000000  8198.000000  8193.000000  819

Data = 0

  8190.000000  8189.000000  8186.000000  8185.000000  8188.000000  8192.000000  8188.000000  8187.000000  8184.000000  818

Data = 1

  8191.000000  8194.000000  8198.000000  8198.000000  8189.000000  8192.000000  8193.000000  8197.000000  8195.000000  818

Data = 0

  8184.000000  8187.000000  8186.000000  8188.000000  8190.000000  8184.000000  8185.000000  8188.000000  8187.000000  819

Data = 1

  8186.000000  8189.000000  8192.000000  8194.000000  8197.000000  8193.000000  8193.000000  8192.000000  8194.000000  819

Data = 0

  8187.000000  8189.000000  8188.000000  8187.000000  8186.000000  8188.000000  8189.000000  8190.000000  8187.000000  818
```

Fig. 5.21: Values for BPSK at receiver

At the receiver, code for detecting these values for BPSK is as shown in Figure 5.22

```
int DetectBPSK()
{

    if( stream2[1]>8190)
    {
        return 1;
    }
    else
    {
        return 0;
    }

}
```

Fig. 5.22: Code for detecting BPSK

5.3.4 SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique.

The SDR platform is supported by highly productive development system emphasizing the value of a structured development process that takes the developer through modeling, system analysis, implementation and verification by bit error rate analysis. SDR platform is a massively parallel processor in which transmitter and a part of receiver is implemented on DSP processor and demodulator is implemented on FPGA. For adaptive modulator-demodulator architecture, a bit error rate analysis is presented. Result shows that for same BER, BPSK requires minimum SNR.

In this work, adaptive software defined radio system for different modulation scheme is designed. In this receiver can select its demodulator in accordance with the signal transmitted by transmitter. The signal to be transmitted will depend upon operator's choice or it depends upon channel conditions. For carrier frequency, fractional 'N' frequency synthesizer [16] is used.

Modulator circuit is implemented on DSP processor while demodulator circuit is implemented on FPGA processor because demodulator circuit is complex and we prefer complex circuit to be implemented on FPGA because component to component difference in FPGA is very less so it will take less time to demodulate.

- ***System level simulations***

In order to characterize the performance of the adaptive modulation techniques, SDR based transmitter - receiver model has been constructed as shown in section 4.5. The channel is assumed to be Gaussian and the noise considered as white. To evaluate the BER for different modulation schemes for each value of SNR, a frame consisting of 2,000 bits is sent from the transmitter having random values 0 and 1. The results below have been obtained from 100 independent iterations of each such frame.

Initially, type of modulation is fed to modulation modifier task. After that if modulation scheme selected is FSK then 1 bit per symbol period is selected to transmit. Then if the input data is 0 then signal of less frequency will be transmitted and if input data is 1 then signal of high frequency will be transmitted. These frequencies are corresponding to the mark and space frequencies. This can be seen in figure 5.23

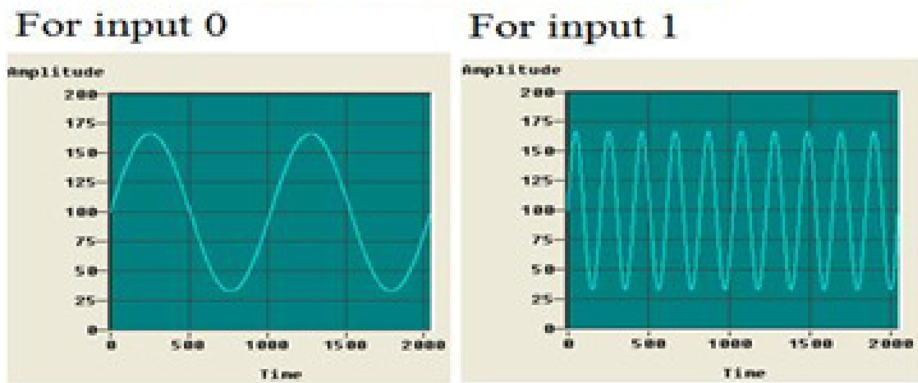


Fig. 5.23: FSK modulation (baseband)

In QPSK, two bits per symbol period is selected to transmit. Then if the input data is 00 then signal of phase shift 0 will be transmitted and if input data is 01 then signal with phase shift 90 will be transmitted. Similarly we have the outputs of other 2 two inputs (i: e for 10 and 11). This can be seen in figure 5.24.

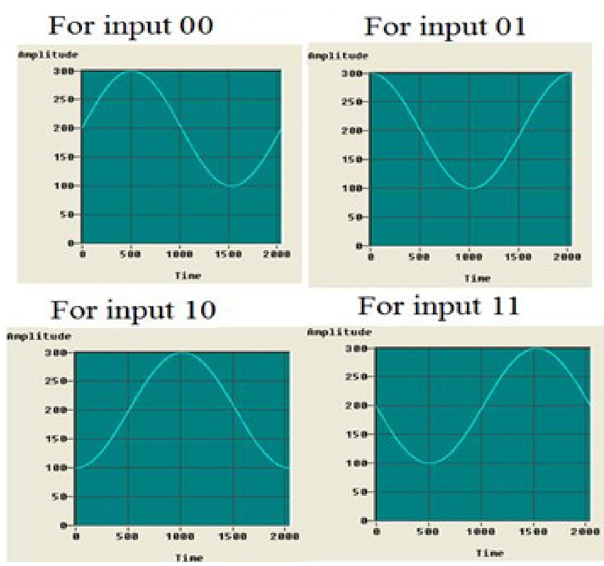


Fig. 5.24: QPSK modulation (baseband)

In BPSK, one bit per symbol period is selected to transmit. Then if the input data is 0 then signal of phase shift 0 will be transmitted and if input data is 1 then signal with phase shift 180 will be transmitted. This can be seen in figure 5.25.

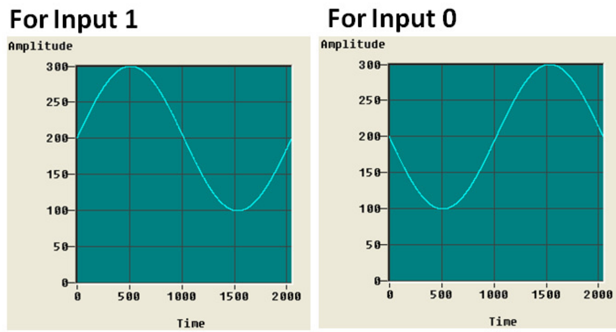


Fig. 5.25: BPSK modulation (BPSK)

Similarly for 8-PSK, three bits per symbol period is selected to transmit. So we have 8 signals of different phase shifts. This can be seen in figure 5.26.

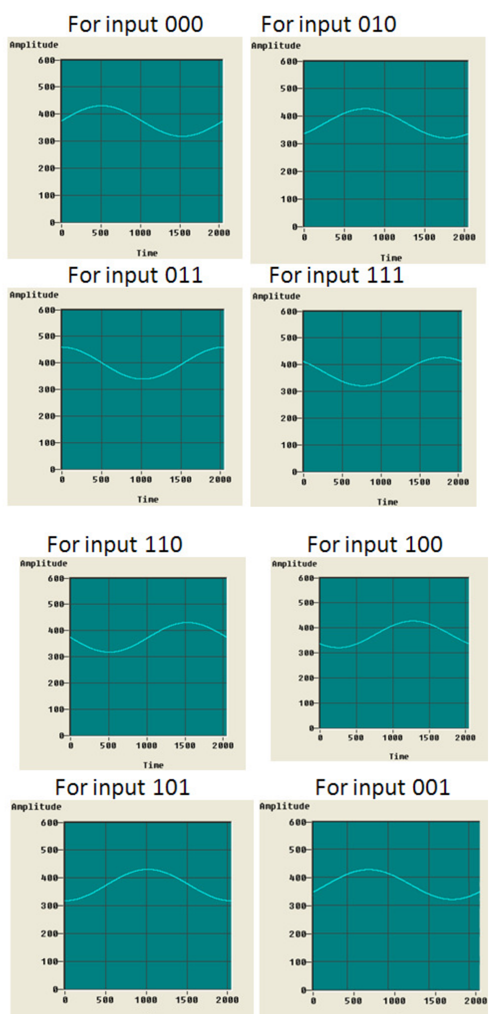


Fig 5.26: 8-PSK modulation (baseband)

At the receiver end, the coded signal is received at the tunable received antenna. The signal is divided into decision regions, one for each transmitted symbol. An error occurs if a symbol is received in a different region from the one it was transmitted. Thus, the maximum allowed error is equal to half the distance between two adjacent symbols. Recovered signal is sent to the Adaptive software defined radio receiver structure where decoding of signal is done. After decoding, the uncoded signal is considered as a reference signal and the decoded signal is compared with it and BER is evaluated. Figure 5.27 shows the adaptive SDR receiver based BER performance for different modulation schemes. From the figure 5.27 it is clear that for same BER, BPSK requires minimum SNR

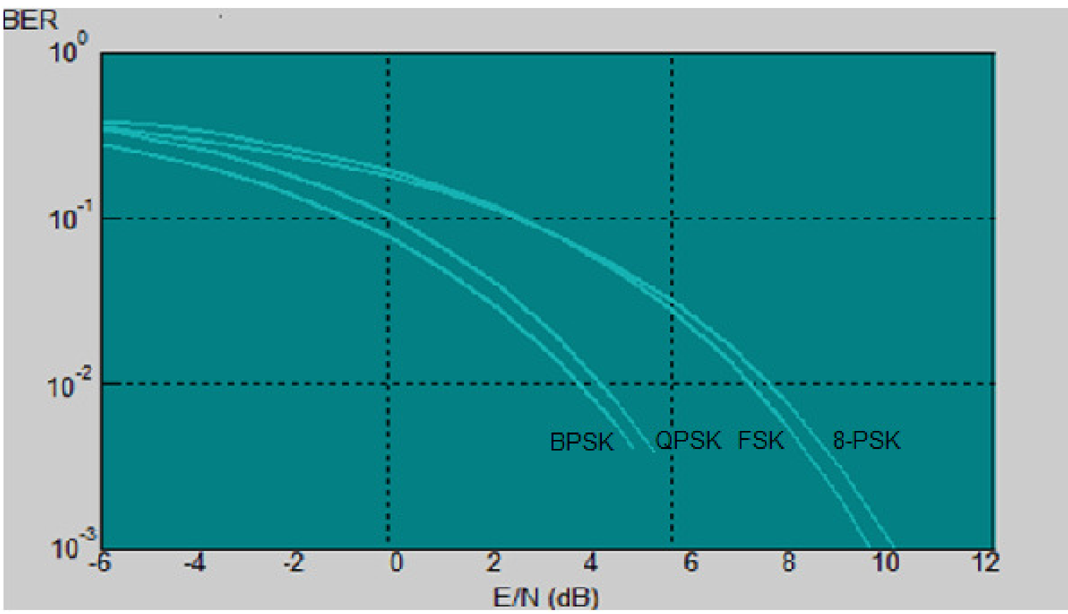


Fig. 5.27: BER Comparison for different modulation schemes in Adaptive SDR receiver.

CONCLUSION

Digital modulation techniques and SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique is presented in this thesis which utilizes the task division scheme for pipelining purpose. This system can be used by military for secured transmission of secret data.

Basically Software defined radio is the base of next generation wireless technology. We can modify our system by just changing the software used in the hardware and hence saving the cost required to change the hardware at each and every place. This also saves the time required to implement the new technology system in practice because we can directly download the software from internet and install on our preexisting hardware.

Basically Software Defined Radio (SDR) system is a useful and adaptable future-proof solution to cover both existing and emerging standards. It provides elements with re-configurability, intelligence and software programmable hardware. In addition, the emerging user requirements on reconfigurable mobile systems and networks are paving the way for the introduction of re-configurability in future mobile systems.

SDR provides an efficient and secure solution to the problem of building multi-mode, multi-band and multifunctional wireless communication devices. We can use Model Based Design methodology and tools to integrate the major development phases like including, simulation, code generation, hardware in the loop testing into one continuous design cycle.

We can implement SDR based application on The SMT-8036 which is a development kit for SDR applications consisting of a DSP coupled with an FPGA connected to two ADCs and a DAC. It is a PCI system based on 3 main modules: C64xx-based module (SMT365-8-2) combined with a dual high-speed ADC/DAC module (SMT370), both plugged on a PCI carrier board (SMT310Q). We can also use Xilinx system generator and Matlab Simulink and various code generation tools for the implementation. Here in our work we have plotted BER for different Digital modulation techniques and we concluded that for same BER, BPSK needs minimum SNR

FUTURE SCOPE

SMT-8036 based implementation of secured Software Defined Radio system for adaptive modulation technique is implemented. In future it can be extended for different channel and space time coding.

REFERENCES

[1] Hüseyin Arslan," Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems", *Springer publication*, Second edition, 2007.

[2] "SDRF Cognitive Radio Definitions", http://www.sdrforum.org/pages/documentLibrary/documents/SDRF-06-R-0011-V1_0_0.pdf Last accessed on June 10, 2011.

[3] Markus Dillinger, Kambiz Madani, Nancy Alonistioti, "Software Defined Radio: Architectures, Systems and Functions" *Wiley & Sons publication*, third edition, 2003.

[4] J. Mitola, "The Software Radios—Survey, Critical Evaluation And Future Directions," *in Proceedings of . National Telesystems Conference (NTC'92)*, IEEE, 1992, pp. 15–23.

[5] Walter Tuttlebee, "Software Defined Radio," *Wiley publication*, second edition, 2002.

[6] "High-speed ADC technology paves the way for software defined radio", http://www.eetimes.com/design/automotive-design/4009968/High-speed-ADC-technology-paves-the-way-for-software-defined-radio.pdf, last accessed on June 10, 2011.

[7] Mansour Ahmadian, Zhila (Jila) Nazari, ed. al. , "Model Based Design and SDR" *proceedings of 2nd IEE/EURASIP Conference*, 2005, pp 8-14.

[8] A. P. Vinod & Edmund M-K. Lai & Amos Omondi (Ed.) " Special Issue On Signal Processing For Software Defined Radio Handsets", *Journal of Signal Processing System*, 2009, pp 1-3.

[9] Adrian Tarniceriu, Bogdan lordache, Silvian Spiridon, "An Analysis On Digital Modulation Techniques For Software Defined Radio Applications", *proceedings of Semiconductor Conference*, 2007, pp 571 - 574.

[10] Raúl Dopico-López , José M. Camas-Albar,Miguel A. Melchor,David Castells-Rufas, "How To Obtain More Powerful SDRs Using Multicore Architectures", *proceedings of the SDR'09 Technical Conference and Product Exposition*, 2009, pp 430-434.

[11] M.Saravanan, Dr.S.Ravi, "SDR Based Multiple Access Technique For Multimedia Applications", *proceedings of Communications and Information Technologies*, 2007, pp 491-494.

[12] Brian A. Dalio and Kevin A. Shelby, "The Implementation Of OFDM Waveforms On An SDR Development Platform Supporting A Massively Parallel Processor" *proceedings of the SDR'09 Technical Conference and Product Exposition*, 2009, pp 447- 451.

[13] Feng Ge and Charles W. Bostian, "SDR Implementation Issues: Rf Front End Nonlinearity And Dynamic Computing Resource Allocation", *proceedings of the SDR'09 Technical Conference and Product Exposition*, 2009, pp 312-315.

[14] David Murotake, Ph.D. and Antonio Martin, "A High Assurance Wireless Computing System (Hawcs™) For Software Defined Radio", *Proceeding of the SDR 06 Technical Conference and Product Exposition*, 2006, pp-116-118.

[15] Shervin Moloudi, Koji Takinami, Michael Youssef, Mohyee Mikhemar, Asad Abidi, "An Outphasing Power Amplifier For A Software-Defined Radio Transmitter", *proceedings of 2008 IEEE Solid-State Circuits Conference*, 2008, pp 568 – 636.

[16] Sabbir A. Osmany, Frank Herzel and J. Christoph Scheytt, "An Integrated Fractional-N Frequency Synthesizer For Software-Defined Radio Applications" *proceedings of Silicon Monolithic Integrated Circuits in RF Systems (SiRF)*, 2010, pp 243-246.

[17] Fabio Garzia, Waqar Hussain, Roberto Airoldi and Jari Nurmi, "A Reconfigurable SoC tailored to Software Defined Radio Applications", proceedings of NORCHIP, 2009, pp 1-4.

[18] Antonietta Stango and Neeli R. Prasad, "Policy-based approach for secure radio software download", *Proceedings of SDR'09 Technical Conference and Product Exposition*, 2009, pg 1-6.

[19] Chris H. Dick and San Jose, "Design and Implementation of High-Performance FPGA Signal Processing Datapaths for Software Defined Radios", www.vmecritical.com/articles/id/?1664, last accessed on June 10, 2011.

[20] George Nychis, Thibaud Hottelier et. Al., "Enabling MAC Protocol Implementations on Software Defined Radios", www.andrew.cmu.edu/user/gnychis/nychis_nsdi09.pdf , last accessed on June 10, 2011.

[21] Stéphane LECOMTE, Samuel GUILLOUARD et. al., "A co-design methodology based on model driven engineering for sdr equipments" *proceedings of the SDR'09 Technical Conference*, 2009, pp 1-6.

[22] Mehrbod Mohajer, Abbas Mohammadi and Abdolali Abdipour, "A Software Defined Radio Direct Conversion Receiver", *proceedings of European technology of wireless technology*, IEEE, 2005, pp 423-426.

[23] Mark Cummings and Todor Cooklev, "Tutorial: Software-Defined Radio Technology", *proceedings of International Conference on Computer Design*, IEEE, 2007, pp 103-104.

[24] Sanjay Pithadia, "Smart Selection of ADC/DAC Enables Better Design of Software-Defined Radio", Texas Instruments, http://focus.ti.com/lit/an/slaa407/slaa407.pdf, last accessed on June 10, 2011.

[25] Lyle Johnson and Mt Vernon, "The High Performance Software Defined Radio Project", http://www.tapr.org/pdf/DCC2006-HPSDRproject-KK7P.pdf, last accessed on June 10, 2011.

[26] S. Weiss, A. Shligersky, S. Abendroth et. al., "A Software Defined Radio Testbed Implementation".
www.eprints.ecs.soton.ac.uk/8275/1/IEEclq03_sdr.pdf, last accessed on June 10, 2011.

[27] S. Glass, V. Muthukkumarasamy and M. Portmann, "A Software-Defined Radio Receiver for APCO Project 25 Signals", http://portal.acm.org/citation.cfm?id=1582395, last accessed on June 10, 2011.

[28] Shinichiro Haruyama and Robert H. Morelos-Zaragoza, "A Software Defined Radio Platform with Direct Conversion: SOPRANO", *Kluwer journal of wireless Personal Communication*, vol 2, 2001, pp 212 -217.

[29] Louis Belanger, "Advanced SDR platform eases multiprotocol radio development" http://rfdesign.com/mag/0701RFDF3.pdf, last accessed on June 10, 2011

[30] Ajay Kr. Singh, Ankita Taneja, et. al.," Software Defined Radio - Talk Through Application", *Proceedings of International Conference on Microwave – 08*, IEEE, 2008, pp 821-825.

[31] Tajinder Manku, Christopher Snyder et.al. , "Dual Mixer Downconversion Architecture Using Complex Mixing Signals: Enabling Solutions for Software Defined Radios", *proceedings of custom integrated circuits conference*, IEEE, 2002, pp 227-234.

[32] T. Nesimoglu, M. A. Beach, J. R. MacLeod and P. A. Warr, "Mixer Linearisation for Software Defined Radio Applications" *Proceedings of Vehicular Technology Conference,* IEEE, Vol-1, 2002, pp 534-538.

[33] Mohamed Ratni, Dragan Krupezevic, et. al., “Broadband Digital Direct Down Conversion Receiver Suitable for Software Defined Radio”, *The International symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, Vol-1, 2002. , pp 100-104

[34] Zhiyu Ru, Eric A.M. Klumperink, and Bram Nauta, “On the Suitability of Discrete-Time Receivers for Software-Defined Radio”, *The International symposium on circuits and systems*, IEEE, 2007, pp 2522-2525.

[35] Howard Patterson, Forrest Scarpitto and Brian Bielick, “Broadband characterization of gan transistors for software defined radio power amplifier applications”, *proceedings of Military communications conference*, IEEE, 2007, pp 1-7.

[36] Franz Hlawatsch, “Analog and Digital Communication Techniques”, *wiley publications*, second edition, 1999.

[37] Bruce A. Fette, “Cognitive Radio Technology”, *Newnes publications*, first edition, 2006.

[38] Fuqin Xiong, “Digital Modulation Techniques”, *ARTECH HOUSE publications*, second edition, 2006.

[39] Wayne Tomasi, “Electronic Communications Systems”, *Dorling Kindersley (INDIA) Pvt. Ltd.*, Fifth edition, 2009.

[40] Álvaro Álvarez, David Blanco, Rafael Palacio et.al. ,“WHERE-adapted Platforms Description”, http://www.kn-s.dlr.de/where/documents/Deliverable51.pdf, last accessed on June 10, 2011.

[41] Hans Heinrich Meinke and Klaus Lange. “Software Defined Radio: components and design issues”, Springer, Berlin, 5th Edition, 1992.

[42] “SMT-349 user manual”, http://www.sundance.com/docs/SMT349UserManual.pdf, last accessed on June 10, 2011.

[43] http://www.3l.com/how-it-works. last accessed on September 2010.