DEVELOPMENT OF A NANOSATELLITE SOFTWARE DEFINED RADIO COMMUNICATIONS SYSTEM

NATASHA GADKARI

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

May 2015

Abstract

Communications systems designed with application-specific integrated circuit (ASIC) technology suffer from one very significant disadvantage - the integrated circuits do not possess the ability of programmability. However, Software Defined Radio's (SDR's) integrated with Field Programmable Gate Arrays (FPGA) provide an opportunity to update the communication system on nanosatellites (which are physically difficult to access) due to their capability of performing signal processing in software. SDR signal processing is performed in software on reprogrammable elements such as FPGA's. Applying this technique to nanosatellite communications systems will optimize the operations of the hardware, and increase the flexibility of the system.

In this research a transceiver algorithm for a nanosatellite software defined radio communications is designed. The developed design is capable of modulation of data to transmit information and demodulation of data to receive information. The transceiver algorithm also works at different baud rates. The design implementation was successfully tested with FPGA-based hardware to demonstrate feasibility of the transceiver design with a hardware platform suitable for SDR implementation.

Acknowledgements

I would like to thank all those who have supported me and encouraged my work through my years at York University. In particular I would like to thank my supervisors, Professor Regina Lee and Professor Sunil Bisnath for their continued support and motivation as I have progressed with my research.

I would like to thank the entire YU-GNSS team at York University, both past and present members, for their friendship, continued motivation, and willingness to assist me whenever I have come across problems. In particular, Thong Thai, whose knowledge of electronics and software has proved most invaluable. I would like to thank Kavita Joshi and Gowry Sinnathambi for providing very needed distractions to otherwise stressful days. I would like to give a special thanks to Surabhi Guruprasad for being an amazing team member for our research projects and providing help whenever needed.

I would like to thank COM DEV and the Natural Science and Engineering Council for their financial support.

I would like to express my gratitude and thanks to Archit Chitnis for keeping me in line and on track as I was working towards the completion of my work and the writing of my thesis. Lastly, I want to thank my parents whose patience and support during my entire life has brought me to where I am today.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ADC	Analog to Digital Converter
AFSK	Audio Frequency Shift Keying
ASIC	Application-Specific Integrated Circuits
ASK	Amplitude Shift Keying
BEE2	Berkeley Emulation Engine
BFSK	Binary Frequency Shift Keying
BPS	Bits Per Second
BPSK	Binary Phase Shift Keying
COSMIAC	Configurable Space Microsystem Innovations and Applications Centre
COTS	Commercial Of The Shelf
CPM	Continuous Phase Modulation
DAC	Digital to Analog Converter
DPSK	Differential Phase Shift Keying
DSP	Digital Signal Processing
EO	Earth Observation
FCS	Frame Check Sequences
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FM	Frequency Modulation
FPGA	Field Programmable Gate Arrays
FSK	Frequency Shift Keying
GNSS	Global Navigation Satellite System
GPP	General Purpose Processor

GPU	Graphics Processing Unit
HDL	Hardware Description Language
ISS	International Space Station
KHU	Kyung Hee University
KUAR	Kansas U. Agile Radio
M-FSK	M-ary Frequency Shift Keying
MODEM	Modulation / Demodulation
OBC	On-Board Computer
OQPSK	Offset Quadrature Phase Shift Keying
PAM	Pulse Amplitude Modulation
PID	Protocol Identifier
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
SDR	Software Defined Radio
SIGMA	Scientific CubeSat with Instrument for Global Magnetic field and Radiation
SNR	Signal to Noise Ratio
TEPC	Tissue Equivalent Proportional Counter
TNC	Terminal Node Controller
UHF	Ultra-High Frequency
USRP	Universal Software Radio Peripheral
VHF	Very-High Frequency
WARP	Wireless Open-Access Research Platform

List of Symbols

S_1	Signal generated with Mark frequency
S_2	Signal generated with Space frequency
A	Amplitude
f_1	Mark frequency
f_2	Space frequency
Φ_1	Initial phase for signal with Mark frequency
Φ_2	Initial phase for signal with Space frequency
T, t	Time period
x_C	Sinusoidal carrier wave
A_C	Amplitude of carrier wave
f_C	Instantaneous frequency
m(t)	Message
$\theta(t)$	Baseband signal
k_f	Frequency sensitivity
FD	Frequency deviation
T_S	Sampling period
S_I	FSK modulated signal
f_i	Frequency of FSK modulated signal
Θ	Phase of signal
l_1^2	Output of Mark frequency correlators
l_2^2	Output of Space frequency correlators
r(t)	Received signal
S_M	Signal generated for Mark frequency

S_S	Signal generated for Space frequency
S_i	FSK modulated signal
x(t)	Complex signal
u(t)	Real signal

1 Introduction

A Software Defined Radio (SDR) is a radio in which the signal is processed entirely on reprogrammable elements (Oliveri, 2011). An SDR integrated with a Field Programmable Gate Array (FPGA) provides an opportunity to update the communication system on nanosatellites, which traditionally are physically difficult to access. Applying this technique to nanosatellites communication system will optimize the operations of the hardware, and increase the flexibility of the system. This thesis will begin with an overview of past nanosatellite missions. Then the motivation and objectives for this research is explained. Last, a thesis outline is provided.

1.1 Nanosatellite Missions

Traditionally, the majority of satellites launched in the past decade have mass greater than 1000 kg. Due to the size and weight of these satellites, the structure and the development of the nanosatellite will be more complex than the small satellites, which increase the development period and manufacturing cost (Rogers et al., 2010). Due to such long development periods and high expenses, space-tested technologies were preferred to mitigate the risk of failure. Therefore the missions with the large satellites limited the scope for research of new technologies.

In the space industry, mass of a satellite is used to distinguish them. Satellites with mass greater than 1000 kg are large satellites; medium satellites have mass from 500 to 1000 kg; mini satellites with mass from 100 to 500 kg; micro satellites have mass from 10 to

100 kg; nanosatellites with mass from 1 to 10 kg; picosatellites have mass from 0.1 – 1 kg; and femto satellites have mass < 100 g (Konecny, 2004).

In recent years, technologies have been advanced in the direction of making smaller and lighter hardware components with higher capabilities. With advances in highly reliable commercial electronics and miniaturization techniques, nanosatellites are becoming popular (Rogers et al., 2010). Their main advantages over traditional satellites are much lower budgets, their modular nature and flexible launch structures. They also have significantly faster development cycles as compared to traditional satellites. The technology provides miniature and reliable satellites conducting single purpose missions (Rogers et al., 2010). Government and private sectors are showing interest in nanosatellite technology. Nanosatellites provide a suitable platform for universities and small companies to develop new space technology and demonstrate in the space field (Rogers et al., 2010). Nanosatellites are suitable for Earth observation and near- Earth missions (Trusculescu, 2012).

Larger satellites are still required for the outer space missions and other targeted missions that require large payloads however the new nanosatellite technology is allowing further research to be conducted for cost effective and commercial off the shelf (COTS) space tools. Figure 1 illustrates an example of a nanosatellite. AAUSAT3 (AAUSAT3, 2012) is the third CubeSat (a class of nanosatellites that conforms to CubeSat specifications published by California Polytechnic State University of 10 X 10 X 10 cm(Oliveri, 2011) built and operated by students from Aalborg University in Denmark. It was launched on 25 February 2013 from Satish Dhawan Space Centre in India on a PSLV rocket.

AAUSAT3 carries two Automatic Identification System (AIS) receivers as the main payload.

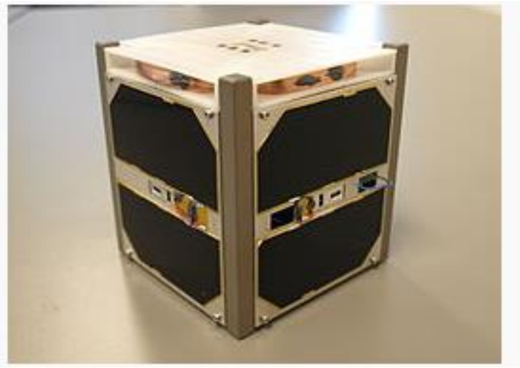


Figure 1 : AAUSAT 3 CubeSat (AAUSAT3, 2012)

Since nanosatellite missions have very constrained budgets, COTS components are used for development. Nanosatellites are used by universities to introduce the space systems engineering process to students and provide hands on training. The training includes constructing different space missions and developing hardware specializations (Rogers et al., 2010). For example Figure 2 illustrates the Sigma (Scientific CubeSat with Instrument for Global Magnetic field and Radiation) CubeSat being developed by the School of Space Research at Kyung Hee University (KHU), Korea in cooperation with the Korea Astronomy and Space Science Institute, York University and the University of New Hampshire to provide their students with training on building nanosatellite subsystems. The payloads include the Tissue Equivalent Proportional Counter (TEPC) and a magnetometer. SIGMA is a 3-unit CubeSat with a mass of 3.2 kg.

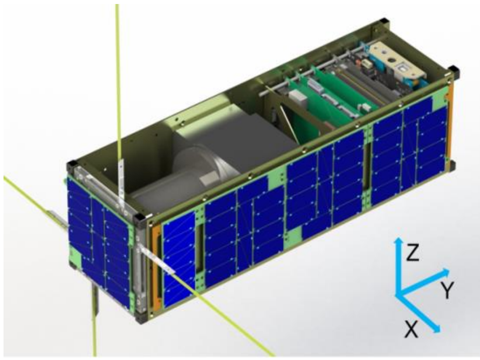


Figure 2 : Proposed SIGMA CubeSat

Nanosatellites provide the opportunity to enable missions which large satellites are unable to accomplish because of their bulk nature. An example is for hardware tested on nanosatellites before being used for future missions. XI-IV developed by University of Tokyo was built for the purpose of testing a working satellite bus for future satellite missions (Klofas, 2008). Nanosatellites can provide a platform for missions which require constellations of satellites for low data rate communications that use low power for operations. Nanosatellites can present a platform for formation flying satellites so that data can be gathered for missions at several points around Earth (Yoon et al., 2014). Nanosatellites also provide the opportunity to be used for inspection of larger satellites and are also used for near-Earth space monitoring and Earth observation missions.

1.2 Challenges of Nanosatellite Missions

Nanosatellite missions face a number of challenges in their development cycle. The constraint of limited mass of less than 10 kg is a major challenge as it limits the hardware that can be used for of satellite subsystems such as power, attitude control and communications. Most nanosatellites are CubeSats whose limited size of 10 cm x 10 cm x 10 cm and mass of less than 10 kg limits the options for payload designed for research purposes. The low cost of the mission also limits the type of hardware that is feasible.

Small satellites especially nanosatellites also face the problem of limited power on a board which is small enough to fit on the satellite and capable enough to provide power to the satellite. Desktop computers and microprocessors cannot fit on small satellites, hence the computing and control of the satellite is a challenge. There is a need to use boards which will do the multiple designated functions and also fit on the satellite. The other challenges faced by nanosatellites are attitude pointing and control, propulsion and communications. This thesis is primarily focussed on the communication challenge of nanosatellites and will be discussed in the rest of the thesis.

1.2.1 Challenges of Nanosatellite Communications System

The two important challenges that are faced by the communications system are structure and power. Structurally, the transceiver which consists of a transmitter and receiver needs to be reduced in size and should not be very heavy. Antennas need to be miniaturized to fit on a nanosatellite. Power is a scarce resource on a nanosatellite. Higher data rates require higher power for the communication system (Homan, 2008). Apart from structural and power constraints, the communications system needs to have a reliable link between the satellite and the ground station and the launch vehicle along with relaying the information efficiently. The communication system also requires higher data rates for effective communication at various frequency bands.

1.3 Research Motivation

A successful nanosatellite mission requires an effective communication system. Historically, communication systems on nanosatellites have been built using Application-Specific Integrated Circuits (ASIC) that are designed to perform the sole function of communications. An example of a communications system built on ASIC is the ISIS Full Duplex Transceiver shown in Figure 3. The ISIS Full Duplex Transceiver (CubeSatShop.com, 2006) is designed for a CubeSat or small satellite, adds telemetry and telecommand capability and can relay data at 1200 bits per second and 9600 bits per second downlink and uses Audio Frequency Shift Keying (AFSK) for uplink. However, with the ever-growing need to use the limited space and mass on nanosatellite as effectively as possible; the motivation for this research thesis is to build a system which can aid in building more flexibility into a nanosatellite communication system in future.

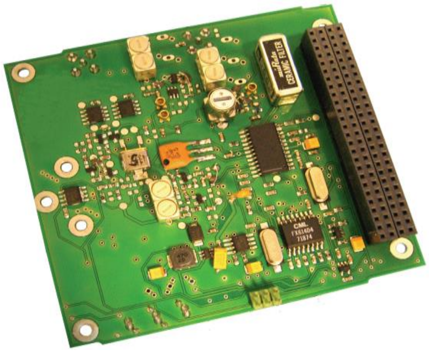


Figure 3 : ISIS Full Duplex Transceiver (CubeSatShop.com, 2006)

This research motivation is realistic and achievable today because of the technology advancement accomplished as part of the software defined everything technology trend (Riveria, 2013), namely Software Defined Radio systems.

1.3.1 Flexibility Afforded by SDR

SDR systems rely significantly on software for their functionality, including baseband functionality, and are known to use encoders, modulators, filters and other such components of a communications system defined and designed in software (Oliveri, 2011).

The main function of an SDR is to provide increased flexibility by implementing a maximum of the communications system code on reprogrammable hardware. Different functionalities of communications system are implemented through software implementation of numerous signal processing elements. SDR's are low-cost and low-mass as there is no requirement of large and expensive hardware due to the use of software for processing (Oliveri, 2011). The radio can be reconfigured for various applications other than communications such as remote sensing, radio occultation, sensor data gathering (Davis et al., 2011) through software implementation of the functions, without having to redesign the entire hardware system.

1.3.2 Benefits of SDR based Nanosatellite Communication System

An SDR-based single hardware unit can receive multiple signals over a large frequency band and process these signals in software and also allows for software-generated signals to be transmitted. These multiple functions can be implemented by changing specific software modules. Signal processing systems are developed and tested easily; and modifications and upgrades are done much more readily on the system, since it is a software module. Moreover, flexible communications protocols are developed to adapt to the system user. Hardware radios are unable to conform to new standards or protocols, but an SDR can be reconfigured to support new standards which are developing or which may develop in the future. Other advantages of SDR’s include the simplicity of quickly testing new technology, and testing individual hardware components by simulating the surrounding components in software (Oliveri, 2011).

The research presented in this thesis is an incremental contribution towards the development of a nanosatellite SDR- based communications system. Oliveri(2011) in his thesis titled “Modular FPGA Based software defined radio for CubeSats” developed a SDR hardware platform called the Configurable Space Microsystem Innovations and Applications Centre (COSMIAC) CubeSat FPGA board which can be fitted on 1U CubeSat(Oliveri,2011). The research in this thesis takes the next step to develop the software design, which does the signal processing for the communication system and presents the research objectives discussed in section 1.4.

1.4 Research Objectives

The research objectives are as follows:

1. To examine available technologies for nanosatellite communications system and consider SDR design as a cost-effective, flexible alternative.
2. To design the software implementation of SDR as a nanosatellite communications system. The proposed design is to perform signal processing for generic communication purpose. This research is an incremental contribution towards the development of a nanosatellite SDR-based communication system, as the

previous work in this field was based on COSMIAC system and tested only with GnuRadio. The proposed design eliminates the use of GnuRadio by implementing in Simulink to allow for easy porting onto the FPGA-based system

3. To implement both receiving and transmission functions of SDR. The proposed design uses software for baseband functionality in signal processing. This research implements a singular modulation scheme to demonstrate the feasibility of an SDR system for nanosatellite communications. The design is implemented on a hardware development platform which is commercial off the shelf, meets the budget constraints and can be enhanced for nanosatellites.

The research is an incremental step in the process of a developing a nanosatellite software defined radio communications system making it unique in the use of SDR for nanosatellite application.

1.5 Thesis Outline

The thesis outline is as follows. The second chapter discusses the background of SDR technology. It also describes the various design options and which design option is suitable for the research objectives to be achieved. The third chapter discusses the nanosatellite communications and link budget analysis. The fourth chapter discusses the hardware platform options. It also talks about which option is chosen and the reasons for picking this option. It describes in detail the hardware platform. The fifth chapter discusses the communications algorithm implemented on the hardware. The sixth chapter discusses the experimental tests and evaluates the performance of the system. The seventh chapter discusses the conclusions from the research and the contributions provided by this thesis. It also discusses the future work required to make this research efficient and ready to be implemented on a nanosatellite.

2 Software Defined Radio Technology

In this chapter, a background on SDR is provided. A literature survey of existing technology is presented as well. In addition, a number of design options are explored. And a brief background on digital communications is provided.

2.1 Software Defined Radio

An SDR is a radio in which the signal is processed entirely on reprogrammable elements. The basic architecture of SDR is shown in Figure 4 attached with the front end and antenna. In this system, all the baseband processing of signals is done in software. As seen in the architecture, the signal is received through the antenna and then converted to the digital form in the front end. These digitized complex baseband data are then processed in the processing engine to extract data frames, which are further sent for processing for the specific application. This process can be implemented for the transmitter as well in reverse order (Oliveri, 2011).

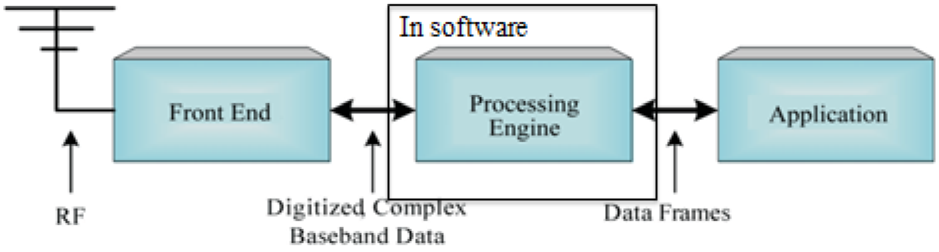


Figure 4: SDR architecture (Oliveri, 2011)

2.1.1 Survey of the Existing SDR Technology

A survey of the existing SDR Technology is presented in this section.

JPL SDR: The SDR has been developed for the Connect project on board the International Space Station (ISS) (Johnson, 2012). This Radio uses S- Band for communication purposes; however, it has the ability to receive L-Band signals as well. The SDR was developed by NASA and JPL.

Table 1 below lists the physical characteristic of the JPL SDR.

Physical Characteristics	
Mass	6.6 Kg
Power	15 W Rx(Typical) + 2W (GPS) + 65 W Tx S Band
Frequencies	S-band, L1 ,L2 and L5
Digital Processing	66 MHz SPARC V8 128 Mbyte SDRAM + 512 MByte Flash 2x Xilinx Virtex II 3Mgate FPGAs SDRAM and Flash on each FPGA

Table 1: Physical Characteristics of JPL SDR (Johnson, 2012)

Figure 5: JPL SDR (Johnson, 2012)

Harris SDR: The Harris SDR has been developed by Harris Engineering Corporation in collaboration with NASA for communication purposes on board the International Space Station (ISS). The Harris SDR is a part of the Connect project and utilizes the Ka-band for communication (Johnson, 2012).

Table 2 lists the Physical characteristic of the Harris SDR,

Physical Characteristics	
Mass	19.2 Kg
Power	100 W
Frequencies	Ka-band
Digital Processing	700 MIP Power PC processor and 4 Xilinx Virtex IV FPGAs

Table 2: Physical Characteristics of Harris SDR (Johnson, 2012)

Figure 6: Harris SDR (Johnson, 2012)

Namuru Software Receiver Platform: The Namuru Software Receiver platform is being developed by the University of New South Wales in Sydney, Australia (Grillenberger, 2008). There have been three versions of the receiver which have been developed so far. Namuru V1, Namuru V2 and Namuru V3. They are being developed for research purposes and have not been flown in any mission so far.

Table 3 lists the Physical characteristic of Namuru V1 and Namuru V2. Table 4 lists the Physical characteristic of Namuru V3.

Physical Characteristics	
Mass	105 grams
Power	7-9 V
Channels	12 channels
Frequencies	L1 RF front ends, L2 up converter
Digital Processing	NiosII soft-core CPU , FPGA Altera Cyclone II EP2C50F484C8 , EPCS64 64-Mbit flash serial Zarlink GP2015 RF chip for GPS L1 upconverter circuit used to configure second front end to GPS L2.

	1 USB 2.0 and 2 RS232 interfaces, 64 MB SDRAM and 8 MB flash memory.

Table 3: Physical Characteristics of Namuru V1 and V2 (Grillenberger, 2008)

Physical Characteristics	
Mass	105 grams
Power	7-9 V
Channels	12 channels
Frequencies	L1 C/A
Digital Processing	Zarlink GP2015 RF FE, Actel ProASIC FPGA, Actel Smart Fusion A2F500 upconverter circuit used to configure second front end to GPS L2. 1 USB 2.0 and 2 RS232 interfaces, 64MB SDRAM and 8MB flash memory.

Table 4: Physical Characteristics of Namuru V3 (Grillenberger, 2008)

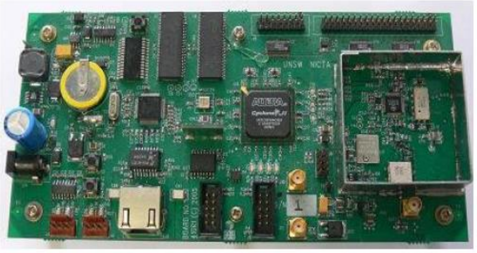


Figure 7: NamuruV1 receiver (Grillenberger, 2008)

Gemini Alpha: This SDR is being developed by the Microsatellites and Space Microsystems Lab of University of Bologna for the ALMASat Earth Observation (EO) mission. The applications of this SDR are for orbit determination and images geo-referencing for both GPS and Galileo constellations (Avanzi and Tortora, 2010). Table 5 lists the Physical characteristic of Gemini Alpha.

Physical Characteristics	
Mass	
Power	5 W
Channels	12 channels
Frequencies	Dual Frequency L1/E5 or E1/L2
Digital Processing	Xilinx Virtex5 FPGA FXT series with a 32-bit PowerPC PPC440 in form of hard processor. External soft FPU can be attached. 64 MB of 200 MHz DDR2 SDRAM , 16 MB of Flash memory, front-ends are based on the Maxim MAX2769 IC Front-end for GPS L1 or Galileo E1 signal, while the second is designed for the L2 signal trough up-conversion from 1227.6 MHz to 1575.42 MHz.

Table 5: Physical Characteristics of Gemini Alpha (Avanzi and Tortora, 2010)

GNSS Software Receiver for MICROSCOPE: This SDR is being developed by Syrlinks of Bruz, France, to be tested on board the scientific satellite MICROSCOPE. It will be utilized for navigation and tracking applications (Grondin et al., 2010). Table 6 lists the Physical characteristic of GNSS Software Receiver for MICROSCOPE.

Physical Characteristics	
Mass	0.9 kg
Power	8 W
Frequencies	L1/E1
Channels	9
Digital Processing	The signal processing functions are split into two main components : an FPGA and a DSP

Table 6: Physical Characteristics of GNSS Software Receiver for MICROSCOPE (Grondin et al., 2010).

2.1.2 Historical Overview of Software Defined Radio for Nanosatellites

Some of SDR's developed are also presented here. These were developed specifically for CubeSat missions, but do not use open source hardware and software and therefore cannot be enhanced for added applications like remote sensing, etc.

CubeSat Software Defined radio by Vulcan Wireless is a UHF transceiver designed for communications with the following specifications: direct to war fighter Communications; ¼ CubeSat form factor; half duplex/full duplex configurations; and on orbit flight heritage (Vulcan Wireless, 2010). Vulcan Wireless also offers Micro Blackbox Transponder which works with fewer protocols and supports S-Band frequencies (Vulcan Wireless, 2010).

2.1.2.1 Configurable Space Microsystem Innovations and Applications Center (COSMIAC)

An SDR for nanosatellites is being developed by Configurable Space Microsystem Innovations and Applications Center (COSMIAC) CubeSat SDR system. COSMIAC operates at University of New Mexico in Albuquerque, NM. The SDR is developed for 1U CubeSat. It is based on the Universal Software Radio Peripheral (USRP) hardware as seen in Figure 8. The system uses the Space Plug and play Avionics (SPA) communication protocol (Oliveri, 2011). The SDR uses open source hardware and software. This system is still under development and will be flown on missions in future. This radio is of interest in this research as it shares the same hardware platform.

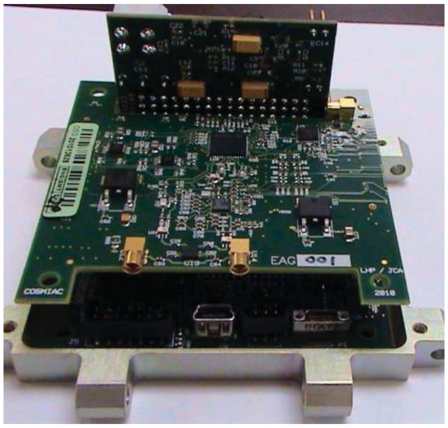


Figure 8: COSMIAC SDR board (Oliveri, 2011)

2.2 Hardware Design Options for Software Defined Radio

SDRs are implemented on a number of hardware platforms, general purpose microprocessors (GPP), digital signal processors(DSP), graphics processing units (GPU), and field programmable gate arrays (FPGAs). In this section a short description of each of these platforms is given and their applicability for SDR.

- General purpose Microprocessors (GPP): These are processors which are found in computers. Intel and AMD devices are common Microprocessors. These devices are optimized to handle the widest possible range of applications. GPP are designed for general purpose applications and therefore are flexible. GPP's processors are designed for speed and multi-purpose usefulness (Oliveri, 2011). An SDR system containing a GPP has fixed hardware computing services and peripheral interfaces. High-level languages are implemented for operations which process incoming and outgoing data (Guo et al., 2012).

Figure 9: AMD General Purpose Microprocessor (X86 CPUS' GUIDE,2010)

- Graphics processing units (GPU): These processors are designed especially for parallel architecture so that they can run vector manipulations and graphical operations. These units are excellent to create images in a frame buffer for display. The parallel structures are efficient for signal processing. The power consumption for GPU is higher than the other platforms for SDR. GPUs are manufactured by AMD and nVIDIA. SDR applications use the multi-core acceleration provided by GPUs along with the abundant parallelism functionality. The application of SDR to GPU's come with many difficulties including architectural complexity, new programming languages and different style of parallelism (Plishker et al., 2011). SDR's use GPUs for high-speed floating-point parallel arithmetic operations (Ahn et al, 2011).

Figure 10: nVIDIA GPU(TECHPOWERUP, 2015)

- Field programmable gate arrays (FPGA): FPGAs are chips that can be configured by the user after manufacture. FPGAs comprise of programmable logic components called "logic blocks" (Oliveri, 2011). FPGAs are configured using hardware description language like VHDL, verilog. Companies that manufacture FPGA’s are Xilinx and Altera.

Figure 11: Xilinx FPGA (AL Electronics, 2011)

- Digital Signal processors (DSP): These processors are designed for specialized operations. They are efficient for mathematical operations and we optimized for narrower set of applications than compared to general purpose microprocessors. Their architecture is specially designed to support the operational needs of digital signal processing. DSPs provide coding flexibility for signal processing functions and a development environment but the arithmetic operation capability does not completely support all the real-time communications operation (Ahn et al., 2011).

Figure 12: Texas Instruments DSP (AL Electronics, 2011)

- FPGA + DSP: FPGAs are important for SDR applications owing to their flexibility and real-time processing capabilities. Increasing number of DSP operations are being implemented on FPGAs including operations such as digital down and up converter, FFT correlators, pulse compressions (for radar processing). FPGAs are suitable for high-speed parallel operation. However, all DSP capabilities cannot be easily implemented on FPGAs. Floating point operations are difficult to implement on FPGAs due to the large amount of memory space needed in the device. DSP and GPP platforms are better for matrix inversion (Rudra, 2004). Therefore a platform with FPGA and DSP or GPP provides a flexible platform for SDR applications.

2.2.1 Comparison between Design Options

Several features of microelectronic platforms are examined to determine a feasible hardware solution for the implementation of SDR with a focus on nanosatellite communications. Table 7 compares the different hardware platforms for SDR.

	GPP	DSP	GPU	FPGA	FPGA+ DSP
Mass	94g (ISIS on-board computer)	Varied depending on type of DSP	930g (Nvidia GTX 480)	0.19 kg	0.20kg (Spartan-3A DSP FPGA)
Digital-Signal Processing Operations	N/A	Efficient	Efficient	Reprogrammable for specialized operations	Efficient
Operations	Efficient	Not very efficient	Not very efficient	Moderate	Moderate
Size	96 x 90 x 12.4 mm	Small (on Integrated circuit)	Large	Large	Large
Power	Moderate	Good	Poor	Moderate	Moderate
Programming Language	C, C++, Java	C, Assembly	CUDA, C	Verilog, VHDL	C, C++, Verliog and VHDL
Flexibility	High	Low	Moderate	High	High
Cost	$6042.20	$600	$500 (Nvidia GTX 480)	$295	$300 (Spartan-3A DSP FPGA)

Table 7: Comparison of hardware platforms for SDR (Oliveri, 2011)

The graph below shows the comparison of the different platforms in terms of processing power and flexibility. The ideal platform for SDR is a combination of the FPGA and

DSP. DSP part of the board is efficient to perform digital signal processing tasks and the FPGA allows the flexibility of performing other operations and tasks. The power consumption of the system is also moderate.

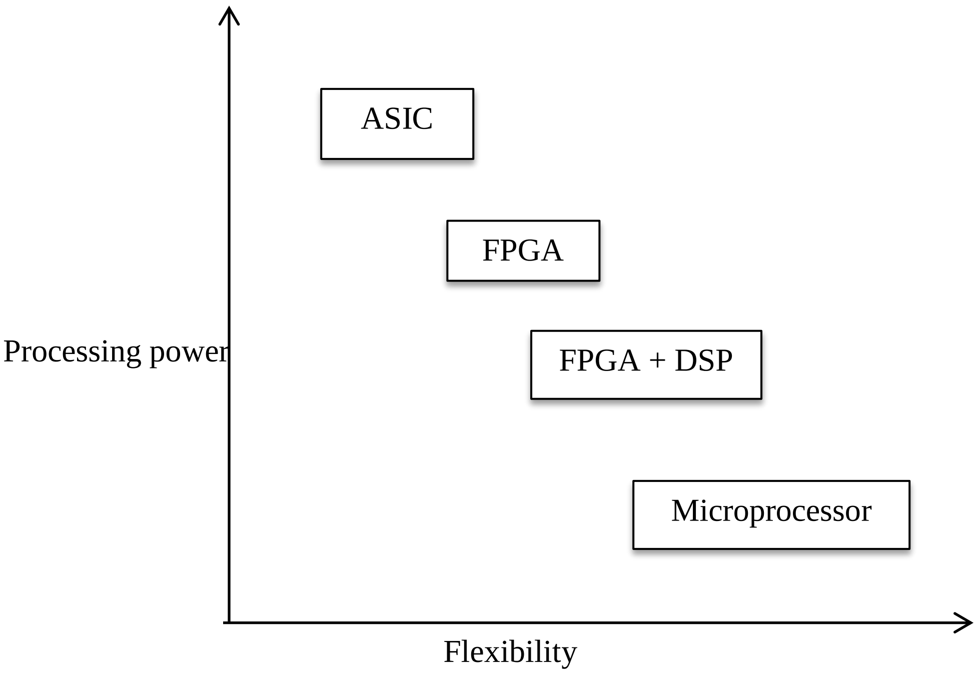


Figure 13: Comparison of Hardware solutions based on power and flexibility (Dovis et al, 2005)

For the purpose of this research communication algorithm will be designed to implement on an FPGA only. This option is chosen so as to take advantage of the flexibility of the FPGA for DSP operations required for signal processing.

2.2.2 Software Defined Radio on FPGA platform

As per the definition of SDR, the signal processing is done on a software reprogrammable element, i.e., in software. From the literature survey, it is seen that the most commonly adopted design software programmable platform is the Field Programmable Gate Array (FPGA). ASIC does have a higher processing speed and uses less power. But FPGAs have various advantages over ASIC (Application-Specific Integrated Circuit). FPGAs are reconfigurable devices which suits the main characteristic of SDR. FPGAs can be reprogrammed multiple times and allow users to define system capability as well as implement parallelization of operations. FPGAs are programmed with a hardware description language, such as Verilog or VHDL. An FPGA-based SDR use more power, but it has the advantage of flexibility and parallel operations to run simultaneously. The system is capable of parallel processing of data, multi-threaded operations and distributed computations of DSP operations. SDR applications on FPGA provide an opportunity to access and update communication system on satellites which are physically difficult to access (Oliveri, 2011). More than one digital signal processing block can be supported by the satellite with the use of an FPGA. Remote access to the firmware of the FPGA on orbit is available to upgrade or make modifications on the signal processing blocks.

There are some concepts of digital communication systems and analog communication systems which are vital to understand the design of the system discussed in Chapter 5. These concepts are discussed in section 2.3.

2.3 Digital Communications System

Digital communication is the process by which digital symbols are changed into transmittable waveforms. The key aspect of a digital communications system is modulation/demodulation system (MODEM) (Wong and Lok, 2004). Modulation is the process by which the signal carrying the digital information is converted to analog waveform before being transmitted (Wong and Lok, 2004). Demodulation is the process in which the analog signal received is converted to a digital format before being processed (Wong and Lok, 2004).

The digital modulation/demodulation techniques available are amplitude shift keying, frequency shift keying, and phase shift keying, continuous phase shift keying and the trellis-coded modulation. Figure 14 shows the different methods to modulate digital data and the variations of each of these methods.

1. ASK: Amplitude modulation of a digital data is called Amplitude Shift Keying (ASK). In this method the variation in amplitude of carrier wave is based on two or more discrete levels. In a binary message there are two levels, zero and one. The modulated binary message has bursts of sinusoid waves. The forms of ASK are Pulse Amplitude Modulation (PAM) and Quadrature Amplitude Modulation (QAM). PAM involves communication using a train of recurring pulses. The message is encoded in the form of amplitude of pulses. QAM involves the modulation of the amplitude of two waves, 90 degrees out of phase with each other (Schwartz, 1990).

2. PSK: Phase modulation of digital data is called Phase Shift Keying (PSK). In this method the phase of the carrier wave is varied. Binary phase shift keying (BPSK) is a form of PSK in which every phase used is assigned a particular binary number. Differential Phase Shift keying (DPSK) varies from basic PSK in that the change in the phases is the important factor here used to modulate / demodulate binary data. High state of PSK contains only one cycle whereas that of DPSK contains one and half cycle. Offset phase-shift keying, also called Offset quadrature phase-shift keying (OQPSK), uses four different values of the phase to transmit. The four values of the phase (two bits) at a time are used to construct a QPSK symbol which allows the phase of the signal to jump by about 180 degrees at a time (NI, 2007).
3. CPM: Continuous phase modulation (CPM) differs from coherent digital phase modulation, because the carrier phase is modulated in a continuous manner as opposed to the carrier phase resetting to zero at the start of every symbol. CPM is applied as a constant-envelope waveform (Wong and Lok, 2004).
4. Trellis-coded Modulation: Coding is a digital function and modulation is an analog function. Typically, most modulation schemes perform these functions separately. In trellis-coded modulation, modulation and coding are combined. The word trellis stands for the use of trellis (also called convolutional) codes (Benedetto et al., 1992).
5. FSK: Frequency shift keying (FSK) modulation scheme is when different frequencies are assigned to the signal (digital symbols). FSK has various

categories depending on the number of digital signals, relation between frequencies and the phase of frequencies (Wong and Lok, 2004). They are as shown in the Figure 21.

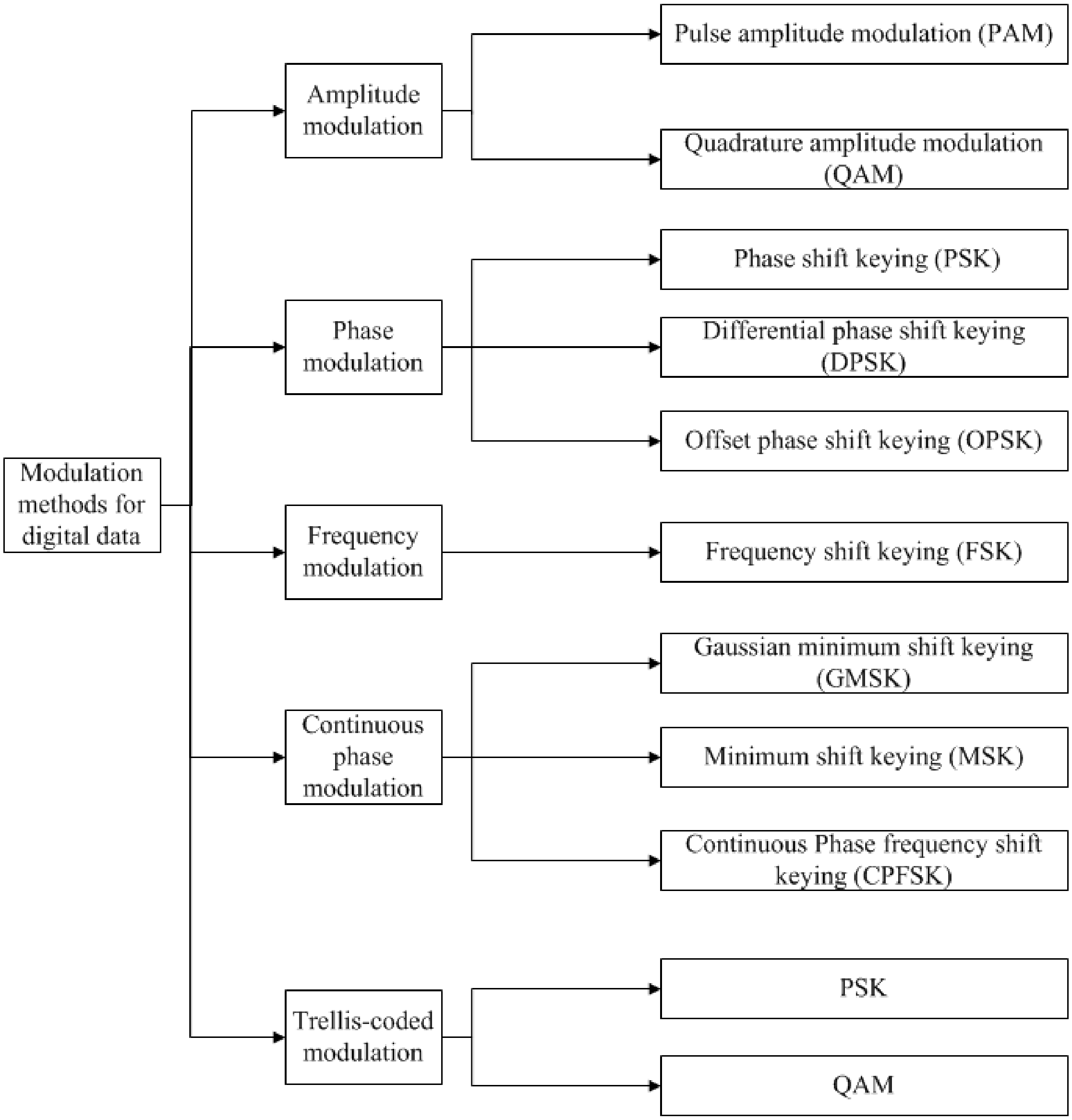

Modulation methods for digital data
Amplitude modulation
Pulse amplitude modulation (PAM)
Quadrature amplitude modulation (QAM)
Phase modulation
Phase shift keying (PSK)
Differential phase shift keying (DPSK)
Offset phase shift keying (OPSK)
Frequency modulation
Frequency shift keying (FSK)
Continuous phase modulation
Gaussian minimum shift keying (GMSK)
Minimum shift keying (MSK)
Continuous Phase frequency shift keying (CPFSK)
Trellis-coded modulation
PSK
QAM

Figure 14: Modulation methods for digital data

FSK is divided into two types Binary Frequency Shift Keying (BFSK) and M-ary Frequency Shift Keying (M-FSK). In M-FSK, the binary data stream is divided into n-tuples of n=log2M bits, i.e., we can send n bits or more than one bits at a time using one of the M signals that are possible. More than two frequencies can be considered in the particular modulation scheme. M-FSK is an orthogonal type of modulation.

In BFSK modulation, the frequency of a continuous carrier wave is shifted to one or two of discrete frequencies called “mark” frequency and the “space” frequency. The mark and space frequencies correspond to binary one and zero, respectively. Mark is the higher radio frequency corresponding to one. In frequency-shift keying, the signals transmitted are represented by:

Marks frequency (binary ones) $s_1(\mathrm{t}) = \mathrm{A}\cos(2\pi f_1 \mathrm{t} + \Phi_1),\ 0 < \mathrm{t} \leq \mathrm{T}$ (1)

Spaces (binary zeros) $s_2(\mathrm{t}) = \mathrm{A}\cos(2\pi f_2 \mathrm{t} + \Phi_2),\ 0 < \mathrm{t} \leq \mathrm{T}$ (2)

where A is the amplitude, $f_1$and f_2 are discrete frequencies, Φ_1 and Φ_2 are initial phases.

This particular system is of discontinuous phase or non-coherent, because the phase of the signal is discontinuous at the switching times and not same at any time. The signal is not continuous at bit transitions (Broendum, 1994).

BFSK can be transmitted coherently as well, which implies phase of each mark or space tone has a fixed phase relationship with respect to a reference signal phase. In this case the initial phases are the same. Non-coherent FSK is easier to generate and independent of phase changes or transitions since the two phases are different, but coherent FSK is capable of superior error performance. In the coherent case, the phase of the transmitted signal remains continuous because the phase of the tones of each symbol is based on the previous symbol phase. Coherent and non-coherent BFSK can be divided into orthogonal and non-orthogonal. Orthogonal signalling is when the inner product of the two signals s_1(t) and s_2(t) is zero (Broendum, 1994).

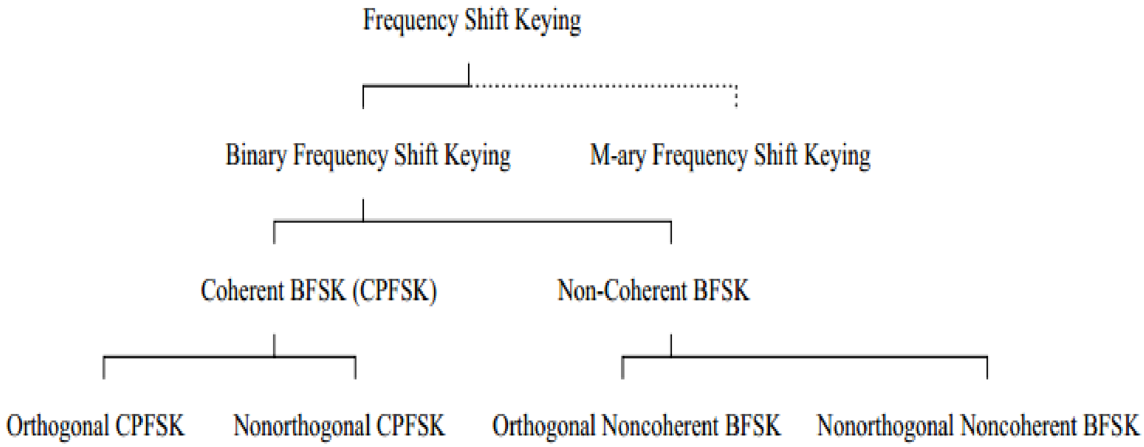

Frequency Shift Keying
Binary Frequency Shift Keying
M-ary Frequency Shift Keying
Coherent BFSK (CPFSK)
Non-Coherent BFSK
Orthogonal CPFSK
Nonorthogonal CPFSK
Orthogonal Noncoherent BFSK
Nonorthogonal Noncoherent BFSK

Figure 15: Types of Frequency Shift Keying (Broendum, 1994)

2.3.1 Frequency Baseband Modulation/Demodulation

Digital communication gives us transmittable waveforms which need to be transmitted or received. For this purpose analog transmission is required. For SDR, the theory of analog communications is used to get a complex baseband signal while transmitting and receiving a baseband signal from a complex signal.

2.3.2 Analog Communication System

In analog transmissions, angle and amplitude modulation are used to transmit data or voice over wire cable, fibre or the atmosphere. By definition, angle modulation involves varying carrier wave angle by an amount proportional to the message signal. Therefore there are two types of angle modulation: frequency modulation and phase modulation (Swiggan, 1998).

Phase modulation: The phase of the carrier signal is varied to match the instantaneous phase deviation, which is the difference between the instantaneous phase and that of the carrier signal and is linearly related to the size of the modulating signal at a given time (Swiggan, 1998).

Frequency modulation: The frequency of the carrier signal is varied to match the instantaneous frequency deviation, which is the difference between the instantaneous frequency and the carrier frequency and is linearly related to the size of the modulating signal at a given time. The FM theory illustrated in Figure 22 can be explained as follows (Swiggan, 1998):

A sinusoidal carrier is represented by:

$$x_c = A_c\cos(2\pi f_c \text{t}) \qquad (3)$$

where, f_c is the instantaneous frequency and A_c is the amplitude of carrier wave.

A baseband signal m(t) is modulated by first integrating the message m(t) with respect to time to get phase θ(t). This can be represented as:

$$\theta(t) = 2\pi f_c t + 2\pi k_f \int_0^t m(\tau)d\tau \qquad (4)$$

k_f – Frequency sensitivity, this indicates how much of the carrier spectrum the input signal should fill out. The frequency sensitivity is related to the frequency deviation by the following equation:

$$k_f = \frac{FD}{A}(2\pi * Ts) \qquad (5)$$

where, FD is the frequency deviation, A is amplitude of the modulating signal and Ts is the sampling period. Phase modulation is required after integrating the signal which consists of a quadrature modulator, which gives out a complex baseband signal (NI, 2014).

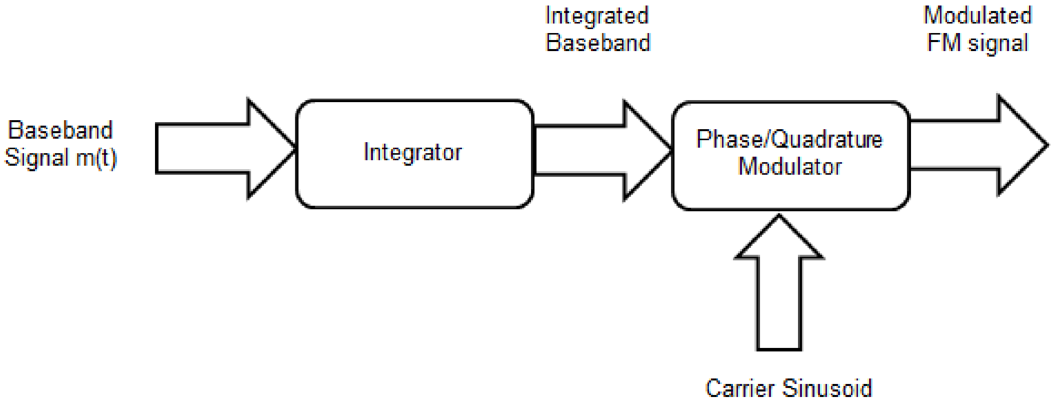


Figure 16: FM Modulation theory

In Figure 17 of the Quadrature Modulator, the I and Q components of the real signal are mixed with carrier signal and carrier signal with a 90 degrees phase offset to give an up-converted signal. The up-converted signal is a complex signal in the baseband form (NI 2014).

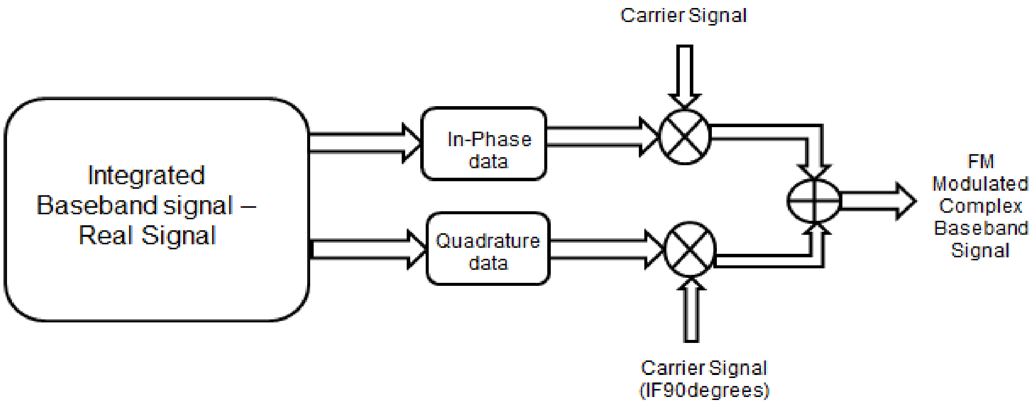

Carrier Signal
Integrated Baseband signal – Real Signal
In-Phase data
Quadrature data
FM Modulated Complex Baseband Signal
Carrier Signal (IF90degrees)

Figure 17: Quadrature modulator

2.3.3 Non-coherent BFSK modulator:

The Figure 18 shows a non-coherent FSK modulator. Conceptually, the FSK scheme involves generating the FSK signal by switching between mark f_1 and space f_2 frequencies. The initial phases of mark and space frequencies are Φ_1 and Φ_2 which are different from each other. Two oscillators generate the two frequency signals s_1 and s_2. The binary data input controls the multiplexer. The amplitude A of the signals is same for both signals (Broendum,1994).

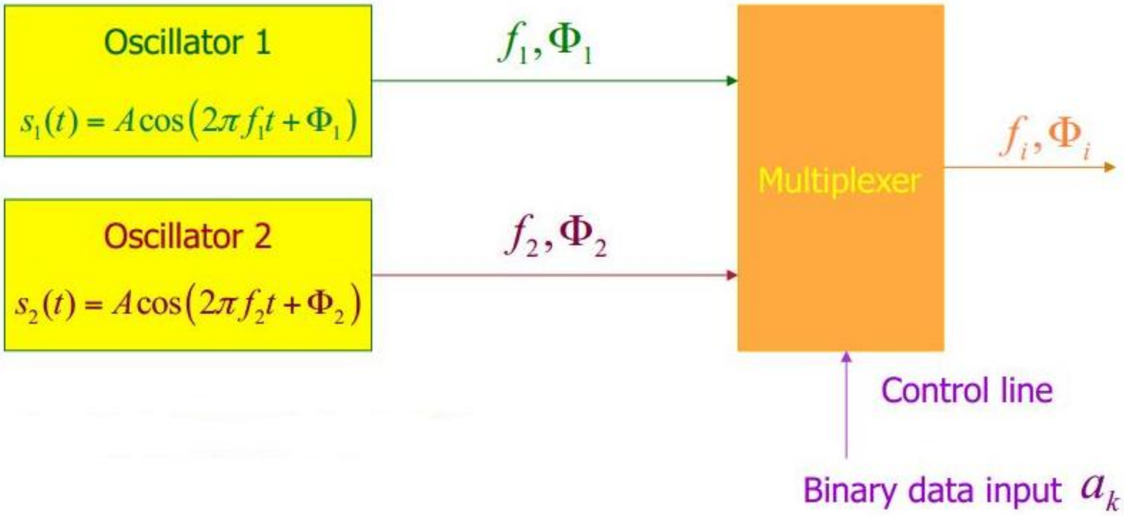


Figure 18: Non-Coherent BFSK Modulator

2.3.4 Non-coherent BFSK demodulator: Correlator Implementation

Theoretically, in a correlator implementation as shown in Figure 19 of a non-coherent BFSK demodulator receiver, the received signal r(t) is divided into in-phase and quadrature components for each frequency component by passing it through the envelope detector. The envelope detector consists of the in-phase and quadrature correlators, integrators and the squarers. Figure 24 depicts a typical non-coherent demodulator where the upper two branches are implemented to detect f_1 and the lower two to detect f_2 (Broendum, 1994).

Ideally the received signal r(t) can be written as

$$s_i\ (t,\theta) = A\cos(2\pi f_i t + \theta),\ \ i=1,2 \qquad (6)$$

$$= A\cos\theta\cos 2\pi f_i t \text{ (In-phase Component)- } A\sin\theta\sin 2\pi f_i t \text{ (Quadrature Component)}$$

If the received signal is Acos($2\pi f_1$t + θ) has a phase of zero then referring to Figure 25 the first multiplication and correlation would produce an output with the highest weight and the second one would yield zero as sin($2\pi f_1$t) is orthogonal to the signal. The third and fourth branches would also produce near-zero outputs, since their reference signals f_2 are also orthogonal to the signal component (Broendum, 1994).

Similarly, if the signal Acos($2\pi f_1$t + θ) has an unknown phase component then, referring to Figure 25, the in-phase component the signal is partially correlated with cos($2\pi f_1$t) and for the quadrature component the signal is partially correlated with sin($2\pi f_1$t). The third and fourth signals will return near-zero outputs due to orthogonality (Broendum, 1994).

After correlation and integration, the output of in-phase correlator is $\frac{AT}{2}$ cosθ and $\frac{AT}{2}$sinθ for quadrature correlators. The output is squared in each branch. The outputs of the first two branches are added and then compared with the sum of the squares of the outputs from the lower two branches (Broendum, 1994).

The received signal corresponds to f_1 or f_2 which is evaluated by the judging unit. If we consider l_1^2 as the output from the first two branches and l_2^2 as the output from the last two branches then the decision is based on the following criteria (Broendum, 1994).

If $l_1^2 > l_2^2$ then the decision is binary bit 1(mark frequency) and if $l_2^2 > l_1^2$ the decision is 0 (space frequency). This type of receiver is called quadrature receiver. In the case of the above case where received signal is considered to be Acos($2\pi f_1$t + θ) , $l_1^2 > l_2^2$ and hence the judging unit which compares the outputs of the two correlators will decide that binary bit 1 is the output (Broendum, 1994).

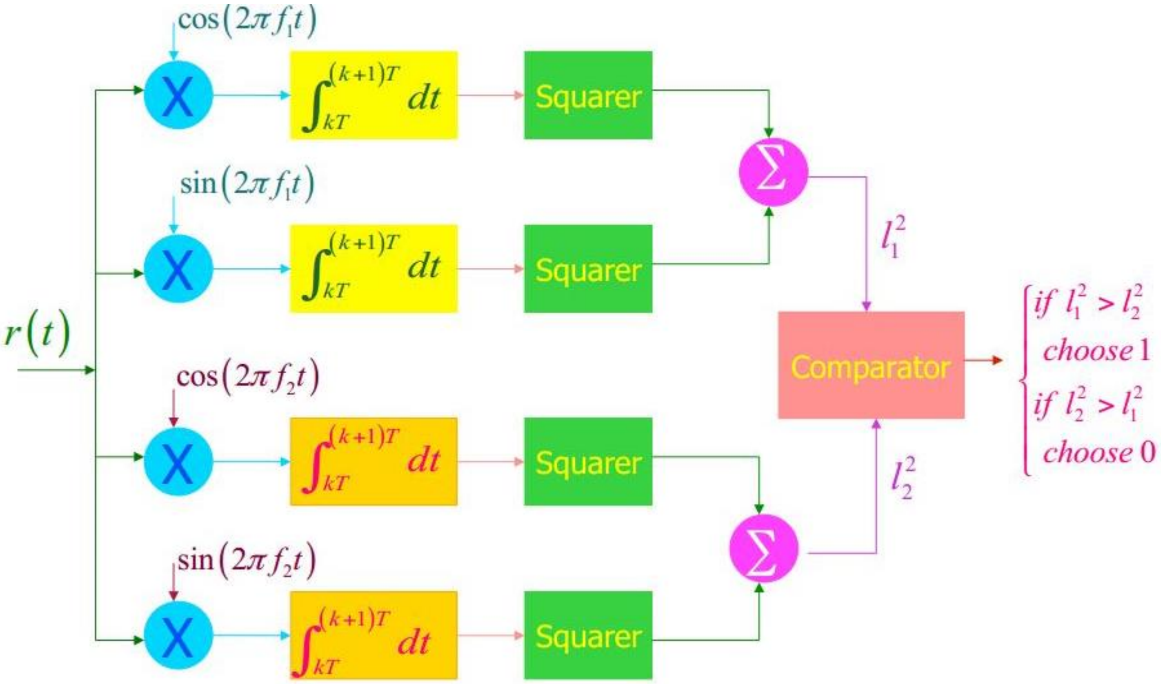


Figure 19: Non-Coherent BFSK Demodulator - Correlator Implementation

3 Nanosatellite Communications and Link Budget Analysis

In this chapter, a background on nanosatellite communications is provided. In addition, a survey on the antennas is performed and the process of data rate selection is also discussed. The various configuration aspects of nanosatellite communications system including the communication protocol and the frequency determination are discussed. The hardware trade-off study for nanosatellite communications system is also provided. The link budget analysis is performed and discussed as well. The benefits of using SDR for nanosatellites are also highlighted in brief in this chapter.

3.1 Nanosatellite Communications

A satellite communications system can be separated in two parts as shown in Figure 20; the space segment and the ground segment. Each segment has three design components; the antenna design, transceiver development, and the communication algorithm design (Crawford et al., 2009).

The ground segment for a typical nanosatellite communications system consists of the computing station to send commands to satellites and to receive data from satellites. The station is connected to the Terminal Node Controller (TNC), which is primarily a device used for radio networks that use the AX.25 Packet protocol. It consists of a microprocessor, a modem, flash memory and software to implement the protocol and also has a command line user interface. A TNC interfaces between a computer and a radio transceiver. The task of the transceiver is to modulate and transmit the analog radio signal containing the data. It also receives the signal and demodulates it. The transceiver is connected to an antenna which transmits and receives the data to and from the nanosatellite (Crawford et al., 2009). The space segment consists of the OBC (On Board computer) which processes data and receives and sends commands. Similar to the ground segment, the computer is connected to a TNC which in turn is connected to a radio transceiver. The transceiver is connected to the antenna (Crawford et al., 2009).

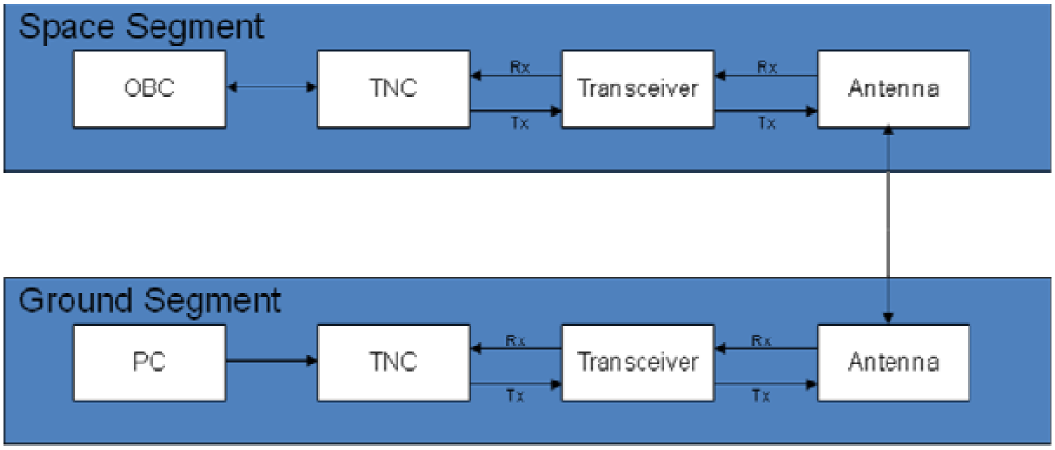


Figure 20: Communication system (Crawford et al., 2009)

The figure below is an example of a communication network for the space segment of the satellite. As mentioned earlier, a successful satellite communication system is capable of receiving information from the ground station and transmitting information to the ground station, to other satellites and also to the launch vehicle. In satellite communications, the uplink refers to the information from the ground station to the satellite and the downlink is vice-versa.

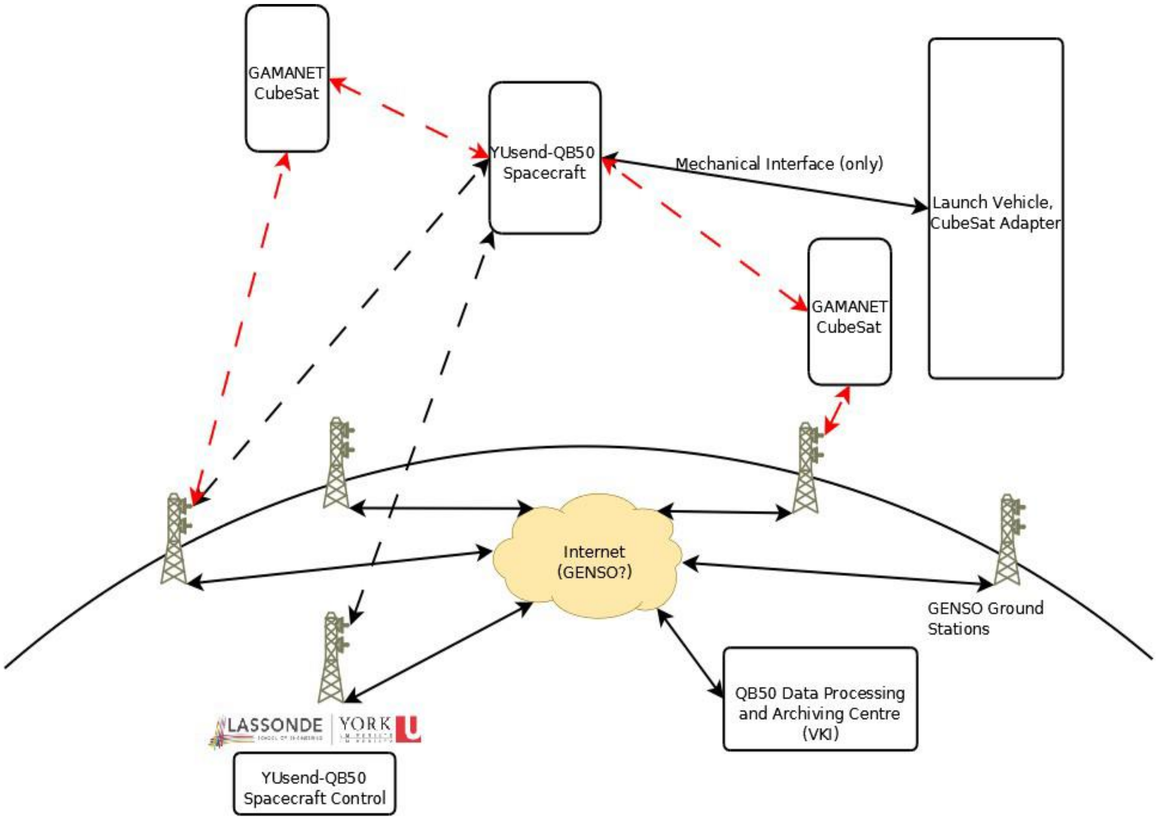

GAMANET CubeSat
YUsend-QB50 Spacecraft
Mechanical Interface (only)
Launch Vehicle, CubeSat Adapter
GAMANET CubeSat
Internet (GENSO?)
GENSO Ground Stations
QB50 Data Processing and Archiving Centre (VKI)
LASSONDE
YORK U
YUsend-QB50 Spacecraft Control

Figure 21: Satellite Communication Network (Crawford et al., 2009)

3.2 Antenna Configuration

A survey of the antenna designs used on previous nanosatellite missions is presented in Table 8. The most suitable configuration is chosen based on factors such as frequency band, impedance mismatch, antenna gain, and radiation pattern. For this research a monopole antenna is used. The monopole antenna is shorter in length, has less weight and the space for mounting the monopole is less.

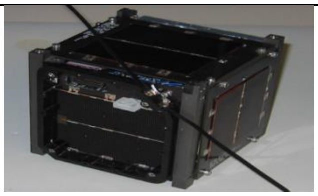
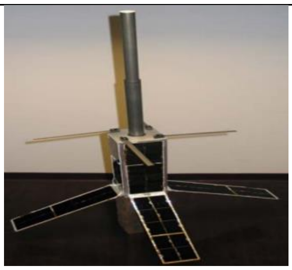
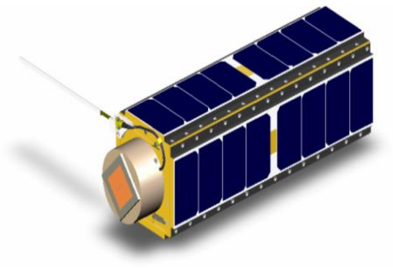
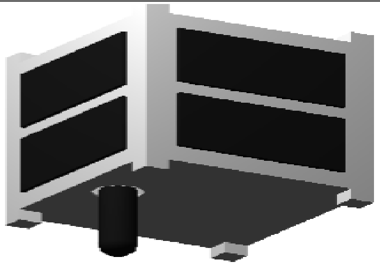
Monopole, Monopole Patch Short length, low weight, less space required, nearly omni-directional, can only transmit linear polarized waves (Mandeep, 2013)	
Dipole , Crossed dipoles, End fed dipole Omnidirectional radiation patter, built by two monopole antennas (Klofas, 2008)	
Turnstile , Canned Turnstile Also known as crossed dipole antenna, two dipole antennas in crossed configuration, transmit circular polarized signal, is an omni-directional antenna. (Klofas, 2008)	
Patch antenna ,Monopole patch Mounted on a flat surface, linearly polarized fields, can be circularly polarized, can be used as arrays of antenna. (Klofas, 2008)	
Helical(cellphone) antenna A conducting wire wound in the form of helix, circular polarization, currently being explored and researched	

Table 8: Antenna Designs (Klofas, 2008)

3.3 Communication System Configuration Aspects

Before describing SDR in detail, other aspects of a communications system that need to be considered and determined are discussed in this section. These include communication protocols, antenna configurations and frequency determination.

3.3.1 Determination of Communication Protocol

Effective communication system entails that the satellite and the ground station must use the same communication protocol. The communication packet protocol which the nanosatellite community often uses is the AX.25 Packet protocol. The protocol is particularly designed for amateur radio operators and is used in amateur radio networks. The protocol conforms to the HDLC and ANSI X3.66 (AX.25, 1998).

The small blocks of data that are sent by the data link layer are referred to as frames. There are three types of frames: information frames carry the data that has to be communicated; supervisory frames supervise the requests for retransmission of lost or corrupted data; and unnumbered frames are used to establish and terminate link connections. Each frame has several fields. We will be focussing on the information frame in the current study. The structure of the information frame is shown in Table 9 (AX.25, 1998).

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N*8 Bits	16 Bits	01111110

Table 9- Structure of information frame (AX.25, 1998)

A flag is used to identify the start and the end of the frames. The flag sequence is 0111111 in binary, which is 7E in hexadecimal. This sequence cannot appear anywhere else inside the frames. The address consists of the source of the frame, i.e., the source call number and the destination of the frame, i.e., the destination call number. The control field of the frame identifies the type of frame. PID Protocol Identifier (PID) is only for the information and unnumbered information frames. It identifies the type of layer 3 protocol. The information field is used to hold the data that have to be communicated and the size is 256 octets long. FCS (Frame-check sequence) is the field calculated by both the transmitting and receiving stations to insure that the frame was not altered during transmission. Bit stuffing is applied to the frames while transmitting. Bit stuffing is the process in which the transmitting station monitors the bit sequence for consecutive five bits to check whether the bits are ones. If five consecutive bits are found then a zero is inserted after the fifth bit. When the frame is received, any zero that follows five consecutive ones is discarded (AX.25, 1998). All fields are sent with the least significant bit first except for the FCS, which sends the most significant bit first.

3.3.2 Frequency Band Determination

Frequency bands are very important for satellite communication architecture. Different frequency bands available have different licensing requirements and applications and are listed in Table 10.

Frequency Band	Range	Applications
UHF(Ultra-high frequency) /VHF(Very-high frequency)	300 MHz to 3000 MHz 30 MHz to 300 MHz	Extensively used for small satellites, for links that requires lower data rates
L-Band	1 GHz to 2 GHz	GNSS satellite systems, telecommunication systems, military
S-Band	2 to 4 GHz	Deep space applications, geostationary orbit, LEO-applications
C-Band	4 to 8 GHZ	Terrestrial microwave radio communications, weather radars, WIFI devices
X-Band	8 to 12 GHz	Military use, coverage of remote areas of world, government and defence use
Ku-Band	12-18 GHz	Satellite communications, handle higher data rates
Ka-Band	23 to 27 GHz	Future missions

Table 10: Frequency Bands

Ultra-high frequency (UHF) / Very–high frequency (VHF), the amateur radio frequency bands do not require permission from the International Telecommunications Union for use. The stations that are utilising the UHF/VHF amateur bands should have a fully licensed amateur radio operator. The UHF band ranges from 300 MHz to 3000 MHz and the VHF band ranges from 30 MHz to 300 MHz. These frequency ranges handle low data rates. UHF can be implemented using low power which requires larger antennas, a disadvantage for small satellites (Elbert, 2008).

L-Band ranges from 1 to 2 GHz. This band is particularly used for GNSS (Global Navigation Satellite System), telecommunication systems and military (Seifu 2008).

S-Band ranges between 2 to 4 GHz. Many satellites use this frequency band for transmission, especially for deep space applications and geostationary orbit missions. Signals are transmitted with low power and therefore reception requires large antennas. The difference between S-band and the amateur frequency bands is that S-band can handle higher data rates (Seifu 2008).

C-Band ranges between 4 to 8 GHz. It was the first band established for satellite communication systems. The C-Band is the frequency range in which there is also terrestrial microwave radio communications assigned. There are a number of similar systems which are located around the world, therefore a chance of interference in this range may arise (Seifu 2008). This range is also used for WI-FI devices and some weather radars (Elbert, 2008).

X-Band ranges between 8 to 12 GHz. The band is used by military due to its advantages of being resistant to rain and interference. It handles higher data rates as compared to-UHF, VHF, L- and S-bands. It can also provide coverage to remote areas of the world. This band is specifically reserved for government and defence use. A section of the X-band is allocated for deep space communications by NASA between ground stations and deep space (Seifu, 2008).

Ku-Band band ranges between 12 to 18 GHz. It is suitable for satellite communications. It uses smaller antennas and can handle higher data rates. However, the power consumption is high and the equipment required for Ku band is expensive (Seifu, 2008).

Ka-Band ranges between 23 to 27 GHz. This band will be used in future missions as it can provide more bandwidth as compared to the other bands. The primary disadvantage of this band is the attenuation caused due to rain and moisture (Seifu, 2008).

UHF/VHF is the band which requires the least power. But it is able to handle only lower data rates. The band selected at this point for the current project is UHF due to minimum restrictions on licensing requirements, power and data rate.

3.3.3 Data Rate Selection

The data rate or baud rate is the definition of the speed of the data which is sent over a serial link. The unit of baud rate is bits-per-second (bps). Standard baud rates are 1200, 2400, 4800, 19200, 38400, 57600, 115200 and specifically 9600 bps as common baud rates for where speed is not important for the link (Bouwmeester, 2010). The data rates considered in this work are 200, 1200 and 1600 bps for testing and performance evaluation purposes. The 9600 bps data rate is the most common data rate used for nanosatellites in the UHF/VHF frequency range (Bouwmeester, 2010) and is therefore the targeted data rate for this work.

3.4 Link Budget

The link budget is a process used to establish whether a communication link is possible by considering parameters including frequency, transmitter signal power, and bandwidth and data rate. The link budget calculates the Signal-to-Noise Ratio (SNR), which is the ration of signal power and noise power (Ps/Pn) at the receiver input (Traussnig, 2007).The link budget includes all gains and losses from baseband input to baseband output. The link margin is a measure of the robustness of the link. As one moves to higher frequencies and one moves from fixed to mobile satellite systems the more difficult the challenge becomes and the higher the link margin must be set to provide reliable service. The link budget analysis is performed using a tool called AMSAT-IARU Link Budget calculator.

This link budget considers the desired bit rate of the system as 9600 bps and uses non coherent FSK modulation. The frequency band used is the UHF frequency band specifically the 437.475 MHz.

3.4.1 Uplink Command Budget

In the uplink command budget shown in table 11, the uplink path is from the ground station to the spacecraft. The value used for the power of the ground station transmitter is a typical value used by ground stations for nanosatellite communications systems working at amateur radio frequency bands. The value is based on specifications given for QB50 a nanosatellite mission currently in the development process.

With the increasing distance between the transmitter and receiver, the power of the signal decreases (all else being equal). The received signal power will be less than the noise that is received / generated by the receiver at some point and a communication link will not be possible. The major reason for the decrease in signal power received is the loss due to the propagation distance known as the path loss or free space loss. In the uplink path from the ground station to the spacecraft the different losses considered are pointing loss, polarization loss, path loss, atmospheric losses, ionospheric losses, rain losses. At the receiver level in the spacecraft, the antenna pointing loss, the antenna gain and the antenna transmission line losses are also taken into account. The values considered in this analysis are typical for a monopole antenna. Also at the transmitter level at the ground station, the transmission line losses and the antenna gain of the transmitter is considered. The ground station antenna considered for this analysis is a Yagi antenna. The link budget calculator provides the values for the losses pertaining to Yagi antenna.

For a desired system data rate of 9600 bps the Energy per bit to Noise Power Density Ratio which is equivalent to the “Signal-to-Noise Ratio is calculated to be 35.. The parameter is the measure of performance for the Uplink from the ground station. The system link margin is calculated to be 20.7 db from the Signal to noise ratio for the uplink. Typically for a low cost system the link margin should be around 10 db

Parameter	Value	Units
Ground Station		
Ground Station Transmitter Power Output	40.0 16.0 46.0	Watts dBW dBm
Ground Station Total Transmission Line Losses	3.4	dB
Antenna Gain	21.4	dBi
Ground Station EIRP	34.0	dBW
Uplink Path		
Ground Station Antenna Pointing Loss	1.0	dB

Gnd-to-S/C Antenna Polarization Losses	0.2	dB
Path Loss:	154.8	dB
Atmospheric Losses	2.1	dB
Ionospheric Losses	0.4	dB
Rain Losses	0.0	dB
Isotropic Signal Level at Spacecraft	-124.5	dBW
Spacecraft (Eb/No Method):--------------Eb/No Method-----------------------		
Spacecraft Antenna Pointing Loss	4.7	dB
Space Antenna Gain	2.2	dBi
Spacecraft Total Transmission Line Losses	1.9	dB
Spacecraft Effective Noise Temperature	268	K
Spacecraft Figure of Merit (G/T)	-24.1	dB/K
S/C Signal-to-Noise Power Density (S/No)	75.3	dBHz
System Desired Data Rate	9600 39.8	Bps dBHz
Command System Eb/No	35.5	dB
Demodulation Method Selected	Non-Coherent FSK	
Forward Error Correction Coding Used	None	
System Allowed or Specified Bit-error-rate	1.0E-05	
Demodulator implementation Loss	1.0	dB
Telemetry Systerm Required Eb/N0	13.8	dB

Eb/No Threshold	14.8	dB
System Link Margin	**20.7**	**dB**

Table 11: Uplink Command Budget

3.4.2 Downlink Telemetry Budget

In the downlink telemetry budget shown in table 12, the downlink path is from the spacecraft to the ground station. The value used for the power of the transmitter of the communications system is of the USRP which is used for this research. Here it is specified at 9.0 W. In the downlink path, the losses considered are similar to the uplink path which includes pointing loss, polarization loss, path loss, atmospheric losses, Ionospheric losses, rain losses. At the receiver level in the ground station the antenna pointing loss, the antenna and the antenna transmission line losses are for an Yagi antenna. For the Transmitter the antenna for the pointing and polarized losses is specified as the monopole antenna which is the antenna selected for this research.

For a desired system data rate of 9600 bps the Energy per bit to Noise Power Density Ratio which is equivalent to the "Signal-to-Noise Ratio is calculated to be 24.5. The parameter is the measure of performance for the downlink from the spacecraft. The system link margin is calculated to be 10.7 db from the Signal to noise ratio for the downlink which is very close to the 10 db margin for a low cost system.

Parameter	Value	Units
Spacecraft		
Spacecraft Transmitter Power Output	9.0 9.5 39.5	Watts dBW dBm
Spacecraft Total Transmission Line Losses	2.2	dB
Spacecraft Antenna Gain	2.2	dBi
Spacecraft EIRP	34.0	dBW
Downlink Path		
Spacecraft Antenna Pointing Loss	6.8	dB
Gnd-to-S/C Antenna Polarization Losses	0.2	dB
Path Loss:	154.8	dB
Atmospheric Losses	2.1	dB
Ionospheric Losses	0.4	dB
Rain Losses	0.0	dB
Isotropic Signal Level at Ground Station	-154.7	dBW
Ground Station (Eb/No Method):--------------Eb/No Method-----------------------		
Ground Station Antenna Pointing Loss	0.5	dB
Ground Station Antenna Gain	21.4	dBi
Ground Station Total Transmission Line Losses	3.7	dB
Ground Station Effective Noise Temperature	466	K

Ground Station Figure of Merit (G/T)	-9.1	dB/K
G.S. Signal-to-Noise Power Density (S/No)	64.4	dBHz
System Desired Data Rate	9600 39.8	Bps dBHz
Telemetry System Eb/No for the Downlink	24.5	dB
Demodulation Method Selected	Non-Coherent FSK	
Forward Error Correction Coding Used	None	
System Allowed or Specified Bit-error-rate	1.0E-05	
Demodulator implementation Loss	0	dB
Telemetry Systerm Required Eb/N0	13.8	dB
Eb/No Threshold	13.8	dB
System Link Margin	**10.7**	**dB**

Table 12: Downlink Command Budget

3.5 Nanosatellite Communications Hardware Trade-off Study

In considering a communication system for nanosatellite, there are three possibilities to be examined, Commercial-of-the-shelf transceivers, modified commercial options/ customized transceivers (Klofas, 2008) and SDR.

3.5.1 Commercial Options

A number of manufacturers have space rated transceivers available for satellite communication applications. These are usually too big for nanosatellites and are very expensive. Due to the popularity of nanosatellites with universities and small scale missions, some companies have started manufacturing communication transceivers especially for nanosatellites. These systems require specialized hardware to implement specific functions. Systems are built with only the hardware necessary for the defined tasks that they are designed for (Klofas, 2008). Hence they are hard to modify or upgrade. Example transceivers include KatySat (PC/104), NanoCom U482C UHF Half-duplex Transceiver, MHX2420 and ISIS Full duplex transceiver; the last two are shown in Figure 22 and 23. These transceivers tend to be expensive for low budget nanosatellite missions. For example, the NanoCom U482C UHF Half-duplex transceiver costs € 8,000.00 and the MHX2420 from Microhard Systems costs $10,000.

Figure 22: NanoCom U482C (CubeSatShop.com,2006)

Figure 23:MHX 2420 (CubeSatShop.com,2006)

3.5.2 Modified Commercial Options / Customized Transceivers

Some COTS systems are unsuitable for use in space since they are designed for use on Earth. These systems could experience problems such as active thermal dissipation since there is no air for the cooling fans in space for the amplifier (Klofas, 2008). Some modifications are made to these transceivers to function in space, for example, removing the packaging to reduce mass and size, increasing transmit power, etc.

Some missions, usually developed by universities build transceivers out of individual components. Such customized systems allow for low development costs, gives hands on experience to students, and allows for control of specifications and requirements. These systems usually consist of a simple TNC combined with a small hand held transceiver (Crawford, 2009). A research group at York University considered a PacComm PicoPAcket TNC shown in Figure 24 and the Yaesu VX-3R handheld transceiver which had its package removed for the communication system, which was assembled to operate in the UHF/VHF frequency. Initial testing of the system showed that the data link was severally limited and thermal control was required for long term functionality in vacuum. The QuakeSat-1 group assembled a custom built-in communications system with a Tekk KS-960 and BayPac BP-96A TNC, which achieved a 422 MB download at 2 W. But the KS-960 appears to be no longer available for purchase and the power consumption is too high for a nanosatellite mission (Crawford, 2009).

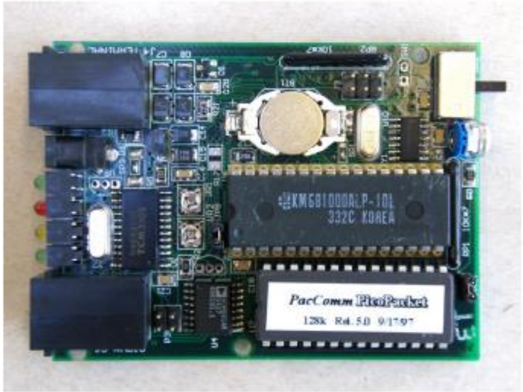


Figure 24: PacComm PicoPacket TNC (Crawford, 2009)

3.5.3 Software Defined Radios (SDRs)

The third option is an SDR. A background of how SDR technology works was given in the section 2.2. These receivers implement all their baseband processing in software. SDR receivers are gaining popularity due to their low cost, flexible functionality, and adaptability to nanosatellites due to reduced size. There are some commercial SDR's which have been optimized for nanosatellite applications. Vulcan Wireless has developed the CubeSat Software Defined Radio (Vulcan wireless, 2010) illustrated in figure 25, which gives access to a number of communication protocols and a data rate of up to 10 Mbps for S-Band and also the Micro Blackbox Transponder (Vulcan wireless, 2010) with similar features. However, these receivers do not use open source hardware and software, therefore they are not modifiable by researchers for different applications (Oliveri, 2011).

Figure 25: CubeSat Software Defined Radio (Vulcan Wireless Inc)

3.6 SDR for Nanosatellite Communications System

SDR concept provides nanosatellite communications the potential for replacing the bulky and expensive hardware required in an ASIC based system. This characteristic is of advantage to a nanosatellite missions since it faces structural challenges as discussed in section 1.2. Albeit being slower in processing capability and consuming more power as compared to an ASIC based system, an SDR system allows nanosatellites to customize the system to adapt to the needs of the user. Different modulation schemes and different protocols can also be implemented. SDR can also be used for different subsystems on a nanosatellite to eliminate the use of certain hardware thus addressing the structural challenge as mentioned earlier. These subsystems include power, camera interface, attitude control systems, sensor interface and also as a processing unit.

This research implements a singular modulation scheme to test the feasibility of an SDR system for nanosatellite communications. It is implemented on a hardware which can be further enhanced to work on a nanosatellite. The research is a secondary step in the whole process of a developing a nanosatellite software defined radio communication system making it unique in the use of SDR for this application.

4 Hardware Test Platform for Implementation of Software Defined Transceiver Algorithm

In Chapter 2, a background on the several platforms available for the implementation of SDR is described. In Chapter 4, the various FPGA platforms are examined to select a feasible hardware platform for the development of SDR for nanosatellites.

A number of hardware options have been developed for the implementation of SDRs by universities and researchers. The most significant ones are listed below.

- USRP1: USRP is one of the most popular platforms for SDR. The first USRP system, released in 2004, was a USB connected to a computer with a low-performance FPGA. USRP1 released in 2005 uses the Cyclone EPIC12 FPGA GPP. USRPs are developed by Ettus Research LTD. Newer USRPs with advanced technologies have been developed and are introduced in a later section.
- Lyrtech SFF SDR: This development platform shown in Figure 26, released in 2006, uses advanced FPGA technology (Xilinx Virtex-4), a low-power general-purpose processor (TI MSP430), and multiple RF frontends to create an advanced development platform but at a high cost of $9900 (MOUSER ELECTRONICS, 2006)

Figure 4: Lyrtech SFF SDR (MOUSER ELECTRONICS, 2006)

- Berkeley BEE2: The Berkeley BEE2 platform released in 2007 is based on the Berkeley Emulation Engine. The platform contains five high-powered Xilinx Virtex2 FPGAs and connects up to eighteen daughter-boards (Oliveri, 2011). BEE2 was released in 2007.
- Kansas U.Agile Radio: The KUAR platform is designed to be a low-cost experimental platform targeted at the frequency range 5.25 to 5.85 GHz. The platform includes an embedded 1.4 GHz general purpose processor and Xilinx Virtex2 FPGA (Minden et al., 2007) as illustrated in Figure 27. All the processing is implemented on the platform. It was released in 2007.

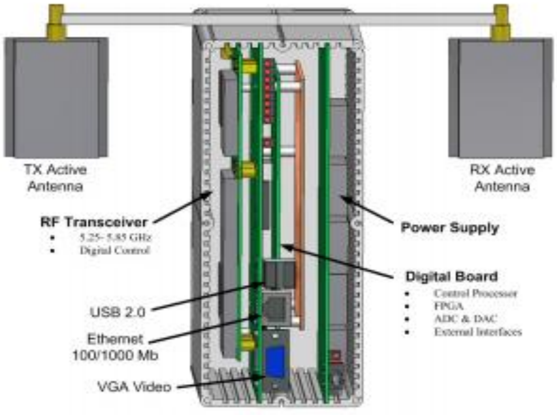


Figure 5: KUAR Radio (Minden et al., 2007)

- Rice University WARP: The developmental platform released in 2008 and developed by Rice University is called the Wireless Open-Access Research Platform (WARP). Figure 28 shows that a Xilinx Virtex 2 FPGA is incorporated in the platform.

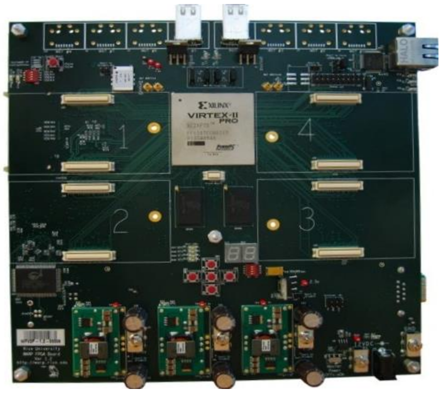


Figure 6: PCB built around a Xilinx XC2VP70 Virtex-II Pro FPGA

- NICT: The Japanese National Institute of Information and Communications Technology (NICT) constructed a SDR platform to test next generation mobile networks in 2011. The platform has two embedded processors, four Xilinx Virtex2 FPGA's as shown in Figure 29. The signal processing is partitioned between the CPU and the FPGA, with the CPU taking responsibility for the higher layers (Harada, 2005).

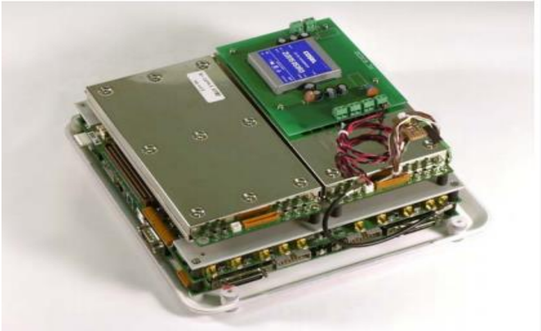


Figure 7: Platform from NICT (Harada, 2005)

Table 13 contains the comparison of FPGA hardware platforms available for SDR implementation.

System	Processor	Released
USRP1	Cyclone EPIC12 FPGA GPP (off board)	2005
Lyrtech SFF SDR	Virtex-4 FPGA MSP430 Microprocessor TI DM6446 DSP	2006
Berkely BEE2	5 Virtex-2 Pro FPGA	2007
Kansas U. Agile Radio	Virtex-2 Pro FPGA Pentium M Microprocessor	2007
Rice University WARP	Virtex-2 Pro FPGA	2008
USRP N210	Spartan 3A-DSP 2400 FPGA	2010
NICT	4 Xilinx Virtex2 FPGA	2011

Table 13: Comparison of FPGA hardware platforms for SDR

4.1 FPGA-Based Hardware

Different FPGA-based platforms examined are: FPGAs, FPGA + DSP, FPGA Development Board and FPGA + hard processors. These platforms are compared based on their physical specifications, cost, adaptability and feasibility for development of a nanosatellite based communication system.

4.1.1 FPGA

Currently, the FPGAs available in the market are from Xilinx, Altera, Lattice and Actel. Xilinx and Altera occupy majority of the market for FPGAs. The different FPGAs are compared based on certain criteria, which are important to researchers developing FPGA based platforms.

4.1.1.1 Digital Signal Processing (DSP) capabilities

Most applications requiring an FPGA use digital signal processing. DSP applications require faster computations. In order to reduce the computation time and to increase efficiency, computations are executed in parallel. Due to flexibility of FPGA structures DSP operations are suitable to be implemented on them. Currently, specially designed DSP processors have been developed and integrated with FPGAs to provide an excellent platform for signal processing applications like Global Navigation Satellite System (GNSS) software receivers. The DSP processors on FPGAs efficiently host algorithms like filtering, compression, FFT and modulation / demodulation. Both Altera and Xilinx provide FPGAs which aid in DSP operations (Šćekić, 2005).

4.1.1.2 Design Implementation

Design implementation allows the Place and Route stage which comprises taking the design using Register-transfer level (RTL) netlist. A netlist describes circuit connectivity. A single netlist consists of a list of connectors; instance, a list of signals for each instance, and also contains the information about attribute; and mapping the logic on the FPGA architecture like LCs (Liquid crystal) and I/O blocks, etc. Once a suitable location is found, the pins are assigned. The tools required for this process are manufactured by the FPGA manufacturers. A configuration file is generated from this process which is then loaded onto the FPGA (Šćekić, 2005). Xilinx, as well as Altera, provide integrated software development tools. Altera's development tool is called Quartus II, and that from Xilinx is called ISE.

4.1.1.3 Development Purpose

Developers are providing development boards with different units installed on it for development purposes. Both Altera and Xilinx provide these boards. Advantages and disadvantages of these boards are discussed in the section 4.1.2.

4.1.1.4 Processors on FPGA

Xilinx offers its own IP microprocessors: 8-bit PicoBlaze, and 32-bit MicroBlaze. Altera offers embedded soft-core RISC 16/32-bit processors Nios/NiosII. These are classified as soft processors embedded in the FPGA. Soft processors are built using FPGAs general-purpose logic (Fletcher, 2005). It is described in a Hardware Description Language or netlist. A soft processor must be synthesized and fit into the FPGA fabric. Other criteria that are considered for comparison between FPGA platforms are fabrication process, logic density, clock management, On-chip memory, I/O compatibility.

4.1.2 FPGA Development Board

FPGA development boards provide a high quality and easy to implement design environment for research and design purposes. A number of kits and software are provided to simplify the design process and reduce time to market. They are easier to interface with PCs and workstations; once the design process has been completed it is easy to port the code to a standalone FPGA. FPGA development boards allow beginners to adapt to FPGA environments as well. FPGA development boards can have FPGAs with its embedded soft processor only, FPGA + DSP or FPGA + an exclusive hard processor. Altera and Xilinx provide several development boards. Some Altera boards Cyclone V SoC Development Kit Cyclone III LS FPGA Development Kit Arria II GX FPGA Development Kit, 6G Edition PROC2S Small Form Factor Prototyping System, Cyclone III FPGA Starter Kit, etc. Examples of Xilinx Development Boards are Avnet Spartan-6 LX9 MicroBoard, Atlys Spartan-6 FPGA Development Kit, Spartan-6 FPGA SP605 Evaluation Kit as illustrated in Figure 30, Spartan-6 FPGA Connectivity Kit, etc.

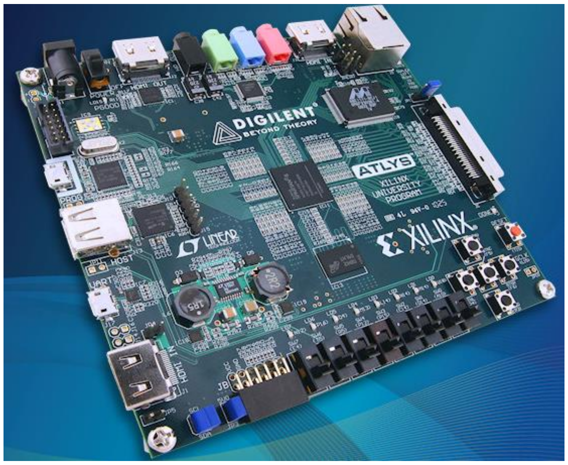


Figure 8: Atlys Spartan-6 FPGA Development Kit (Digilent)

4.1.3 FPGA + DSP

As mentioned in the previous sections, manufacturers are developing DSP units integrated on FPGA boards. Specialized FPGA boards and development boards are available for this purpose. Some of the hardware available are Xtreme DSP Development Platform — Spartan-3A DSP 3400A Edition, Avnet Spartan-6 FPGA DSP Kit, Avnet Kintex-7 FPGA DSP Kit with High-Speed Analog. These boards are efficient to carry out applications that require DSP operations and also allow the added flexibility of implementing any other application on the FPGA. For example, communication system which requires DSP operations along with, e.g., radio occultation or remote sensing operations.

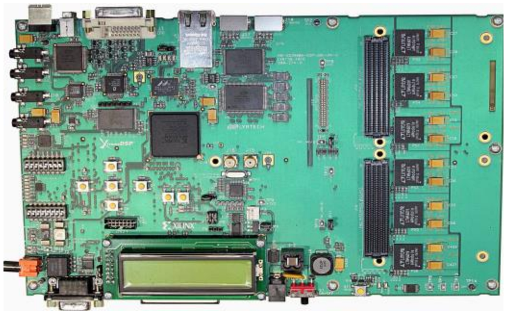


Figure 9: Spartan-3A DSP 3400A Edition (Xilinx)

4.1.4 FPGA + hard processor

Xilinx and Altera also produce FPGA families that have hard processors embedded in the FPGA. Hard processors are physical processors built from dedicated silicon. These processors have faster processing speeds since they are not limited by fabric speed (Fletcher, 2005). Therefore they are suitable for applications which require higher processing speeds like GNSS software receivers. Hard processors are fixed and cannot be modified. It can still take advantage of custom logic in FPGA by fabric speed (Fletcher, 2005). Examples of such processors are ARM922T^{TM} inside the Altera Excalibur family and the PowerPC^{TM} 405 inside the Xilinx Virtex-II Pro and Virtex-4 families.

4.1.5 Universal Software Radio Peripheral (USRP)

Ettus Research LLC produces the USRP series which include USRPE (Embedded Series), USRPB (Bus series), USRPX (X series) and the USRPN (Networked Series) series (Ettus Research, 2014). These platforms use cheaper hardware at low power.

USRPB series replaces the first USRP1 series. It uses the Spartan6 FPGA as illustrated in Figure 32. It is apt for experimentation of basic low cost SDR applications. It uses USB to interface with a PC. This series consists of the USRP B210 and USRP B200 boards.

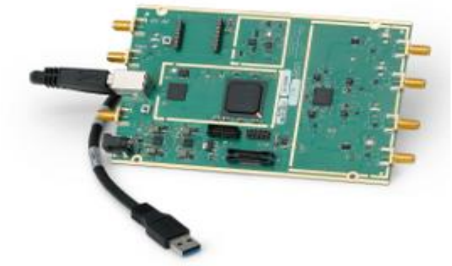


Figure 10: USRP B210 (Ettus Research, 2014)

USRPX series has improved on the USRP N series. It uses faster Xilinx Kintex-7 FPGA as illustrated in Figure 33, for high performance DSP unit. It is a high performance SDR platform and is compatible with a large number of supported development frameworks, reference architectures and open source projects.

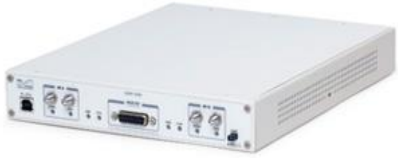


Figure 11: USRP X300 (Ettus Research, 2014)

USRPN and USRP E series are the second generation platform released in September 2008. Both series includes a Xilinx Spartan 3A-DSP 3400 FPGA device. The USRP uses open source hardware and software licenses, making them ideal for academic environments. They allow for the implementation of flexible and powerful software radios. USRP N210 as illustrated in Figure 34, allows for custom FPGA functionality to implement baseband signal processing modules. USRPN210 can be used with host PC to develop the software radio. Developing with a host-based platform typically involves less risk and will require less effort to optimize various pieces of the software radio. The code can be then ported easily to USRP E100/E100, which is appropriate as a standalone system.

Figure 12: USRP N210 (Ettus Research, 2014)

The USRP E100/E110 as illustrated in Figure 35 is ideal for applications that require mobile transceivers. The next step would also be to use the processor on the FPGA to act as a host, therefore, making the system a modular and standalone without the requirement of any other host.

Figure 13: USRP E100 (Ettus Research, 2014)

4.1.6 Hardware options for Nanosatellites Communication System

Table 14 contains the comparison between different FPGA based development platforms for SDR applications. The comparison is between FPGA, FPGA Development Board and USRP.

	FPGA (Spartan-3A DSP FPGA)	Development Board (Atlys Spartan-FPGA Development Board)	USRP
Size	95.89 mm x 90.17 mm x 15.24 mm	27.94 cm x 22.86 cm x 2 cm	22x16x5 (with outer box) 14 cm x 14 cm X 5cm (FPGA + daughter board)
Mass	Approximately 0.20 kg	Quite heavy	1.2 kg with outer box)
Power	1.5W @ 5 VDC	19 W	Maximum of 9 W
Cost	$300	$419	$2140
Integration with RF front end	Have to connect to a different front end	Need a front end	integrated with front end
Open Ended software	Xilinx ISE development	Xilinx ISE development, Webpack, Chipscope.	GnuRadio, Matlab Simulink
Development purposes	Slightly harder.	Easier for development	Easier for development
Standalone System.	Yes	No	Yes

Table 14: Comparison between different development platforms for SDR development

Considering the size 95.89 mm x 90.17 mm x 15.24 mm of the FPGA; it is also able to fit on the small satellite. The size development board is quite big with a size of 27.94 cm x

22.86 cm x 2 cm, since it has various features which aid in development of a design. Therefore this cannot be flown on a nanosatellite. The USRP albeit not being as small as the FPGA can still fit on the nanosatellite. Similarly, for the mass the development board is too large, but the FPGA and USRP are able to fit on nanosatellite. A front end has to be purchased and then integrated with the FPGA and the development board, but the USRP already has a front integrated, connected and working with the code making it easier for the developer to focus on the design. For development purposes, testing the FPGA is harder. Development boards provide features such as LCD displays, audio inputs and outputs to test operations. USRP is good for development purposes as well as it uses open source tools like Simulink, etc; which also aid in testing. An FPGA by itself and an USRP can be a standalone system, i.e., it does not require a PC or host connected to it all the time. A development board requires to be connected to the host at all times. The USRP N210 is chosen as a platform for this research as it is an excellent platform to develop a SDR based communication algorithm.

4.2 Universal Software Radio Peripheral (USRP)

The architecture of USRPN210 is illustrated in Figure 36. This architecture is typical for a second generation platform. The USRP N210 consists of a motherboard which contains four Analog to Digital Converters (ADCs), Digital to Analog Converters (DACs), a Spartan 3A-DSP 3400 FPGA from Xilinx. It supports four daughter boards, two for receiving and two for transmitting, on which the RF front ends are implemented. The Gigabit Ethernet interface serves as the connection between the USRPN210 and the host computer.

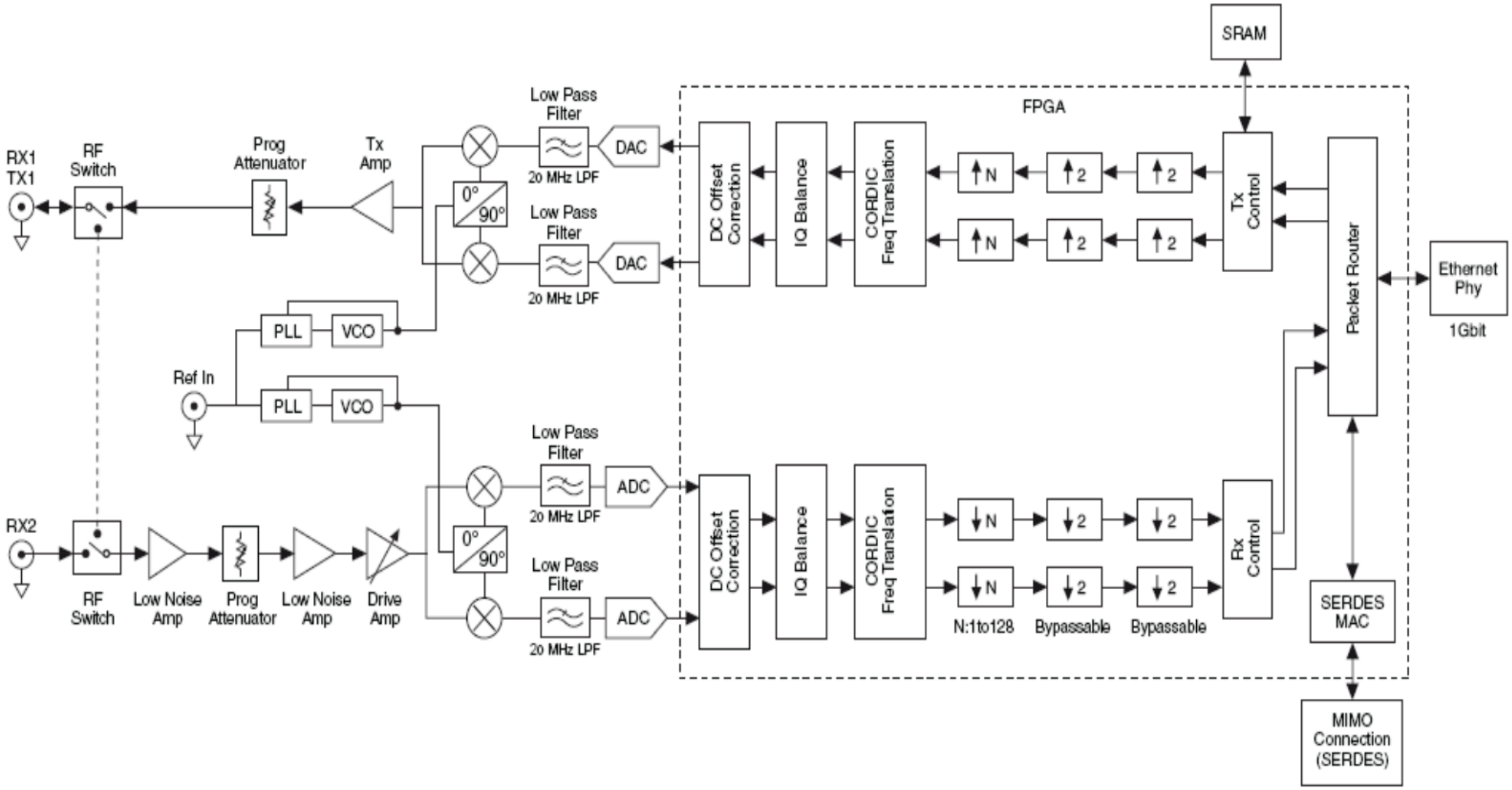


Figure 14: Architecture of USRP N210 (Malsbury and Ettus, 2013)

The USRP Hardware Driver™ is the official driver for all Ettus Research products. The RF front-ends convert the analog signal to digital and vice versa. The FPGA has digital signal processing modules already programmed on the USRP N210. Figure 37 illustrates the modules implemented on USRP N210. On the receiver side, the FPGA section has the frequency translation from Intermediate Frequency (IF) to baseband of the digitized signal. The FPGA also has the digital down-converter module, which allows for the samples to be down-sampled, so that they can be sent to the host PC over the Ethernet interface. The receiver control module adds timing information to the samples and checks them against the sampling rate. The samples are then packed in frames which are then send to the host via the packet router. Similarly, on the transmitter side, the data are sent from the host to the transmitter control via the packet router. The transmitter control uses a FIFO buffer which is placed in an external RAM chips. The packets are decoded from here and also checked to ensure that the frames sample data and timing is correct. These samples are then sent to the digital up converter where up-sampling occurs. The frequency translation module implements the conversion from baseband to intermediate frequency. This module sends the data to the front end where the signal is converted to analog and then transmitted through the antenna (Enevoldsen et al., 2011)

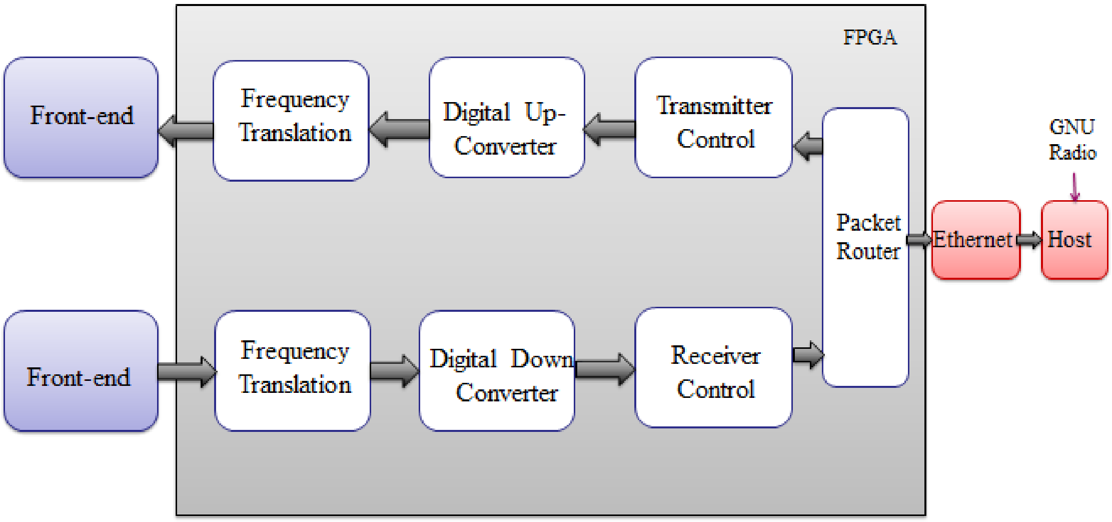


Figure 15: Modules of USRP N210

4.2.1 Hardware Platform Configuration using GnuRadio

The USRPN210 platform also consists of the GNU Radio software and the RF front ends. The USRP is integrated with the GNU radio which runs on Windows and Linux to be used as a Radio. The GNU Radio software allows for the compression/decompression, encoding/decoding, modulation/demodulation for the purpose of signal processing of the data (Dabcevic, 2011). The GNU Radio uses a signal source while transmitting and a signal sink while receiving. The GNU Radio framework allows for simple implementation of powerful signal processing systems (Oliveri, 2011). The disadvantage of this interface is that it requires host computers for design implementation. These computers cannot fit on a nanosatellite. Since a GNU radio interface cannot be utilized, there is a need for the digital signal processing to be implemented on the USRP itself.

4.2.2 Customization of USRP

The USRP needs to be customized to eliminate the use of the GnuRadio, It is also possible to use the inbuilt signal processing blocks to create receiver and transmitter. Since the USRP allows for customization of the FPGA code different modules, which do the digital signal processing are added on the FPGA of the USRP (Dabcevic, 2011). As seen in the Figure 38, the Custom RX- Receiver and TX-Transmitter modules highlighted in are implemented on the USRP. These modules involve the digital signal processing functions necessary to receive and transmit data. Chapter 4 discusses the architecture of the transmitter and receiver design implemented in these customized modules.

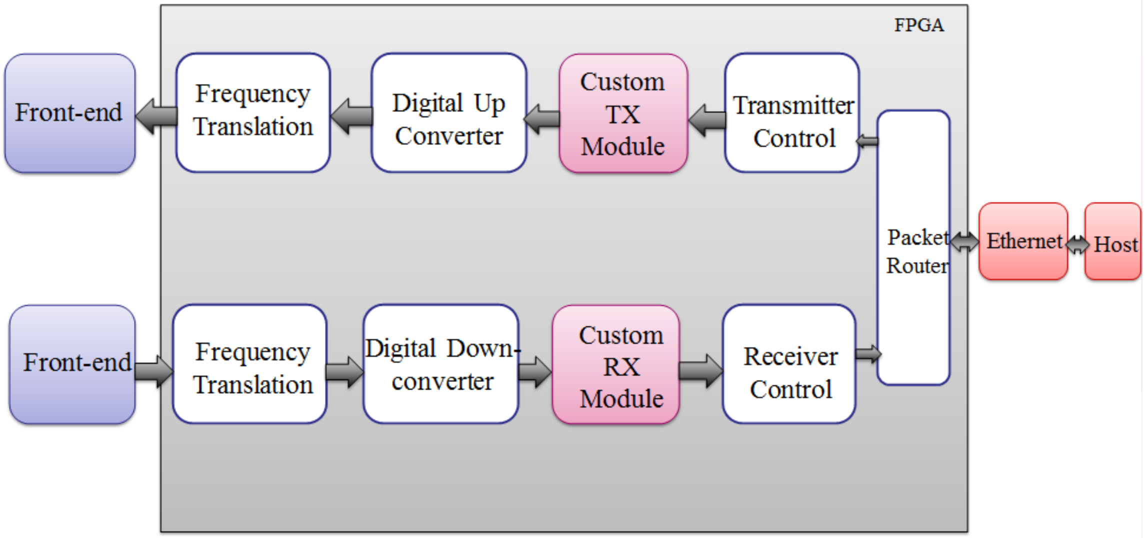


Figure 16: Customization of USRPN210

4.2.3 Software Toolbox for FPGA Implementation

Earlier sections discussed that the USRP has the flexibility of modifying the structure that it has already in-built. A common method to customize the RX signal path and TX signal path is to use the Xilinx ISE software directly, provided by the Xilinx FPGA manufacturers. The software can be implemented on the Linux based systems.

An alternate method is to use Communications System Toolbox Support Package for USRP Radio. This package is a platform for SDR applications. This platform uses Matlab and Simulink. It helps to design and test SDR systems. It also uses HDL coder for Targeting FPGA with USRP hardware. It has functions to connect Simulink to the USRP UHD driver. The package is integrated with Xilinx ISE Software using System Generator (MathWorks).

System Generator tool is used to design high performance DSP Systems. It integrates with the software's RTL and MATLAB and hardware components of a DSP system. System Generator provides system modelling and automatic code generation using Simulink and Matlab. The tool allows the generation of HDL code for Xilinx FPGA's using Xilinx-specific blocks in Simulink.

Simulink is an environment which uses block diagrams for simulation and model-based design. It is also integrated with Matlab which allows using Matlab functions and Matlab algorithms. Simulink is used extensively in control theory and digital signal processing.

5 Implementation of Software Defined Transceiver Design

A platform suited for a nanosatellite specifically a 1U CubeSat was built by Oliveri (2011) called the COSMIAC system. The next research step is the software implementation of SDR as a nanosatellite communications system. The design performs the signal processing for the communication system. In this chapter, the algorithm for this design is explained. The modulation / demodulation schemes used for this purpose have been implemented in simulation previously but not with a SDR platform. This implementation aspect of the signal processing methods is crucial to the incremental contribution in the development of the software defined radio nanosatellite communication system. The transceiver algorithm is fully modelled and implemented in software using the Simulink tool. The aforementioned environment provides the required flexibility for the implementation and validation of the SDR design. In section 5.2 and 5.3, the transmitter and the receiver have been implemented.

5.1 Transceiver Algorithm Design Specifications

NanoCom U480 is a half-duplex UHF transceiver system for space applications with limited resources. NanoCom U480 consists of a transceiver which does all the analog signal processing and a modem which does all the baseband signal processing. The modem used is CMX469A FSK/MSK Modem. The transceiver is designed to emulate the NanoCom U480 transceiver system in software. Referring to Figure 38, for the purpose of this research, the analog signal processing is performed by the front end WBX TX/RX board, which is attached to the FPGA on the USRP board. The transceiver design implements the baseband signal processing in software. The design specifications of the algorithm are:

1. The algorithm implements the transmitter and receiver chains which perform modulation and demodulation respectively.
2. The algorithm is compatible with the modules already present on the USRP for the hardware implementation of the design. The algorithm is designed to fit in the custom TX and RX module (from Figure 38).
3. The signal is generated, received and processed entirely in software.
4. The design algorithm easily adapts to the changes in the baud rates.

5.1.1 Selection of Modulation/Demodulation Scheme

As discussed in section 2.4, the digital communication schemes namely PSK, CPM, and trellis-coded modulation require phase tracking. In this research the, a functioning algorithm with a simpler configuration is implemented, therefore the phases are not being considered. The many advantages of FSK over ASK include better noise resistance. ASK is prone to noise which affects the amplitude of the signal. FSK is less susceptible to errors as compared to ASK since the receiver looks for specific frequency changes over a number of intervals so noise spikes can be ignored. The type of FSK being implemented is BFSK modulation scheme with two frequencies which correspond to one and zero. The message to be communicated is a binary message, which has either a zero or one in the message. The modulation scheme implemented is the non-coherent BFSK modulation. Given the objective is to validate the software defined communication system designed for a nanosatellite, a non-coherent BFSK is a better fit than a coherent system due to its simpler configuration on the development platform (fewer parameters) and shorter development duration.

5.2 Transmitter Implementation

Referring to section 5.1.1, non-coherent BFSK modulation is being implemented in the design algorithm explained in section 4.2.1 and 4.3.1. From the transmitter, the signal is sent to the digital up-converter of the USRP.

5.2.1 Transmitter Architecture

The modulator in the transmitter design has two phases shown in Figure 39.In the first phase the modulator generates a signal using non-coherent FSK modulation. The second phase consists of resampling and FM baseband modulation.

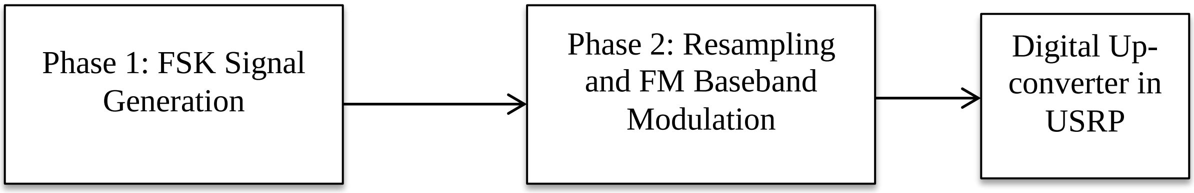

Phase 1: FSK Signal Generation
Phase 2: Resampling and FM Baseband Modulation
Digital Up-converter in USRP

Figure 17: Transmitter Algorithm Architecture

In the first phase the modulator generates a signal corresponding to the information being sent as illustrated in Figure 40. The binary data input message is generated by the Bits Generator block in Simulink. In the algorithm, the signal is generated using two oscillators each corresponding to mark f_1 and space f_2 frequencies. The two discrete frequencies chosen are 2100 Hz as mark frequency and 1300 Hz as space frequency, the initial phases of mark and space frequencies are Φ_1 and Φ_2 which are not related to each other and are considered to be zero for this particular design. The signal generated by the oscillators is represented by s_M (Mark)=cos(2*π*2100*t), s_S (Space)=cos(2*π*1300*t) where t is the sampling period. The algorithm implementation designs the oscillators using DSP Sine wave generator in Simulink. The signal generated is controlled by the binary data input. The control is achieved by designing a multiplexer, which takes in the information in the form of binary data. The multiplexer switches between the mark frequency and space frequency depending on the bit in the binary data input. A one bit corresponds to the mark frequency and the zero corresponds to the zero frequency. The multiplexer is designed with switch block in Simulink. The signal generated is FSK modulated and is represented by s_I(t)=cos(2*π*f(i)*t) where f(i)(frequency) depends on the information message bit. The signal is then sent to the second phase of the modulator

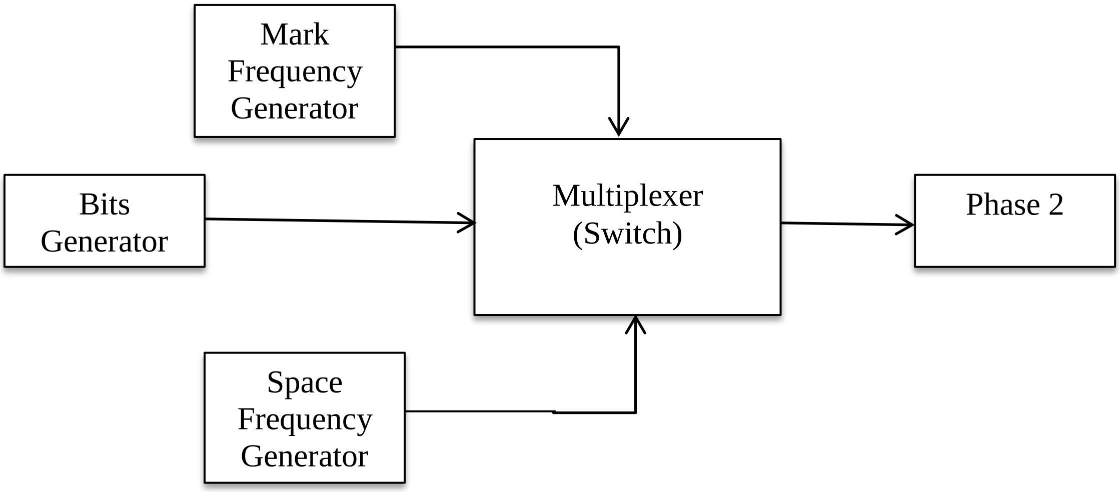


Figure 18: Transmitter Algorithm Phase 1 FSK Signal Generation

The second phase of the algorithm takes the FSK modulated signal generated from the first phase, and modulates further to make the signal compatible with the USRP as illustrated in Figure 41. In the algorithm, firstly the signal is resampled to convert the sampling frequency of the design to the USRP sampling rate of 200 kHz. A resampler is needed to achieve this function. In the algorithm implementation of the second phase the signal is directed through an FIR interpolation block in Simulink to achieve the resampling. The conversion rate depends on the sampling frequency of the design which in turn depends on the data rate at which the transmission is occurring. The resampling filter is designed using the MFILT object from the DSP System Toolbox in Simulink. For the example, if the baud rate is 200 bits per second, sampling frequency is 8000 Hz of the FSK generated signal, the conversion rate is 25 so that it gives 200 kHz (25*8000 Hz).

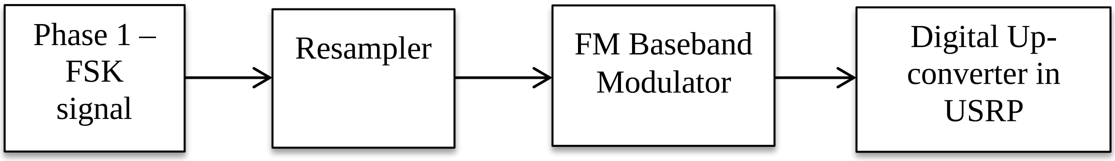


Figure 19: Transmitter Algorithm Phase 2 Signal Transmission

Next, the signal generated is converted to complex baseband signal. The conversion is so required because the signal needs to be compatible with the USRP Digital up-converter which requires a complex baseband signal. The USRP works with a complex baseband signal which it further converts to RF signal which is further transmitted via antenna. In the algorithm, the conversion is done using the frequency baseband modulation in the design. The real signal from the first phase of the modulator is represented as u(t). The complex signal x(t) can be produced in the following way

$$x\ (t) = \text{e\textasciicircum}(2*\pi*j*K_f*\text{integral}(u(s)ds)) \qquad (7)$$

where K_f – Modulation index or frequency sensitivity.

Referring to Figure 42, the frequency baseband modulator is designed with an integrator and a phase modulator whose function in this case is conversion of a real signal into a complex signal (i.e., Quadrature modulator)(SGR, 2010). In the implementation of the design, the integration is done by the digital filter more specifically an IIR (Infinite Impulse Response) filter. The signal then is multiplied with the frequency sensitivity gain as per equation 7. The multiplication is achieved in the implementation by passing the signal from the filter to the frequency sensitivity gain which has a value of 0.0785. This value is calculated using equation 5 from section 2.4 with the values of amplitude 1, frequency deviation 2.5 kHz and the sampling period of the USRP (200,000 Hz). Conversion of the signal into a complex signal is achieved using quadrature modulation, which in the Simulink implementation is the Magnitude-Angle to Complex converter where the magnitude is taken as 1.

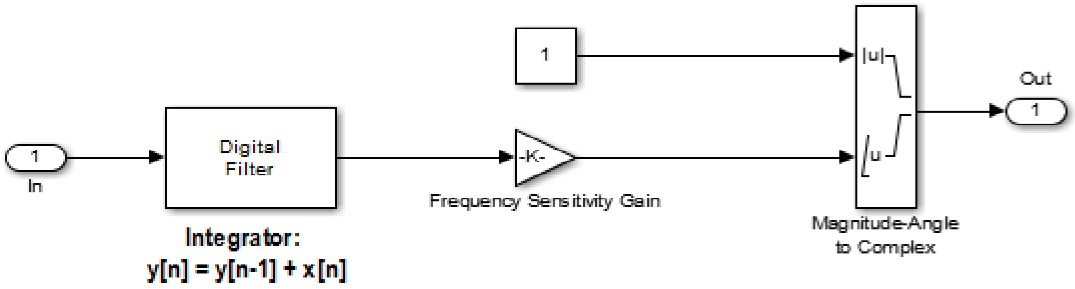


Figure 20: Transmitter Module – Frequency Baseband Modulator

The signal is then sent to the digital up converter which is already in the USRP. The transmitted message is sent from the host computer to the USRP via the transmitter control. After modulation and up conversion the frequency is converted to analog signal which is transmitted by the front end using the antenna.

5.3 Receiver Implementation

Non-coherently generated signal can only be non-coherently demodulated. Therefore, for the purpose of this research development the non-coherent BFSK demodulator is implemented on the receiver. The signal once received via the front-end connected to the USRP, digitized and down-converted by the down converter present in the USRP goes through the receiver algorithm described in section 5.3.1.

5.3.1 Receiver Architecture

Referring to figure 43, the receiver algorithm has two phases. The first phase consists of frequency demodulation to baseband form, resampling of the signal and filtering of noise as illustrated in Figure 44. The second phase of the algorithm is designed using a correlator implementation of a non-coherent BFSK demodulator in the receiver

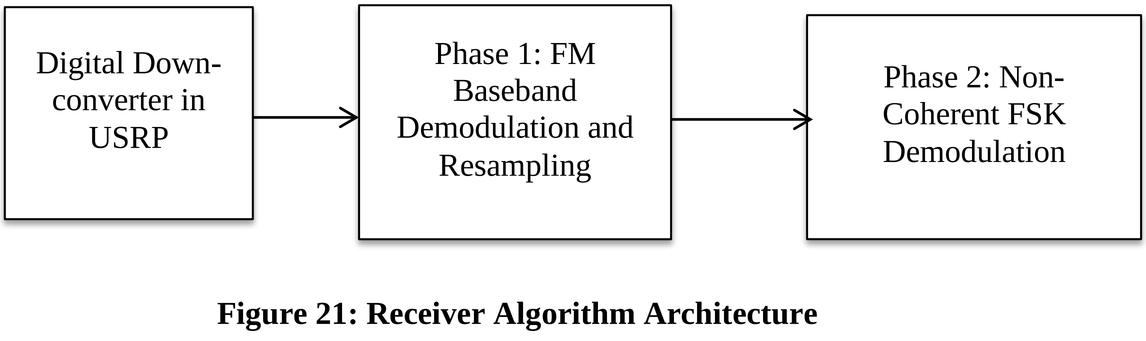

Digital Down-converter in USRP
Phase 1: FM Baseband Demodulation and Resampling
Phase 2: Non-Coherent FSK Demodulation

Figure 21: Receiver Algorithm Architecture

The frequency demodulation to the baseband form is achieved in the design by performing differentiation to get the frequency change of the carrier wave. The output signal from the USRP digital down-converter is fed to the FM demodulator which performs differentiation to get the baseband signal.

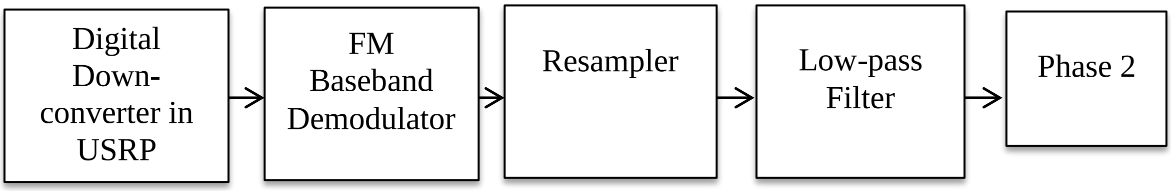


Figure 22: Phase 1 -Receiver Architecture

In Figure 45, the algorithm implementation of FM Baseband Demodulator in Simulink is illustrated; the delayed signal is multiplied with the conjugate of the signal. The signal is then converted to a real baseband signal using a Complex to Magnitude-Angle block from Simulink to get the output as a phase angle of the input signal.

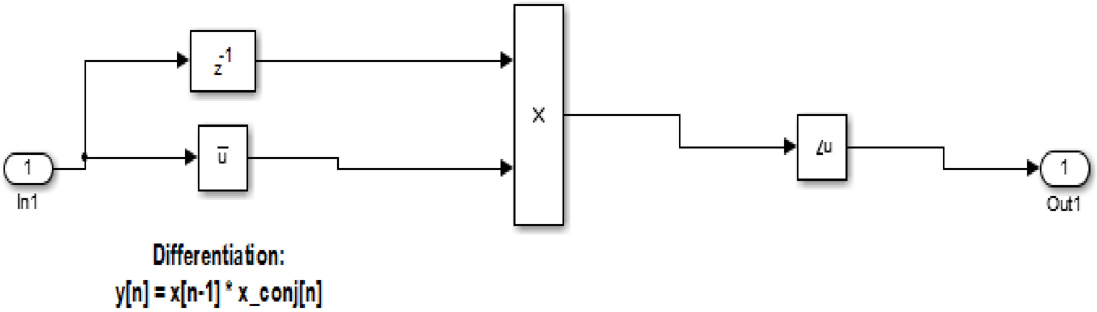


Figure 23: Frequency Baseband Demodulator

Secondly in the first phase of the algorithm, signal is resampled from 200 kHz to the sampling frequency of the design as shown in Figure 40. The algorithm implementation in Simulink uses the FIR decimation to achieve the resampling.

The received signal is contaminated with noise while being transmitted over air. There is thermal interference and other sources of noise which use electromagnetic waves. The algorithm design includes a filter to remove the noise. A low pass filter is used for this purpose. A low-pass filter allows signals with frequencies lower than a certain cut off frequency and reduces the effect of higher frequencies on the signal. The low-pass filter filters out the high-frequency components of a signal, letting us focus on the low frequencies we may be interested in. The low frequencies keep most of their strength. The high frequencies are reduced. At a certain frequency, called f_{3db}, the filtered strength of the frequency is exactly 3 decibels less than the original (or, about 70%). In the design implementation a Low-Pass Filter block is used in Simulink. In Simulink the low-pass filter has the filter response illustrated in Figure 46. The filter response is for the following specifications, input sampling frequency of 8000 Hz and cut-off frequency of 2500 Hz. The low-pass filter filters out the frequencies above 2500 Hz so that it includes the mark and space frequencies, 2100 Hz and 1300 Hz, respectively.

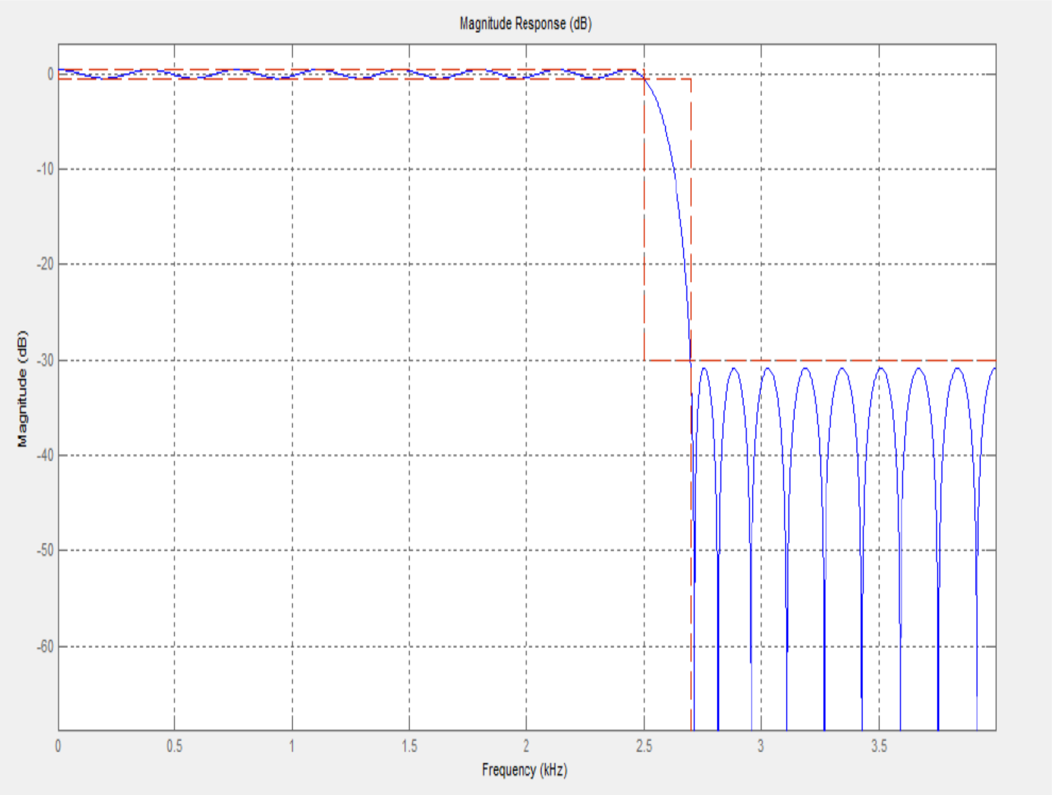


Figure 24: Filter response of Low-pass filter

The second phase of the algorithm is designed using a correlator implementation of a non-coherent BFSK demodulator in the receiver. Referring to the Figure 47, the upper two branches are implemented to detect f_1 (mark frequency) and the lower two to detect f_2(space frequency). Figure 43 illustrates the non-coherent demodulation implemented in the phase 2 of the algorithm design. Referring to equation 6 the received signal r(t) is represented as s_i (t,θ) = Acos(2πf_it + θ), i=1,2 , where Acosθcos(2πf_it) is the in-phase component and Asinθsin(2πf_it) is the quadrature component.

In the equation 6, the in-phase component is the real part of the complex signal and the quadrature components is the imaginary part of the signal. In the algorithm, the signal received from the low-pass filter represented by r(t), is divided into in-phase and quadrature components for each frequency component by passing it through the respective correlators. The in-phase correlator is designed using real part of the expected mark or space frequency and the quadrature correlator is designed using the imaginary part of the expected mark or space frequency. The signal is multiplied with the in-phase correlator to get the in-phase part of the signal and the quadrature correlator to get the quadrature or imaginary part of the signal. In the algorithm implementation, the correlators are implemented in Simulink by multiplying the signal from the low-pass filter to the real and imaginary parts of the DSP sine wave generated, expected mark and space frequencies of 2100 Hz and 1300 Hz.

Next the signal from the correlators is fed to the integrator which is implemented in the Simulink using the Integrate and Dump. The signal is then squared. The signal from the integrator is squared to get rid of the phase components. In Simulink the Math module specified for squaring is used for this purpose. The outputs of the in-phase and the quadrature branches are added. The received signal corresponds to either f_1 mark frequency or f_2 space frequency, evaluated by the judging unit. The judging unit is designed by comparing the outputs from the correlators of the first two and the lower two branches as shown in Figure 47. The judging unit implemented by comparing the output of the sum of the first two branches and the sum of the lower two branches in the algorithm using a Relational Operator block in Simulink. Next, based on the decision of the aforementioned comparison the algorithm judges whether a bit one or bit zero is the output of the demodulation. If the sum of the first two branches is higher than the sum of the lower two branches then a one bit is the output and if the vice versa is true then a zero bit is the output. The implementation of the judging unit is done by a Simulink model called Relay. The data output from this relay are the final demodulated signals which are stored in a file.

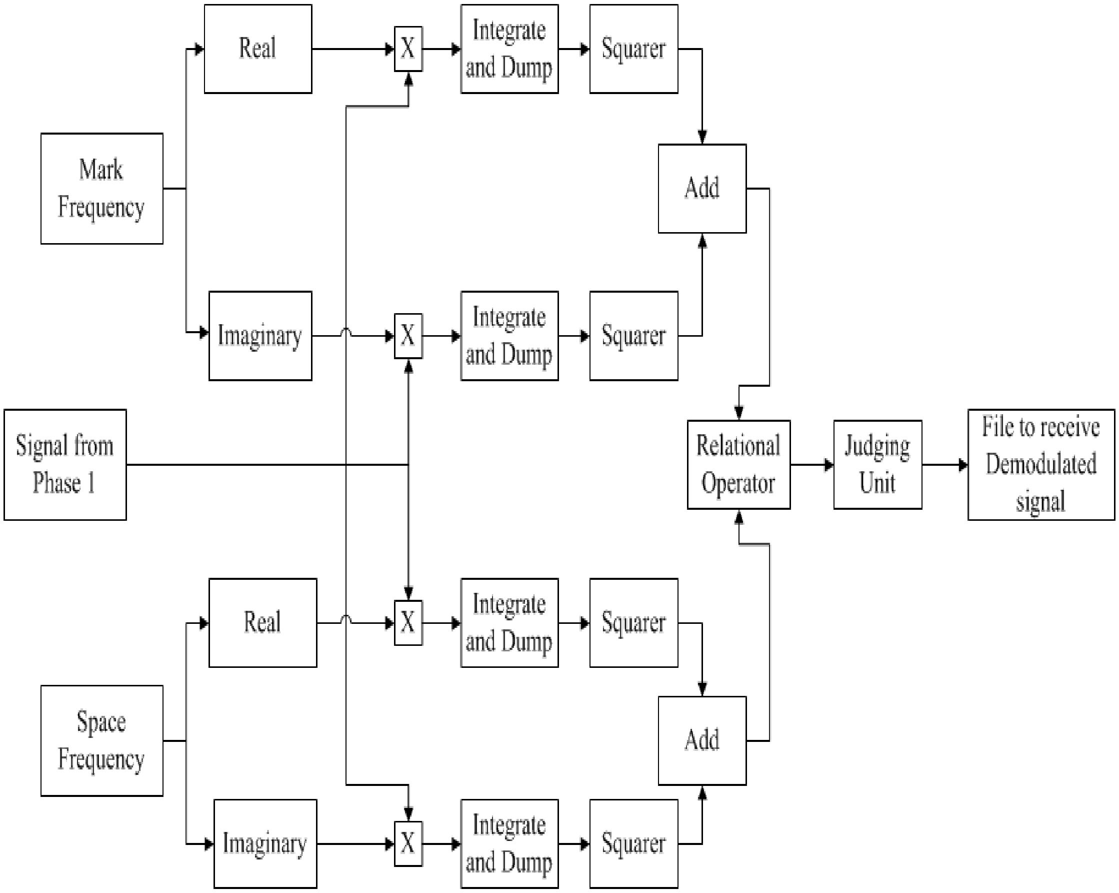


Figure 25: Phase 2- Receiver Architecture – Non-coherent FSK Demodulation

6 Transceiver Algorithm Performance Evaluation

The research in this thesis is an incremental contribution towards the development of a nanosatellite SDR-based communication system. Chapter 6 describes the tests performed to establish whether the software implementation satisfies the research objectives discussed in section 1.4. Tests are performed in two phases to evaluate the algorithm. The tests are performed to demonstrate that the transceiver design implementation works as a transceiver using a singular modulation scheme. The tests also aim to establish Simulink as a substitute to the GnuRadio while using the hardware with the design. A third test is performed to test whether the design can be completely imported on the FPGA of the USRP N210. With the implementation of the transceiver algorithm in software and its successful tests the research aims to demonstrate the feasibility of an SDR based system for nanosatellite communications.

6.1 Test Setup

To fulfill the research objectives, the functionality of the transceiver needs to be tested. The evaluation can be accomplished by transmitting ones and zeros as the binary information message through the transmitter and receiving the information using the receiver and checking whether the information matches.

The tests are done in two phases. The first test phase involves the testing of the design in simulation to validate the functionality of the transmitter and receiver as illustrated in

Figure 48 of the test setup. This is performed in Simulink. The data generated by the transmitted design are saved on file and compared with the data received from the receiver design saved on file. The comparison is done using an Error Rate block in Simulink. The test is performed for three different baud rates of 200 bits per second, 1200 bits per second and 1600 bits per second.

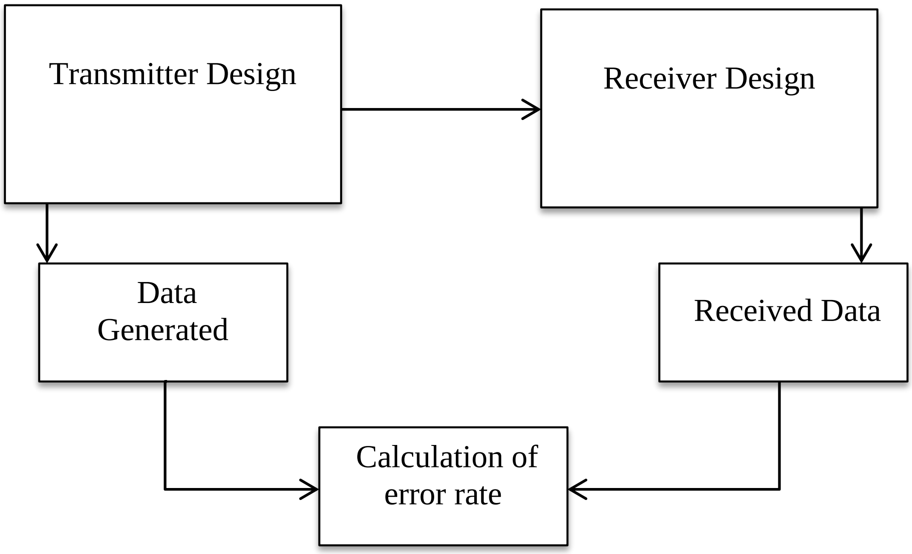


Figure 26: Test Phase 1 – Communication link in Simulink

The second phase of testing has two parts to it as shown in figure 49 and 50. Firstly, referring to Figure 49, the transmitter design is connected to an SDRUTx block available in Simulink (Refer to section 4.2.3), which connects the transmitted algorithm implemented in Simulink to the USRP transmitter chain. The design when run transmits data through the USRP. The transmitter sends the data at the UHF frequency of 437.475 MHz. The data are collected by a Baofeng radio at the UHF frequency of 437.475 MHz while connected to a host computer. The Baofeng radio sends the data to the host computer which saves the data. The data generated by the transmitter are saved on file to compare with the received data.

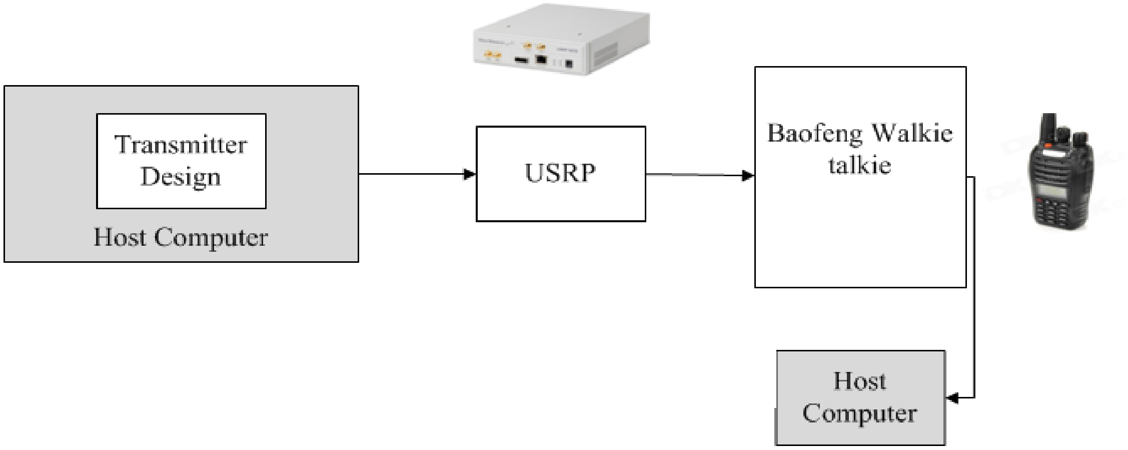


Figure 27: Test Phase 2 – Transmitter Setup

Secondly, referring to Figure 50, the receiver design is connected to a SDRURX block available in Simulink which connects the receiver algorithm to the receiver chain on the USRP. The data saved on the computer from the first part of this phase are transmitted through the Baofeng Radio again at a UHF frequency of 437.475 MHz, which is received by the USRP receiver chain and the data are sent to the receiver design implemented on Simulink, which gives us the demodulated data signal. The received data are saved on file which is once again compared with the transmitter generated data to do error analysis. This test simulates how the algorithm would work with the USRP but does not have the algorithm on the FPGA in the USRP. The baud rates of 200 bits per second and 1200 bits per second are tested.

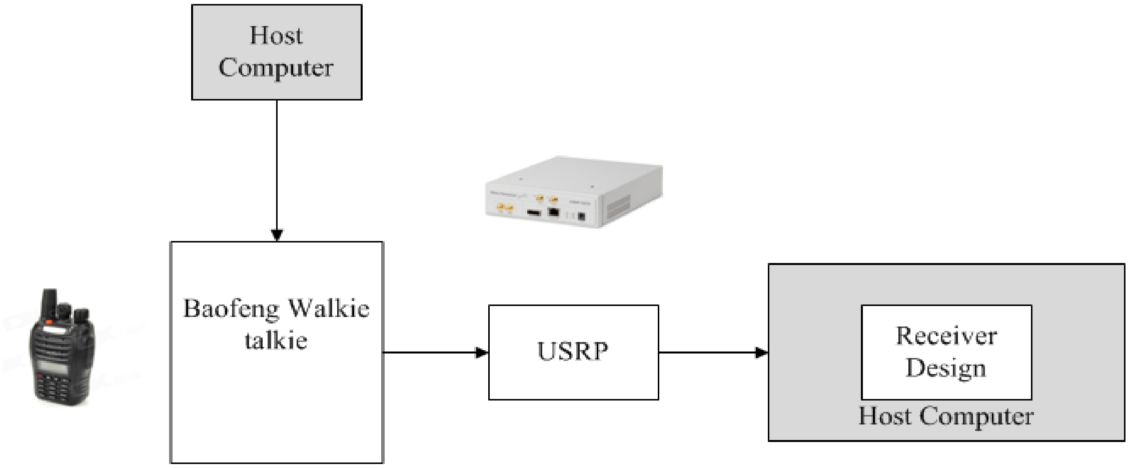


Figure 28: Test Phase 2 – Receiver Setup

The third test involves using the Communications System Toolbox Support Package for USRP Radio to target the FPGA on the USRP. The aim of this test is to verify whether parts of the transmitter and receiver design can be imported on the FPGA as illustrated in Figure 51 and Figure 52. The data generator in the transmitter which is implemented by the Bernoulli Bits Generator in Simulink is not a part of the design that needs to be imported on the FPGA. Similarly in the receiver design the file to receive the demodulated data are not the part of the design that needs to be imported on the FPGA.

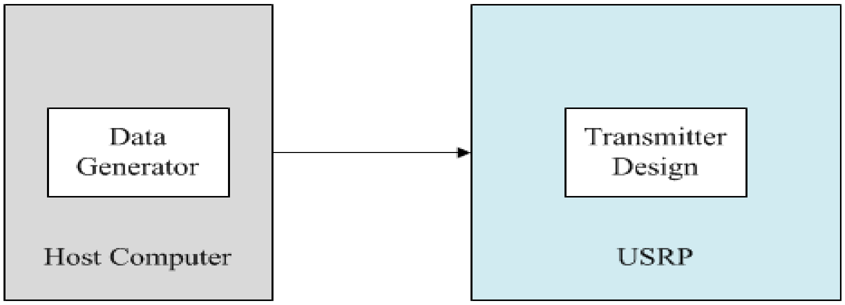


Figure 29: Test 3 – Transmitter on FPGA

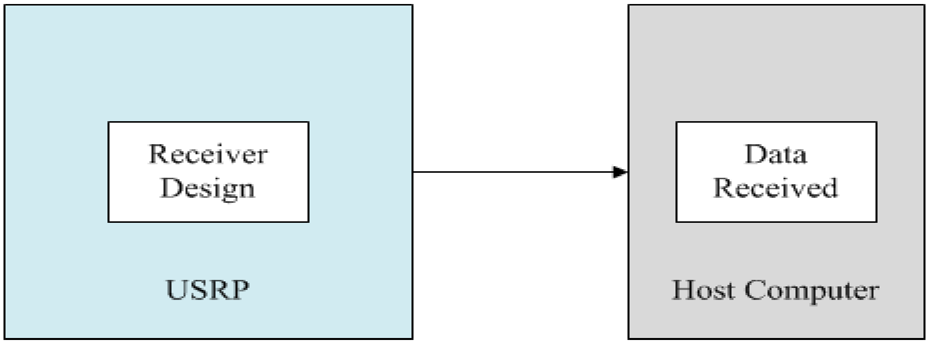


Figure 30: Test 3 – Receiver on FPGA

6.2 Transceiver Design performance Evaluation

The two phases of the performance evaluation are analysed in the sections below. Data is compared from the receiver and the transmitted to determine a successful communication link and the error rates.

6.2.1 Phase 1 – Transmitter Receiver Link Simulation Results

As discussed earlier the simulation is tested with different data rates of 200, 1200 and 1600 bits per second as the generic software based communication system should be able to work with different baud rates. Table 15 shows a summary of all the tests done in this phase, the number of data symbols being sent and the error rates. The signal generated by the transmitter contains a mixture of ones and zeros. From table 13 it is seen that data transmitted from the transmitter design is identical to the received demodulated data without any errors. The successful transmission and reception at different baud rates shows that the transceiver design is capable of functioning at different baud rates. The 1200 baud rate is one of the standard baud rates of an amateur band communication system. Signal processing is being performed since the information data received by the receiver are identical to the information data sent by the transmitter. The non-coherent BFSK modulation scheme is implemented successfully by transmitter in simulation to provide a signal to be transmitted. The non-coherent BFSK demodulation is applied by the receiver successfully in simulation since the received information matches original information provided by the transmitter. The successful communication link established also shows that changes can be made quickly to the algorithm by changing the different baud rates, adding and FM baseband modulation / demodulation implementation to the algorithm thus establishing the flexibility of the design which an ASIC based system is unable to provide. The target rate of 9600 bps was tested with the transceiver code, the modulation/demodulation scheme implemented for this code does not suit the desired system data rate. Therefore a coherent scheme needs to be implemented which is explained in the section 6.3.1.

Baud Rate (bits per second)	Time (seconds)	No of Data Symbols	Error rate
200	1	201	0 %
200	25	5001	0 %
1200	1	1201	0 %
1200	25	30001	0 %
1600	1	1601	0 %

Table 15: Phase 1 test cases with results

6.2.2 Phase 2- Tests of Transceiver Design with USRP

The tests for the second phase of testing are performed for 200 bits per second and 1200 bits per second. Figure 53 compares the transmitted data with received data for 200 bits per second baud rate. The plot comprises of the subtraction of the received data from the transmitted. When the data transmitted is identical to the received data the comparision gives out a zero. The peaks show the errors in the comparision. From the plot it is seen that the received data has a few errors. The errors correspond to the first test case in table 16. A total of 4 errors are detected which gives a error rate of 2% for a total of 201 data symbols transmitted and received. Table 16 shows a summary of all the tests done in this phase along with error rates.

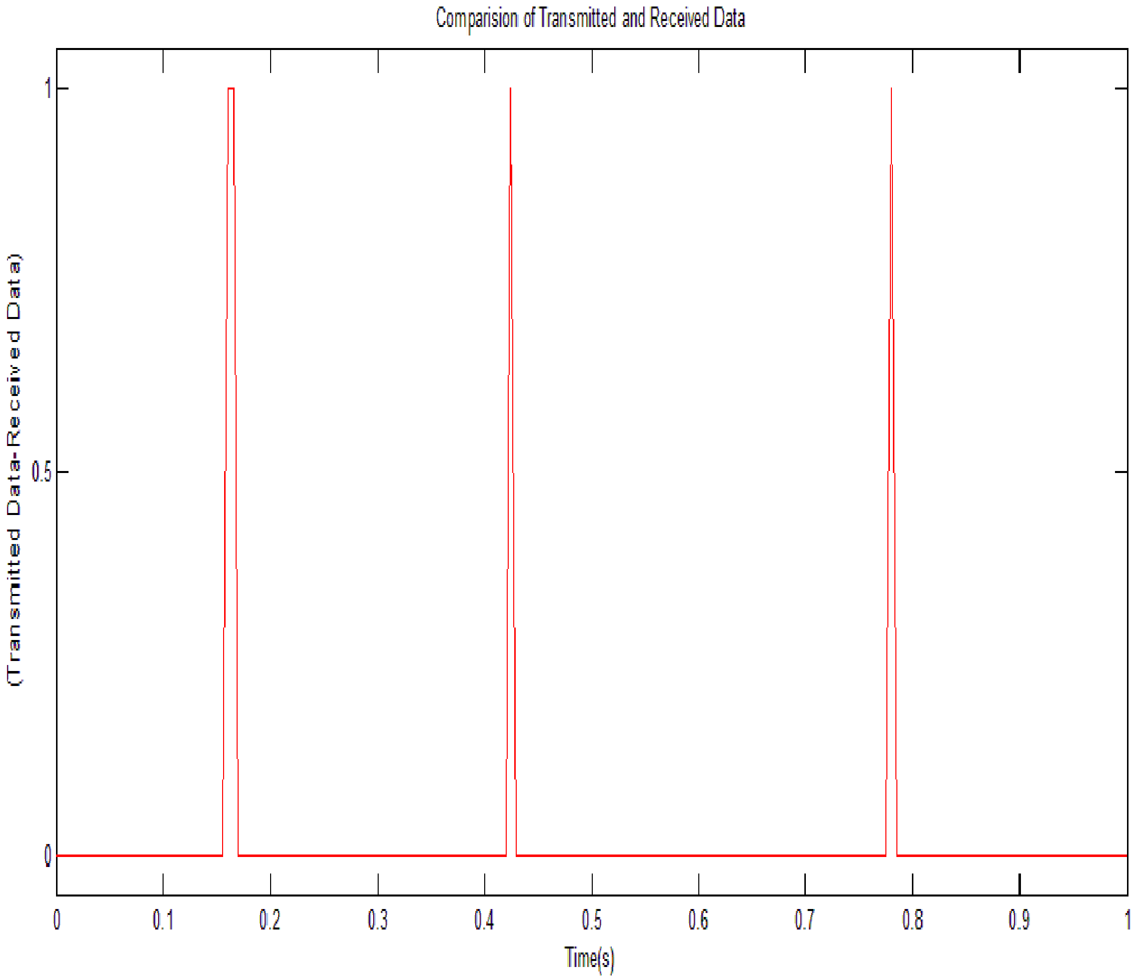


Figure 31: Comparison of transmitted and Received Data for 1 second at 200 baud rate

The successful communication link verifies the feasibility of the transceiver algorithm designed with a FPGA based hardware platform which in this case is the USRP N210. The successful communication link is indicated by the successful transmission and reception of data of the USRP N210 with the use of the transceiver algorithm for signal processing. The successful communication link also shows the compatibility of the transceiver code with the previously present modules in the USRP. The error rate column shows the low errors in the communication link. As per the link budget described in section 3.4, the bit error rate allowed for the system is 1.0E-05. The error rates shown in the last column of Table 16 are quite high for the system due to the lack of time synchronization with the USRP based transmitter and receiver. Similarly, comparing to Table 15 the errors are higher because the transmitter and the receiver and are not clock synchronized along with the lack of phase synchronization of the signal. Even though the error rates do not match the specified error rate for the system, they establish that the transceiver works with albeit with a few errors. As per the successful transmission and reception, the transceiver design feasibility with hardware suitable for SDR implementation is established for all the test cases of this phase.

Baud Rate (bits per second)	Time (seconds)	No of data symbols	Error rate
200	1	201	2 %
200	7.5	1501	6.5 %
1200	0.2	241	13.6%
1200	2	2401	11.8 %

Table 16: Phase 2 test cases with results.

6.2.3 Targeting the FPGA on USRP N210

As discussed in section 6.1, for the third test some parts of the algorithm design are imported on the FPGA since the blocks are compatible with the HDL coder in Simulink. Blocks that are not compatible with the HDL coder are the resampler in the transmitter and the FIR decimation block for receiver. All the blocks where imported on the FPGA and the USRP was tested but a successful communication link was not established because some of the blocks imported on the FPGA don't work with the frames on the FPGA (Digital Filters in Simulink). New blocks need to be designed with same functionality to be able to work with frames to establish a successful communication link.

6.3 Error analysis of Transceiver Design

The higher data rates produce higher error rates. These errors are reduced with synchronization which includes timing recovery and carrier phase recovery explained in section 5.3.1. Referring to table 15, the error rate for 1600 baud rate is the highest.

It is also noticed that there is high error rate when the design is run with the USRP. The major contributing factor to these errors is transmitter-receiver synchronization as explained in section 6.3.1.

6.3.1 Frame Synchronization

Frame synchronization is important for error free data reception. It requires the transmitter and receiver to be synchronized in terms of the end and beginning of the frame and in the phase of the signal.

Timing recovery: The objective of timing recovery is to obtain symbol synchronization. The demodulation in receiver needs to know the beginning and the end time of the signal. Clock recovery tries to synchronize the receiver clock with the symbol rate transmitter clock to obtain samples at appropriate time. The receiver needs to know the sample frequency and where to take the samples within each symbol interval. The two quantities need to be determined by the receiver to achieve symbol synchronization which are sampling frequency and sampling phase. Time timing recovery comprises estimation of the timing error and the timing corrections (Dick et al., 2000). Different algorithms which implement timing recovery are Gardner, Mueller and Mueller and Early – Late gate Phase recovery and tracking.

Carrier recovery (Phase recovery): The objective is to remove frequency offset for the signal to be processed in baseband form. For this purpose two parameters that need to be estimated are carrier frequency offset and carrier phase offset. The received signal is then corrected as per these estimates (Dick et al., 2000). Carrier recovery can be accomplished with Phase Locked Loop or a feedforward digital carrier recovery technique. Costas loop, N-ary PSK Costas Loops, Digital phase-locked loop, decision-directed carrier recovery loop also performs phase coherent supressed carrier reconstruction and synchronous data detection within the loop.

6.4 Physical Specifications of the hardware

Table 17 contains the physical specifications of USRP N210. The USRP N210 is a development platform on which the communication algorithm has been implemented. The USRP N210 requires the use of a host computer. For the communication system to be a modular standalone the use of host computer is not needed. To enhance the hardware to fit for nanosatellite system, the algorithm code is easily ported to USRP E100/E100 which is appropriate as a standalone system.

Physical Specifications	
Mass with the case	1.2 kg with the case
Size	22 x16 x 5 cm
Maximum Power Consumption	13.8 W maximum with the current daughterboard

Table 17: Physical Specifications of USRP N210

7 Conclusions and Future work

In this chapter a summary of the research work done for this thesis is provided. The research objectives accomplished are discussed along with the contributions, which this study provides in the area of software based nanosatellite communications system. Future recommendations to build a complete software based transceiver for nanosatellite communication system are presented as well.

7.1 Summary

A survey is done of the hardware for nanosatellite communications system to determine that SDR is a flexible, cost effective solution for nanosatellite communications system. A software based transceiver algorithm is developed and implemented for the purpose of nanosatellite communications. A survey is presented of the various hardware platforms available. A survey of FPGA based platforms is also presented to determine the suitable development platform for this research. USRP N210 is selected to implement the transceiver design algorithm. A communication algorithm for a transceiver is designed using non-coherent binary frequency shift keying modulation scheme. This design is implemented on the USRP N210 development platform. Simulink is the software tool used for this purpose and also to test the design. Implementation in Simulink provides a unique functionality for testing in simulation and in hardware. The design successfully sends and receives information bits in simulation and also with the hardware. Parts of the design in simulation are also targeted on the FPGA. A link budget analysis is done to determine the feasibility of the system. Some of the objectives accomplished are presented in the following sections.

7.1.1 Transceiver Functionality

The software implemented on the USRP N210 functions as a transceiver whose function is to transmit data and receive data. The transmitter is implemented using non-coherent binary frequency shift keying modulation scheme. The receiver also uses non-coherent binary frequency shift keying demodulation scheme. The transceiver fulfills the primary objective of establishing a communication link by successfully transmitting and receiving data using the design. The communication algorithm is first tested successfully in simulation only. The design is also tested with the USRP successfully. Compatibility of the design with the already present modules on the USRP is achieved with the successful communication link. The USRP transmits and receives successfully at UHF frequencies. Data rates of 200 bits per second, 1200 bits per second and 1600 bits per second are also tested with the design successfully. The flexibility of the design code is demonstrated successfully with the implementation of the design with the already present modules, the implementation of different data rates and the added configurability of adding or removing functions of encoding, modulation and other signal processing blocks. The flexibility of adding more functionality is essential of a software defined radio system.

7.1.2 Software Implementation of SDR for nanosatellite communications system

The communication system transceiver design uses software for baseband signal processing. The USRP N210 is used with host PC to develop the software for the system.

The USRP uses open source hardware and software licenses. The communication system design is implemented in software on the USRP with the use of the Communications System Toolbox Support Package for USRP Radio. The package uses Matlab /Simulink which simulate the design, tests the design with USRP and also target the FPGA on the USRP using HDL coder. USRP UHD driver is used to test the design in Simulink with the USRP. The package is integrated with Xilinx ISE Software using System Generator. The use of Matlab/Simulink eliminates the use of GnuRadio.

7.1.3 Transceiver Design Hardware Implementation

USRP N210 implements both transmitter and receiver design of communication system successfully. The USRP N210 development platform hardware is a commercial off the shelf product with the cost of $2140 which remains within the cost of a nanosatellite mission development. The USRP has a daughterboard with a front end attached. A monopole antenna is connected to the USRP N210. The FPGA has the following digital signal processing modules, digital down-converters and up-converters, frequency translators and transmitter and receiver control modules already programmed on the USRP N210. USRP has the flexibility of modifying the structure that it has already inbuilt. Therefore a communication algorithm is added and tested successfully to the already present software modules. Referring to section 3.2, the USRP is used normally as a radio using GnuRadio. The disadvantage of GnuRadio is that the software cannot be implemented on the FPGA. A new method is used to simulate the design, test it and port it on the FPGA using Simulink.

7.2 Contributions

The research in this thesis is an incremental contribution towards the development of SDR for nanosatellite communication system using open source hardware and software. Oliveri(2011) developed an agile SDR hardware platform which can fit on 1U CubeSat. The SDR hardware platform developed is called the Configurable Space Microsystem Innovations and Applications Centre (COSMIAC) CubeSat FPGA board. The research in this thesis takes a step further on the software development aspect and presents the following novel contributions:

1. The available technologies for nanosatellite communications system were examined and SDR design was established as a cost-effective, flexible alternative;
2. The software implementation of SDR as a nanosatellite communications system was done. The design performed signal processing for generic communication purpose. The design eliminated the use of GnuRadio by implementing in Simulink to allow for easy porting onto the FPGA-based system
3. The design supports both receiving and transmission. The design uses software for baseband functionality in signal processing. A singular modulation scheme to demonstrate the feasibility of an SDR system for nanosatellite communications is tested. The design is implemented on a hardware development platform which is commercial off the shelf, meets the budget constraints and can be enhanced for nanosatellites.

These contributions create the foundation to develop a flexible software based communication system using a FPGA platform.

7.3 Future Work Recommendations

The performance and the implementation need to be improved to build complete software based communication system for nanosatellites. Future recommendations are discussed to aid in getting closer to building a complete standalone modular system for a nanosatellite.

7.3.1 Improve Design Performance by Reducing Errors

Referring to the error analysis in section 6.4 there is a need for algorithm performance enhancement by reducing the errors. Section 6.4 lists the reason which gives rise to errors as the need for transmitter receiver synchronization. Higher data rates produce higher error rates which can be reduced with transmitter receiver synchronization as well.

A coherent FSK implementation as explained in section 5 has a lower probability of errors because it includes phase tracking. Therefore changes should be made to the non-coherent BFSK transceiver algorithm to implement coherent BFSK scheme to reduce the error probability.

7.3.2 Modular standalone Software Defined Radio

For the communication system to be a modular standalone the use of host computer is not needed. The USRP N210 is a development platform on which the communication algorithm has been implemented. The USRP N210 requires the use of a host computer. For this purpose, the algorithm code should be easily ported to USRP E100/E100 which is appropriate as a standalone system (Ettus Research).The embedded computer has a fully functional Linux distribution with GNU Radio and other software used for signal processing installed. The USRP E110 uses the same FPGA as the USRP N210. Since the USRP is not tested for space qualification, environmental testing should be done to make it flight ready. Aspects of environmental testing are discussed in section 6.3.3.

Hardware sent to space is affected by the space radiation. There are some space rated FPGA's like the Radiation-Hardened, space grade Virtex-5QV. But these cannot be adapted to a smaller satellite like a nanosatellite. Therefore, once a complete prototype is built it can be transferred to a custom built radiation hardened FPGA system fit for a nanosatellite.

7.3.3 Protocols and Space Environment Testing

Referring to Section 3.3.1 communication protocols should be implemented on the SDR for proper synchronized information transmission and reception. The implementation should be an added block in the algorithm design to decode the protocol and output information frames as per the protocol.

The hardware also needs to go through environmental testing. Environmental testing includes vacuum chamber testing and thermal testing. Environmental testing is done to examine whether the hardware will perform in vacuum environment and if it will perform with space temperatures and temperature variations. Environmental testing also includes vibration testing. The hardware also needs to be tested to determine whether it will function under high vibration conditions known due to flight launch and satellite spin.

8 References

AAUSAT3 (2012), *AAUSAT3- the third CubeSat from Aalborg,* http://www.space.aau.dk/aausat3/

Ahn, C., Kim, J., Ju, J., Choi, J., Choi, B., & Choi, S. (2011), *Implementation of an SDR platform using GPU and its application to a 2× 2 MIMO WiMAX system,* Analog Integrated Circuits and Signal Processing, 69(2-3),pp. 107-117.

ALTelectronics (2011), *XC3S400-4TQ144C,* http://altelectronics.co.uk/shop/surplus-parts/semiconductors/fpga/cpld/xc3s400-4tq144c/prod_8.html

Avanzi, A.; Tortora, P.,(2010**)** *Design and implementation of a new spaceborne FPGA-based dual frequency GPS and Galileo software defined receiver,* Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on , vol., no., pp.1,7, 8-10.

AX.25 Link Access Protocol for Amateur Packet Radio (1998, July), *Version 2-2,* Rev.

Bekkeng, J. K., *Radiation effects on space electronics,* Universidad de Oslo, http://www.uio.no/studier/emner/matnat/fys/FYS4220/h11/undervisningsmateriale/forelesninger-vhdl/Radiation%20effects%20on%20space%20electronics.pdf

Benedetto, S., Mondin, M., & Montorsi, G. (1992), *Performance evaluation of trellis-coded modulation schemes,* GLOBECOM-NEW YORK, pp.1785-1785.

Bouwmeester, J., & Guo, J. (2010), *Survey of worldwide pico-and nanosatellite missions, distributions and subsystem technology,.* Acta Astronautica, *67*(7), pp. 854-862.

Brockie, Peter (2007), *Apple MPEG Media System*, http://www.forcedperfect.net/hardware/cards/applempegmediasystem/

Broendum. J.(1994), *Frequency Shift keying*, Msc Thesis

Crawford,T., Hughes, M., Ketcheson, D., & Kuruvilla, M. (2009), *YuSend-1 nanosatellite communication subsystem*.

CubeSatShop.com (2006), *NanoCom U482C Half Duplex Receiver*, http://www.cubesatshop.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=36&category_id=5&option=com_virtuemart&Itemid=67

CubeSatShop.com (2006)**,** *ISIS UHF downlink /VHF uplink Full Duplex Transceiver*, http://www.cubesatshop.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=11&category_id=5&option=com_virtuemart&Itemid=67&vmcchk=1&Itemid=67

Dabcevic, K. (2011), *Evaluation of software defined radio platform with respect to implementation of 802.15. 4 zigbee*, School of Innovation, Design and Engineering

Davis, L. M., Haley, D., Pollok, A., & Chen, Y. (2011), *Satellite links and software defined radio for remote communications and sensor data gathering*, Defense Applications of Signal Processing (DASP), Australia, pp.1-5.

Dick, C., Harris, F., & Rice, M. (2000*), Synchronization in software radios. Carrier and timing recovery using FPGAs*, In Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on ,pp. 195-204.

Digilent, *Atlys Spartan 6 FPGA Development Board,* http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,836&Prod=ATLYS&CFID=6213559&CFTOKEN=eda57d21938ec1f6-66D78514-5056-0201-0295F3697755FE57

Dovis, F., Spelat, M., Mulassano, P., and Leone, C. (2005) , *On the Tracking Performance of a Galileo/GPS Receiver Based on Hybrid FPGA/DSP Board*, Proc, 2005 ION GNSS Conf., Institute of Navigation, Long Beach, CA.

Downey, J. A., Reinhart, R. C., & Kacpura, T. J. (2012), *Pre-flight Testing and Performance of a Ka-band Software Defined Radio*, American Institute of Aeronautics and Astronautics paper.

Elbert, B. R. (2008), *Introduction to satellite communication*, Artech house Third Edition.

ELECTROFRIENDS (2012), *DPSK Modulation and Demodulation,* http://electrofriends.com/articles/electronics/analog/dpsk-modulation-and-demodulation/

Ellis, P., Jaris, S., Ahn, A. D. I. S., & Lu, Y. (2011), *Implementation of Software-Defined Radio Using USRP Boards*, Bradley University.

Enevoldsen, S. V., Johansen, J. G., & Pucci, V. (2011), *Analysis and architectural mapping of an fft algorithm into an already existing fpga firmware of a low-cost cots sdr peripheral*, Aalborg University.

Ettus Research, *Application Note Low-Cost, Wide-Band RF Recording and Spectral Analysis with a USRP Device*,

http://www.ettus.com/content/files/kb/application_note_usrp_and_hdsdr_spectrum_monitoring.pdf

Ettus Research, *Application Note Selecting a USRP Device*,
http://www.ettus.com/content/files/kb/application_note_selecting_a_usrp.pdf

Ettus Research, *Application Note Design Process for Smart, Distributed RF Sensors Ettus Research*,
http://www.ettusresearch.com/content/files/kb/application_note_distributed_rf_sensors.pdf

Ettus Research, *USRP N210*, https://www.ettus.com/product/details/UN210-KIT

Fletcher, B. H. (2005, March), *FPGA embedded processors: revealing true system performance*, In Embedded Systems Conference.

G. J. Minden, J. B. Evans, L. Searl, et al.(2007), *KUAR: a flexible software-defined radio development platform*, in Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '07), pp. 428–439, Dublin, Ireland, April 2007.

Grillenberger, A., Rivas, R., Markgraf, M., Mumford, P.J. and Parkinson, K.J. (2008),*THE NAMURU RECEIVER AS DEVELOPMENT PLATFORM FOR SPACEBORNE GNSS APPLICATIONS*, Navitec 2008, Noorwijk, The Netherlands.

Grondin, M.; Belasic, M.; Ries, L.; Issler, J-L; Bataille, P.; Jobey, L.; Richard, G.(2010), *A new operational low cost GNSS software receiver for microsatellites,* Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on , vol., no., pp.1,5, 8-10.

Guo, P., Qi, X., Xiao, L., & Zhou, S. (2012), *A novel GPP-based Software-Defined Radio architecture*, Communications and Networking in China (CHINACOM), pp. 838-842.

Harada, H. (2005, November), *Software defined radio prototype toward cognitive radio communication systems*, In New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on, pp. 539-547.

Homan, D., & Young, Q. (2008), *The Challenges of Developing an Operational Nanosatellite,* Lockheed Martin Space Systems Company.

Hsiong, W., Huntzicker, S., King, K., Lee, A., Lim, C., Wang, J., ... & Jahn, J. M. (2009, August), *Performance and area tradeoffs in space-qualified FPGA-based time-of-flight systems*, In Electronic Measurement & Instruments, 2009. ICEMI'09. 9th International Conference on (pp. 2-1).

Johnson, S. K., Reinhart, R. C., Kacpura, T. J.(2012), *CoNNeCT's Approach for the Development of Three Software Defined Radios for Space Application,* Aerospace Conference 2012 IEEE, Big Sky, MT.

Klofas, B., Anderson, J., & Leveque, K. (2008, April), *A survey of cubesat communication systems*, In 5th Annual CubeSat Developers' Workshop.

Konecny, G. (2004, July*), Small satellites–A tool for Earth observation?*, InXXth ISPRS Congress-Commission (Vol. 4).

Malsbury, J., & Ettus, M. (2013, August), *Simplifying FPGA design with a novel network-on-chip architecture*, In Proceedings of the second workshop on Software radio implementation forum, pp. 45-52.

Mandeep, J. S., & Mustapha, H. (2013). *Design and Analysis of Dipole and Monopole Antenna for CUBESAT Application*, Research Journal of Applied Sciences, Engineering and Technology 6(17), pp. 3094-3097

MathWorks (2014), *Digital Modulation*,
http://www.mathworks.com/help/comm/ug/digital-modulation.html

MathWorks (2014), *FPGA Targeting with USRP N210 Hardware,*
http://www.mathworks.com/help/supportpkg/usrpradio/examples/fpga-targeting-with-usrp-r-n210-hardware.html

MathWorks (2014)*, USRP Support Package from Communication Package Toolbox,* http://www.mathworks.com/hardware-support/usrp.html

Maurer, R. H., Fraeman, M. E., Martin, M. N., & Roth, D. R. (2008), *Harsh Environments: Space Radiation*, Johns Hopkins APL Technical Digest, 28(1), 17.

MicroHard Systems, *Spread Spectrum 2.4GHz Industrial Wireless Modem*, http://www.microhardcorp.com/MHX2420.php

Military Embedded Systems (2012), *Rad-hard SRAM FPGAs enable vast improvements in space*, http://mil-embedded.com/articles/rad-hard-vast-improvements-space-exploration/

MOUSER ELECTRONICS (2006), *Lyrtech SFF SDR Development Platform*, http://www.mouser.com/search/refine.aspx?Ntk=P_MarCom&Ntt=136078009

NI, (2007), *OQPSK.*http://www.ni.com/white-paper/5487/en/

NI. (2014), *Frequency Modulation (FM).*

Oliveri, Steven J. (2011), *Modular FPGA Based software defined radio for CubeSats*, Masters Dissertation, Worcester Polytechnic Institute.

Plishker, W., Zaki, G. F., Bhattacharyya, S. S., Clancy, C., & Kuykendall, J. (2011, May), *Applying graphics processor acceleration in a software defined radio prototyping environment*, In Rapid System Prototyping (RSP), 2011 22nd IEEE International Symposium on , pp. 67-73.

Podosinov Vlodymyr S. (2011),*A Hybrid DSP and FPGA System for Software Defined radio Application*, Masters Dissertation, Department of Electrical Engineering, Virginia Polytechnic Institute and State University

Rivera, J. (2013), *Gartner Identifies the Top 10 Strategic Technology Trends for 2014.*

Rogers, A. Q., & Summers, R. A. (2010), *Creating Capable Nanosatellites for Critical Space Missions*, Johns Hopkins APL Technical Digest, 29(3), 283-288.

Rudra, A. (2004), *FPGA-based applications for software radio*, RF Design Magazine, pp. 24-35.

Šćekić (2005), O. *FPGA Comparative Analysis*, Belgrade.

Seifu, K. Z., & Law, C. T.(2008), *Communication Satellites: Frequency Band limitations Versus Implementations*.

Skey, K., Bradley, J., & Wagner, K. (2006, October), *A reuse approach for FPGA-based SDR waveforms*, In Military Communications Conference, 2006, MILCOM 2006. IEEE, pp. 1-7.

TECHPOWERUP (2014), *NVIDIA GEForce 6600 GT AGP*, http://www.techpowerup.com/gpudb/115/geforce-6600-gt-agp.html

Telecommunications & integrated applications ESA, *SATELLITE FREQUENCY BANDS*, http://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/Satellite_frequency_bands

Traussnig, W. (2007), *Design of a communication and navigation subsystem for a CubeSat mission*, University of Graz, Austria

Tresvig, J. L. (2010), *Design of a prototype communication system for the CubeSTAR nano-satellite*, Department of Physics, University of Oslo

Trusculescu, M., Pandele, A., Dragasanu, C., Piso, M. I., & Balan, M. (2012), *Nanosatellites: The Tool for Earth Observation and Near Earth Environment Monitoring*, INTECH Open Access Publisher.

Vulcan Wireless Inc (2010)**,** *Software-Defined Radio and Sensors*, http://www.vulcanwireless.com/products/csr-sdr-uu

WARP WIRELESS OPEN –ACCESS RESEARCH PLATFORM *WARP FPGA Board Overview* http://warpproject.org/trac/wiki/HardwareUsersGuides/FPGABoard_v1.2

Watermeyer, K**.** (2007), *Design of a hardware platform for narrow-band Software Defined Radio applications*, Department of Electrical Engineering, Doctoral dissertation, University of Cape Town.

Wong, T. F., & Lok, T. M. (2004), *Theory of Digital Communication*, University of Florida.

X86 CPUS' GUIDE **(**2010), *AMD 29050-40 PGA*, http://www.x86-guide.com/en/cpu/AMD-29050-40-PGA-cpu-no4259.html

Xilinx, *XtremeDSP Development Platform Spartan-3A DSP 3400A Edition*, http://www.xilinx.com/products/boards-and-kits/HW-SD3400A-DSP-DB-UNI-G.htm

Yoon, Z., Frese, W., Bukmaier, A., & Brieß, K. (2014), *System design of an S-band network of distributed nanosatellites*, CEAS Space Journal, 6(1),pp. 61-71.