2011

A Software Radio Based Ionosonde Using GNU Radio

Yan Nie

A Software Radio Based Ionosonde Using GNU Radio

(Thesis format: Monograph)

by

Yan Nie

Graduate Program
in
Department of Electrical and Computer Engineering
Faculty of Engineering

2

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor

____
Dr. John W. MacDougall

Supervisory Committee

____
Dr. Alan Webster

____
Dr. Don R. Moorcroft

Examiners

____
Dr. Alan Webster

____
Dr. Serguei Primak

____
Dr. Brown Peter

____

The thesis by

Yan Nie

entitled:

A Software Radio Based Ionosonde using GNU Radio

is accepted in partial fulfilment of the
requirements for the degree of
Master of Engineering Science

Date______________________________ ______________________________
Chair of the Thesis Examination Board

Abstract

The Canadian Advanced Digital Ionosonde (CADI) is used to study and investigate the structure and motion of the ionosphere.

The main components of CADI are implemented in microcontroller based digital logic. Due to the increased speed and reconfiguration capability of modern FPGAs (Field Programmable Gate Arrays) and ADCs (Analog-Digital Converters), this project is aimed to develop an ionosonde based on an open source radio software platform, GNU Radio, in conjunction with its hardware support, the Universal Software Radio Peripheral (USRP). The lowest cost FPGA, Cyclone EP1C12Q240C8 is selected to control Analog Digital Converter to transmit a HF (High Frequency) signal and to decimate and down convert the signal at the receiver side, which is controlled by a USB controller, and to load the FPGA configuration image through GNU radio platform. To obtain a good range resolution and low noise level, the signal is modulated by a 113-bit Legendre sequence with a preamble packed at front for synchronization. Based on the experimental results, the discussion and conclusion are included.

Keywords: GNU Radio, Ionosonde, USRP, FPGA

Acknowledgements

I would like to extend my deepest appreciation to my supervisor, Prof. John. W. MacDougall for his patient guidance of my thesis project, and his kindness, understanding, and encouragement.

Much appreciation go to the GNU Radio develop group for their online response of the questions related to my project. Thanks go to Tomasz Rybak for his help through my projects. I would like also thank staffs in Electronic shop of Electrical Engineering Department, and Phin Perquin and Brian Dalrymple in Physics department for their support.

Special thanks to my parents for their constant moral support.

Table of Content

List of Tables

List of Figures

Acronyms

AD	*Analog-Digital*
ADC	*Analog-Digital-Converter*
AM	*Amplitude Modulation*
ASICs	*Application Specific Integrated Circuits*
API	*Application Programming Interface*
BPSK	*Binary Phase Shift Key*
CADI	*Canadian Advanced Digital Ionosonde*
CIC	*Cascaded Integrator-Comb*
CW	*Continuous Wave*
DDC	*Digital Down Conversion*
DDS	*Direct Digital Synthesizer*
DEMUX	*Demultiplexer*
DSP	*Digital Signal Processors*
DUC	*Digital Up Conversion*
EM	*Electromagnetic*
EUV	*Extreme Ultraviolet*
FIFO	*First-In-First-Out*
FM	*Frequency Modulation*
FPGA	*Field Programmable Gate Array*
GRC	*GNU Radio Companion*
GPL	*General Public License*
GPS	*Global Positioning*
GUI	*Graphical User Interface*
HDTV	*High-Definition Television*
HF	*High-Frequency*
IF	*Intermediate frequency*

ISL	*Integrated Sidelobe Level*
LO	*Local Oscillator*
MCU	*Microcontrollers*
MIMO	*Multi-Input Multi-Output*
NCO	*Numerical Controlled Oscillator*
OFDM	*Orthogonal Frequency Division Multiplexing*
PGA	*Programmable Gain Amplifier*
P-P	*Peak-to-Peak*
PPS	*Pulse Per Second*
PRI	*Pulse Repetition Interval*
PSL	*Peak Sidelobe Level*
QoS	*Quality of Service*
RF	*Radio Frequency*
RTL	*Resistor Transfer Level*
Rx	*Receiver*
SDR	*Software Defined Radio*
SNR	*Signal-to-Noise Radio*
SSB	*Single Side Band*
SWIG	*the Simple Wrapper Interface Generator*
TEC	*Total Electron Content*
Tx	*Transmitter*
UHD	*Universal Hardware Driver*
USRP	*Universal Software Radio Peripheral*

Chapter 1

Introduction

1.1 Introduction to the Earths Ionosphere

The ionosphere is the partially ionized layer of the upper atmosphere, which reflects HF radio waves, yielding the possibility of long-distance radio communications. The ionosphere approximately resides in the altitude ranging from 50 km to 600 km and higher. It is primarily created by photoionization by extreme ultraviolet (EUV) light from the sun and x-ray photons, which is partially absorbed by the atmosphere and dislodges an electron from the neutral atom or molecule during a collision in the earth's atmosphere. Impact ionization by energetic charged particles is the other ionization process involved in the formation of the ionosphere. The variation in composition, electron density, ionization source, variability and dynamic effects causes the low-altitude ionosphere to be structured in three layers or regions, which are named D region (50km to 90km), E region (90km to 150km), and F region (150km and 600km). The electron density in the typical daytime ionosphere varies with altitude and experiences an exponential decline above about 250km. The daytime F region has two peaks demonstrated in F region, resulting in the F region being split into two separate regions, F_1 (150km to 250km) and F_2 (250 km to 600km), which are identified in Fig 1.1. During nighttime (Fig. 1.2), F_1 region and F_2 region unite into a single F region. This F region is the only major surviving region in the ionosphere during nighttime. However, a peak in electron concentration sporadically occurs at mid-latitudes approximately covering the altitude between 90km and 130km, which is called the Sporadic E region.

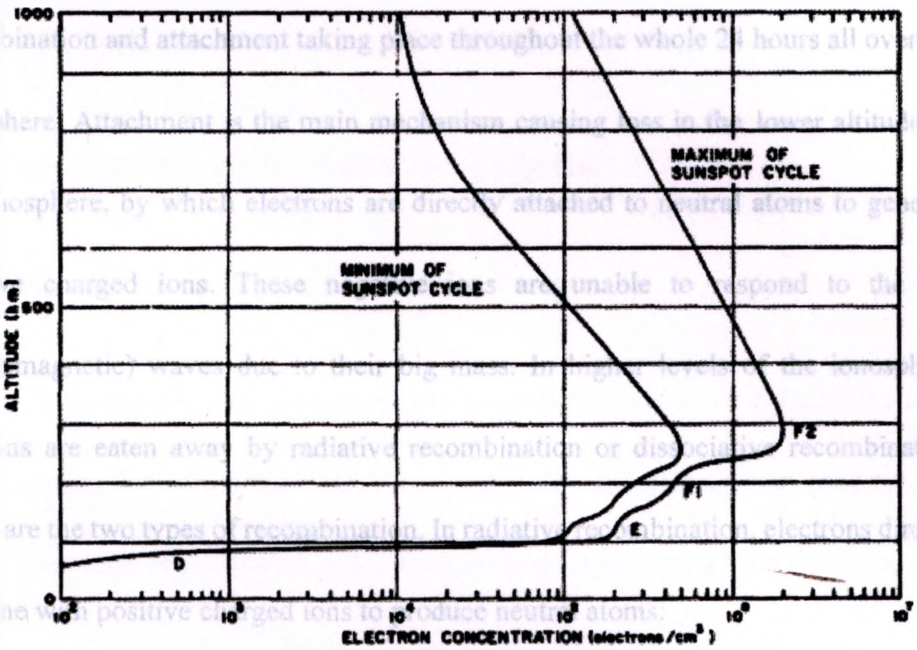


Fig. 1.1 Normal electron distributions at daytime [Johnson, 1965]

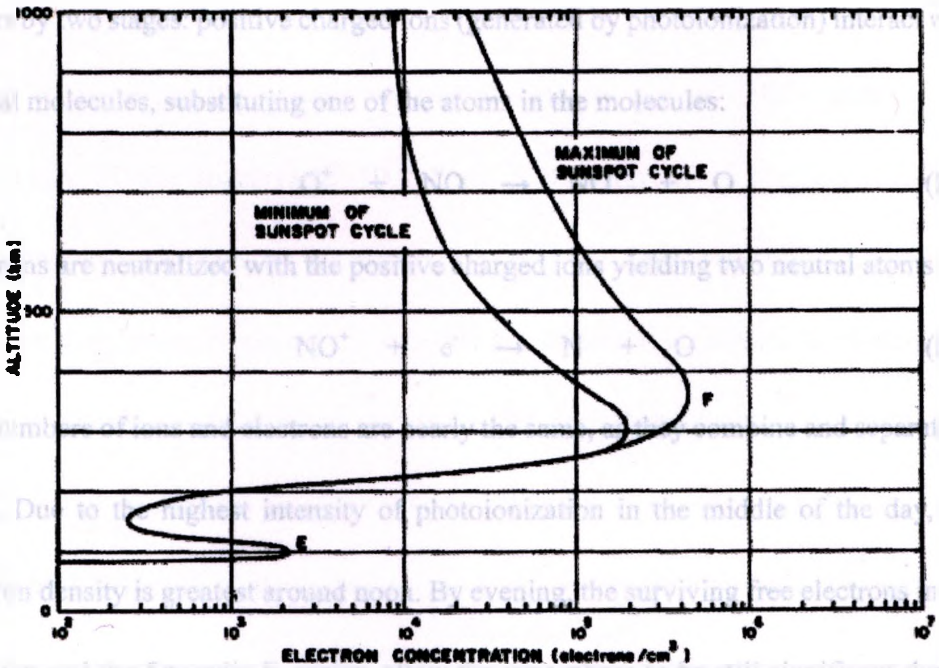


Fig. 1.2 Normal electron distributions at nighttime [Johnson, 1965]

The variations of electron density in daytime and at nighttime are both the results of the imbalance between photoionization that occurs only during the day and the recombination and attachment taking place throughout the whole 24 hours all over the ionosphere. Attachment is the main mechanism causing loss in the lower altitudes of the ionosphere, by which electrons are directly attached to neutral atoms to generate negative charged ions. These negative ions are unable to respond to the EM (electromagnetic) waves due to their big mass. In higher levels of the ionosphere, electrons are eaten away by radiative recombination or dissociative recombination, which are the two types of recombination. In radiative recombination, electrons directly combine with positive charged ions to produce neutral atoms:

$$O^{+} + e^{-} \rightarrow O + h\gamma \qquad \textbf{(1.1)}$$

where $h\gamma$ represents a photon. In dissociative recombination, the loss of electrons occurs by two stages: positive charged ions (generated by photoionization) interact with neutral molecules, substituting one of the atoms in the molecules:

$$O^{+} + NO \rightarrow NO^{+} + O \qquad \textbf{(1.2)}$$

Electrons are neutralized with the positive charged ions yielding two neutral atoms:

$$NO^{+} + e^{-} \rightarrow N + O \qquad \textbf{(1.3)}$$

The numbers of ions and electrons are nearly the same, as they combine and separate in pairs. Due to the highest intensity of photoionization in the middle of the day, the electron density is greatest around noon. By evening, the surviving free electrons in the F region and the Sporadic E region allow the ionosphere to be still significant during nighttime. The ionospheric characteristics also exhibit significant variations in location, local time, solar activities (solar sunspot cycle, solar wind), and magnetic storm time [Hunsucker, 1991].

1.2 Ionospheric radio wave propagation

1.2.1 Plasma Frequency and Critical Frequency

In terms of plasma physics, the ionosphere can be considered as a cold lightly ionized plasma superposed within the Earth's magnetic field. The plasma frequency f_N is the natural frequency of oscillation of the free electrons which have been displaced from the heavy positive ions and allowed to move freely thereafter [Davies, 1969], which is given by:

$$f_N^{2} = \frac{Ne^2}{4\pi^2 \varepsilon_0 m} = 80.5N \ \ (Hz^2) \qquad \mathbf{(1.4)}$$

Where N represents the number of electron density per m^3 (MKS units), ε_0 is permittivity of free space (8.85×10^{-12} F/m), e is charge of an electron (1.6×10^{-19} coulombs), and m is rest mass of electron (9.1×10^{-31} kg). Radio waves with frequencies less than the plasma frequency in a region are reflected, at vertical incidence. The maximum plasma frequencies are defined as the critical frequencies (*fc*) of different layers in the ionosphere. A wave whose frequency is equal or smaller than this frequency is reflected in the corresponding layer, otherwise it penetrates to the layer above.

1.2.2 Gyro-frequency

In a uniform external magnetic field **B**, electrons and ions execute a spiral gyration moving in a helix with axis direction along the magnetic field **B**, with a frequency, which is named gyro-frequency f_H, and is given by:

$$f_H = \frac{|e|}{2\pi m}\mathbf{B} \qquad \textbf{(1.5)}$$

where $|e|$ and m are the charge and mass of the particle, respectively. Since the mass of electrons is much smaller than the mass of ions, the electron gyro-frequency affects the propagation of HF waves in the ionosphere more than ion gyro-frequency. The effect of the electrons in propagation will be shown by equations 1.6 to 1.9.

1.2.3 Collision frequency of electrons

In the ionosphere, electrons and heavier particles (ions and neutrals) affect each other by effective collisions. An increasingly number of collisions gives the ionosphere more equilibrium. Simply defined, the collision frequency is the number of collisions between particles each second. The collision frequency is contingent on the density, velocity and mass of particles. The absorption of radio waves is due to the loss of energy because of collisions of electrons and particles in the ionosphere. Since the collision frequency is higher in the D region, this means the absorption is relatively high in this region.

1.2.4 Refractive index

In a neutral medium with a uniform external magnetic field, the complex refractive index n of radio wave is given by the Appleton formula [Davies, 1990]:

$$n^2 = (\mu - i\chi)^2 = 1 - \frac{X}{1 - iZ - \frac{Y_T^2}{2(1-X-iZ)} \pm \left[\frac{Y_T^4}{4(1-X-iZ)} + Y_L^2\right]^{1/2}}, \qquad (1.6)$$

where:

f = frequency of the "exploring wave"

f_N = plasma frequency

f_H = gyro-frequency = $\frac{B_0|e|}{2\pi m}$

f_L = the transverse component of gyro-frequency = $\frac{B_0|e|}{2\pi m}\cos\theta$

f_T = longitudinal component of gyro-frequency = $\frac{B_0|e|}{2\pi m}\sin\theta$

υ = electron collision frequency

θ = angle between the magnetic field and direction of phase propagation

$X = (f_N / f)^2$, $Y = f_H / f$, $Y_L = f_L / f$, $Y_T = f_T / f$, $Z = \upsilon / f$,

χ = absorption coefficient

In the absence of the magnetic field, $Y_L = Y_T = 0$, the refractive index is:

$$n^2 = 1 - \frac{X}{1-iZ} \qquad (1.7)$$

With the negligible absorption, in the E region and the F region, the electron collision frequency can be neglected, and the refractive index is given:

$$\mu^2 = 1 - \frac{2X(1-2X)}{2(1-X) - Y_T^2 \pm \left[Y_T^2 + 4(1-X)^2 Y_L^2\right]^{1/2}} \qquad (1.8)$$

To reflect a plane wave, the vertical incidence reflection condition can be obtained by setting $\mu = 0$, which yields $X = 1$ when positive sign is considered, and $X = 1 - Y$ or $X = 1 + Y$ when negative sign is taken. The positive sign gives reflection independent of the magnetic field for a wave reflected at any height, and is named the ordinary wave consequently. The wave that depends on the magnetic field, by using the negative sign, is called the extraordinary wave.

For an ordinary plane radio wave in the E region or F region, which means that both collision and magnetic field are neglected, the refractive index becomes:

$$\mu = \sqrt{1 - (\frac{f_N}{f})^2} \tag{1.9}$$

where f_N is plasma frequency, and f is radio wave frequency.

1.2.5 The reflection condition

Considering a horizontally plane-stratified ionosphere with negligible magnetic effects and collision depicted in Fig. 1.3, applying the Snell's law for a radio wave oblique incident on the ionosphere with angle ϕ_0 gives:

$$\mu_0 \sin \phi_0 = \mu \sin \phi = \mu_r \sin \phi_r \tag{1.10}$$

The oblique incident radio wave reflects when $\phi_r = 90°$. The refractive index below the ionosphere, μ_0, can be set to 1 for simplification. Consequently, the reflection condition of an oblique incident radio wave is $\mu_r = \boldsymbol{sin}\ \phi_0$, which becomes $\mu_r=0$ for vertical incidence.

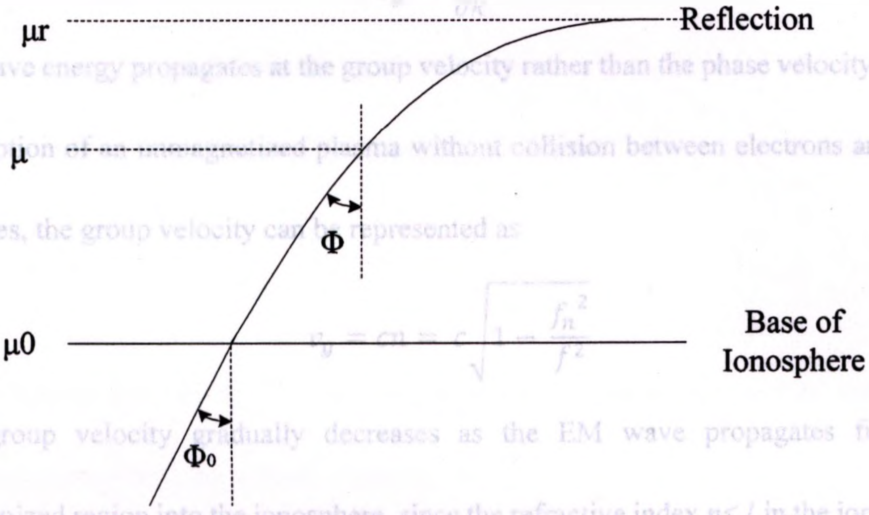


Fig. 1.3 Reflection of a Radio Beam in Horizontally Plane-stratified Ionosphere

1.2.6 Phase velocity and group velocity

The phase velocity of a wave is the speed at which any fixed phase of the wave is travelling in the transmitting direction. By definition, the phase velocity of a light wave is represented as:

$$v_P = {}^{\omega}/_{k} = {}^{1}/_{\sqrt{\mu\varepsilon}} \qquad \textbf{(1.11)}$$

Where ω = the angular frequency of the traveling wave

k = the wave number of the travelling wave

μ = permeability (in free space = $4\pi \times 10^{-7}$ Henrys/m)

ε = permittivity (in free space = $1/36\pi\times 10^{-9}$ Farads/m)

in free space υ_P equals to the speed of light, $v_{P_0} = c = {}^{1}/_{\sqrt{\mu_0\varepsilon_0}} = 3\times 10^8 m/s$

The energy transport is through a wave envelope governed by the group velocity which is given as:

$$v_g = \frac{\partial \omega}{\partial k} \qquad \textbf{(1.12)}$$

The wave energy propagates at the group velocity rather than the phase velocity. On the assumption of an unmagnetized plasma without collision between electrons and other particles, the group velocity can be represented as

$$v_g = cn = c\sqrt{1 - \frac{f_n^{\ 2}}{f^2}} \qquad \textbf{(1.13)}$$

The group velocity gradually decreases as the EM wave propagates from the non-ionized region into the ionosphere, since the refractive index *n<1* ~~in the~~ ionosphere. With increasing height for a wave vertically travelling within the ionosphere, the group velocity declines to 0 as the electron distribution grows denser. When the group velocity becomes 0, the frequency of the wave equals the plasma frequency at this height, and the wave is reflected.

1.2.7 Virtual height

Assume a radio wave with the frequency of f_v vertically propagates into the ionosphere at the speed of light, it penetrates through all layers of the ionosphere when the wave frequency is higher than the plasma frequency. Conversely, the radio wave with the frequency lower than plasma frequency can suffer fairly heavy absorption in D region at daytime.

HF (High-Frequency, 3-30MHz) radio techniques are commonly used for the ionosphere probing. Radio waves whose frequencies equal the plasma frequencies in the ionosphere can be reflected back to the earth, and the altitude of the reflection point is called the 'real height'. Since the radio wave travels slower than the velocity of light in the ionosphere according to equation (1.9), the virtual height is greater than the real height as depicted in Fig 1.4.

If the measurable travelling time and velocity of the radio wave are *t* and the velocity of light is *c* respectively, the virtual height is:

$$H_v = \frac{1}{2}ct \qquad \textbf{(1.14)}$$

In oblique propagation, shown in Fig 1.4, the frequency f_o of a wave reflected at a given real height can be derived from the frequency f_v of a radio wave reflected at the same real height:

$$f_o = f_v \sec \varphi_o \qquad \textbf{(1.15)}$$

where ϕ_o is the oblique incident angle in Fig 1.4.

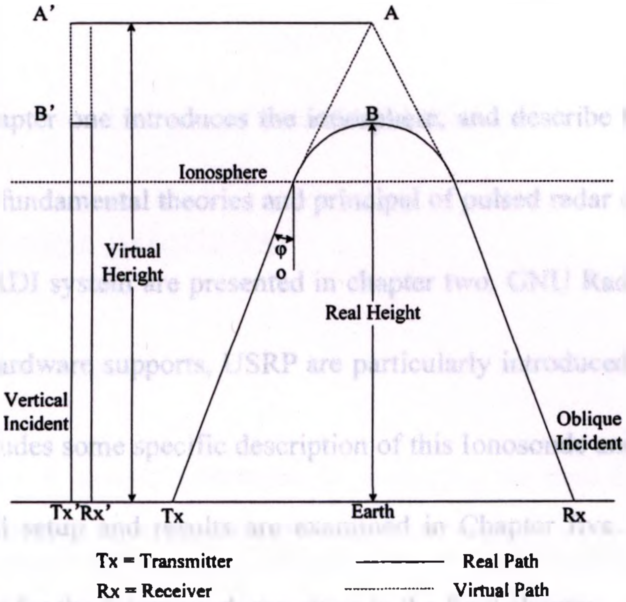


Fig. 1.4 A Comparison of Real Height and Virtual Height

1.3 Overview of thesis objective

In order to explore and study the motion and structure of the ionosphere, CADI (Canadian Advanced Digital Ionosonde), an ionospheric radar, has been developed to measure the virtual height, drift velocity, angle of arrival, echo intensity, etc. The major component of CADI is developed in microcontroller based digital logic. Due to the main feature of GNU Radio software toolkit, open source, and the increased speed and reconfiguration ability of the modern FPGAs (Field Programmable Gate Array) and ADCs (Analog-Digital Converters) on the hardware support of GNU Radio, USRP (Universal Software Radio Peripheral), the objective of this project is develop and test a low cost, low power, and software radio based Ionosonde to which is equivalent to CADI in terms of performance, to replication the typical CADI measurements.

In this thesis, chapter one introduces the ionosphere, and describe the motivation of this project. The fundamental theories and principal of pulsed radar design and a brief illustration of CADI system are presented in chapter two. GNU Radio in conjuncture with one of its hardware supports, USRP are particularly introduced in chapter three. Chapter four includes some specific description of this Ionosonde and its development. The experimental setup and results are examined in Chapter five. Conclusions and recommendations for the future work are given in the final chapter.

Chapter 2

RADAR design principles and CADI system

2.1 Radar design principles

2.1.1 Range to a target

The target's range, *R*, can be determined by measuring the time delay, *Δt*, from the radar's transmission to the detection of the target's echo. Assuming that radio waves travel at the velocity of light, with the assumption of the monostatic radar (transmitter and receiver are co-located at the same place), the range of the detected target is represented as:

$$R=\frac{c\times\Delta t}{2} \qquad \textbf{(2.1)}$$

where *Δt* is the round-trip time delay. This is considered as a fundamental theoretical relation utilized in ionosondes (ionospheric radars) and other RADARs to achieve height measurement.

Commonly, a pulsed radar transmits and receives a sequence of pulses of RF (Radio Frequency) energy. The range ambiguity is associated with the Pulse Repetition Interval (PRI), T, indicated in Fig 2.1. To avoid the range ambiguity the PRI needs to suffice for a pulse transmission and echo returning in terms of time. Actually, the maximum non-ambiguous range is:

$$R_{max}=c\times\frac{T}{2} \qquad \textbf{(2.2)}$$

For a monostatic radar utilized for ionosphere probing, the range measurement corresponds to measuring the virtual height of a plane of plasma whose plasma is equal to the radar frequency. Similarly, range measurement in multi-static ionospheric radar (transmitter and receiver deployed at different places so that radio waves propagate at oblique incidence into the ionosphere) is also a detection of the half length of the virtual path.

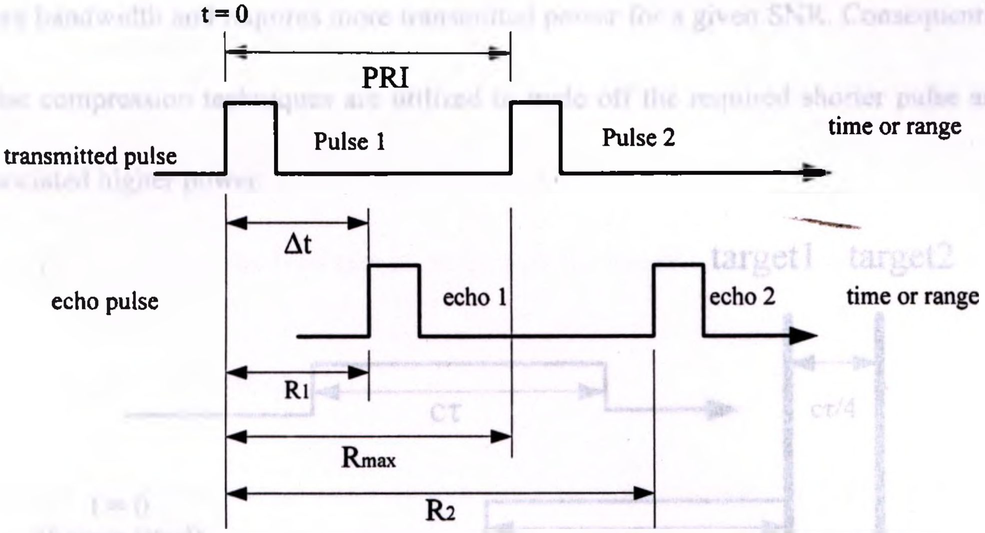


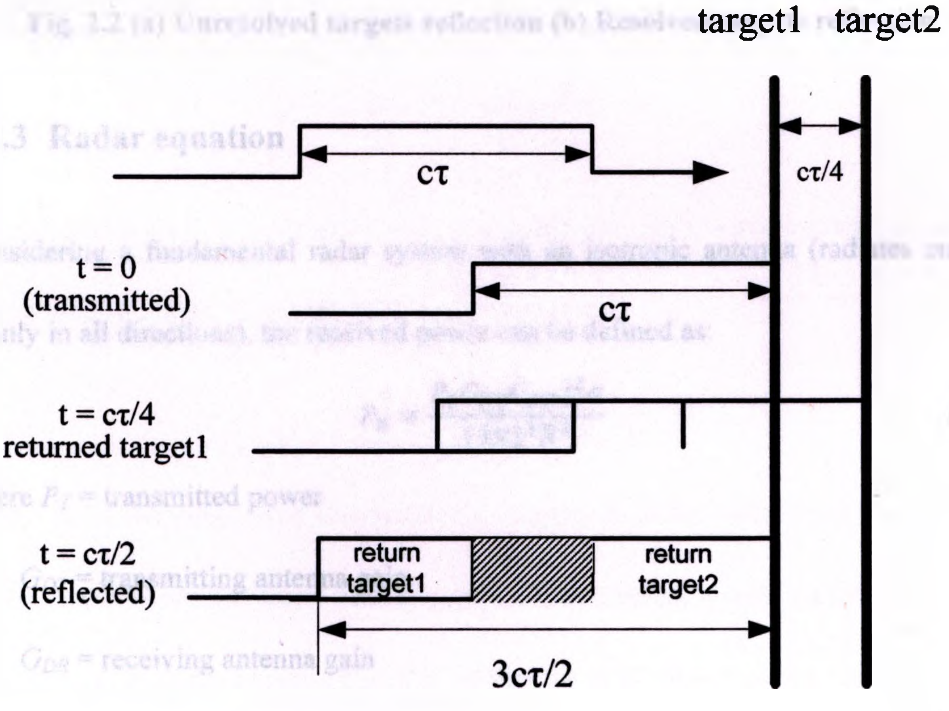
Fig. 2.1 Fundamental principle of range measurement in pulsed radar

2.1.2 Range resolution

Range resolution, denoted as *ΔR*, is a range on precision evaluation criterion for a radar system. It indicates the radar system's detection ability two targets to be seen as distinct objects, which is obtained by measuring the smallest range difference of two detectable targets. When two targets are separated by less than *cτ/2*, (τ is the pulse width), the echoes overlap each other. The processes of the reflection of the two pulse signals from the unresolvable targets are illustrated in Fig. 2.2 (a). The distance of two resolvable targets needs to be greater than *cτ/2* to avoid the ambiguities of their echoes (Fig. 2.2 (b)). Therefore, the range resolution can be described as:

$$\Delta R = min\left[\frac{1}{2}c(\Delta t_2 - \Delta t_1)\right] = \frac{c\tau}{2} = \frac{c}{2B} \qquad \textbf{(2.3)}$$

where B = 1/τ is the bandwidth of the pulse. Minimizing the range resolution to improve the radar performance requires reducing the pulse width, which occupies more bandwidth and requires more transmitted power for a given SNR. Consequently, pulse compression techniques are utilized to trade off the required shorter pulse and associated higher power.



(a)

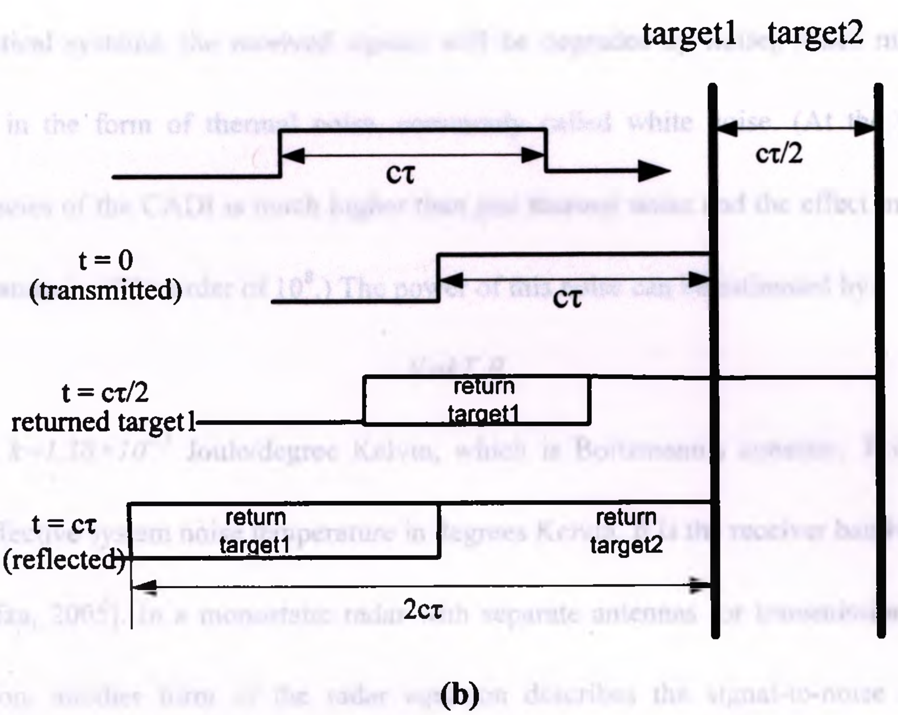


Fig. 2.2 (a) Unresolved targets reflection (b) Resolved targets reflection

2.1.3 Radar equation

Considering a fundamental radar system with an isotropic antenna (radiates energy evenly in all directions), the received power can be defined as:

$$P_R = \frac{P_T G_{DT} G_{DR} \lambda^2 \sigma}{(4\pi)^3 R^4} \qquad \textbf{(2.4)}$$

where P_T = transmitted power

G_{DT} = transmitting antenna gain

G_{DR} = receiving antenna gain

λ = wavelength of transmitted signal

σ = effective area of an isotropic target

R = the range of the detected target

In practical systems, the received signals will be degraded by noise, which mainly occurs in the form of thermal noise, commonly called white noise. (At the MHz frequencies of the CADI is much higher than just thermal noise and the effect in rose temperature is of the order of 10^8.) The power of this noise can be estimated by:

$$N=kT_sB \tag{2.5}$$

where: $k=1.38\times10^{23}$ Joule/degree Kelvin, which is Boltzmann's constant, T_s is the total effective system noise temperature in degrees Kelvin, B is the receiver bandwidth [Mahafza, 2005]. In a monostatic radar with separate antennas for transmission and reception, another form of the radar equation describes the signal-to-noise radio (SNR):

$$\frac{S}{N}=\frac{P_TG_{DT}G_{DR}\lambda^2\sigma}{(4\pi)^3R^4kT_sB} \tag{2.6}$$

If a radar is bistatic or multistatic, due to the different radio wave travelling distance for transmitter and receiver, the SNR can be written as:

$$\frac{S}{N}=\frac{P_TG_{DT}G_{DR}\lambda^2\sigma}{(4\pi)^3R_1{}^2R_2{}^2kT_sB} \tag{2.7}$$

where: R_1 is the range from transmitter to a target, R_2 is the range from the target to the receiver. As can be clearly seen from this SNR equation, increasing the transmitted power can effectively increase the SNR so that the detectability of the radar system is improved. However, increasing the power of radar systems is costly. Pulse compression techniques are extensively utilized to address this radar power dilemma. Long pulse coding, 1019-bit Legendre sequence and 13-bit Barker code have been implemented in CADI, which thereby achieves higher SNR and range resolution.

2.1.4 The Doppler shift

Radar systems utilize Doppler frequency to measure the radial velocity (range rate) of a moving target. Doppler shift characterizes the shift in the center frequency of a waveform due to the target motion with respect of the source of radiation [Mahafza, 2005].

Consider a radio wave with wavelength λ incident on a moving target at relative velocity v_d in the direction with the angle θ between radar line of sight and the target. The Doppler shift is approximated as:

$$f_d = \frac{2v_d}{\lambda} \cos\theta \tag{2.8}$$

As shown, the sign of the Doppler shift determines the direction of the target motion (negative Doppler targets moving away from the radar).

2.1.5 Pulse compression technique

As mentioned in the previous discussion, improving the range resolution for a radar can be accomplished by applying short pulses. However, using short pulse in radar system diminishes its average transmitted power, so that the SNR of the system will be degraded. A good detection radar requires high SNR which requires high average transmitter power. For a short pulse the average transmitted power will be low. Therefore, pulse compression techniques are introduced to eliminate this dilemma of desiring long pulse for obtaining higher system SNR and short pulse for improving the range resolution of a given radar.

Pulse compression is mainly implemented by frequency modulation and phase modulation. Because of its simplicity of implementation, Binary Phase Shift Key (BPSK) modulation is commonly utilized in radar system. Fig 2.4 depicts a general BPSK system. Consider a pulse with duration of τ divided in time into N chips, each of which is modulated either +1 or -1 with duration of $\tau' = \tau/N$, which respectively corresponds to the nominal carrier phase or π phase shift on a continuous wave (CW). And the amplitude of the kth chip is notated as a_k. Based on pulse compression techniques, applying the modulated pulse to a matched filter or correlator generates a compressed pulse with small effective duration. The autocorrelation is defined in discrete form: $\rho(m)=\sum_{k=1}^{N} a_k a_{k+m}$, where m is the index of the autocorrelation function, $-(N-1)\leq m\leq (N-1)$. In pulse radar receiver, the output of the matched filter is the autocorrelation result of the modulated pulse. The mainlobe of the result determines the time delay of an echo and the corresponding range of a target. Accordingly, a high detectability radar requires that the sidelobes be suppressed to the greatest extent. A criterion, *peak sidelobe level* (PSL) is introduced to evaluate the maximum sidelobe to the mainlobe ratio. Similarly, *the integrated sidelobe level* (ISL) represents the ratio of energy contained in sidelobe and mainlobe.

$$PSL = 10\,log\left[\frac{max\left(\rho_m{}^2\right)}{\rho_0{}^2}\right] \tag{2.9}$$

$$ISL = 10\,log\left[\sum_{i=-(N-1)}^{N-1} \frac{{\rho_m}^2}{{\rho_0}^2}\right] \qquad \textbf{(2.10)}$$

where: $m \neq 0$, and ρ_0 is the mainlobe level. Low PSL and ISL require that the chips of the pulse are coded by a sequence which gives a good autocorrelation result.

The goodness of compressed binary phase coding is determined by the random sequence of the phase for the individual subpulses. [Mahafza, 2005] Specifically, in CADI, 7-bit Barker code and 13-bit Barker code [Gao, 1991], and 1019-bit Legendre code [Huang, 2003] have been implemented, producing low PSL and improving the range resolution. These three sequences are defined as:

7-bit Barker: +1, +1, +1, -1, -1, +1, -1

13-bit Barker: +1, +1, +1, +1, +1, -1, -1, +1, +1, -1, +1, -1, +1

Legendre code:

$$L(n)=\left(\frac{n}{N}\right) = \begin{cases} +1; & n \text{ is a quadratic residue mod } N \\ -1; & \text{otherwise (n is a quadratic non-residue mod } N) \\ 1\, or -1; & n = 0 \end{cases}$$

where: N is an odd number represents the length of the sequence; n is the index of the sequence. In this project, 113-bit Legendre code is selected as the biphase coding sequence to implement pulse compression and to obtain a comparative high range resolution and SNR improvement. The SNR of a compressed pulse can be derived from radar equation 2.6 as:

$$\frac{S}{N} = N\frac{P_T G_{DT} G_{DR} \tau' \lambda^2 \sigma}{(4\pi)^3 R^4 k T_s} \qquad \textbf{(2.11)}$$

which indicates that increasing the total pulse length of the N chips upgrades the SNR N times for a given radar without change of radar parameters by utilizing pulse compression. Since the bandwidth is unchanged, the range resolution remains unchanged. A comparison of the specifications of pulse coding sequences implemented in CADI is listed in Table 2.1.

Table 2.1 Comparison on specification of sequences implemented in CADI

	Single Pulse	7-bit Barker	13-bit Barker	113-bit Legendre	1019-bit Legendre
Peak sidelobe level (unit)	N/A	1	1	7	24
PSL(dB)	N/A	-16.902	-22.2789	-24.159	-32.5593
ISL(dB)	N/A	-9.1204	-11.4871	-7.1171	-7.7358
SNR improvement (dB)	0	8.451	11.1	20.5	30.1

2.2 CADI system

The Canadian Advanced Digital Ionosonde (CADI) is a HF coherent pulse ionospheric radar which is used for the research and surveillance on the structure and motion of ionosphere. Experiencing several revisions, CADI has been developed as a small-size, low-cost, flexible, and multi-functional digital ionosonde by applying the phase coding technique, solid state electronic and PC technique. It can observe the phase and amplitude of the echoes, the echo delay (virtual height) as a function of frequency (the ionogram), the angle of arrival, the Doppler shift, and the electron density and drift can be derived from these measurements. The CADI network has been deployed to provide observation data for studies on polar cap convention, gravity waves, the auroral zone, the Sporadic-E region, and meteor tails etc. The location of the operating CADI network in Canada is shown as the red dots in Fig 2.3. In this figure, the red dots gives the locations where GPS (Global Positioning System) receiver for TEC (Total Electron Content) study and the CADI locate; the blue dots shows the location of only GPS receiver deployed; and the green dots are the proposed stations for CADI or GPS deployment in future.

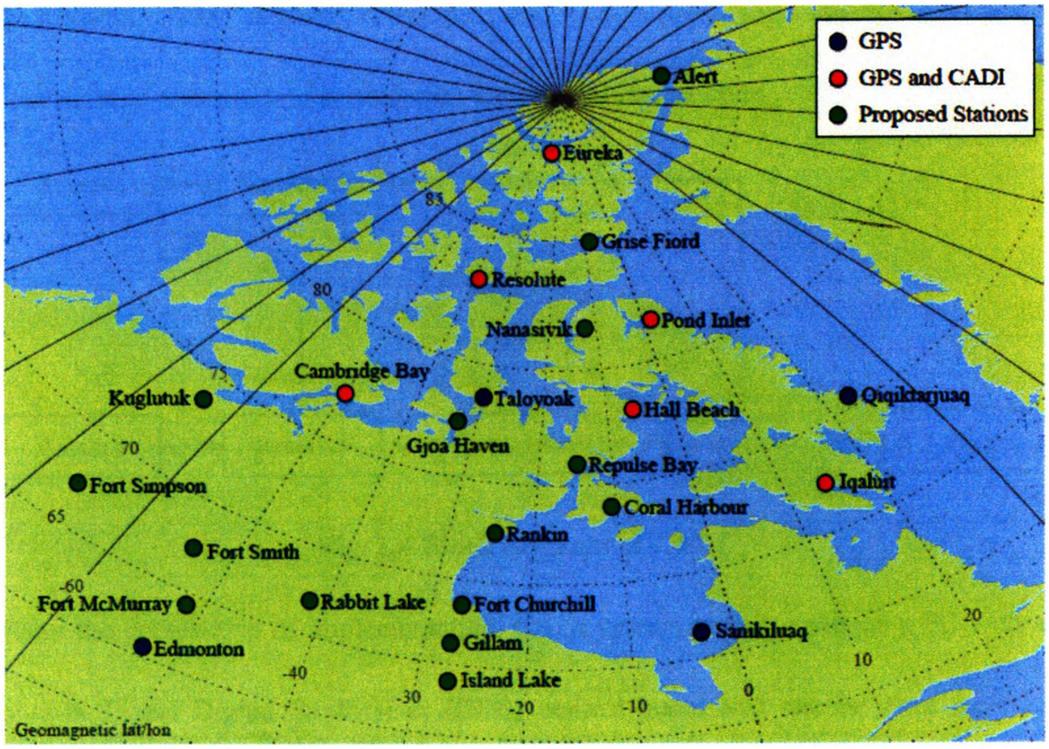


Fig. 2.3 Location of the operating CADI network in Canada

2.2.1 General description of the CADI system

The CADI system comprises of a transmitter and four receivers or less. The block diagram of the system is shown in Fig. 2.4. Due to the application of the pulse compression technique, the Barker correlator plays the most important role in the system, generating Barker code as the source code during signal transmission and reproducing the original code for use in correlation. After feeding the Barker code to the modulator, the BPSK modulated signal are up converted to a certain frequency ranging from 1MHz to 20MHz and are vertically transmitted by the HF band transmitter through a dipole half wavelength antenna or other types of antennas. The receiver obtains the echo signals reflected from ionosphere and down converts it to the IF frequency. The in-phase component and quadrature component are coherently demodulated and cross-correlated with the original Barker code.

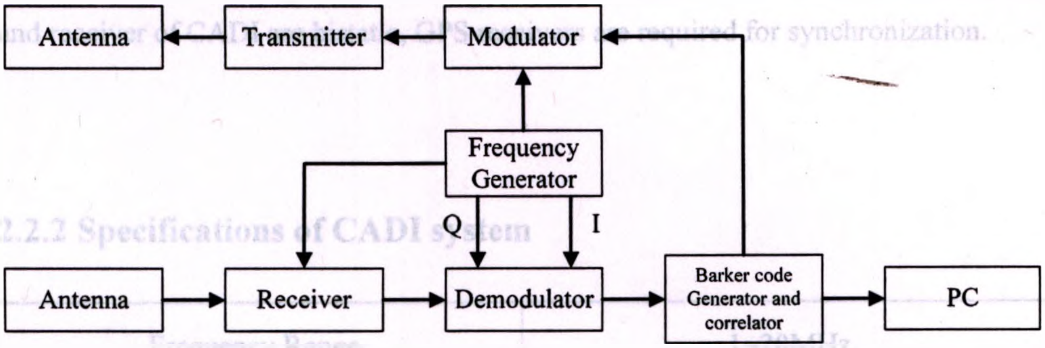


Fig. 2.4 Block diagram of CADI

From the perspective of implementation, CADI system mainly contains a PC with a plug-in Direct Digital Synthesizer (DDS) control board and one or more receiver boards, a power amplifier, antennas and GPS receivers. The system diagram of CADI system with one receiver board is shown in Fig. 2.5, where ‘Transmitter’ and ‘Receiver’ are respectively represented as ‘Tx’ and ‘RCVR’.

A standard PC performs as an interface for system configuration and data processing. By directly receiving the commands from the PC, the microcontrollers (MCU) provide the principal control for the whole system. These three MCUs in a one receiver only system are selected from Motorola 68HC11 family in the old version and 68HC12 family in the latest version. The MCU on DDS board works as a Barker generator and a modulator in the system while the other two microprocessors on the receiver board sample the demodulated signals and implement the correlation. DDS plays the role as a Local Oscillator (LO), generating the carrier signal at RF frequency for the transmitter and providing LO signals for Digital Down Conversion (DDC). The power amplifier amplifies the RF signal up to 600W at the transmitter side. When transmitter and receiver of CADI are bistatic, GPS receivers are required for synchronization.

2.2.2 Specifications of CADI system

Frequency Range	1~20MHz
Transmitter power	Max. 600W
Pulse width (chip duration)	40μs
Pulse Repetition Frequency	20 or 40Hz
External GPS trigger	Available
Range Resolution	6km
Maximum Measurement Range	1020km
Pulse compression	Complementary, Barker code
Number of Receivers	Up to 4

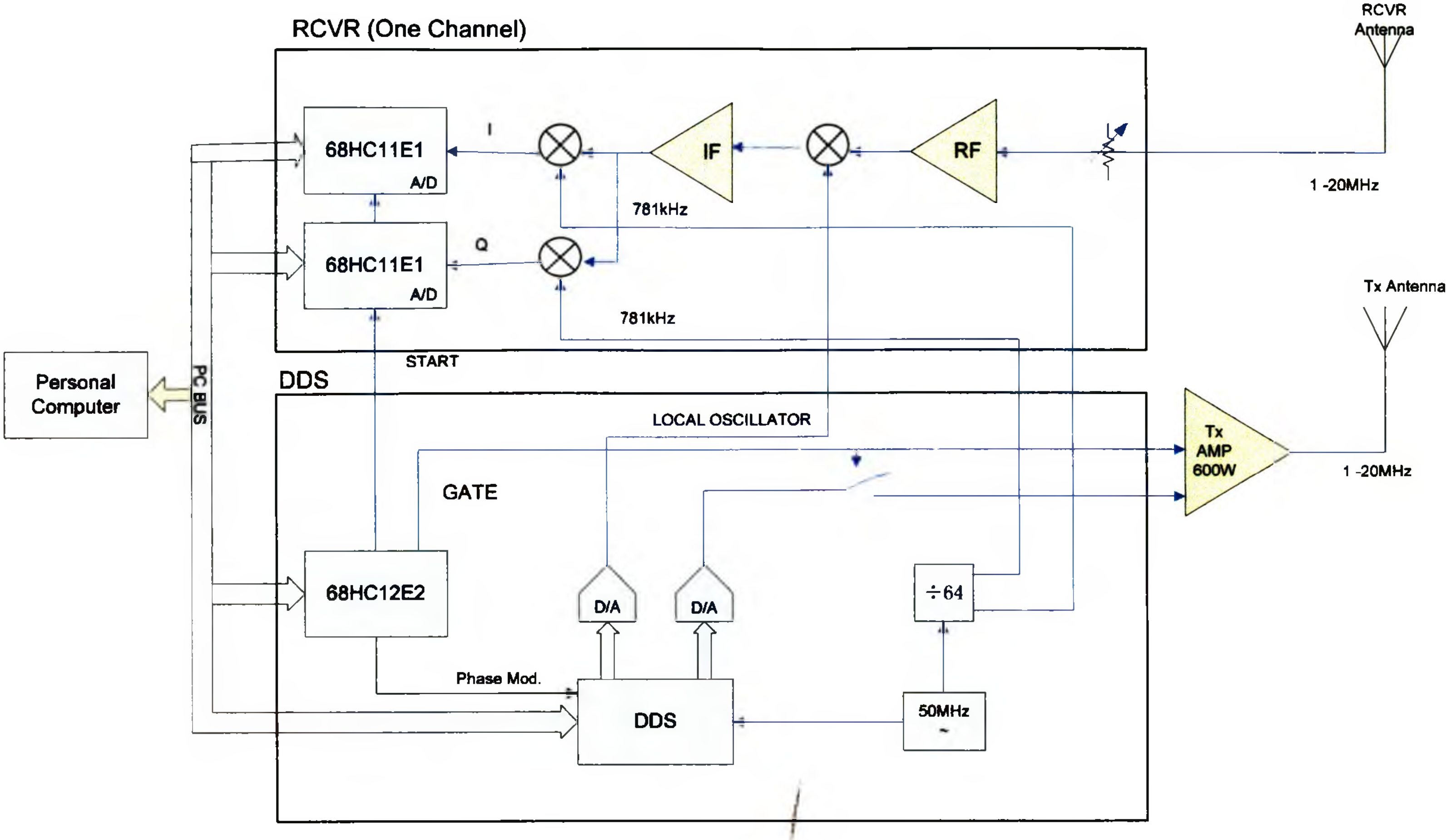


Fig. 2.5 System diagram of CADI system [MacDougall, 2000]

2.2.3 Signal processing in the transmitter and receiver of CADI

2.2.3.1 Signal in transmitter

By applying pulse compression technique and biphase modulation in the CADI, the signal transmitted can be represented as:

$$S(t) = B(t)\, cos(2\pi f_c) \tag{2.12}$$

where: *B(t)* is the compressed sequence with required amplitude

f$_c$ is the carrier frequency and also the RF frequency

2.2.3.2 Echo detection at receiver

As a coherent radar system, CADI obtains the amplitude and phase of the echoes by detecting the in-phase (I) and quadrature (Q) components of the received signals. The signal detection network is the part of Fig 2.6 after the signal is down-converted to the IF (Intermediate frequency). The IF signal can be represented as:

$$R(t) = A(t)\, cos\big(2\pi f_0 t + \phi(t)\big) \tag{2.13}$$

where: *A(t)* is the envelop magnitude of the IF signal; *f*$_0$ is the IF frequency; $\phi(t)$ is the phase shift, in CADI specifically, this is the Doppler shift.

Accordingly, I and Q reference components of the IF signal are determined as:

$$I_0(t) = cos\, 2\pi\, f_0 t \tag{2.14}$$

$$Q_0(t) = sin(-2\pi f_0 t) \tag{2.15}$$

The I and Q signals can be obtained after splitting and mixing with the reference:

$$\mathrm{R(t)} \times \mathrm{I_0(t)} = \frac{1}{2}\mathrm{A(t)}\left[\cos\left(4\pi \mathrm{f_0 t} + \phi(t)\right) + \cos\left(\phi(t)\right)\right] \tag{2.16}$$

$$R(t) \times Q_0(t) = \frac{1}{2}A(t)\left[-\sin\left(4\pi f_0 t + \phi(t)\right) + \sin\left(\phi(t)\right)\right] \tag{2.17}$$

After low-pass filtering the higher harmonics, the I and Q signals are sampled by utilizing A/D (Analog-Digital-Converter). The digitized I and Q signals are:

$$\mathrm{I_n} = \frac{1}{2}\mathrm{A(t)}\cos(\phi(t)) \tag{2.18}$$

$$\mathrm{Q_n} = \frac{1}{2}\mathrm{A(t)}\sin(\phi(t)) \tag{2.19}$$

Deriving from the $\mathrm{I_n}$ and $\mathrm{Q_n}$, the amplitude and phase of the received signal yield by the expression:

$$C_n = \sqrt{I_n^2 + Q_n^2} \tag{2.20}$$

$$\theta = tan^{-1}\left(\frac{I_n}{Q_n}\right) \tag{2.21}$$

This detection process ideally doesn't have any DC offset, gain imbalance and non-orthogonality between the two channels. Consequently, some calibration needs to be included in real implementation.

Chapter 3
GNU Radio and USRP

GNU (which is Unix-like) Radio is a free software development framework licensed under the GNU GPL (General Public License) for rapid composition of software defined radios using provided signal processing run-time and signal processing primitives integrated with one of its available hardware support, USRP board. This project aims to achieve the goal of Software Defined Radio (SDR), making the software part of radio system as closely integrated with the RF front end as possible, and to be independent in terms of vendor scheme. The inexpensiveness and versatility of GNU Radio allow it is widely used for hobbyist, academic research and industry on wireless communications support and radio systems development, such as High-Definition Television (HDTV) receiver, IEEE802.11 WiFi Stack, RFID, Active/Passive Radar, OFDM, FM/AM/SSB Radio receiver, etc.

3.1 Software Defined Radio

The concept of Software-Defined Radio (SDR) was coined by Joe Mitola in 1991, "a radio whose channel modulation waveforms are defined in software". In other word, it refers to a series of reprogrammable and reconfigurable radio systems, so that various functionalities satisfying the requirements of different radio systems can be implemented by the same piece of hardware at different time. The exact definition of SDR is controversial and there is no definitive level of the reconfigurability that needs to be reached to satisfy the requirements of a SDR. A better working definition of SDR is "a radio that is substantially defined in software and whose physical layer behavior can be significantly altered through changes to its software" [Reed, 2002].

An ideal SDR system consists of an antenna, an Analog-to Digital Converter (ADC) or a Digital-to-Analog Converter (DAC), and a computer at transmitting path and receiving path respectively, which can be depicted as Fig 3.1

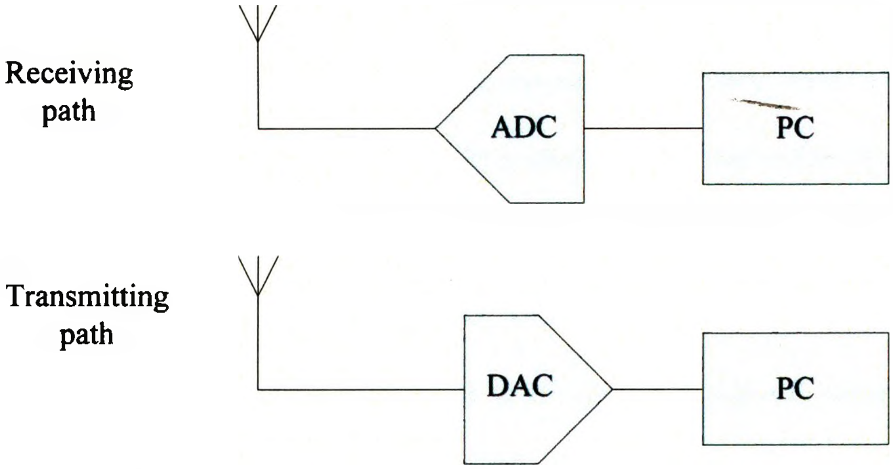


Fig. 3.1 Block diagram of an ideal SDR system

The ideal SDR system, however, requires all the components to be ideal, which means antennas need to handle all range of carrier frequencies, the sample frequency of ADC or DAC can be greater than twice of a frequency in the frequency range of interest, and the power of computer can suffice for various communication protocol and signal processing demand.

Due to the consideration of technique limitation and cost, therefore an ideal system cannot be completely implemented in practice. The SDR system model illustrated in Fig 3.2 is more practical.

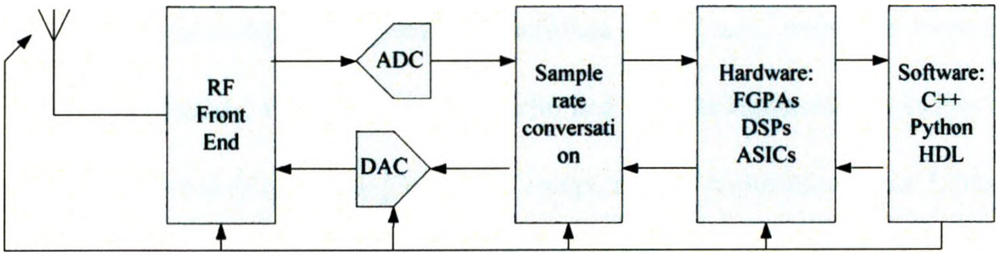


Fig. 3.2 Practical SDR system model

In order to augment the performance of SDR systems, RF front ends, known as superheterodyne mixing stages, and Field Programmable Gate Array (FPGA) have been introduced. The receiver detects signals by a smart antenna that utilizes its gain versus direction characteristic to minimize the interference and noise. The antenna offers the same functionality for the transmitter. The newly introduced RF front-end plays the role of converting the RF to IF and IF to RF at receiver side and transmitter side respectively. IF signals is digitized by the ADC at a lower data rate than would be necessary for the RF frequency, which can decrease the computational complexity of the PC. Lower data rate DAC provides a similar benefit for PC when transmitting. The computational complexity demands still need hardware devices in the stages before data processing in the PC. Such hardware, as FPGAs, Digital Signal Processors (DSPs), or application specific integrated circuits (ASICs), perform signal processing that includes Digital Down/Up Conversion (DDC/DUC) and decimation/interpolation filter etc. Required sample rate needs to accommodate the data rate of ADC/DAC.

The flexibility and reconfigurability of software radio means that the radio system can be updated rapidly nearly without degradation of desired Quality of Service (QoS).

3.2 GNU Radio

To achieve the goal of SDR, integrating the software part of radio system as close to the RF end as possible, GNU Radio, a powerful and flexible framework, provides a series of signal processing building blocks to compose SDR application under Linux, Unix, Windows, or Mac OS X (Operating System X). Besides the basic signal processing blocks, GNU Radio offers hardware interface, Graphical User Interface (GUI, such as GNU Radio Companion (GRC)), utility libraries. The GNU Radio toolkit is developed to use a graphical design method. The radio system development methodology is to first draw a flow graph which is structured by signal processing blocks and the associated ports of these blocks. The data transportation between blocks is facilitated by connecting the input port of a block to the output port of the previous block in the custom radio applications. Conceptually, once the blocks of the whole system are hooked together properly, the datastream flows from signal source, throughout the signal processing blocks, to signal sink.

As a hybrid Python/C++ programming toolkit, the block-based GNU Radio is built using C++ for the performance-critical signal processing blocks, and utilizing Python for such non performance-critical functions as flow graph construction, policy, and run-time management. This allows the radio system applications to obtain the benefits not only from the highly optimized signal processing primitives in C++, but also the free and open source programming language, Python. Since most of developers opt to build custom radio systems in Python, and Python only handles the links between signal processing blocks and run-time, GNU Radio utilizes a wrapper interface, SWIG (the Simple Wrapper Interface Generator), to allow all the blocks to be directly called in system construction in Python. The main branches of pre-defined modules in the GNU Radio signal processing primitive library are:

① gr: the main GNU Radio library. Almost all the commonly used blocks are encapsulated, including signal sources and sinks, simple type conversions, modulations, and filters, etc. In addition, some high-level modules with particular high-level functionalities are derived from these gr- modules.

② usrp: USRP board associated signal sinks and sources, controls

③ audio: Soundcard related signal sinks and sources, controls. Audio signals are capable to be sent and received to and from a soundcard.

④ blks2: additional blocks in Python including such common-used tasks as modulators and demodulators, extra filters and resamplers, etc.

⑤ optfir: routines for FIR filter design.

⑥ plot_data: data plotting with Matplotlib.

⑦ wxgui: a submodule, providing utilities to create Graphic User Interface (GUI) for the flow graph of the radio system.

⑧ eng_notation: functions to handle numbers' representation in engineering notation.

⑨ eng_option: an extension of Python built-in module, optparse, particularly for eng_notation.

⑩ gru: miscellaneous utilities, mathematical operation, etc.

After gluing the signal processing blocks together, if the custom application is a real-time application, the datastream and hardware configuration bitstream are loaded into available hardware support, USRP board, through a USB connection. The architecture of the whole GNU Radio framework is depicted in Fig 3.3.

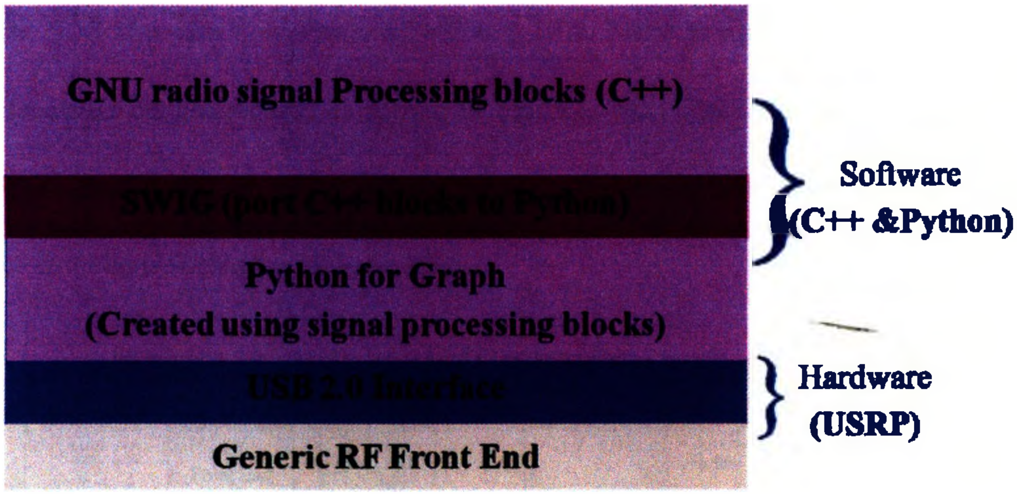

GNU radio signal Processing blocks (C++)
SWIG (port C++ blocks to Python)
Python for Graph
(Created using signal processing blocks)
USB 2.0 Interface
Generic RF Front End
Software
(C++ &Python)
Hardware
(USRP)

Fig. 3.3 Architecture of GNU Radio

3.3 The Universal Software Radio Peripheral (USRP)

The USRP hardware was initially developed by Mr. Matt Ettus, who teamed with Eric Blossom whose brainchild is GNU Radio, to provide the software support of the USRP. The design of USRP tends to allow engineers to rapidly design and implement powerful, flexible software radio systems. Specifically, the USRP enables general purpose computers to function as high bandwidth software radio [Hamza, 2008]. In essence, the USRP serves as a digital baseband and IF section of a radio communication system, integrated with associated RF front-end daughterboard to compose a complete radio system. The fundamental philosophy of this design is doing such waveform-specific signal processing, as modulation and demodulation in the computer, while all of the high-speed general purpose operations are implemented using a FPGA. Typical high-speed general purpose operations are Digital Up Conversion (DUC) and Digital Down Conversion (DDC), decimation and interpolation [Hamza, 2008].

The SDR structure is implemented utilizing GNU Radio along with the USRP and the associated daughterboard, as shown in Fig 3.4. The USRP offers DAC/ADC and FPGA functionalities, and a series of daughterboards perform different functions as RF front-ends within the SDR systems.

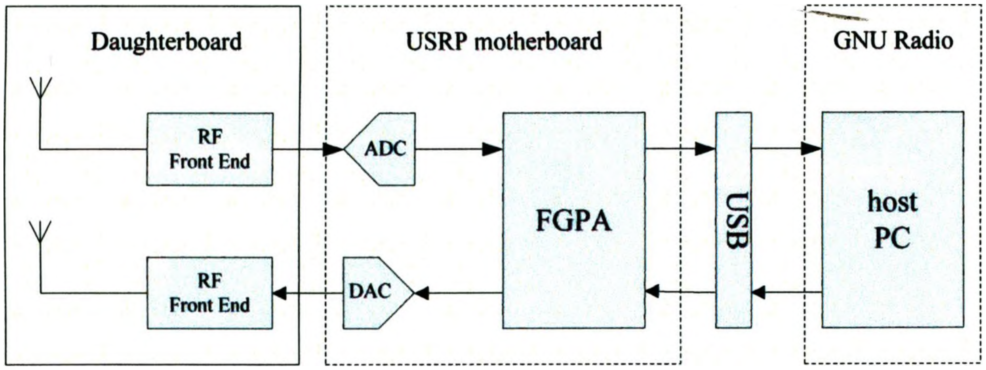


Fig. 3.4 SDR implementation based on GNU Radio

Fig 3.5 shows the pictures of the motherboard of the USRP and Fig 3.6 depicts the block diagram of the USRP motherboard and associated daughterboards. The USRP contains four 12-bit high-speed Analog-to-digital converters (ADCs) sampling at 64M Samples/sec, and four 14-bit high-speed Digital-to-Analog converters (DACs) sampling at 128M Samples/sec. These four outputs and four inputs of DACs and ADCs are connected to the master chip of the USRP, the FPGA. The FPGA was

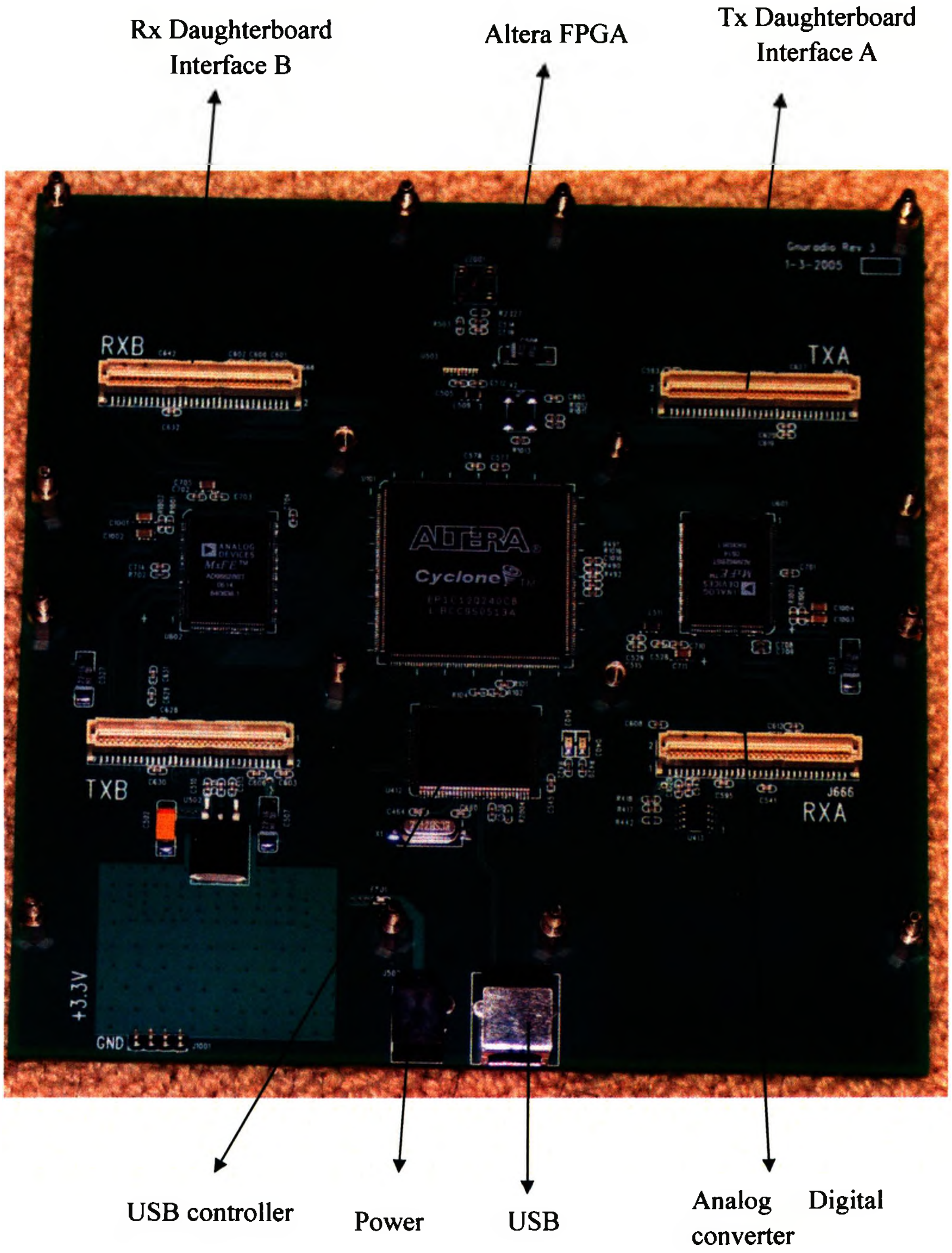


Fig. 3.5 Motherboard of the USRP

selected from the lowest cost series of Altera FPGA, which is Cyclone EP1C12. The configuration bitstream of the FPGA is loaded by GNU Radio through a USB bus, controlled by a USB interface chip, Cypress FX2.

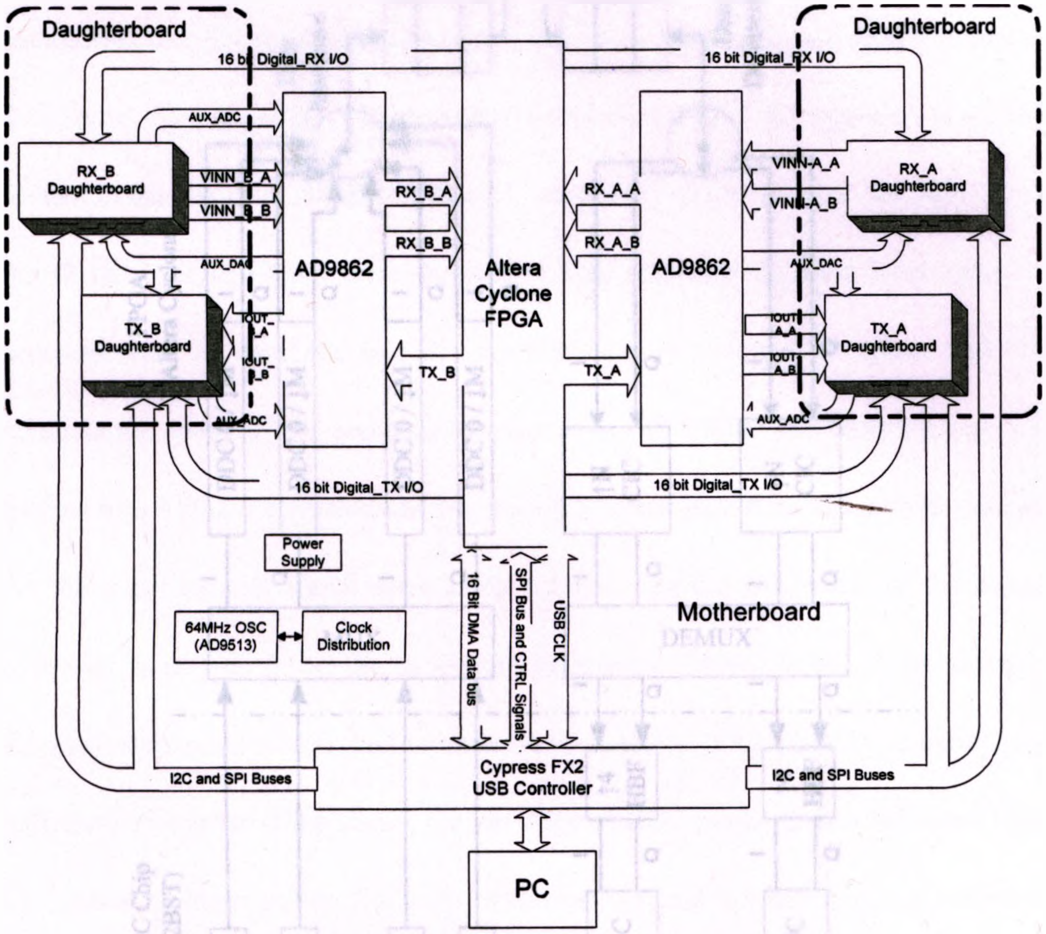


Fig. 3.6 Block diagram of the USRP

3.3.1 The USRP motherboard

The FPGA chip on the USRP plays a significant role in the USRP system since both ADCs and DACs are connected to FPGA and all of the DDC and DUC, decimation and interpolation are performed in the FPGA. The main functionalities implemented on USRP motherboard are shown in Fig 3.7.

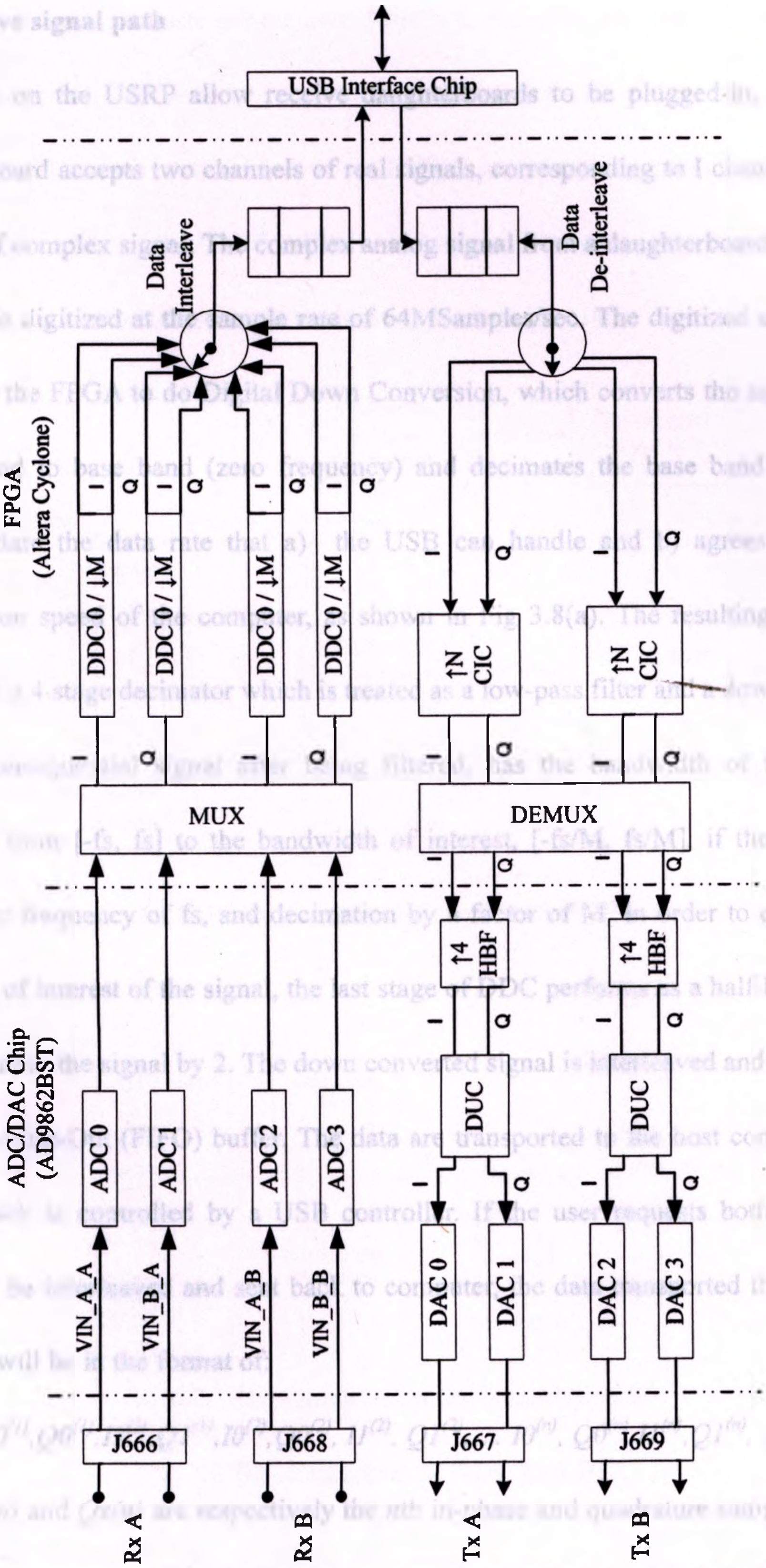


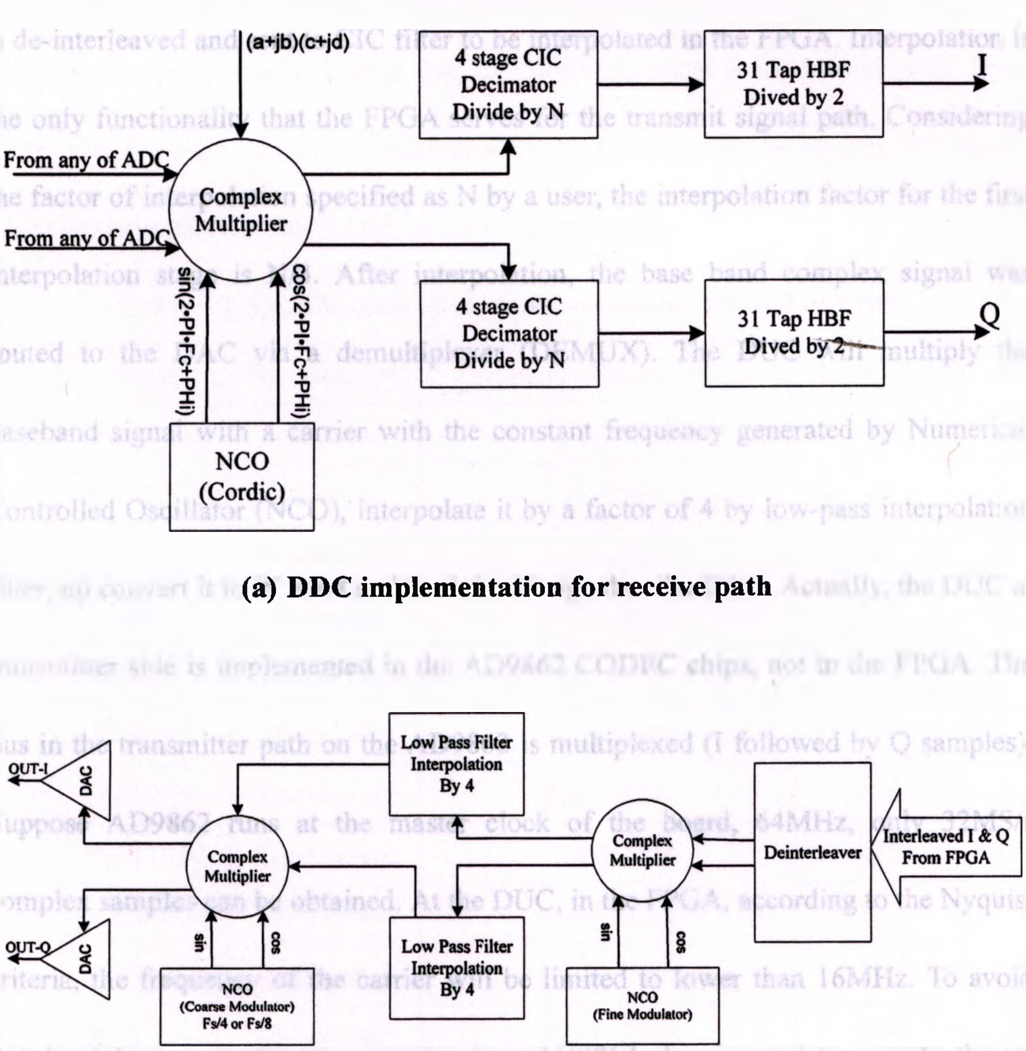
Fig. 3.7 USRP receive and transmit signal paths

The receive signal path

Two slots on the USRP allow receive daughterboards to be plugged-in, and each daughterboard accepts two channels of real signals, corresponding to I channel and Q channel of complex signal. The complex analog signal from a daughterboard is sent to DAC to be digitized at the sample rate of 64MSamples/sec. The digitized signal will be sent to the FPGA to do Digital Down Conversion, which converts the signal from the IF band to base band (zero frequency) and decimates the base band signal to accommodate the data rate that a) the USB can handle and b) agrees with the computation speed of the computer, as shown in Fig 3.8(a). The resulting signal is filtered by a 4 stage decimator which is treated as a low-pass filter and a down sampler. As the consequential signal after being filtered, has the bandwidth of the signal narrowed from [-fs, fs] to the bandwidth of interest, [-fs/M, fs/M], if the signal is sampled at frequency of fs, and decimation by a factor of M. In order to extract the half-band of interest of the signal, the last stage of DDC performs as a half-band filter by decimating the signal by 2. The down converted signal is interleaved and pushed to a First-In-First-Out (FIFO) buffer. The data are transported to the host computer via USB, which is controlled by a USB controller. If the user requests both channels signals to be interleaved and sent back to computer, the data transported through the USB bus will be in the format of:

$$I0^{(1)},Q0^{(1)},I1^{(1)},Q1^{(1)},I0^{(2)},Q0^{(2)},\ I1^{(2)},\ Q1^{(2)},\ \ldots\ I0^{(n)},\ Q0^{(n)},I1^{(n)},Q1^{(n)},\ \ldots$$

where $Ix(n)$ and $Qx(n)$ are respectively the nth in-phase and quadrature samples of the output of decimator x. The USB interface samples the data by 16-bit in IQ format, which means each complex sample uses 4 bytes in the memory. The sustainable data rate of the USB interface can maintain 32MB/sec, which limits the maximum effective total spectral bandwidth to about 8MHz. In order to accommodate both the operation frequency of the USRP, 64MHz, and this maximum spectral bandwidth, the data needs to be decimated at a certain rate in the range [8, 256].



(a) DDC implementation for receive path

(b) DUC implementation for transmit path

Fig. 3.8 Block diagram of DUC and DDC

The transmit signal path

Similarly to the receiver, at the transmitter path, the signal experiences a similar processing as at receiving path, but in reverse. The interleaved data for I and Q channel are pushed into FIFO on the USRP in the same format as the format of the down converted signals sent to computer via USB:

$$I0^{(1)},Q0^{(1)},I1^{(1)},Q1^{(1)},I0^{(2)},Q0^{(2)}, I1^{(2)}, Q1^{(2)}, \ldots\ I0^{(n)}, Q0^{(n)},I1^{(n)},Q1^{(n)}, \ldots$$

where *Ix(n)* and *Qx(n)* are respectively the *n*th in-phase and quadrature data. The data is de-interleaved and sent to CIC filter to be interpolated in the FPGA. Interpolation is the only functionality that the FPGA serves for the transmit signal path. Considering the factor of interpolation specified as N by a user, the interpolation factor for the first interpolation stage is N/4. After interpolation, the base band complex signal was routed to the DAC via a demultiplexer (DEMUX). The DUC ~~will~~ multiply the baseband signal with a carrier with the constant frequency generated by Numerical Controlled Oscillator (NCO), interpolate it by a factor of 4 by low-pass interpolation filter, up convert it to IF band and feed the IF signal to the DAC. Actually, the DUC at transmitter side is implemented in the AD9862 CODEC chips, not in the FPGA. The bus in the transmitter path on the AD9862 is multiplexed (I followed by Q samples). Suppose AD9862 runs at the master clock of the board, 64MHz, only 32MS/s complex samples can be obtained. At the DUC, in the FPGA, according to the Nyquist criteria, the frequency of the carrier will be limited to lower than 16MHz. To avoid this limitation, a digital up converter in a AD9862 chip is used to operate the up conversion.

3.3.2 Daughterboards

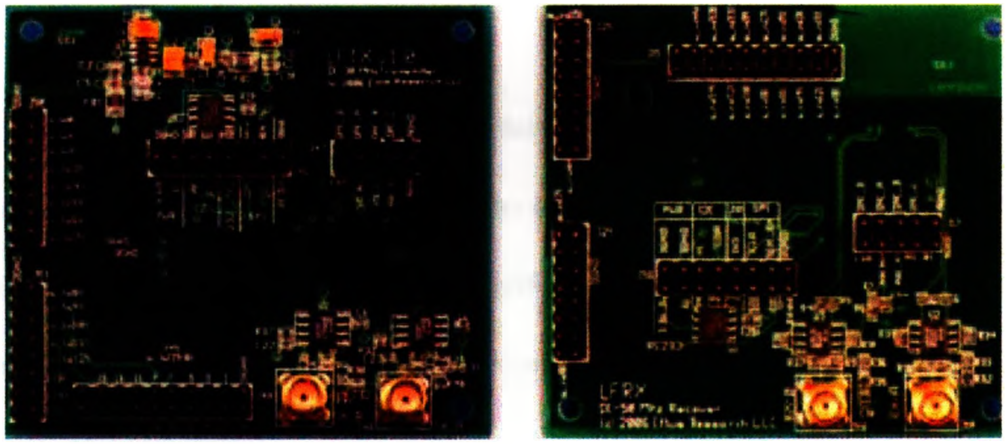
There are four slots in the motherboard available for daughterboards to be plugged into, maximum two transmitter (Tx) boards and two receiver (Rx) boards. Two types of daughterboards are offered from Ettus Research, transceiver daughterboard and transmitter (Tx) and receiver (Rx) daughterboards. Table 3.1 illustrates some specification of all daughterboards that are currently available for development. GNU Radio identifies the board and automatically set up the system by reading the type of the board stored in I2C EEPROM on the daughterboards.

Table 3.1 Specification of all daughterboards offered in Ettus

Name	Functionality	Frequency Range (MHz)	Cost (USD)
Basic Tx	Transmitter	1~ 250	75
Basic Rx	Receiver	1~ 250	75
LFTX	Transmitter	0~ 30	75
LFRX	Receiver	0~ 30	75
TVRX	Receiver	50~ 870	100
DBRX2	Receiver	800~ 2400	150
RFX900	Transceiver	800~ 1000	275
RFX1200	Transceiver	1150~ 1450	275
RFX1800	Transceiver	1500~ 2100	275
RFX2200	Transceiver	2000~ 2400	275
RFX2400	Transceiver	2300~ 2900	275
XCVR2450	Transceiver	2400~ 2500 and 4900~ 5850 (Dual Band)	400
WBX	Transceiver	50~ 2200	450

LFTX and LFRX (Low Frequency TX/RX)

The LFTX and LFRX daughterboards mainly contain two differential amplifiers, an I2C EEPROM, and two 30MHz low pass filters on each board, shown in Fig 3.9. They perform signal amplification utilizing differential amplifiers on daughterboards to extend their frequency response down to DC, so that the frequency of the signals of ADC inputs and DAC outputs can be converted to DC. LFTX and LFRX also contains 30MHz low pass filter for anti-aliasing. Each of the LFTX/LFRX includes logic analyzer connecters for the 16 general purpose IO pins, which helps to debug FPGA design by providing access to internal signals.



(a)LFTX (b)LFRX

Fig. 3.9 LFTX and LFRX daugtherboards

Chapter 4

System Design

In this chapter, the design of Software Radio Based ionosonde prototype using GNU Radio is discussed. The discussion starts with talking about the system requirements that determined the whole design. This is followed by how the design meets these requirements from the aspects of software and hardware.

4.1 Design requirements

4.1.1 Hardware requirements

The key point of this development is not to design a new hardware for ionosonde, but to use "off-the-shelf" universal digital hardware and software for digital signal processing in the system. Power amplifier and pre-amplifiers are required at the transmitter output and the receiver input to match the power requirements of an ionosonde to the standard digital hardware modules. Modifications on the hardware could be made when required, but not to violate the characteristic of this "off-the-shelf" hardware component of GNU Radio, USRP, which means the modifications should be relatively simple and implementable.

4.1.2 Software requirements

As an ionospheric radar, the ionosonde system should have the capabilities of transmitting and receiving modulated pulse signals, such that the time delay between the transmission and reception can be exactly measured. Consequently, the range, the power, and the frequencies of the plasma are obtained by the measurements of time delay and the power of the final correlation results. For the purpose of time delay measurements, the ionosonde system is required to exhibit time-coherence. The time-coherence is demonstrated in a stream of digital data samples if time values are assigned to data samples, so that the difference in time values assigned to two samples is equal to the time difference between the actual times when the digital samples are converted to or from an analog signal at transmitter side or receiver side. If the ionosonde is time-coherent, the discrete data signal accurately represents its analog counterpart, so that the coherence-time can be precisely measured ~~by~~ the received digital data. If the transmitter and receiver data stream can be synchronized and each of them is time-coherent, the time-delay between the pulse transmission and reception can be determined. The pulse utilized in this project is coded by a long sequence, which is 113-bit Legendre code. Being a comparison group, a 13-bit Barker code is also used as a coding approach to modulate the single pulse. Additionally, the transmitter and receiver are co-located at the same place. In order to avoid interference between the ground wave from the transmitter and the echoes from the ionosphere, the transmitter and receiver need to be scheduled to work alternatively in terms of time.

4.2 System design

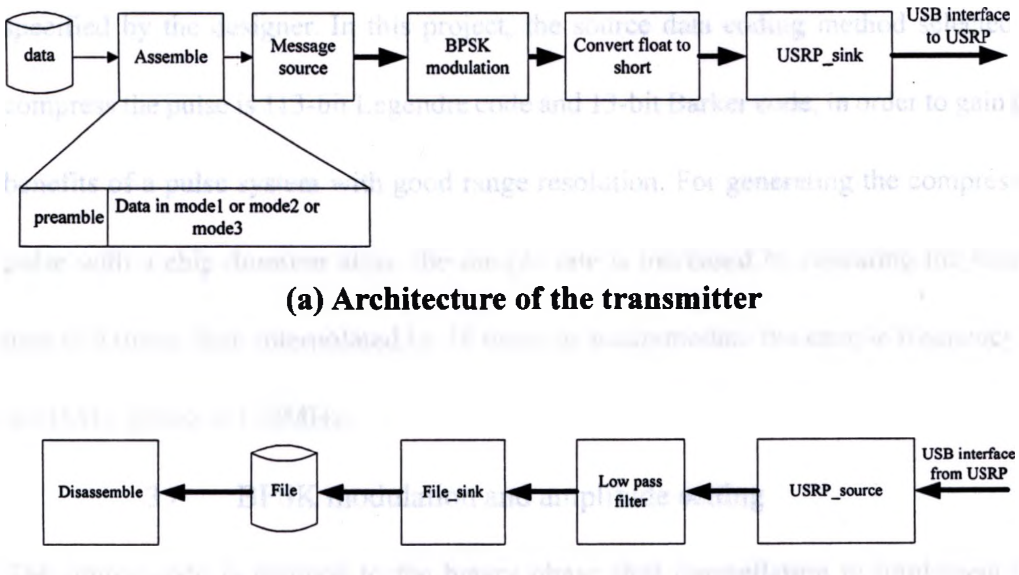
4.2.1 Software architecture overview

Discussed in Chapter 3, the GNU Radio uses a signal processing block based the transmission and reception can be exactly measured. Consequently, the range, the power, and the frequencies of the plasma are obtained by the measurements of time delay and the power of the final correlation results. For the purpose of time delay measurements, the ionosonde system is required to exhibit time-coherence. The time-coherence is demonstrated in a stream of digital data samples if time values are assigned to data samples, so that the difference in time values assigned to two samples is equal to the time difference between the actual times when the digital samples are converted to or from an analog signal at transmitter side or receiver side. If the ionosonde is time-coherent, the discrete data signal accurately represents its analog counterpart, so that the coherence-time can be precisely measured by the received digital data. If the transmitter and receiver data stream can be synchronized and each of them is time-coherent, the time-delay between the pulse transmission and reception can be determined. The pulse utilized in this project is coded by a long sequence, which is 113-bit Legendre code. Being a comparison group, a 13-bit Barker code is also used as a coding approach to modulate the single pulse. Additionally, the transmitter and receiver are co-located at the same place. In order to avoid interference between the ground wave from the transmitter and the echoes from the ionosphere, the transmitter and receiver need to be scheduled to work alternatively in terms of time.

4.2 System design

4.2.1 Software architecture overview

Discussed in Chapter 3, the GNU Radio uses a signal processing block based framework, which implements the critical signal processing primitives in C++ and glues them together in Python. Therefore, almost all of the developments based on the GNU Radio are based on drawing a flow graph to connect associated processing blocks together and scheduling the data transmission and reception to and from this graph. The software architectures constructed as the transmitter and the receiver of the Ionosonde are shown in Fig 4.1. The 113-bit Legendre code is read and modulated to feed into the USRP board at the transmitter side, while the receiver obtains the received data from USRP board, low-pass filters the signals and stores the magnitude of the complex signals in a data file. Nearly all of the signal processing components are implemented in C++ and are called by the main program, written in Python. The functional details in every block will be stated in the following sections.



(a) Architecture of the transmitter

(b) Architecture of the receiver

Fig. 4.1 Software architecture of the Ionosonde

(a)Functional details in blocks of the transmitter's architecture

1) Assemble

The data are assembled as a packet of preamble followed by one of the three transmission modes of data. The mode of data transmission is determined by the frequency of the plasma in different layer of the Ionosphere. To start, a 13-bit Barker code, whose correlation result can provide relatively precise time mark for synchronization, is selected as the preamble of the packet. The configurations of this three-mode data transmission will be described in the following section.

2) Message source

The Message source module allocates a FIFO buffer in the memory ~~in~~ PC and copies the data need to be transmitted to the memory. The data stream copied to the computer memory is fed into the next module in the system. The size of buffer allocated is specified by the designer. In this project, the source data coding method selected to compress the pulse is 113-bit Legendre code and 13-bit Barker code, in order to gain the benefits of a pulse system with good range resolution. For generating the compressed pulse with a chip duration 40μs, the sample rate is increased by repeating the source data 640 times, then interpolated by 16 times to accommodate the sample frequency of the DAC, which is 128MHz.

3) BPSK modulation and amplitude setting

The source code is mapped to the binary phase shift constellation to implement the BPSK modulation. The constellation utilized in this project is [-1, 1, 0], where the first two elements is for BPSK implementation. The last one is used for transmission suspension, which gives zero final output to stop transmitting for certain period of time. The amplitude is set by multiplying all of the source codes by a constant value, 1500 units.

4) Data Conversion: float to short

Since the type of digital data in the USRP can only be short and complex and the signal that the CADI transmitted is a real signal, therefore, the float type of the source code is converted to complex short type data.

5) USRP sink: configuration of USRP

The USRP sink block pushes all the data into the USRP board with all the setting of the parameters for the configuration of FPGA and DAC, including the DUC frequency, gain, interpolation rate, and FPGA configuration bitstream etc. All the details related to the USRP sink and the USRP source will be stated in the following hardware section, since the USRP configuration involves the operation of USRP.

(b) Functional details in blocks of the receiver's architecture

1) Low-pass filter

In the receiver, after low pass filtering the demodulated signal will be correlated with the original signal in order to detect the echo arrival. The filter is a Blackman low-pass filter and is used to eliminate noise. The narrower is the bandwidth of the low-pass filter, the more the received signal is distorted. However, the SNR after correlation is reduced along with increasing of the bandwidth *B* and decreasing of the mainlobe level of the correlation. [Gao, 1991] Thus it turns out that a proper bandwidth is fairly significant for the trade-off between the PSL and SNR to improve the ability of the echo detection. The bandwidth of the band pass filter in the CADI based on Barker 13 sequence is 35 KHz. [Gao, 1991] Therefore, the cut-off frequency for the low pass Blackman is selected half of this bandwidth: 17.5 KHz.

2) File sink

After filtering, the modulated signals are sunk into a data file in complex data form in order to obtain the amplitude and phase of the received signal.

3) Disassemble

When the recorded data is processed in Matlab, the desired portion of the data is selected and combined together to reassemble the whole 113-bit Legendre Code as the data input of the correlation. The starting point of data disassembling is determined by the correlation result of the preamble, which uses a 13-bit Barker code to mark the start time in this project.

4.2.2 Time-coherence

The conceptual operation of this GNU Radio based Ionosonde application is depicted in Fig. 4.2. The Ionosonde transmits a compressed pulse waveform that is read from a data file. After going through the whole transmit signal processing blocks designed by the developer, these waveform data are fed into a software First-In-First-Out (FIFO) buffer controlled by the gr- block in the GNU Radio in the general purpose computer. The gr-block facilitates the data transfer to the FIFO on the FPGA, the main chip of the USRP, through the USB interface. When the transmit signal path is enabled for running, the data in FIFO will start to be pushed into the USRP signal processing blocks for transmission. The same scenario exists on the receiving path, but operating in the opposite direction.

The data rate of signal processing on the host computer, USB controller, and the USRP signal processing are different and without synchronization. However, both data stream in transmission path and receiving path are time-coherent, due to the feature of FIFO buffers. FIFOs provide an interface to get data organized and manipulated in an asynchronous system. When a FIFO is full, an overrun will occurred if the data producing system continues to provide data for storage. In such a scenario, the FIFO simply ignores the incoming data. Conversely, an underrun will happen, when the FIFO is empty and the data consuming system keeps requesting data from the FIFO. In this situation, the data consuming system will receive an empty data set.

If the asynchronous system depicted in Fig. 4.2 operates without any overrun and underrun occurring in FIFO, the data throughout the signal processing blocks will be time-coherent. Accordingly, the time delay of the data reception with respect to the time of signal transmission could be measured by the sample rate of receiver and transmitter. However, it is noteworthy that time-coherence cannot guarantee the synchronization between transmitter and receiver. This issue will be handled by the scheduling scheme developed in next section.

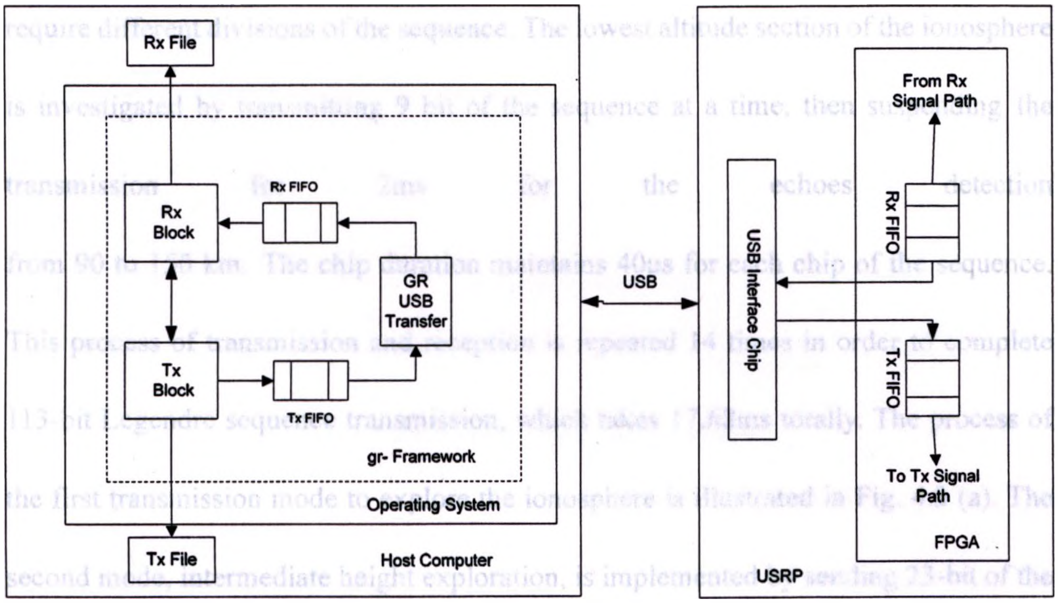
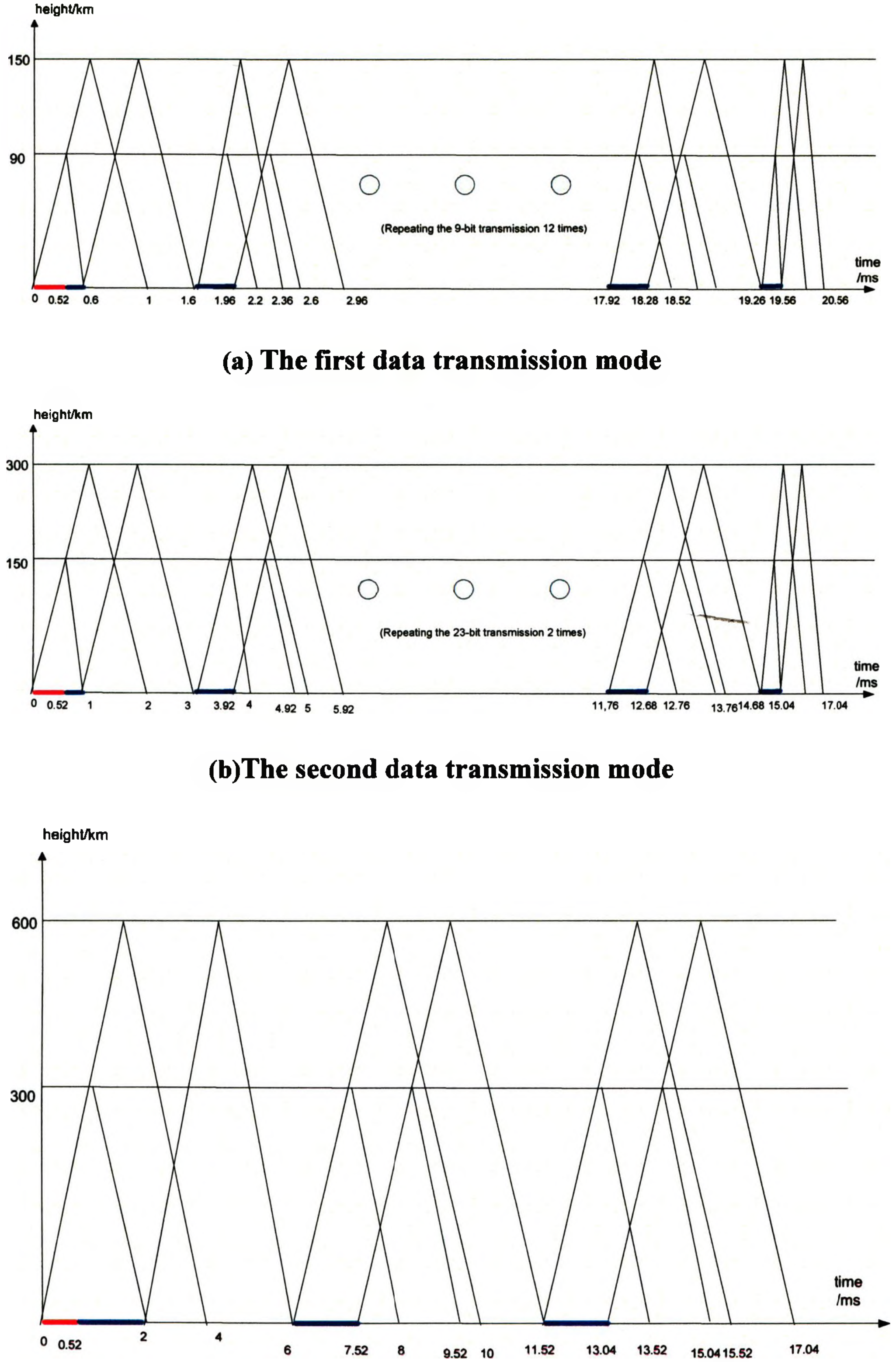


Fig. 4.2 Conceptual operation architecture of the Ionosonde software

Preventing underruns and overruns, the transmitter and receiver signal processing blocks need to provide and consume data requested by the gr- module in time every time when this block is called. In GNU Radio, a warning sign is shown every time when overrun or underrun occurs.

4.2.3 Transmission scheduling

To avert the loss of any echoes from the ionosphere and avoid the blanketing effect of the ground wave directly received from the transmitter, the transmitter sends the 113-bit Legendre code portion by portion and stops signal transmitting between two transmission slots for echo reception. The ionosphere is split into three sections in altitude, 90 km to 150 km, 150 km to 300 km, and 300 km to 600 km. Exploring these three layers of ionosphere is implemented by three modes of transmission, which require different divisions of the sequence. The lowest altitude section of the ionosphere is investigated by transmitting 9 bit of the sequence at a time, then suspending the transmission for 2ms for the echoes detection from 90 to 150 km. The chip duration maintains 40μs for each chip of the sequence. This process of transmission and reception is repeated 14 times in order to complete 113-bit Legendre sequence transmission, which takes 17.68ms totally. The process of the first transmission mode to explore the ionosphere is illustrated in Fig. 4.3 (a). The second mode, intermediate height exploration, is implemented by sending 23-bit of the code at a time, and suspending the transmission 4ms for the echoes reception. This process is repeated 5 times, which takes 17.04ms for this second ~~mode~~ transmission, depicted in Fig. 4.3(b). The highest altitude is detected by transmitting 38-bit of the sequence at a time then followed by a 8ms transmission stop for echo reception. This transmission and reception process repeats 3 times, which takes 17.04ms to finish the transmission of the whole sequence. Fig. 4.3(c) describes this mode of transmission. To synchronize the transmitter and the receiver, a preamble, which currently selected as a 13-bit Barker code, is plugged into the very beginning of each mode of data transmission, assembled as a data packet. The preamble transmission takes 0.52ms to be finished, which forces the first time transmission to be only be the first 2 bits, 12 bits, and 37 bits of the of the a, b, c sequence respectively for each of the three transmission modes, then followed by the transmission scheme. Therefore, the Pulse Repetition Interval (PRI) varies from different mode of data transmission but all pulse repetition frequencies are higher than 20Hz, which satisfies the requirement of the CADI.



(a) The first data transmission mode

(b)The second data transmission mode

(c) The third data transmission mode

Fig. 4.3 Transmission scheduling scheme for the Ionosphere exploration

4.2.4 Scheduling scheme implementation

4.2.4.1 Data packet encapsulation

The basic idea is to use a long Legendre sequence for the coding. Because the long Legendre means that the transmitter would be on while one is trying to receive, the Legendre is split into smaller pieces. The transmission scheduling scheme is implemented by these different formats of data packet encapsulation first. For the different height ranges there are three modes of data transmission corresponds to the data packet in Fig. 4.4 (a), (b), (c) respectively. The stop sign in each data packet is selected as '02' which can be mapped to 0 in the constellation in order to generate the zero amplitude signal as disabling transmission and the number of bit of the stop sign is decided by the transmission suspension time. The scheme of data packets is as follows:

13 bit Barker code	First 12 bit of Legendre code	Stop sign	9 bit of Legendre code	Stop sign	● ● ● 9-bit +stop sign by 9-bit + stop sign	9 bit of Legendre code	Stop sign	The last 3 bit of Legendre code	Stop sign

(a) mode 1 data packet

13 bit Barker code	first 12 bit of Legendre code	Stop sign	23 bit of Legendre code	Stop sign	● ● ● 23-bit +stop sign by 23-bit + stop sign	23 bit of Legendre code	Stop sign	The last 9 bit of Legendre code	Stop sign

(b) mode 2 data packet

13 bit Barker code	first 37 bit of Legendre code	Stop sign	38 bit of Legendre code	Stop sign	38 bit of Legendre code	Stop sign

(c) mode 3 data packet

Fig. 4.4 Data packet encapsulation formats

4.2.4.2 Data packet transmission

When the transmitter starts working, the transmitter checks its state and responds accordingly. When no data has been pushed in the software FIFO in the transmitter, the state is set as WAIT, waiting for data to be pushed in. After setting the transmission start flag, the system transits to the PREAMBLE state. In this state, the system starts to transmit the preamble signal, which currently is a 13-bit Barker sequence using 640 samples per chip in order to maintain the same chip rate as for the Legendre code. Once the preamble signal has been transmitted, the system transits to the TRANSMIT state, where the Legendre sequence transmits in the one of the three modes determined by the transmitting frequency. After finishing the Legendre sequence transmission, the transmitter system will move to the SUSPEND state for a certain amount of time for the echo reception. Having done the whole procedure of a cycle of transmission, the transmitter system is configured back to the PREAMBLE state to maintain the transmission endlessly running until any keyboard interruption is caught. Fig. 4.5 shows the flow chart of this scheduling scheme implementation.

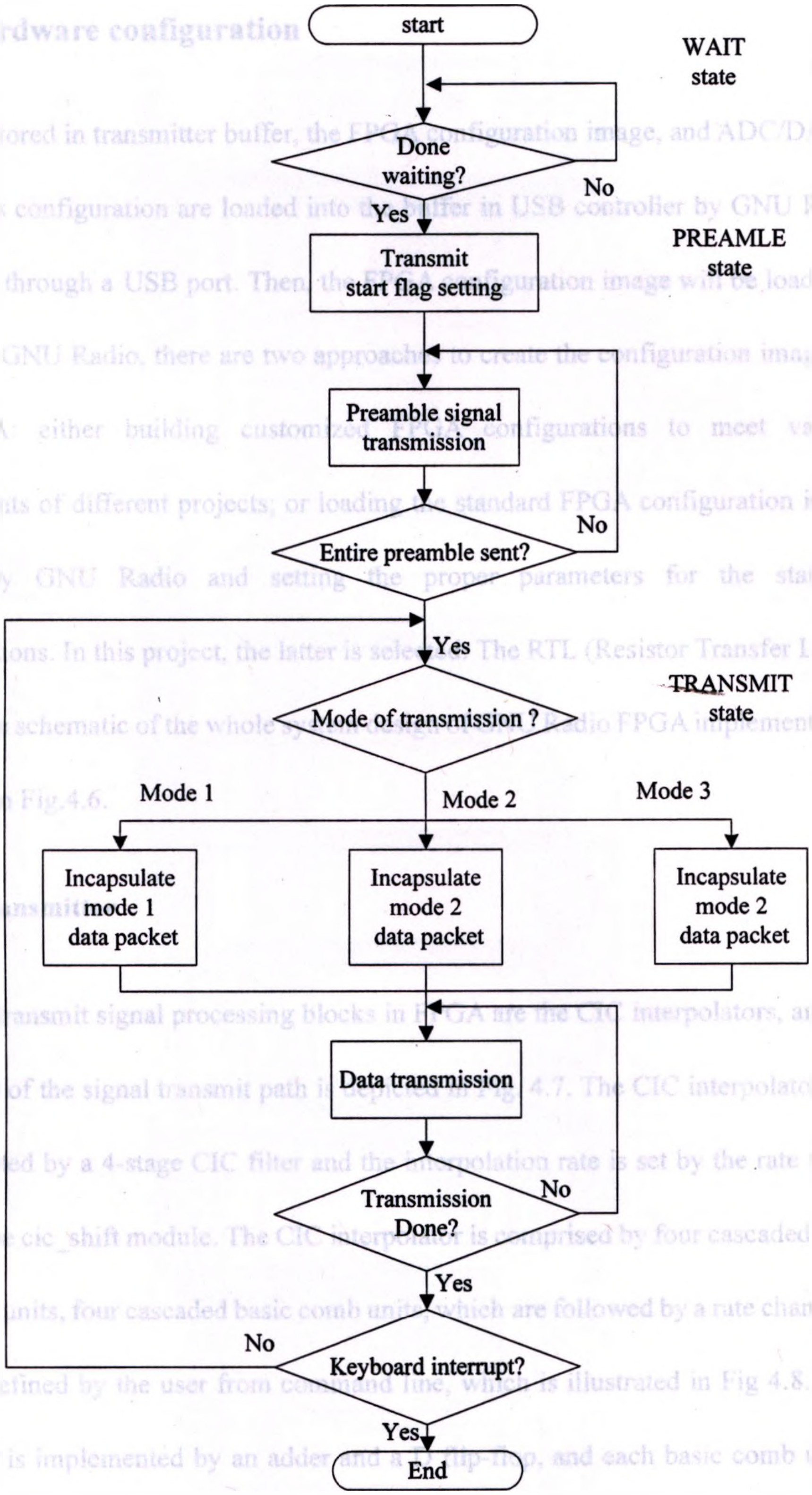


Fig. 4.5 Flow chart of transmission scheduling scheme

4.2.5 Hardware configuration

The data stored in transmitter buffer, the FPGA configuration image, and ADC/DAC parameters configuration are loaded into the buffer in USB controller by GNU Radio primitives through a USB port. Then, the FPGA configuration image will be loaded in FPGA. In GNU Radio, there are two approaches to create the configuration image for the FPGA: either building customized FPGA configurations to meet various requirements of different projects; or loading the standard FPGA configuration image created by GNU Radio and setting the proper parameters for the standard configurations. In this project, the latter is selected. The RTL (Resistor Transfer Level) view of the schematic of the whole system design of GNU Radio FPGA implementation is shown in Fig.4.6.

4.2.5.1 Transmitter

The only transmit signal processing blocks in FPGA are the CIC interpolators, and the RTL view of the signal transmit path is depicted in Fig. 4.7. The CIC interpolators are implemented by a 4-stage CIC filter and the interpolation rate is set by the rate signal fed into the cic_shift module. The CIC interpolator is comprised by four cascaded basic integrator units, four cascaded basic comb units, which are followed by a rate change by a factor defined by the user from command line, which is illustrated in Fig 4.8. Each integrator is implemented by an adder and a D flip-flop, and each basic comb unit is created by an adder and two D flip-flop register, which respectively illustrated in Fig. 4.9(a) and Fig. 4.9(b).

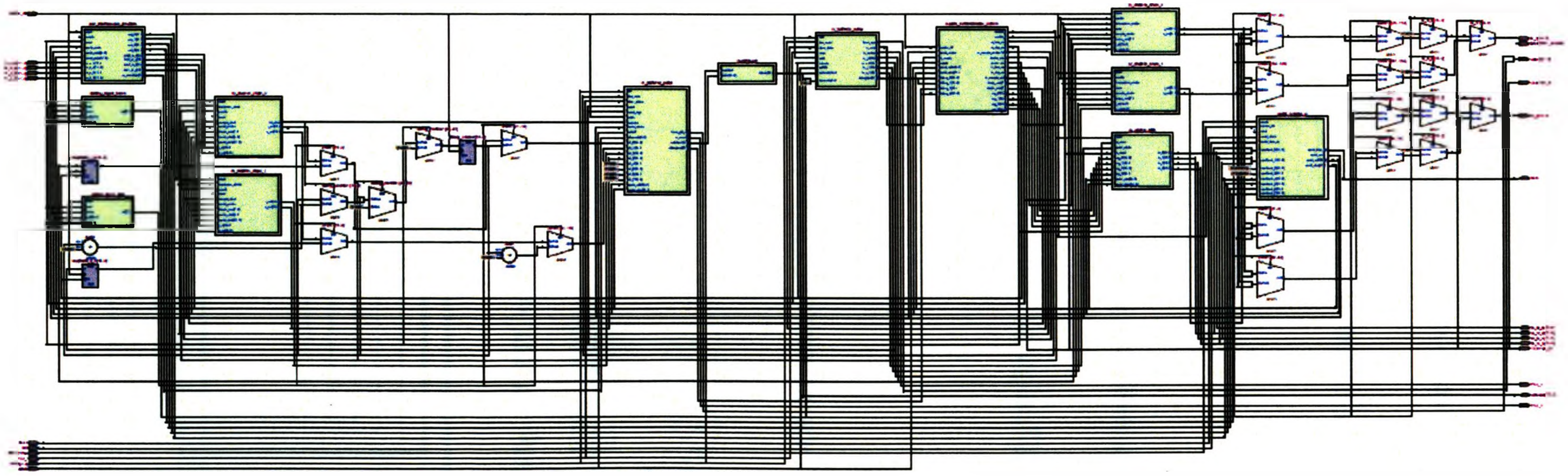


Fig. 4.6 RTL view of standard FPGA configuration in GNU Radio

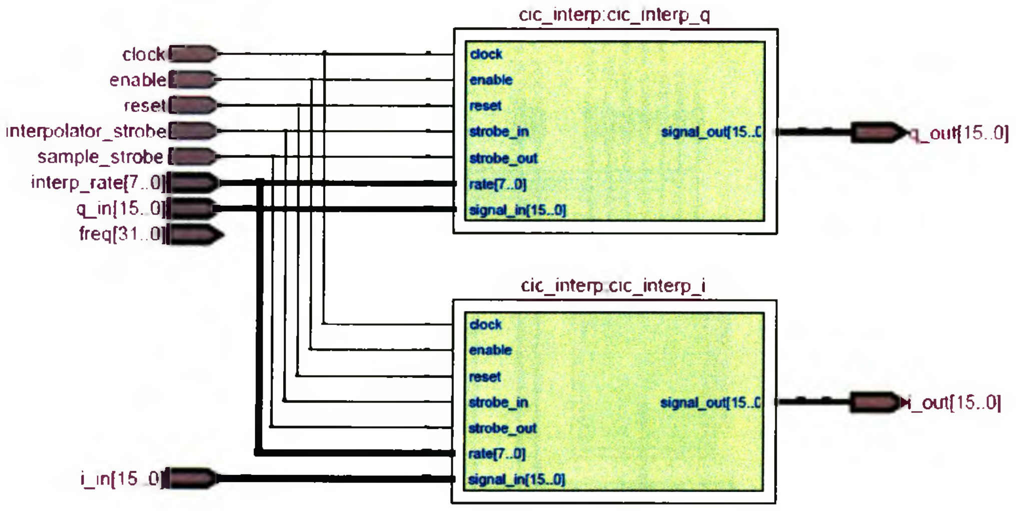


Fig. 4.7 RTL view of signal transmit path

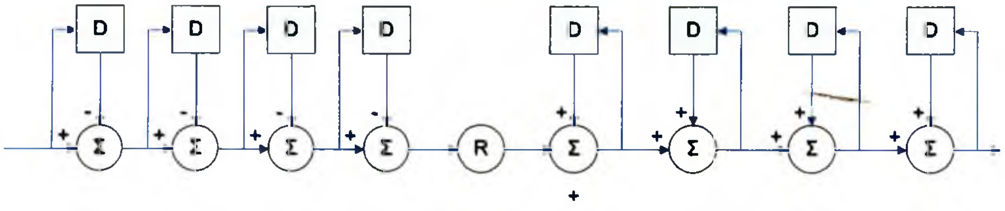
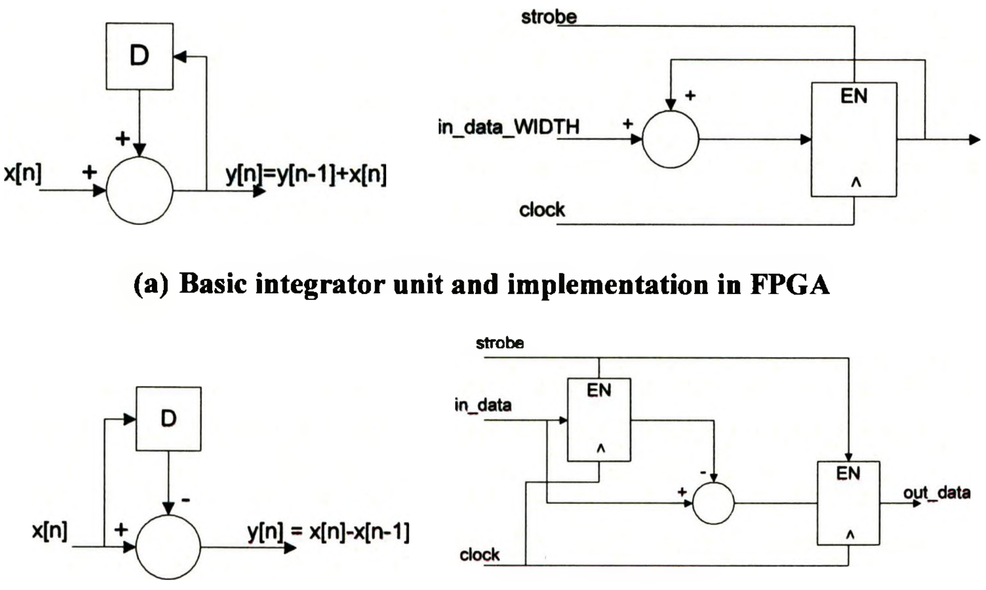


Fig. 4.8 Structure of CIC interpolator in FPGA



(a) Basic integrator unit and implementation in FPGA

(b) Basic comb unit and implementation in FPGA

Fig. 4.9 Basic Integrator Unit and Basic Comb Unit

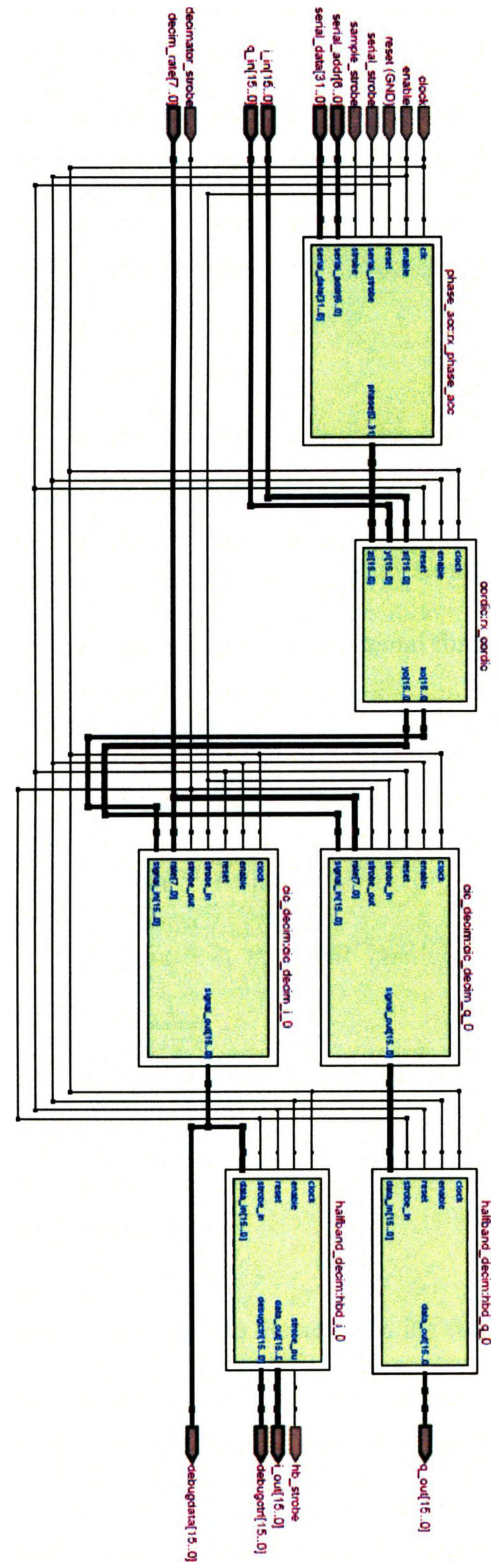


Fig. 4.10 RTL view of signal reception path

4.2.5.2 Receiver

At the receiver side, DDC is implemented in the rx_chain module by CORDIC and PHASE_ACC (phase accumulation) modules, and followed by a two CIC decimators and two halfband decimators. The RTL view of the schematic of the signal reception path is shown in Fig. 4.10.

The rx_chain module utilizes the CORDIC to implement the DDC. The CORDIC Algorithm, an acronym of Coordinate Rotation Digital Computer Algorithm, is an iterative solution for trigonometric and other transcendental functions that only use add and shift to implement. The incremental expression of this algorithm is:

$$x_{i+1} = K_i[x_i - y_i d_i 2^{-i}] \qquad 4.1$$

$$y_{i+1} = K_i\,[y_i + x_i d_i 2^{-i}] \qquad 4.2$$

$$z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \qquad 4.3$$

Where: $K_i = \cos(\tan^{-1} 2^{-i}) = \frac{1}{\sqrt{1+2^{-2i}}}$,

(K$_i$=1 in GNU Radio cordic module)

$$d_i = \begin{cases} +1, & z < 0 \\ -1, & \text{otherwise} \end{cases}$$

D flip-flops perform the shift and adders do the addition for the implementation of one stage of this algorithm in FPGA. The phase iteration is achieved by PHASE_ACC module to feed the frequency for this algorithm. This CORDIC stage is iterated 12 times, to achieve the final result of this algorithm and implement the DDC.

The CIC decimators in the signal reception path perform similarly as the CIC interpolators in the transmitting path, but in reverse order, which means signals pass the comb stages first and are fed into the integrator units afterward. The sample rate changes finally, and it specifically defined as 256 in this project, which reach the highest decimation rate in order to minimize the size of the received data file.

All of the parameters configured for the signal processing blocks in FPGA are written in registers built in FPGA by the register setting (reg_setting) module defined in master control (master_ctrl) module. This master control module generates the strobe for interpolation, decimation, and transmission sampling. The reference clocks sent to the i/o pins on daughterboards are produced in this control module as well. There are another two main modules implemented in the standard FPGA configuration: rx_buffer storing the received data temporally, serial_io setting parameters defined by user from command line or default defined parameters through the serial bus.

The output signals from FPGA are fed into the next the stage which uses a signal processing chip, AD9862, which mainly performs the DUC and Digital-Analog Conversion for the signal transmission path and Analog-Digital Conversion for the signal reception path. The functional diagram of this AD (Analog-Digital) chip is shown in Fig. 4.11. The DUC is implemented by a NCO (Numerically Controlled Oscillator) and a 4 × interpolation filter is performed by using a 39 tap filter followed by a 15 tap filter in the transmission path in this AD chip. The mode of the interpolation filter is configured by writing the Tx DIGITAL register. The three NCO TUNING WORD registers set the 24-bit frequency tuning word for the NCO in the modulator stage of the Tx path. The full scale of the tuning word produces the maximum tuning frequency, $f_{DAC}/4$, which is equivalent to 32MHz, with a resolution of $f_{DAC}/ 2^{26}$. A PGA (Programmable Gain Amplifier), which provides a 20dB maximum continuous gain range with 0.1 dB steps to both Tx channels and the gain is set by a 8-bit TxPGA GAIN register follows the Digital-Analog Conversion stage in the transmitter path. The PGA at receiver side performs pretty much the same as at the transmitter side, gain ranging from 0dB to 20dB with 1dB steps. The low-pass decimation filter shown in the Functional Block Diagram is bypassed in GNU Radio. All of the ~~AD~~ registers are written by "write_9862" module in GNU Radio to define the parameters for the functions performed in the AD.

4.3 Whole system block diagram

According to the software and hardware architecture of the Ionsonde system described above, the architecture of the whole system shown in Fig. 4.12.

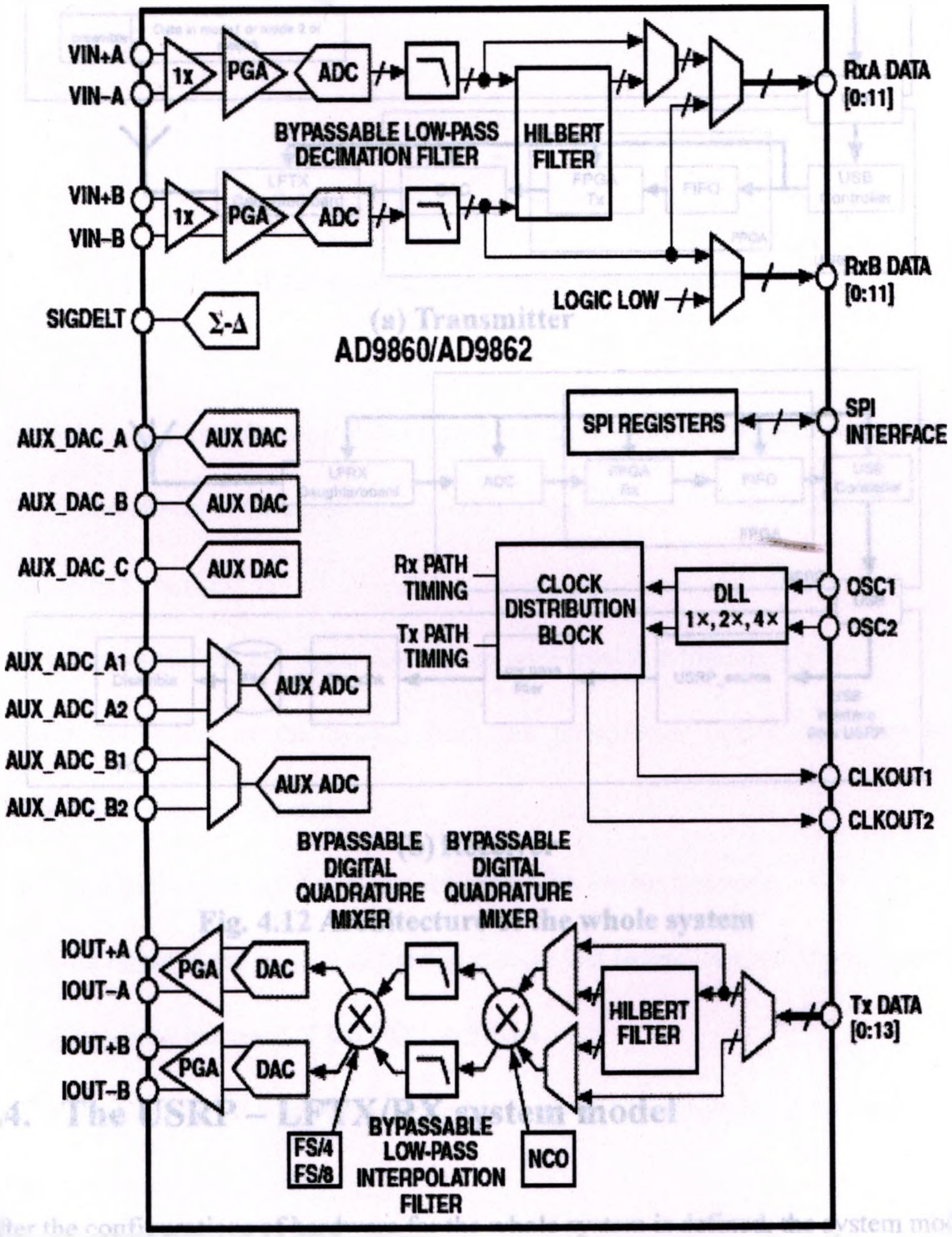
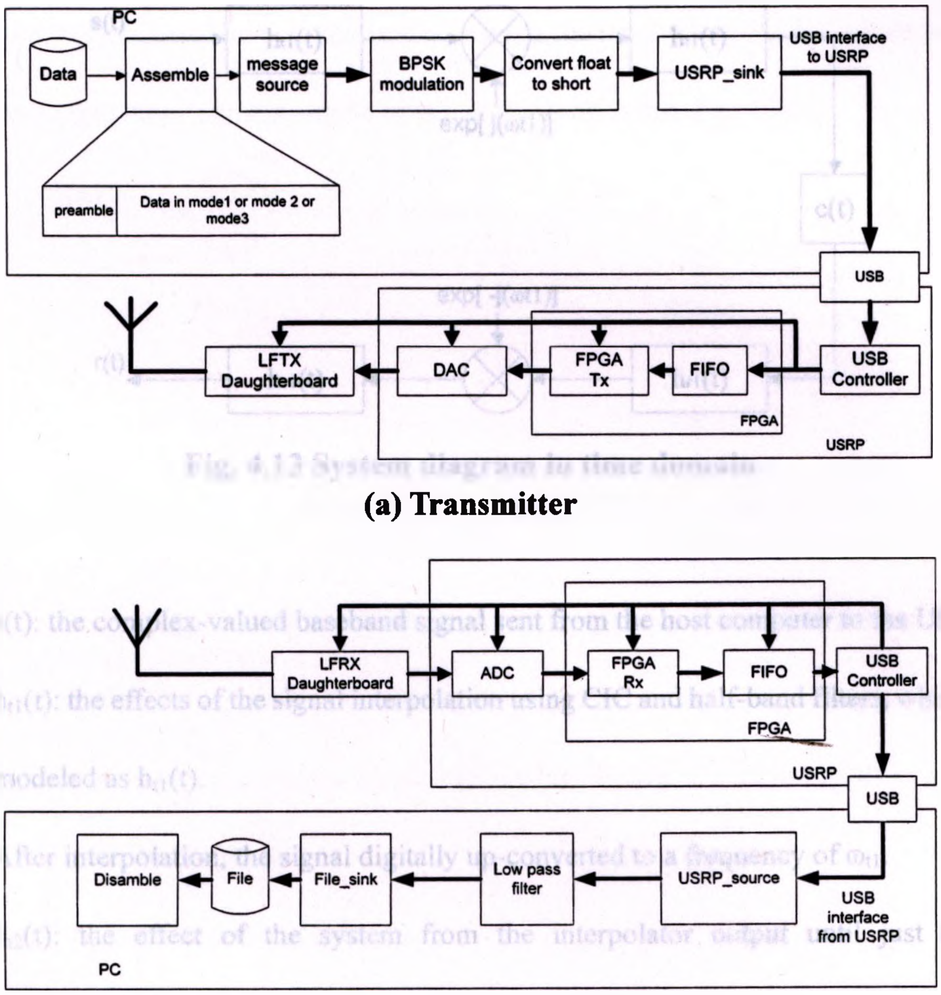


Fig. 4.11 Functional diagram of AD9862



(a) Transmitter

(b) Receiver

Fig. 4.12 Architecture of the whole system

4.4. The USRP – LFTX/RX system model

After the configurations of hardware for the whole system is defined, the system model can be derived by working from Fig. 3.7 and Fig. 3.8. A system level block diagram of the Ionosonde waveform signal path is stated in Fig.4.13, and the text following this figure.

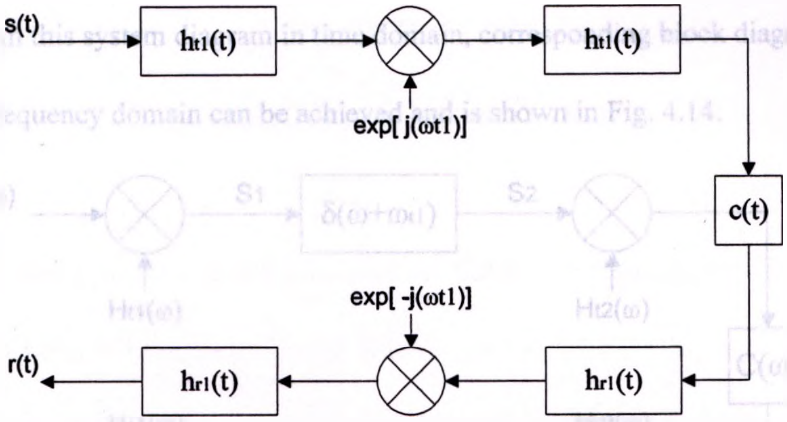


Fig. 4.13 System diagram in time domain

1. s(t): the complex-valued baseband signal sent from the host computer to the USRP.
2. $h_{t1}(t)$: the effects of the signal interpolation using CIC and half-band filters, which is modeled as $h_{t1}(t)$.
3. After interpolation, the signal digitally up-converted to a frequency of ω_{t1}.
4. $h_{t2}(t)$: the effect of the system from the interpolator output until just after transmission by the antenna.
5. c(t): model that the radiated signal interacts with the environment.
6. h_{r2}: the effect of the system on the signal from just before reception by the antenna until reaching the input of the DDC in FPGA.
7. The received signal is down converted to baseband by multiplying the signal with the frequency of ω_{r1}.
8. $h_{r1}(t)$: The resulting baseband signal is then decimated using CIC decimation filter and half-band filter. The effect of the system in this stage is modeled as $h_{r1}(t)$. The resulting signal is sent to computer through USB.

Derived from this system diagram in time domain, corresponding block diagram of the system in frequency domain can be achieved and is shown in Fig. 4.14.

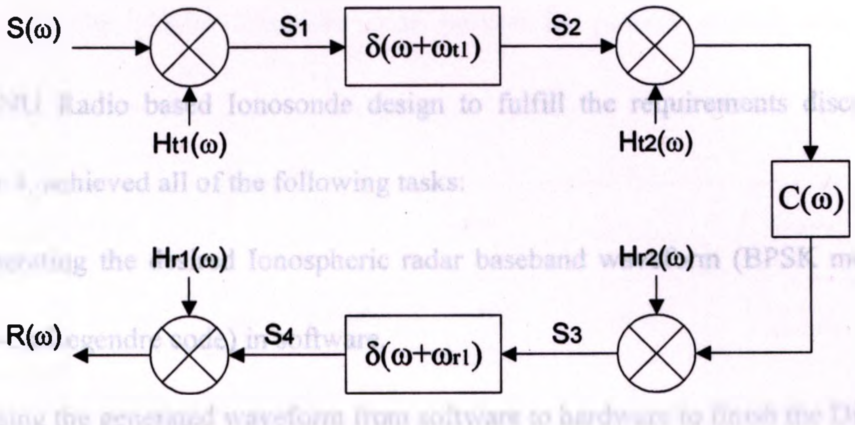


Fig. 4.14 System diagram in frequency domain

In this scenario, the system transfer function can be derived by analyzing the signal at each point in the signal processing path indicated by the signal sign in Fig.4.14:

$$S_1(\omega)= S(\omega)H_{t1}(\omega)$$

$$S_2(\omega)= S(\omega+\omega_{t1})H_{t1}(\omega+\omega_{t1})$$

$$S_3(\omega)= S(\omega+\omega_{t1})H_{t1}(\omega+\omega_{t1})H_{t2}(\omega)C(\omega)H_{r2}(\omega)$$

Let $C_e(\omega) = H_{t2}(\omega)C(\omega)H_{r2}(\omega)$:

$$S_3(\omega)= S(\omega+\omega_{t1})H_{t1}(\omega+\omega_{t1})C_e(\omega)$$

$$S_4= S(\omega+\omega_{t1}-\omega_{r1})H_{t1}(\omega+\omega_{t1}-\omega_{r1})C_e(\omega+\omega_{r1})$$

Since $\omega_{t1}=\omega_{r1}$:

$$S_4= S(\omega)H_{t1}(\omega)C_e(\omega+\omega_{r1})$$

According to the analysis of the system transfer function, the system can be approximated as a linear time-invariant system, which proves that system's impulse response can be used to improve range resolution.

CHAPTER 5

Experimental set up and results

This GNU Radio based Ionosonde design to fulfill the requirements discussed in Chapter 4, achieved all of the following tasks:

1. Generating the desired Ionospheric radar baseband waveform (BPSK modulated 113-bit Legendre code) in software.
2. Passing the generated waveform from software to hardware to finish the Digital Up Conversion and Digital-to-Analog Conversion
3. Transmitting the analog RF signal
4. Receiving the returned signals in hardware
5. Processing the received signal: Digital Down Conversion, and data decimation, noise suppression etc.
6. Passing the received signals from hardware to software through USB port
7. Doing data analysis on the desired portion of recorded data to get the delay of the received signal

All of the capabilities that this Software Defined Radio based Ionosonde should satisfy were tested in the laboratory and at the field site, and the results are stated in the following sections of this chapter. A conceptual block diagram of the whole SDR based Ionosonde architecture is shown in Fig. 5.1, which also depicts the portion of the system corresponding to each of the seven tasks of the whole system as noted by the labeled numbers in the figure.

The transmitter and receiver of the Ionosonde system are tested by two steps: 1) continuously transmitting the whole 113-bit Legendre code without scheduling; 2) transmitting the 113-bit Legendre code portion by portion implemented by the scheduling scheme.

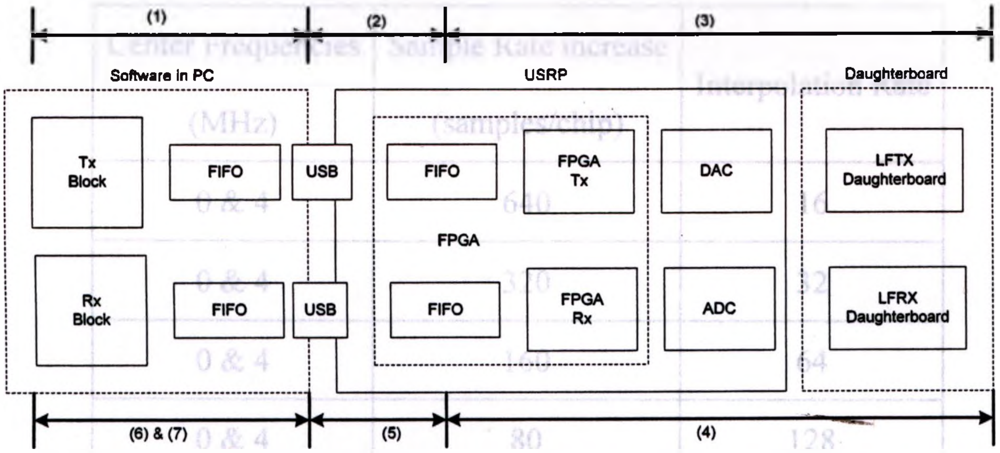


Fig 5.1 Conceptual block diagram of the Ionosonde

5.1 Laboratory test

5.1.1 Continuously transmitting the code without scheduling

5.1.1.1 Transmitter

Task 2 and 3 are tested by injecting the RF output of LFTX daughterboard into an oscilloscope to demonstrate the waveform of the baseband signal and the RF signal, and comparing the waveform with expected results implemented in simulation. The test set up is shown in Fig. 5.2.

The parameters configuration of the transmitted signal for testing is listed in Table 5.1. The corresponding baseband and passband signal waveforms demonstrated on the oscilloscope are illustrated in Fig 5.3 with respect to different sample rates. The expected baseband signal and RF signal are shown in Fig 5.4, for the comparison with the transmitted signal.

Table 5.1 Parameters configuration of the transmitted signal

Center Frequencies (MHz)	Sample Rate increase (samples/chip)	Interpolation Rate
0 & 4	640	16
0 & 4	320	32
0 & 4	160	64
0 & 4	80	128
0 & 4	40	256
0 & 4	20	512
0 & 4	10	512

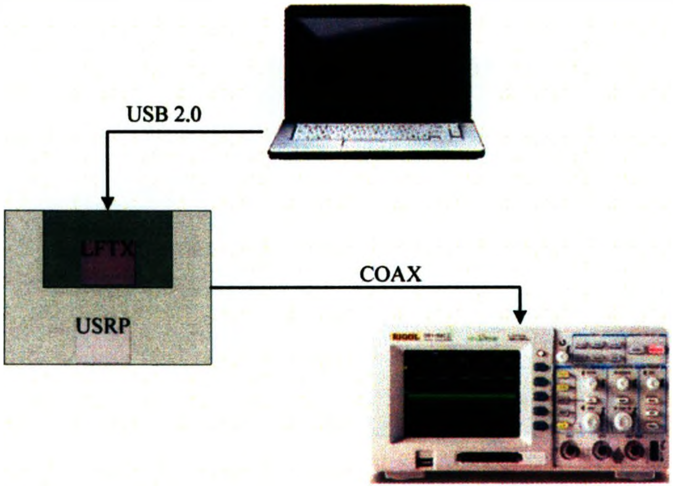
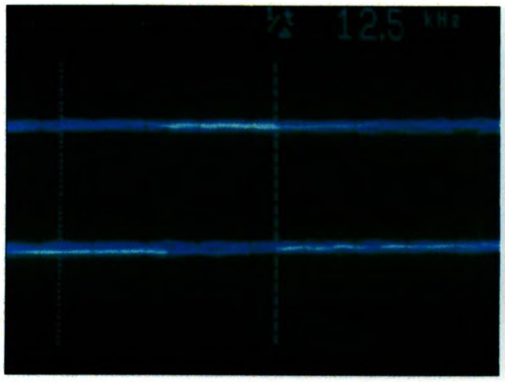
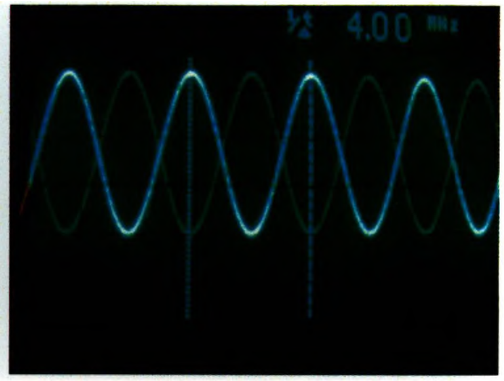


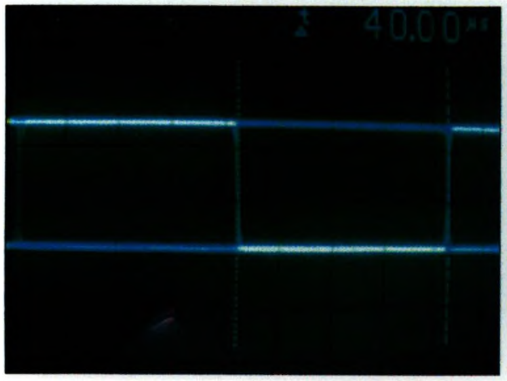
Fig. 5.2 Hardware configuration for transmitted waveform measurements



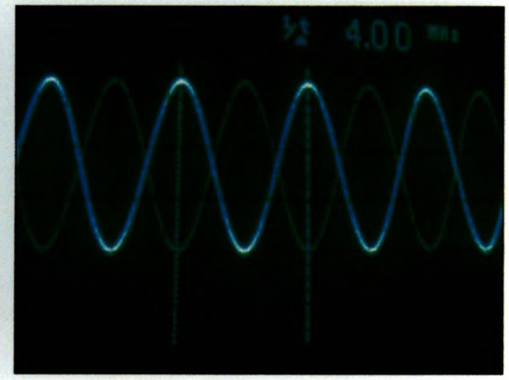
(a.1) 0MHz, 640samples/chip, interp_rate:16



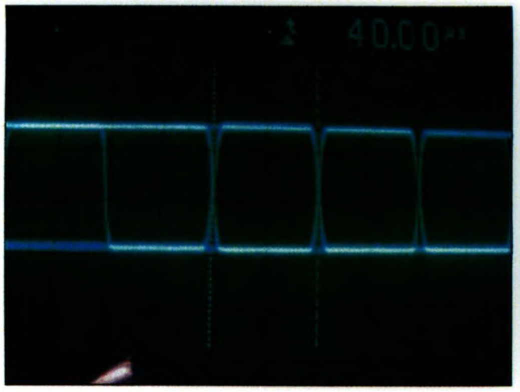
(a.2) 4MHz, 640samples/chip, interp_rate:16



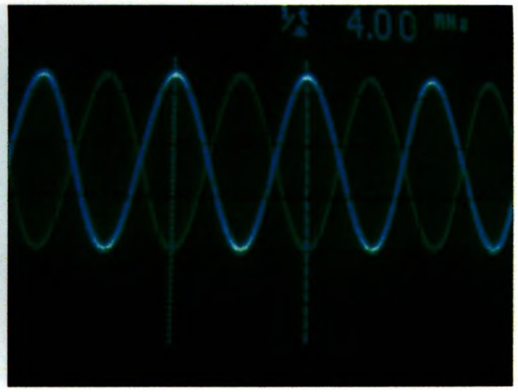
(b.1)0MHz, 320 samples/chip, interp_rate:32



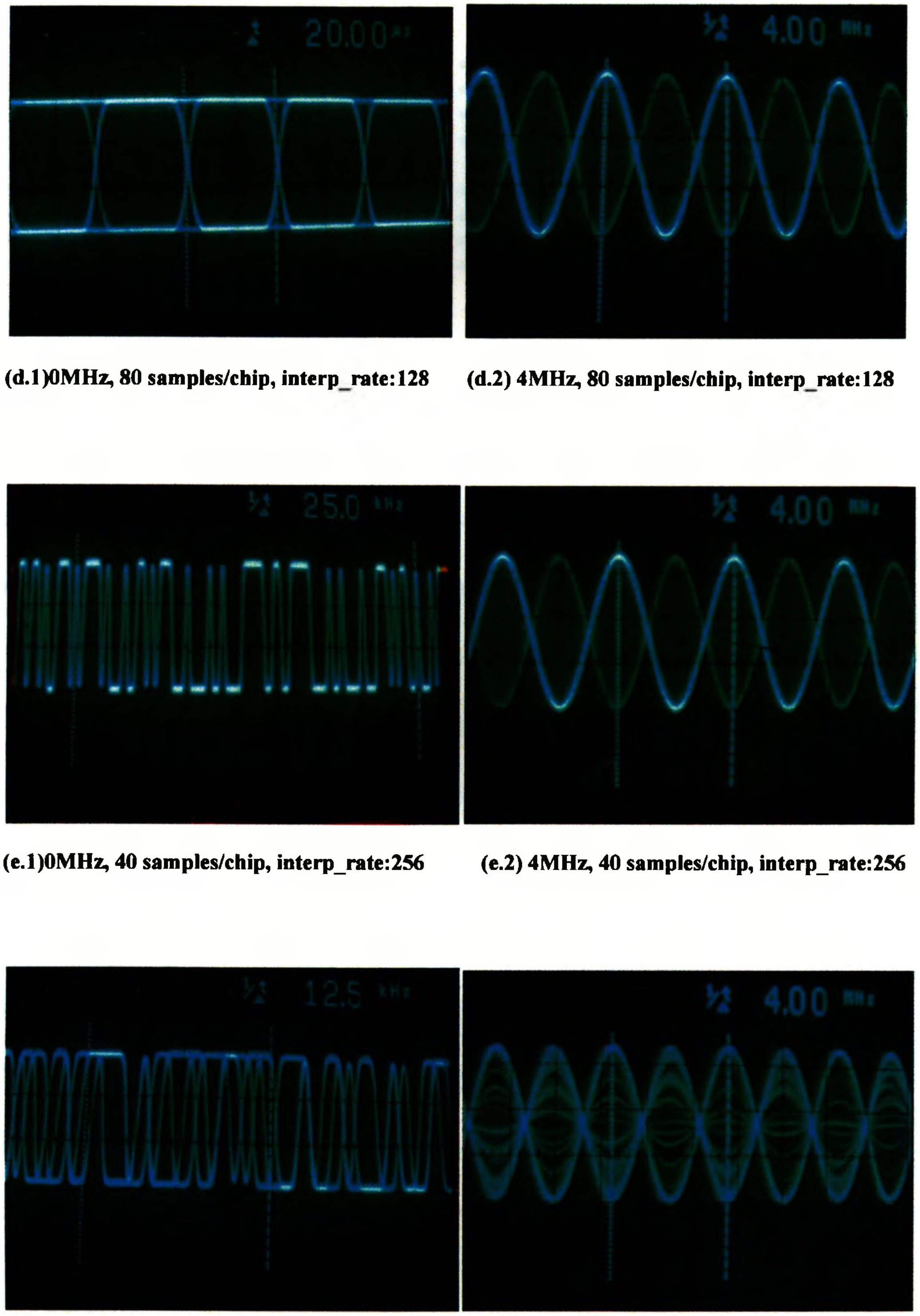
(b.2) 4MHz, 320samples/chip, interp_rate:32



(c.1)0MHz, 160 samples/chip, interp_rate:64



(c.2) 4MHz, 160 samples/chip, interp_rate:64

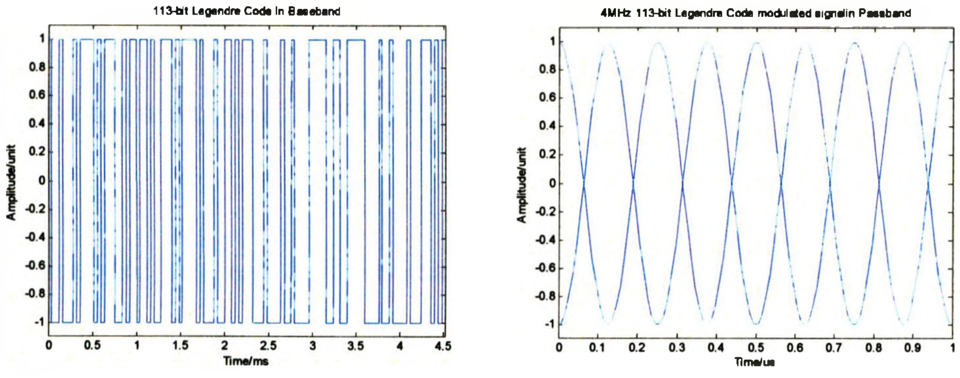


(d.1)0MHz, 80 samples/chip, interp_rate:128 (d.2) 4MHz, 80 samples/chip, interp_rate:128

(e.1)0MHz, 40 samples/chip, interp_rate:256 (e.2) 4MHz, 40 samples/chip, interp_rate:256

(f.1)0MHz, 20 samples/chip, interp_rate:512 (f.2) 4MHz, 20 samples/chip, interp_rate:512

Fig 5.3 Actual transmitted signal waveforms in baseband and passband



(a) Transmitted signal in baseband **(b) Transmitted signal in passband**

Fig 5.4 Expected transmitted signal in baseband and in passband

The signal expected to be transmitted is a square wave modulated by 113-bit Legendre code in baseband and a phase modulated plain sinusoid wave in passband. According to the comparison of the actual transmitted signal waveforms with the expected signal waveforms corresponding to different parameters settings, the transmitted signal gradually distorted when the number of samples per chip was decreased and the interpolation rate was decreased. Therefore, the number of the samples per chip of the Legendre code needs to be adjusted to achieve the pulse duration (one chip rate) of the Legendre code, τ = 40μs, and the bandwidth of its spectral mainlobe, $2/\tau$ = 50kHz. Considering the sample rate of DAC in the transmission path and the range restriction of the interpolation rate, [4, 512], the number of the samples contained in one chip is configured as 640 samples/chip to obtain a nice 40μs square wave signal in baseband.

5.1.1.2 Receiver

Task 4, 5, and 6 implemented in the receiver are tested by directly connecting the RF output of the LFRX daughterboard to the RF output of the LFTX transmitter daughterboard, and analyzing the recorded data. The hardware configuration for receiver test is illustrated in Fig. 5.5.

The recorded data will be correlated with the original Legendre code in Matlab to obtain the signal reception time. The correlation result is shown in Fig. 5.6, where the spikes show the times that when the whole Legendre code was received at the receiver side when transmitter is continuously sending the whole Legendre sequence without any scheduling scheme but just repeatedly. As the receiver is randomly enabled to receive signal, the time of echo reception is all relative to the time of ground wave reception. The time difference between the adjacent spikes is 4.52ms which equals the duration of the whole Legendre code, since the Legendre code is continuously transmitted and directly received with zero-delay. This agrees with the feature of FIFO, time-coherence, discussed in Chapter 4, and verifies that the synchronization between transmitter and receiver could be implemented by inserting a preamble (Barker code) in the data packet and the time delay of echo reception could be directly measured by time difference between the directly received preamble and the echo time. The amplitude of the correlation maxima shows the amplitude of the received signal.

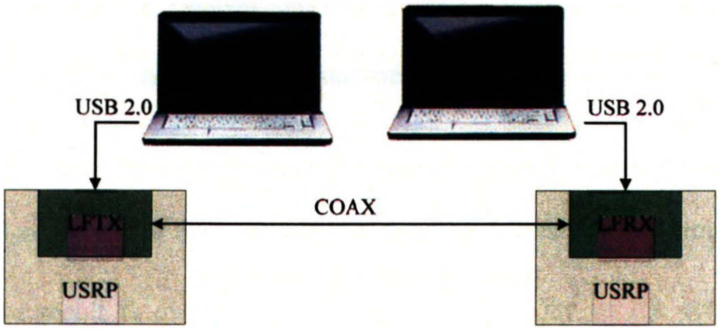


Fig. 5.5 Hardware configuration for receiver testing

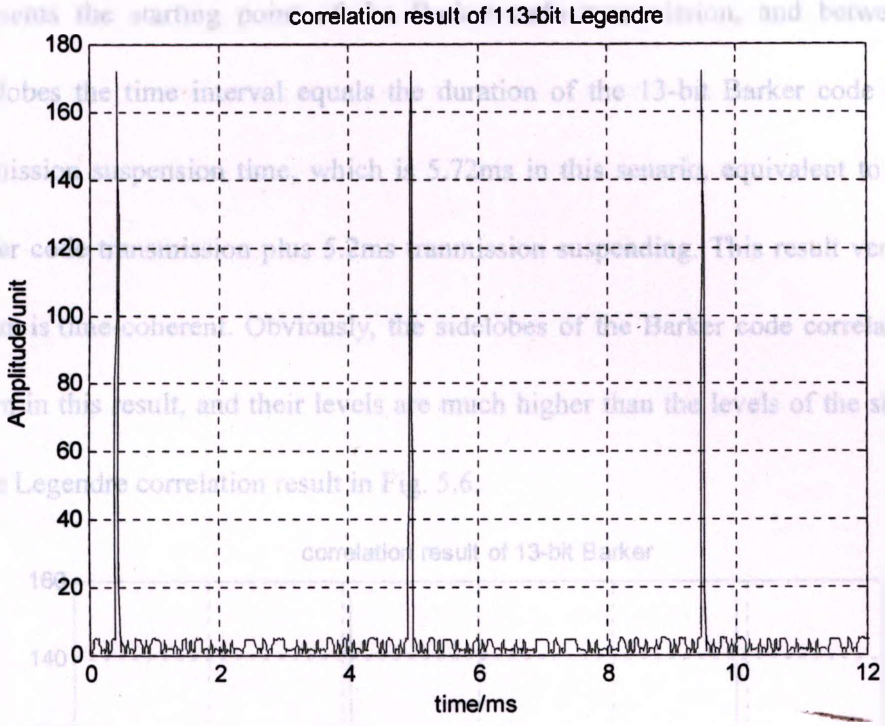


Fig. 5.6 Correlation result without scheduling scheme

5.1.2 Transmission with scheduling scheme

After testing the functionalities of the transmitter and the receiver, a transmission scheduling scheme is plugged into the system for the echo collection. During the laboratory experiments, the hardware setting is exactly the same as the hardware configuration for the receiver functionality testing. It is assumed that plasma keeps high stability which means the vertical velocity of plasma is quite slow, such that the echo collective time doesn't change the measurement in height or phase.

As a comparison, a 13-bit Barker code followed by 5.2ms transmission suspension is tested and the result is given in Fig. 5.7. Each mainlobe of the correlation result represents the starting point of the Barker code transmission, and between two mainlobes the time interval equals the duration of the 13-bit Barker code and the tansmission suspension time, which is 5.72ms in this senario, equivalent to 0.52ms Barker code transmission plus 5.2ms tranmission suspending. This result verifys the system is time-coherent. Obviously, the sidelobes of the Barker code correlation are shown in this result, and their levels are much higher than the levels of the sidelobes of the Legendre correlation result in Fig. 5.6.

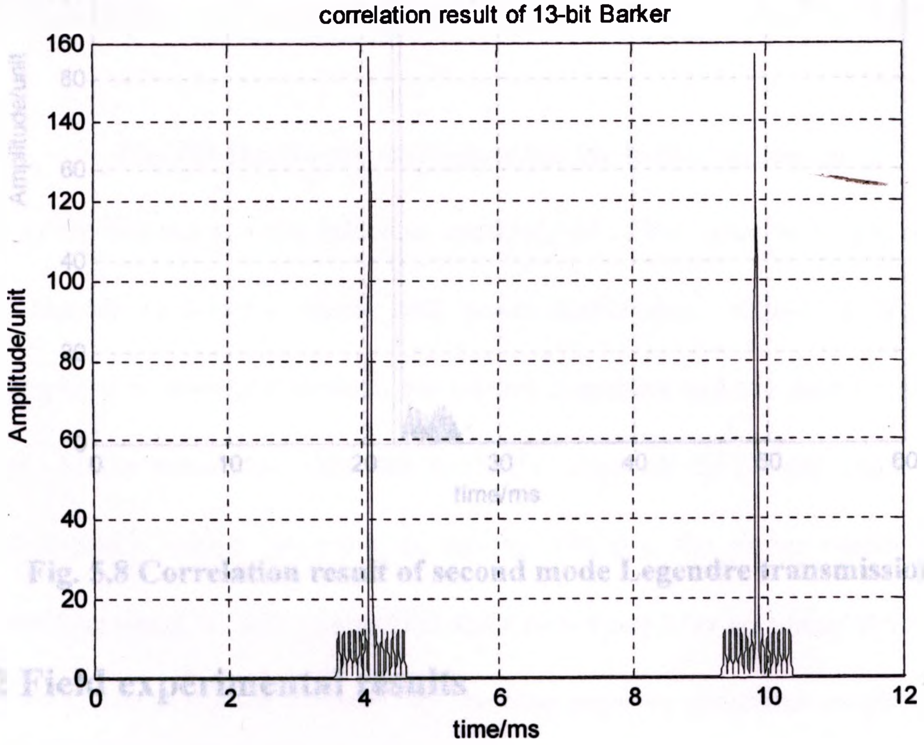


Fig. 5.7 Correlation result of 13-bit Barker code

In order to determine the starting point of whole packet of the data transmission, the received data is correlated with the preamble, which currently is 13-bit Barker code. In this project, only the second height range (discussed in Chapter 4) is tested, Fig. 5.8 depicts the correlation result of the second mode Legendre code transmission with preamble plugged in for synchronization. The first spike of the correlation result represents the preamble starting transmission point. The second one is the beginning of the transmission of Legendre code. Obviously, the time interval between these two spikes is equal to the duration of 13-bit Barker code, 0.52ms.

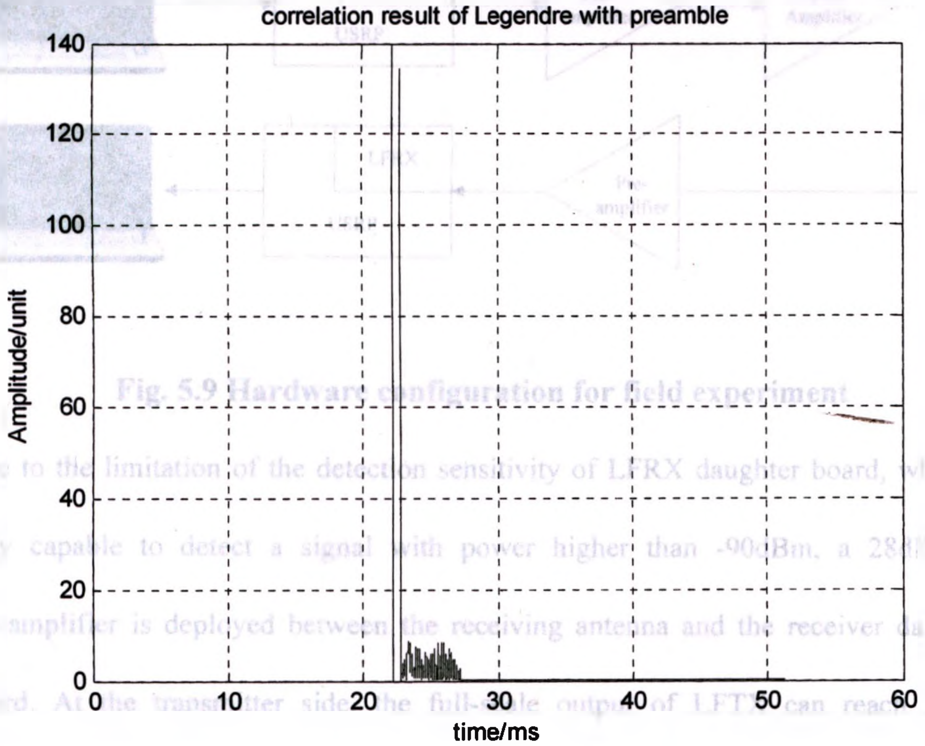


Fig. 5.8 Correlation result of second mode Legendre transmission

5.2 Field experimental results

5.2.1 Experimental setup

The setup for the field experiment is shown in Fig. 5.9. The transmitter transmits data packet and works at 6MHz, and the receiver detect the echo also at 6MHz. A half wavelength dipole receiving antenna is deployed. The length L of the dipoles can be estimated by the following equation. [ARRL handbook, 1991]

$$L(ft)=\frac{492\times 0.95}{f(MHz)}=\frac{468}{f(MHz)} \tag{5.1}$$

When working at 6MHz, the length is 78 feet, or 23.774 meters.

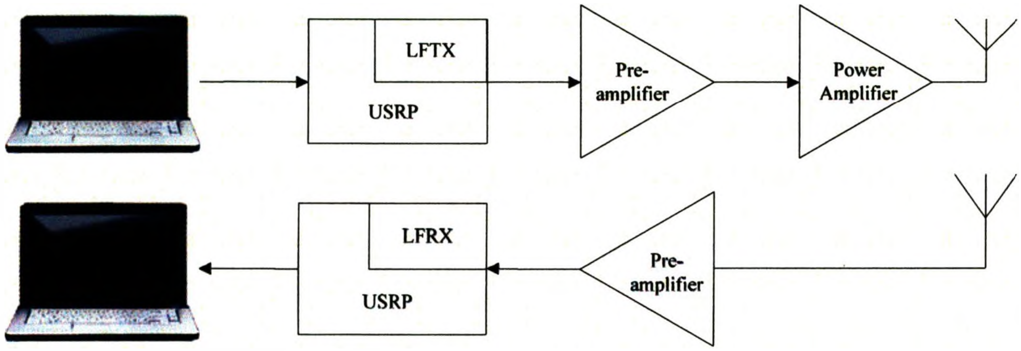


Fig. 5.9 Hardware configuration for field experiment

Due to the limitation of the detection sensitivity of LFRX daughter board, which is only capable to detect a signal with power higher than -90dBm, a 28dB gain pre-amplifier is deployed between the receiving antenna and the receiver daughter board. At the transmitter side, the full-scale output of LFTX can reach 2Vp-p (peak-to-peak) voltage. In order to achieve 30Watts, the power output of the transmitting signal, a 24dB preamplifier and a power amplifier are placed followed by the LFTX. The amplitude of signal from the final output of transmitter daughter board was adjusted to 1.4Vp-p.

5.2.2 Echoes from the Ionosphere

The signal were transmitted and received at the same place at a field site. Basically, the data processing is just to correlate the received signal with the original code and give a plot shows the amplitude as a function of height (from the correlation time delay). As the echo was very weak on the testing day (as shown by the CADI located at the same place), only Barker code processing methods was experimented with. Thus the experiments just transmitted Barker 13 coded signal on 5.5MHz operation frequency with transmitted power of 30W, then suspending the transmitter for 26ms. Fig.5.10 gives the experimental result, which takes the average of signal power every corresponding time in each transmitting cycle for one minute at the observation time in UT. In this result, the origin of x-axis is the time and height of ground wave received, and the echo was returned from 438km in altitude, which agrees the ionogram result that CADI gave at the same time.

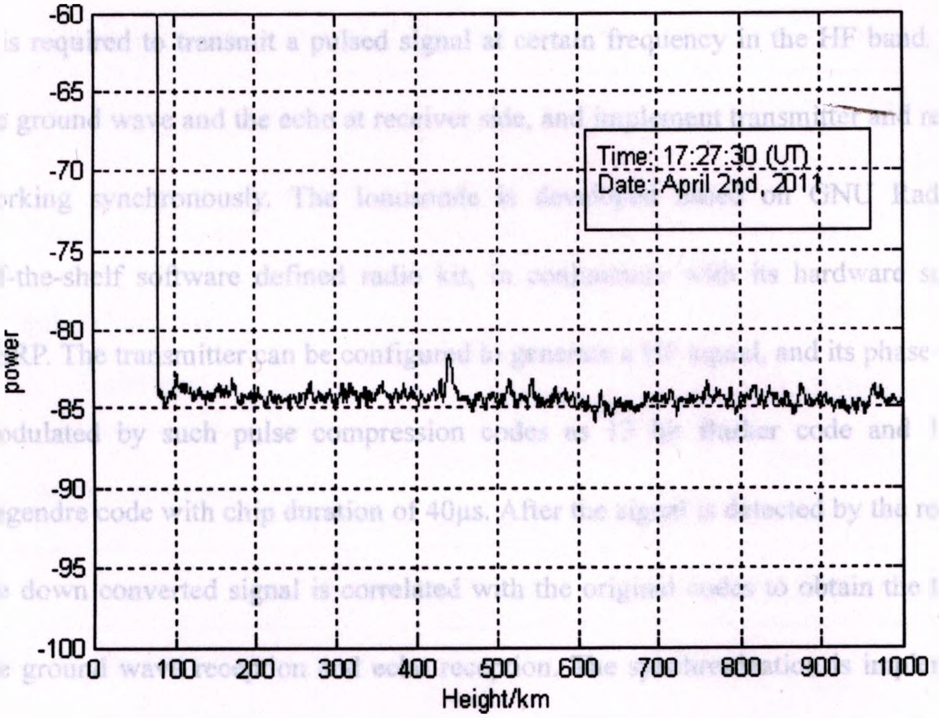


Fig. 5.10 Field experimental result of Barker 13 coded signal at 5.5MHz

Chapter 6

Conclusion and recommendation

This chapter will present the conclusions for this project. Recommendations for the future work will be given.

6.1 Conclusion

The prototype Software GNU Radio based Ionosonde has been developed and tested. It is required to transmit a pulsed signal at certain frequency in the HF band, detect the ground wave and the echo at receiver side, and implement transmitter and receiver working synchronously. The Ionosonde is developed based on GNU Radio, an off-the-shelf software defined radio kit, in conjuncture with its hardware support, USRP. The transmitter can be configured to generate a HF signal, and its phase can be modulated by such pulse compression codes as 13 bit Barker code and 113 bit Legendre code with chip duration of 40μs. After the signal is detected by the receiver, the down converted signal is correlated with the original codes to obtain the time of the ground wave reception and echo reception. The synchronization is implemented by a scheduling scheme where a preamble is inserted in the front of the scheduling scheme in the Legendre scenario. The time delay between the echo and ground wave reception can be calculated knowing the sample rate of the received signal, due to the time-coherence feature of FIFO buffers in the receiver, which is confirmed by the laboratory experiments.

To test the Iononsonde, a standard CADI radar system was operating simultaneously for comparison. The field experiment results agree with those obtained from the CADI system. The Ionosonde works properly and can be used to measure some ionospheric parameters, such as virtual height, phase delay, amplitude etc.. Based on these measurement results, other parameters associated with the Ionosphere can be further deduced, such as Doppler frequency and so forth. A comparison on the specification of CADI and this GNU Radio based Ionosonde is shown in Table 6.1.

Table 6.1 Comparison of CADI and the GNU Radio based Ionosonde

	CADI	the GNU Radio based Ionosonde
Transmitter Power	600W	30W
Pulse Compression Technique	13 bit Barker	113 bit Legendre or 13 bit Barker
Main component	Microcomputers (MCU MC68HC11A1)	Altera FPGA Cyclone IEP1C12
Cost	US $30000	US $2000
External Synchronization devices	GPS	None
Pulse width	40μs	40μs

6.2 Recommendations

6.2.1 Synchronization

The transmitter and receiver can only work on one frequency in the range from 1MHz to 20MHz, which cannot satisfy the requirement of the Ionsonde, which does a frequency sweeping through the whole range of plasma frequencies in the ionosphere. The current implemented synchronization between transmitter and receiver are only capable to set the time of the transmitted signal by utilizing a preamble, which cannot allow the transmitter to actually talk directly to the receiver to precisely enable and disable the receiver and ensure no signal is lost. Newer models of the USRP being currently developed for the GNU Radio use the Universal Hardware Driver (UHD), which is aiming to provide a host driver and Application Programming Interface (API) for current Ettus Research products, such as USRP1, USRP2, USRP-N2 series and USRP-E1 series. UHD emulates some feature on timing issues in software to support the new API: 1) Setting the current device time; 2) Getting the current device time; 3) Transmitting at a specific time; 4) Receiving at a specific time; etc. The FPGA on our USRP1 has insufficient space to support these advanced features, only some of the features are emulated in software, and some are still not offered, such as notification on late transmit packet, notification on late stream command, etc. Therefore, USRP 2, a newer version of hardware support of GNU Radio is also recommended for future work. USRP2 is built based on the USRP1 and has some new features. The comparison of major specifications between USRP1 and USRP 2 is listed in Table 6.2.

To implement the synchronization between transmitter and receiver by USRP2 and the features on timing that UHD builds for USRP 2, data samples can be time stamped, and this time is calibrated to the 1 PPS (Pulse Per Second) signal. In most general scenario, the PPS input is directly fed into the FPGA, which means the frequency that PPS works on can be programmed to generating higher frequency synchronization pulse. USRP 2 provides a built-in MIMO (Multi-input Multi-output) expansion interface, which can synchronize multiple USRP2s through connecting them by a MIMO cable.

Table 6.2 Comparison of the major specifications of USRP1 and USRP2

	USRP 1	USRP2
Interface	USB 2.0	Gigabit Ethernet
FPGA	Altera Cyclone IEP1C12	Xilinx Spartan 3 2000
RF bandwidth to/from host	8MHz @ 16 bit	25MHz @ 16 bit
Cost	$700	$1400
ADC samples	12 bit, 64MS/s	14 bit 100MS/s
DAC samples	14 bit, 128MS/s	16 bit 400MS/s
Daughter board capacity	2 Tx, 2 Rx	1 Tx, 1 Rx
SRAM	None	1 Megabyte
Power	6V, 3A	6V, 3A
PPS(Pulse Per Second)	None	One available
External reference lock	None	10MHz
FPGA configuration loading approach	Automatically loaded through runtime	Stored in a standard SD card

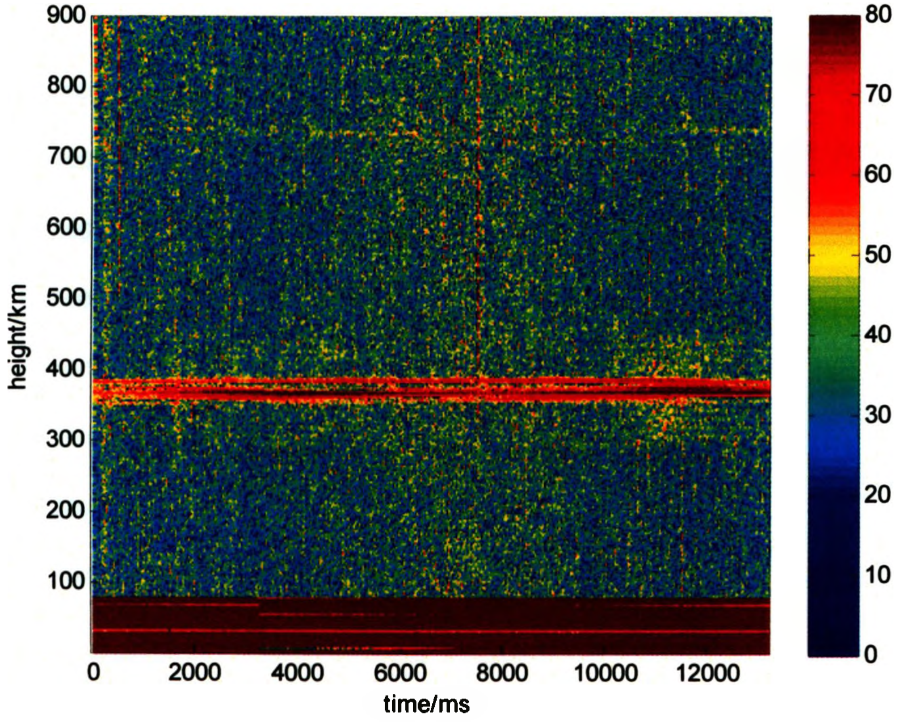
6.2.2 Signal power conversion

The signal power and amplitude given in the experimental results only use the scale of bit units in the digital side of ADC/DAC. This cannot give a high-precision relation between the numerical results in digital world-view and what in the analog world by voltage or in dB scale. Therefore, a "calibration curve" that converts the numerical results into absolute power numbers is required, which can be implemented by a calibration source with calibrated precision attenuators.

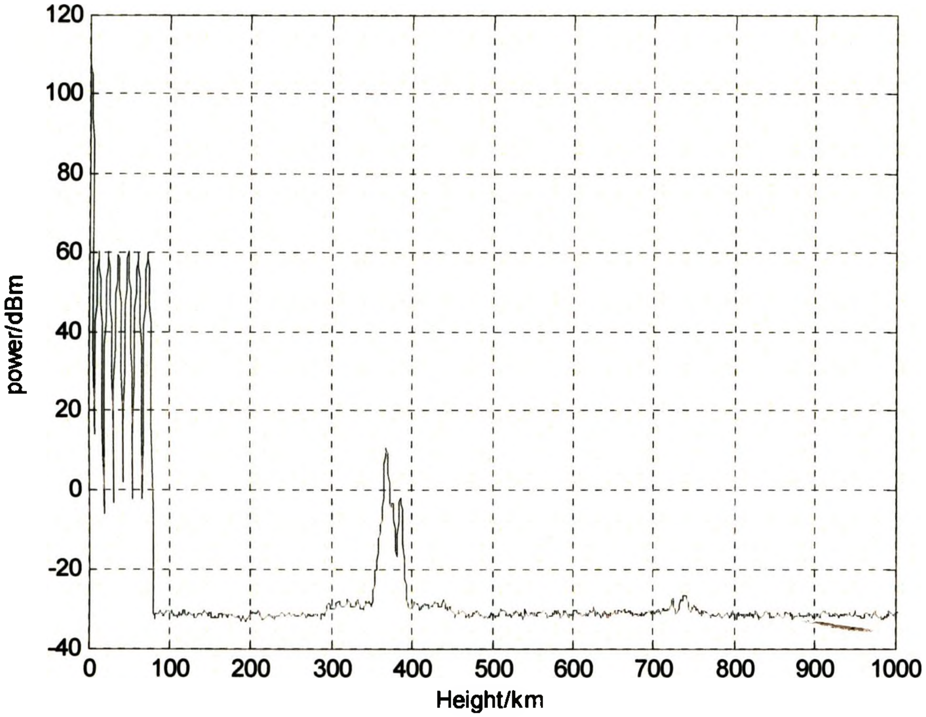
Appendix

1. Field experimental result of 13-bit Barker code

More field experiments were done at April 13th, 2011. The field experiments, using 13-bit Barker code was applied as the coding method with operating frequency 6MHz, was repeated 8 times starting at the different Universal Times. Correlation results are shown in Fig. A.1 to A.6. Part (a) of each Figure gives the plot of power (color) versus virtual height (from the correlation time delay as a function of the observation time starting from the UT given by the title of each figure), while part (b) is the plot averaging the height and power of each echo at every time point in part (a). With time going, the echoes slightly vary ranging from 340km to 370 km in altitude, and the echoes gradually became stronger, as from 20:07:00 (UT), the sidelobes of the correlation results of echoes start to appear and the second hop echoes were returned at around 690km to 700km. The ordinary wave and extraordinary wave were split in the test results in Fig.A.1 and Fig. A.2.

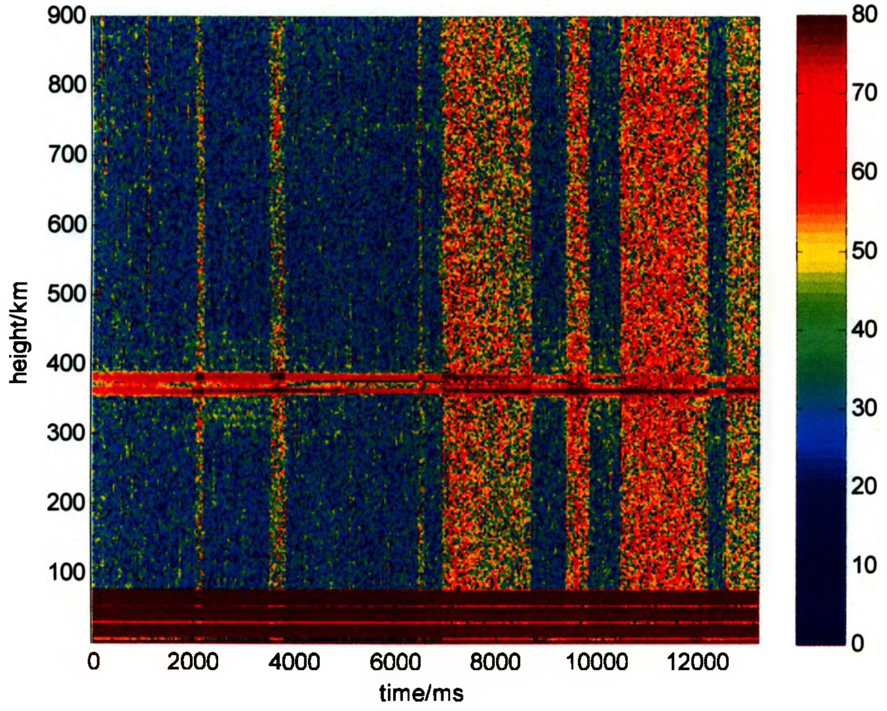


(a)

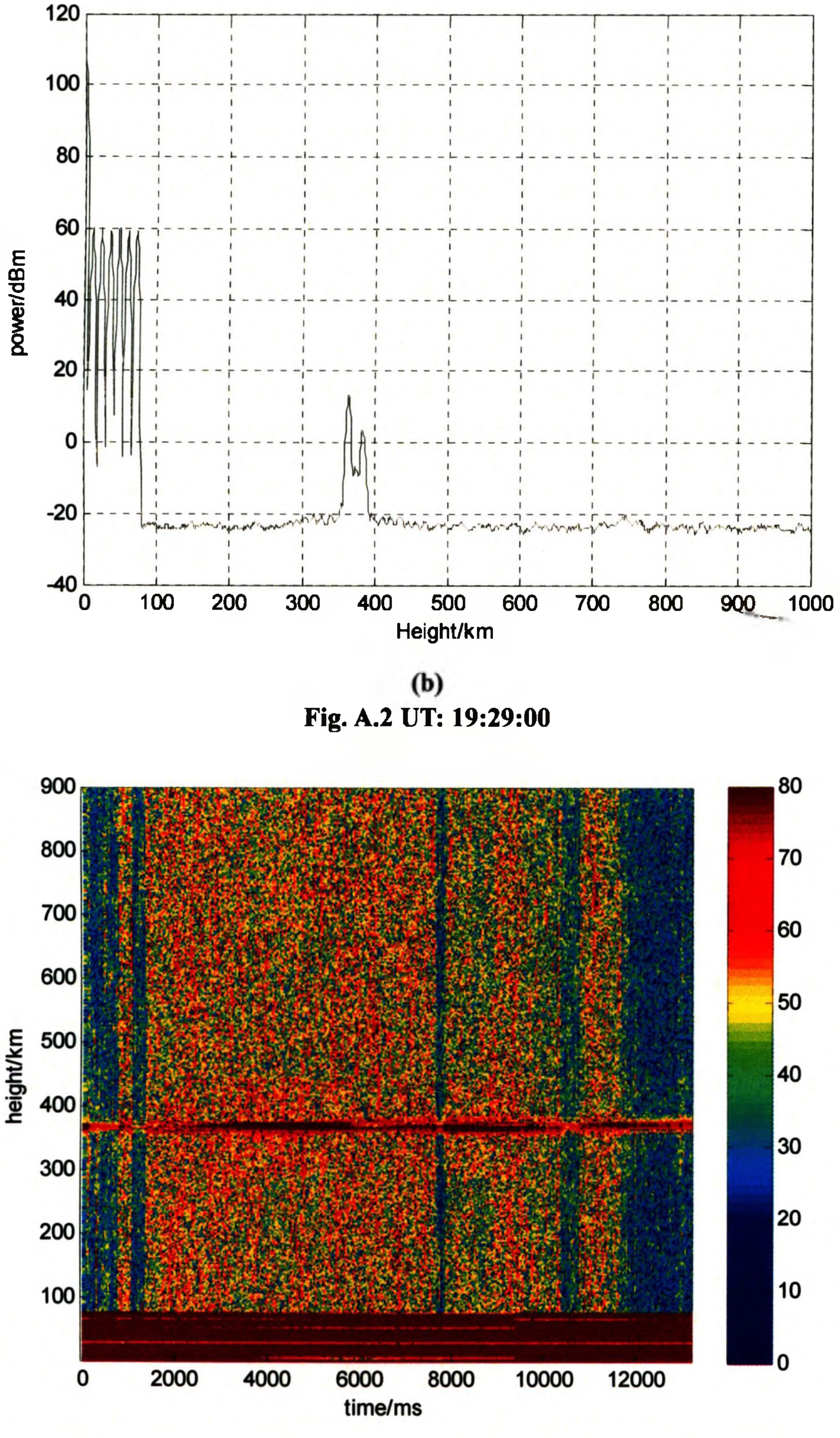


(b)

Fig. A. 1 Universal Time (UT): 19:25:30



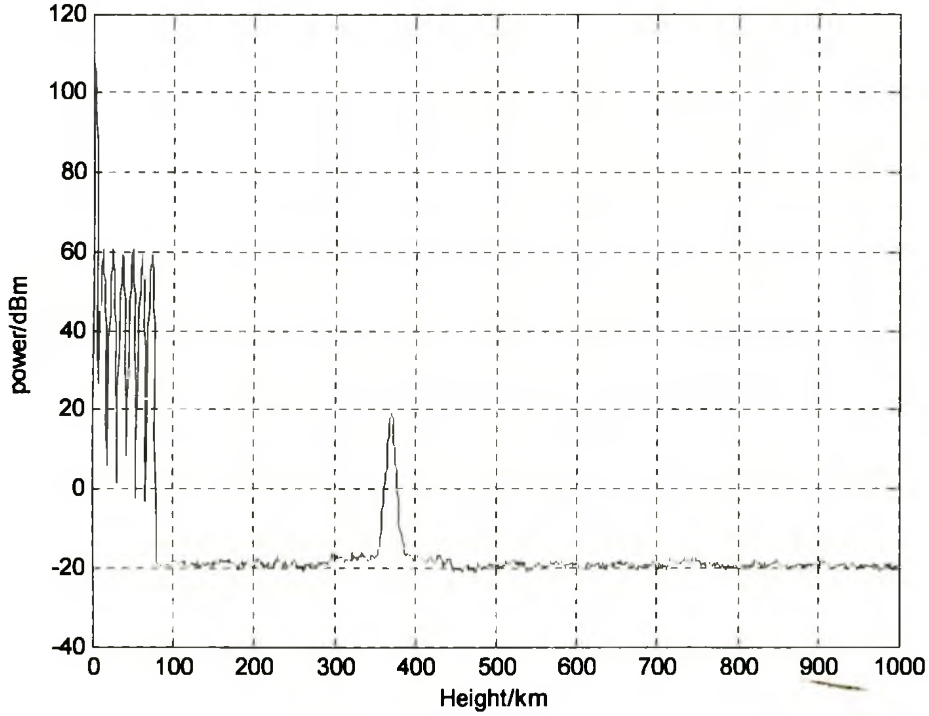
(a)



(b)

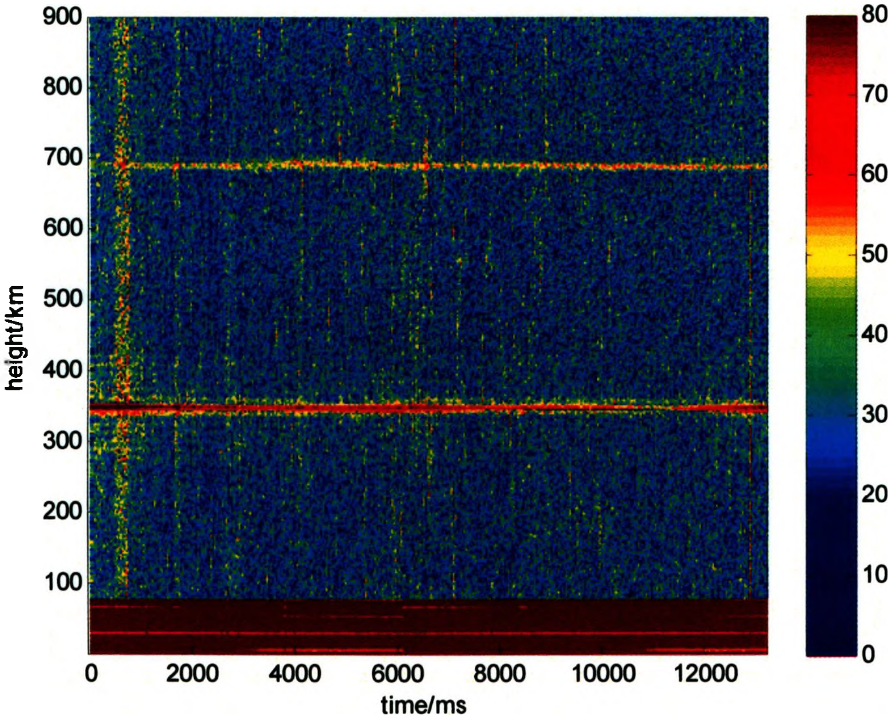
Fig. A.2 UT: 19:29:00

(a)

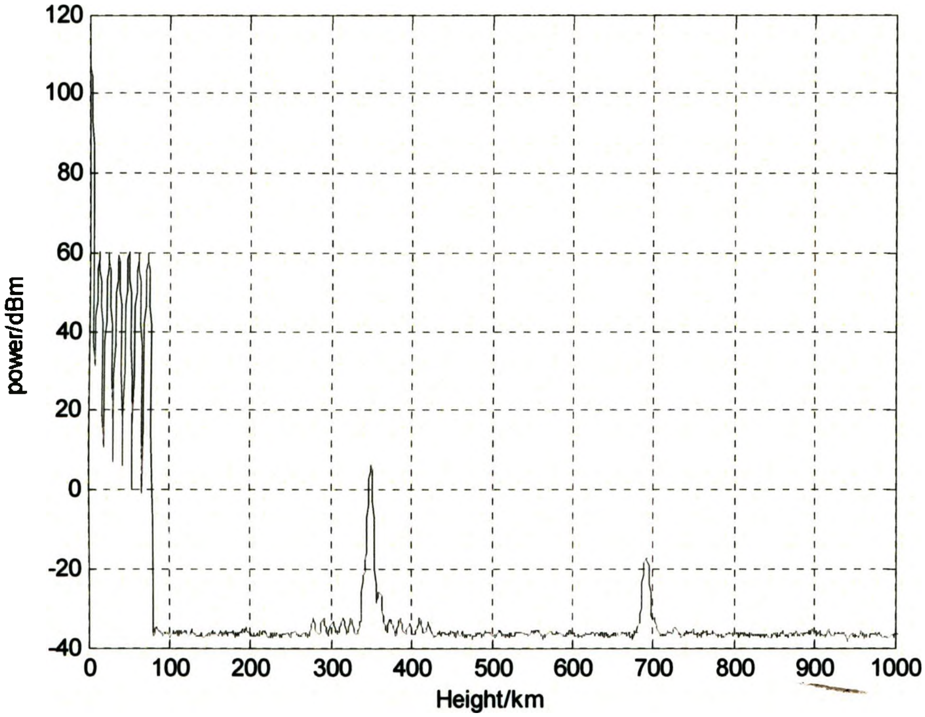


(b)

Fig. A.3 UT: 19:31:30

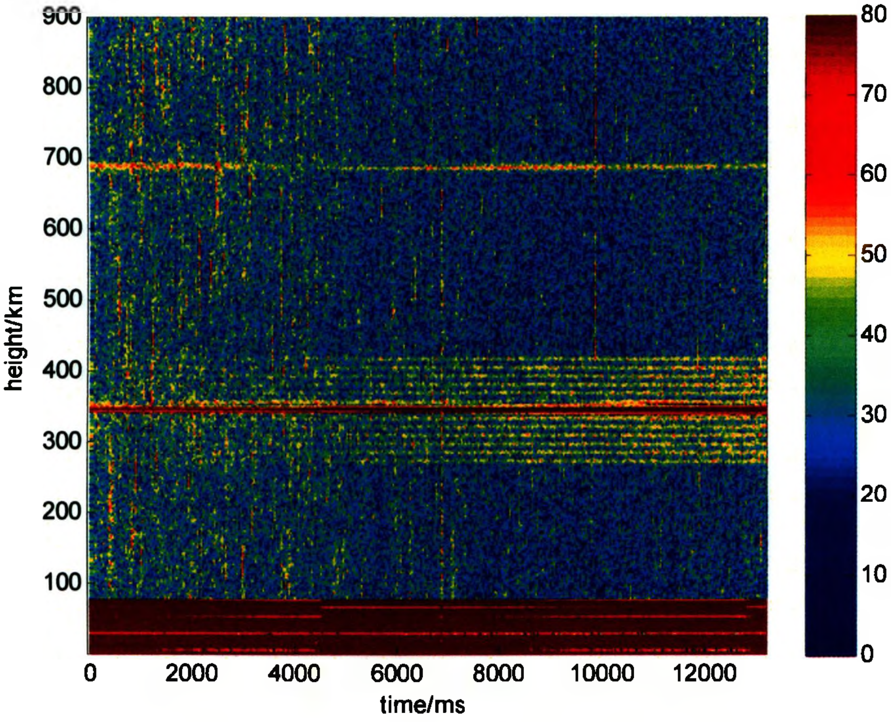


(a)

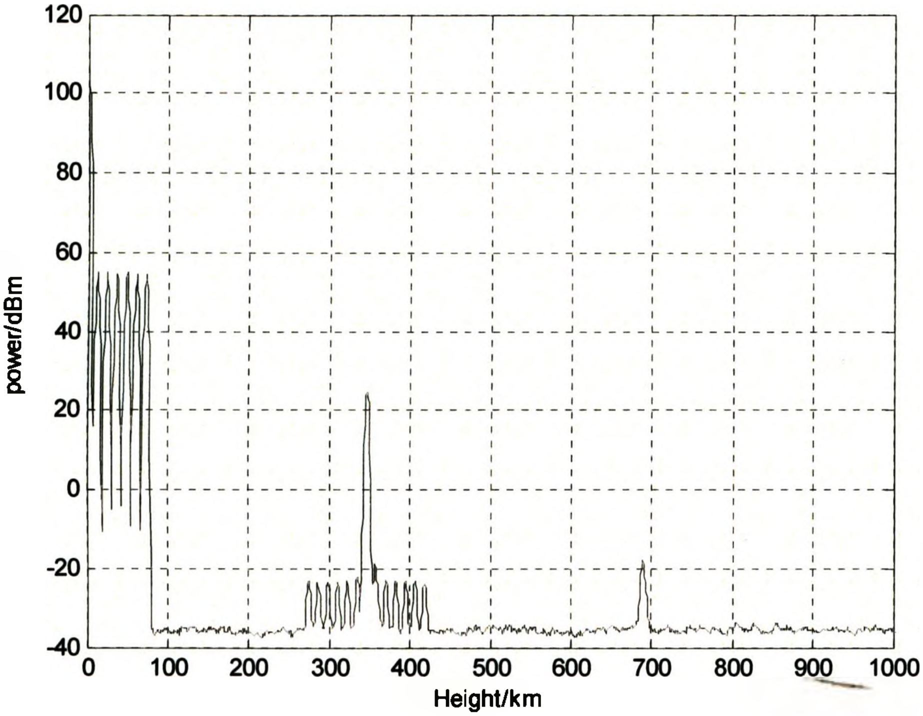


(b)

Fig. A.4 UT: 20:07:30

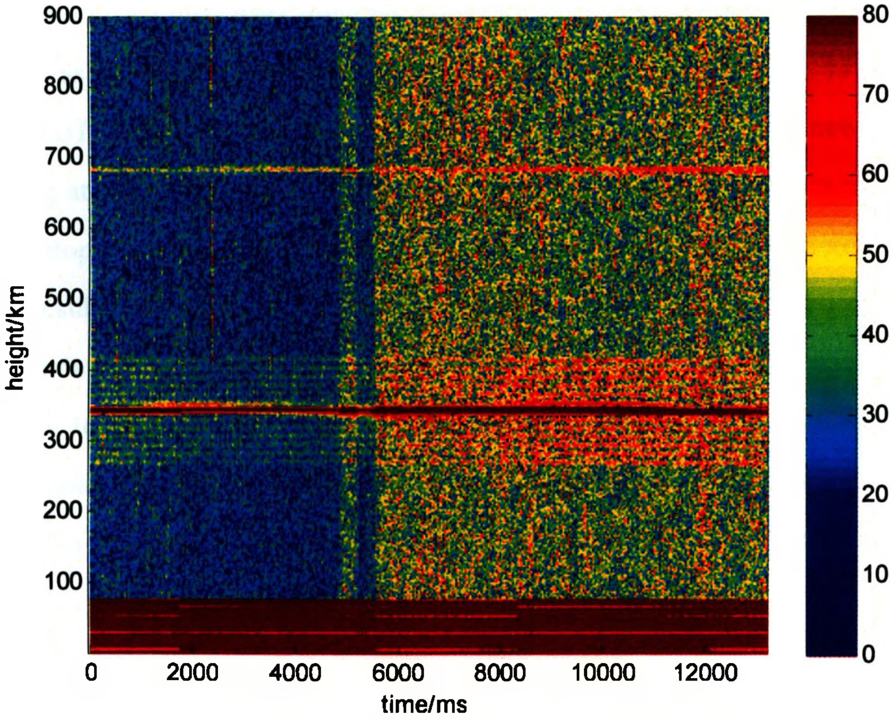


(a)

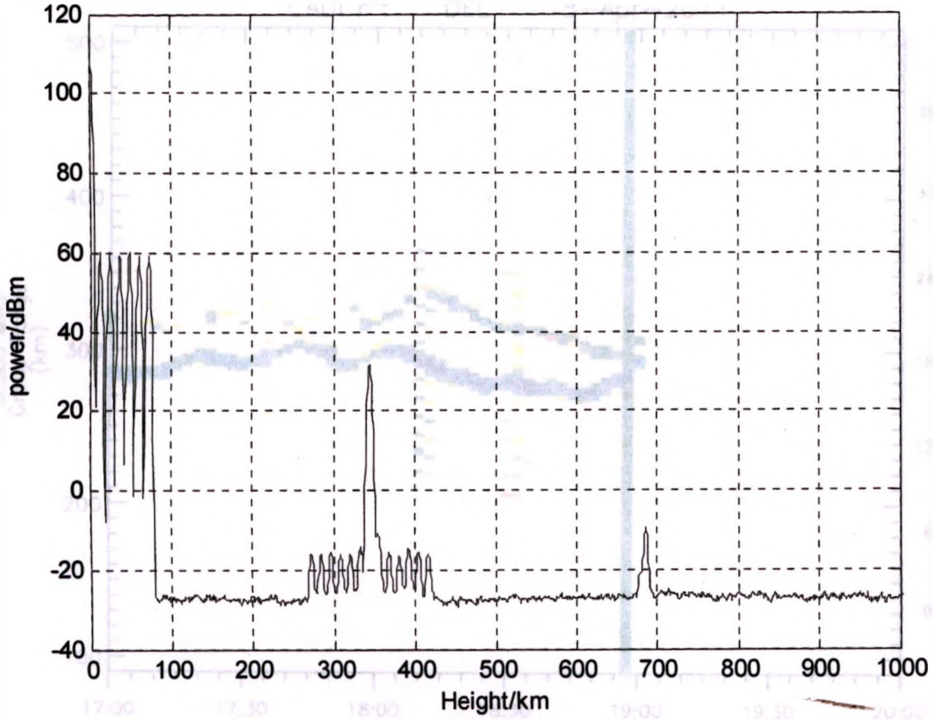


(b)

Fig. A.5 UT: 20:09:00



(a)



(b)

Fig. A.6 UT: 20:10:30

Some more field tests were done on April 18th, 2011, using the same data processing is done as April 18th, 2011. The CADI was running at 6MHz from 17:00 (UT) to 19:00 (UT) with results for comparison shown in Fig. A.7. The new Ionosonde working at 6MHz was tested starting from 19:24:30. Starting only a few minutes after CADI stopped, so that we assume the plasma didn't move very much vertically. The same results of virtual height of the plasma at 6MHz given by CADI and the new Ionosonde are therefore expected. The virtual height of the plasma at 6MHz given by the new Ionosonde is around 280km to 290km which agreed with CADI, ranging from 260km to 300km. The weak second hop was returned at around 560km to 580km in the experiment started at 19:31:00 (UT). With time the echo became stronger.

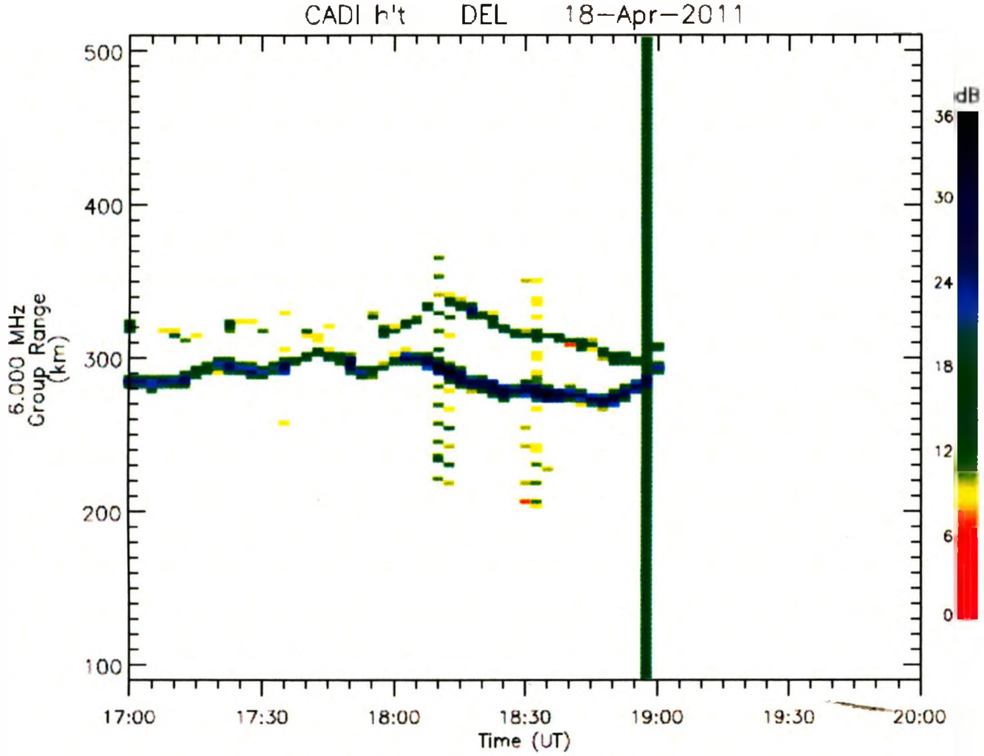
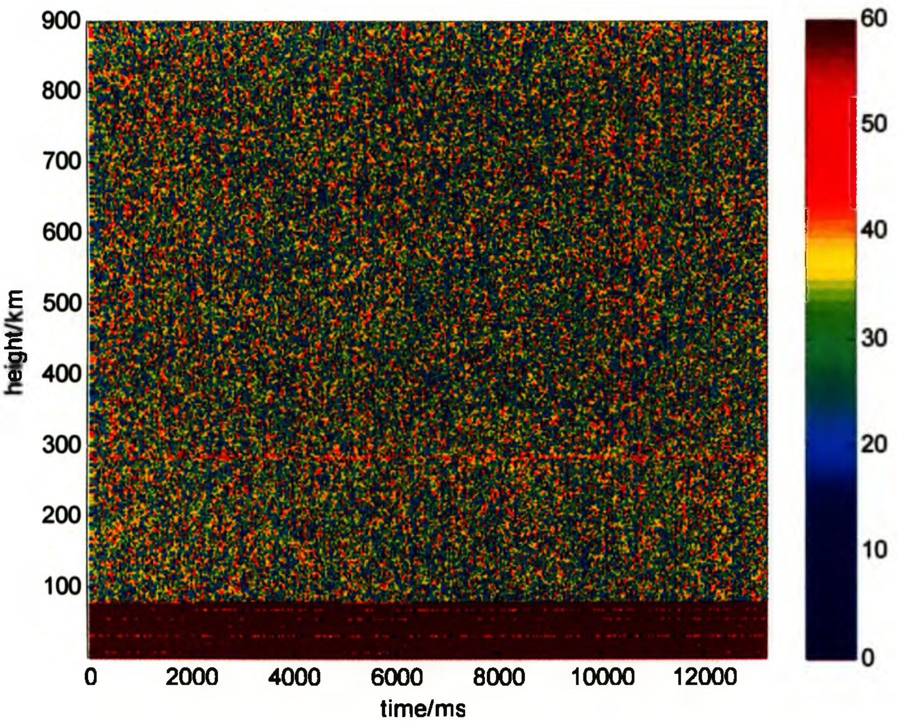
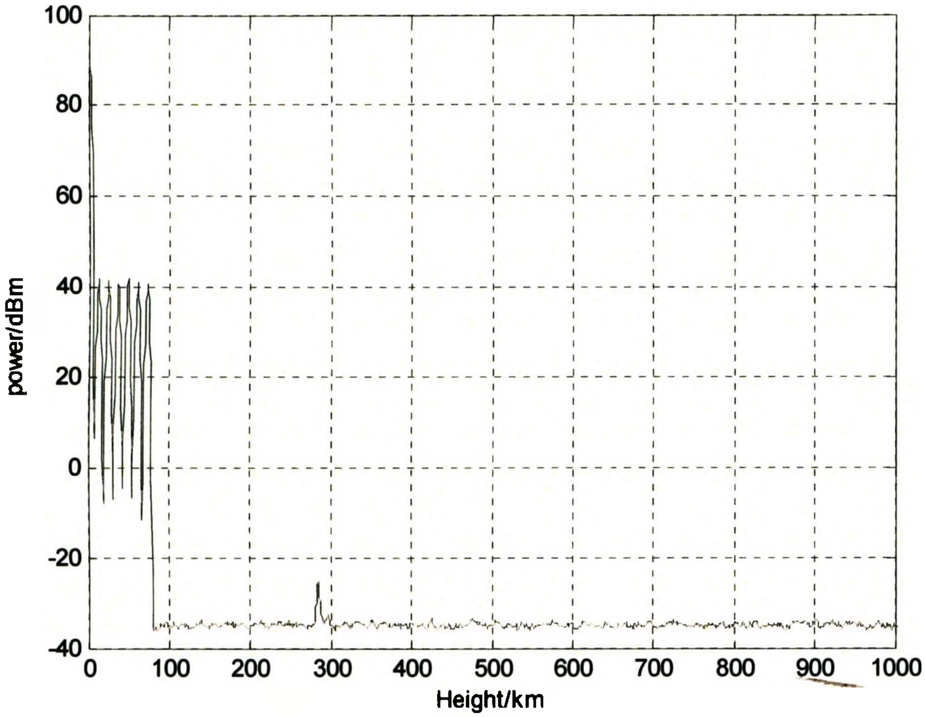


Fig. A.7 CADI echo plot @ 6MHz

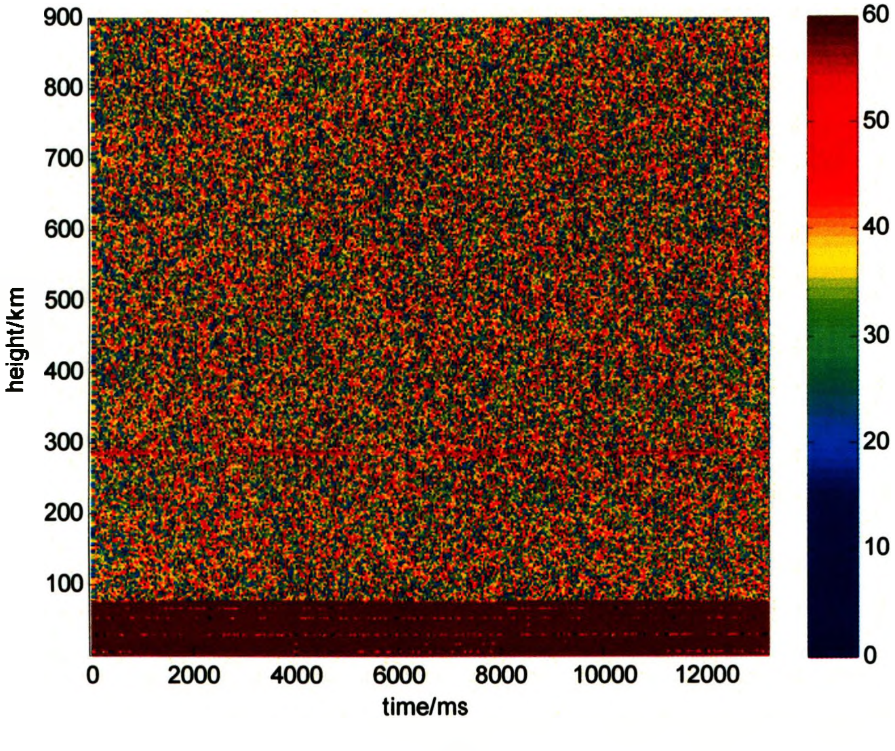


(a)

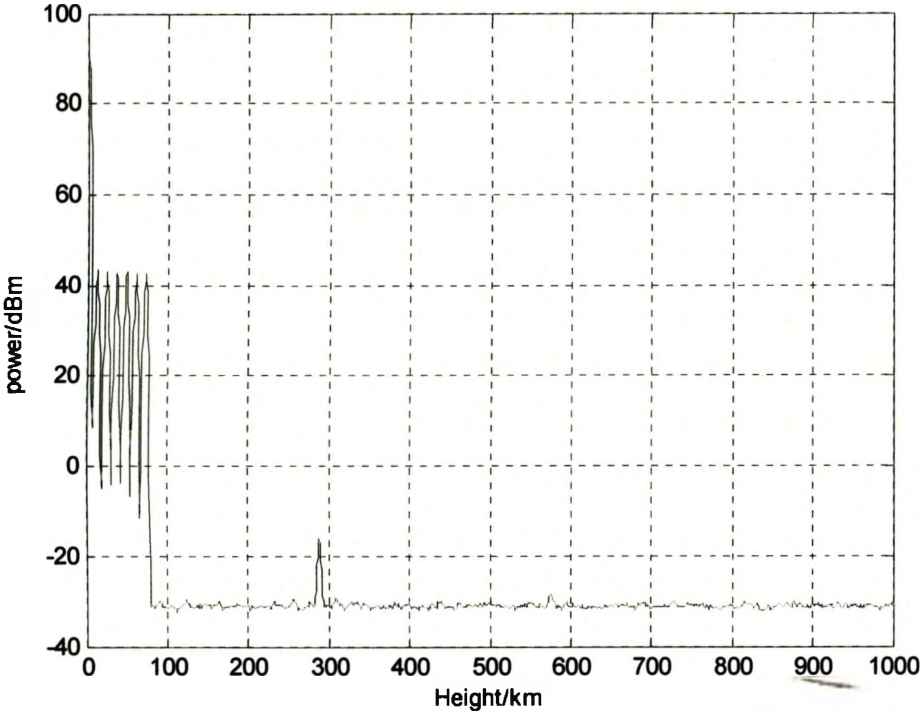


(b)

Fig. A.8 UT: 19:24:30

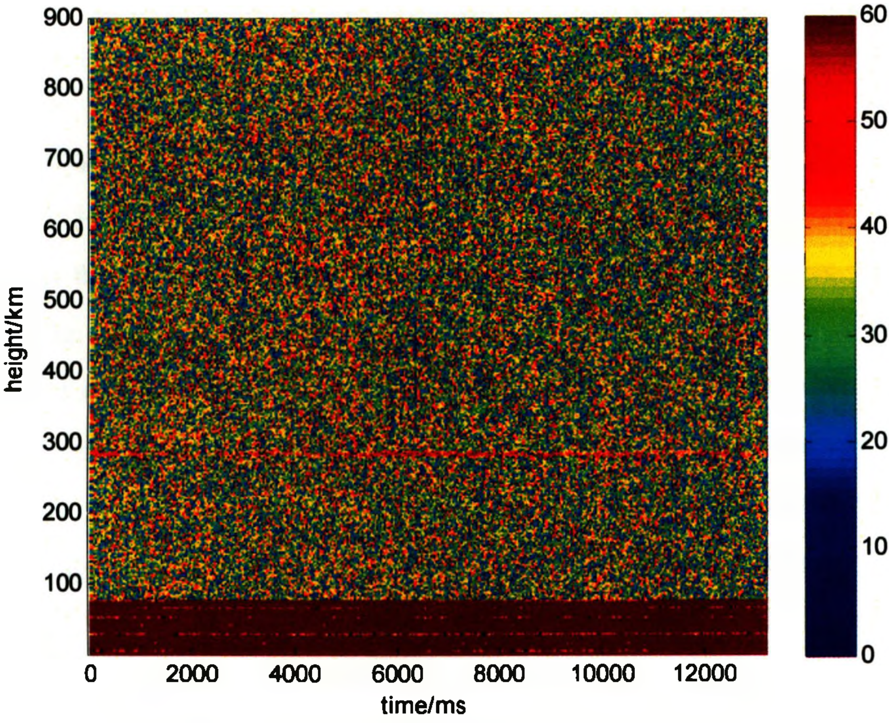


(a)

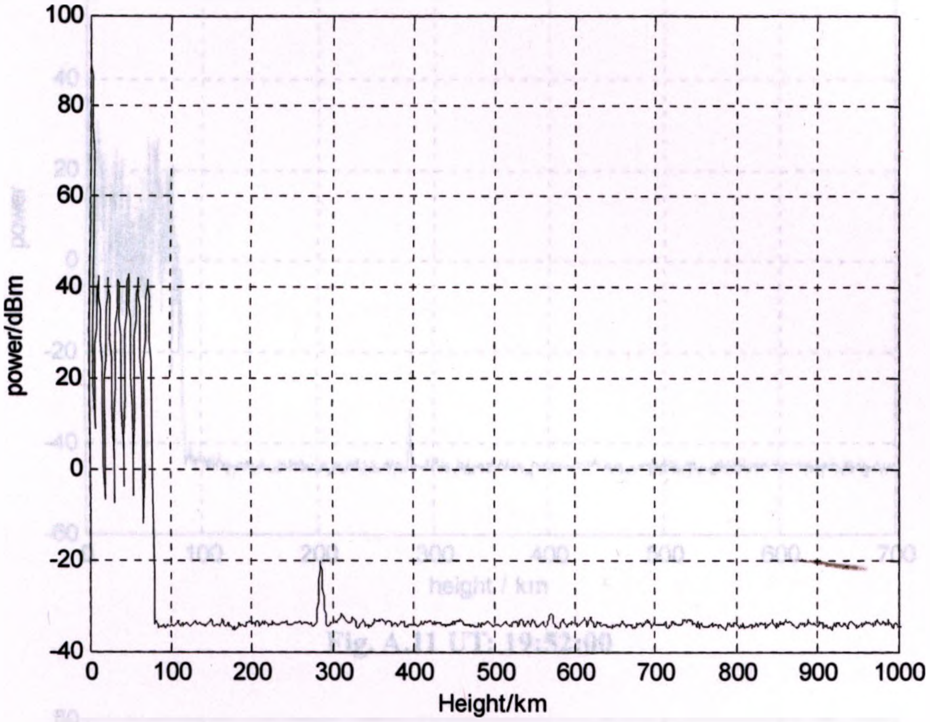


(b)

Fig. A.9 UT: 19:31:00



(a)



(b)

Fig. A.10 UT: 19:33:00

2. Experimental results of Legendre coding (5μs pulse width)

Using 113-bit Legendre coding, another approach was used to avoid the interference between the ground wave and the echo. This was to reduce the pulse width of the code. Fig. A. 11 and A.12 show the field experimental result, which is the average power of several hundred transmission cycles versus virtual height, when a 5μs pulse width Legendre code is applied as the coding method. The new Ionosonde was tested at 6MHz operating frequency on April 18th, 2011, just after the experiment of Barker 13 coding at the same day, hence the same virtual height of plasma at 6MHz is expected. The virtual height of plasma given by this 5μs pulse width Legendre coding method is around 280km, which agrees with the virtual height range CADI showed.

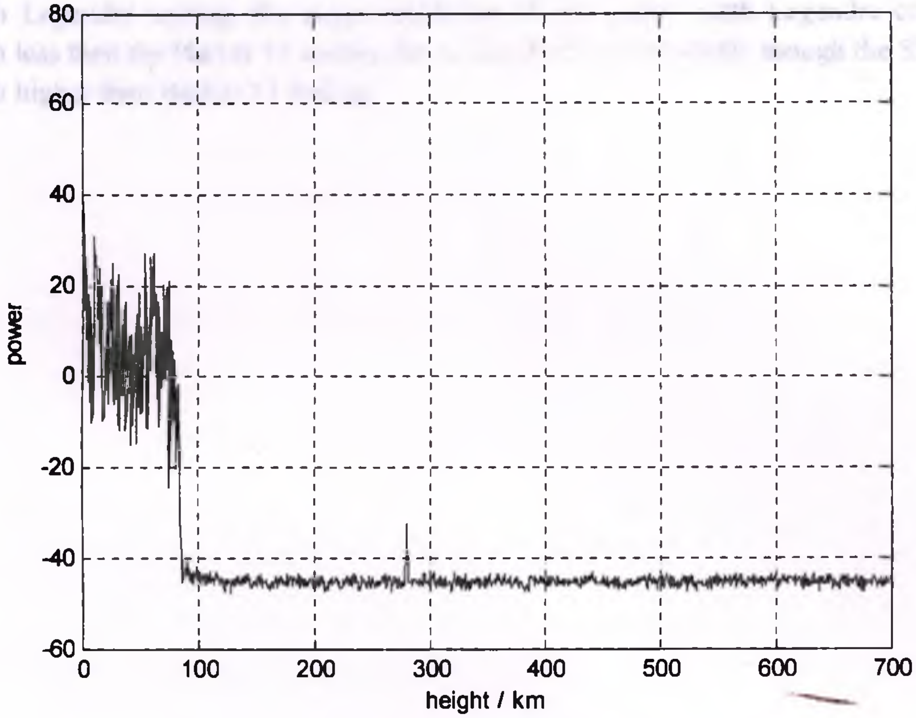


Fig. A.11 UT: 19:52:00

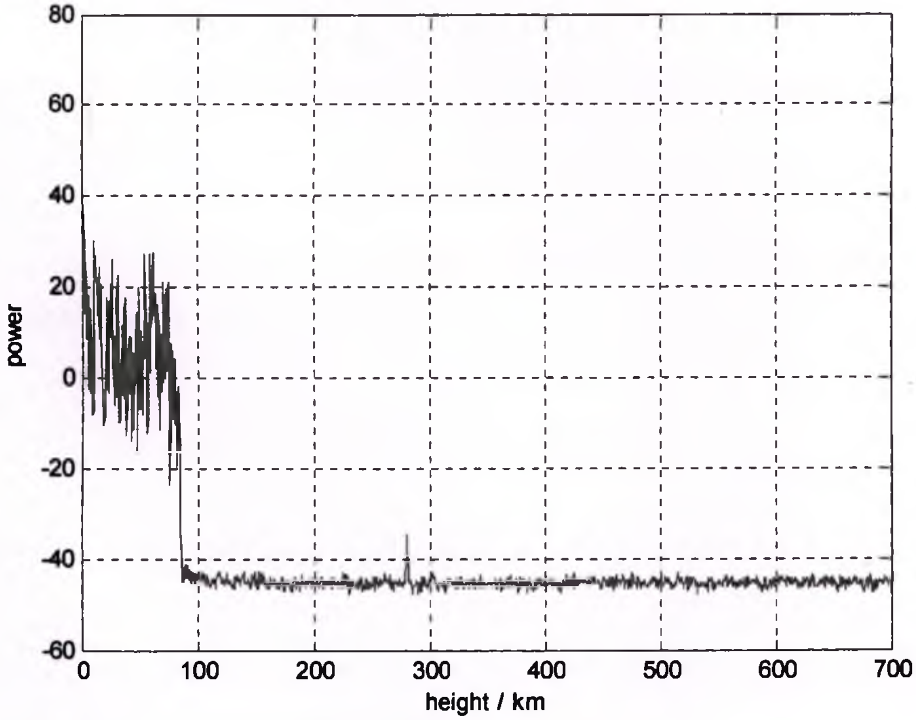


Fig. A.12 UT: 19:53:30

Comparing with the experimental result of Barker 13 coding method and 5μs pulse width Legendre coding, the range resolution of 5μs pulse width Legendre code is much less then the Barker 13 coding due to the shorter pulse width, though the SNR is much higher than Barker 13 coding.

References

Blossom, E., *GNU Radio: Tools for exploring the Radio Frequency Spectrum*, Linux Journal, Vol.? No. 122, June 2004.

Chapman, S.J., *MATLAB Programming for Engineers*, Thomson, 2004.

Davies, K., *Ionospheric Radio Waves*, Blaisdell publishing Company, 1969.

Davies, K. *Ionospheric Radio*, Peter Peregrinus Ltd., 1990.

Ettus Research LCC http://www.ettus.com/, 2011.

Edde, B., *RADAR Priciples, Technology, Applications*, PTE Prentice Hall, 1993.

Hunsucker, R.D., *Radio Techniques for Probing the Terrestrial Ionosphere*, Spinger - Verlag, 1991.

Gao, S. X., A pulse compression ionospheric radar, M.S. thesis, Dept. of Elect. Eng., The University of Western Ontario, 1991.

GNU Radio http://gnuradio.org/redmine/wiki/gnuradio, 2011.

Hamza, F. A., *The USRP under 1.5X Magnifying Lens!* June 2008.

Huang, J., Pulse coding for ionospheric radar, M.S. thesis, Dept. of Elect. Eng., The University of Western Ontario, 2003.

Hargreaves, J.K., *The Upper Atmosphere and Solar-Terrestrial Relations*, Uan Nostrand Reinhold, 1979.

Johnson, F. S., *Setellite Environment handbook*, Stanford University Press, 1965.

Nathanson, F.E., *Radar Design Princilples*, Mc Graw Hill, 1990

Joseph Mitola, III, Software Radio Architecture: Object Oriented Approaches to Wireless Systems Engineering. John Wiley and Sons, 2000.

Kingsley, S., Quegan, S., *Understanding Radar Systems*, 1992.

Li, G., A Dual Frequency Transmitter for Enhanced Polar Outflow Probe (e-POP) Mission, M.S. thesis, Dept. of Elect. Eng., The University of Western Ontario, 2007.

Lin, M., *Digital System Designs and Practices*, John Wiley & Sons, 2008.

Levanon, N., *Radar Priciples*, John Wiley & Sons, 1988.

Mahafza, B.R., *Radar Systems Analysis and Design Using MATLAB*, Chapman & Hall, 2005.

MacDougall, J., CADI software, hardware and Plotting manual, 2000.

Peebles, P. Z., *RADAR Principles*, John Wiley & Sons, 1998.

Ratcliffe, J.A., *Sun, Earth and Radio*, MacGraw-Hill, 1970.

Skolnik, M.I., *Introduction to Radar Systems*, Mc Graw Hill, 2001.

Skolnik, M. I., *RADAR Handbook*, Mc Graw Hill, 2008.

Wolfgang, L. D., etc., *The ARRL Handbook for Radio Amateurs 68th*, Published by the American Radio Relay League, Newingto, 1991.

VITA

Name:	Yan Nie
Post-secondary	Beijing Technology and Business University
Education and	Beijing, P. R. China
Degrees:	2002 - 2006 B.E.
	The University of Western Ontario
	London, Ontario, Canada
	2008 – 2011 M.Sc.
Honours and	University Scholarship
Awards:	2002-2006
Related Work	Teaching Assistant
Experience	The University of Western Ontario
	London, Ontario, Canada
	2009-2010