MEE 07:05

Unmanned Aerial Vehicle (UAV) Communications[1]

Shahid Mahmood

Master Thesis

Master of Science in Electrical Engineering

Blekinge Institute of Technology, Sweden

March 5, 2007

[1]This work has been conducted under an occupational training scholarship at the Institute for Telecommunications Research (ITR), University of South Australia, and was sponsored by the Sir Ross and Keith Smith Foundation.

External Advisor

Prof. Bill Cowley.
Institute for Telecommunications Research (ITR).
University of South Australia.
SPRI Building, Mawson Lakes Campus.
South Australia,
Adelaide.
Email: Bill.Cowley@unisa.edu.au
Phone: +61 8 8302 3858

University Advisor

Prof. Hans-Jürgen Zepernick.
Blekinge Institute of Technology.
School of Engineering.
SE - 371 25 Ronneby, Sweden.
Email: hans-jurgen.zepernick@bth.se
Phone: +46 457 385718

Compiled at

Institute for Telecommunication Research (ITR).
University of South Australia.
SPRI Building, Mawson Lakes Campus
South Australia 5095,
Adelaide.

DEPARTMENT OF SIGNAL PROCESSING

Blekinge Institute of Technology.
School of Engineering.
SE - 371 25 Ronneby.
Sweden.
Internet: www.bth.se
Phone: +46 457 3850 00
Fax: +46 457 271 29

Abstract

This project deals with a small Unmanned Aerial Vehicle (UAV) communication system. The considered distance between UAVs is 100 m to 10 km. These small UAVs are usually used for civil purposes like fire fighting, for farmers etc. This report presents the modem architecture for a small UAV radio network, which works with flexible transmission data rates (62 kbps to 744 kbps) by using an Orthogonal Frequency Division Multiplexing (OFDM) technique with adaptive resource allocation. It includes the instructions and architecture of an Fast Fourier Transform (FFT) processor for a single radix - 2 butterfly engine. The architecture is being modeled in Altera Hardware Description Language (AHDL). For logic synthesis we have used Altera functions. An OFDM technique with adaptive modulation and coding is used to readily achieve variable data rates and to provide a multipath resistant solution. We also investigate different FPGAs' characteristics including memory, multiplexers and logic cells, to identify a low complexity and low cost solution. A Cyclone II FPGA is considered likely to implement the processing required for small UAV communication with the ground station and between UAVs.

Acknowledgements

This work is an outgrowth of my research work at the Institute for Telecommunications Research (ITR) Adelaide, South Australia, under the supervision of Professor William G. Cowley. My work was regarding the modem design for small Unmanned Aerial Vehicles (UAVs). First of all I would like to give profound thanks to my advisor Prof. William G. Cowley for his outstanding guidance and support during my stay at ITR. He is a fascinating researcher and a thoughtful mentor, always eager to share his expertise with his students. I benefited greatly by working under his guidance. His friendship and encouragement have been invaluable throughout my research work at ITR. I am especially grateful for his limitless patience during my trying times.

It is also pleasure to acknowledge Dr. Ian Holland who provided assistance in editing my thesis and gave me important advice regarding OFDM work. I also would like to thank Marc Mirza and Marc Lavenant for sharing with me their considerable knowledge regarding the Altera Hardware Descriptive Language (AHDL) and gave me lots of beneficial advice throughout my research work in ITR.

I want to pay special thanks to Professor, Dr. Hans-Jürgen Zepernick who provided me the opportunity to complete my research work in a world renowned research

institute, regarding telecommunications and satellite research.

Finally, I would like to thank my parents for their constant prayers, love and support, and for the encouragement to pursue my goals.

Contents

Abstract ii

Acknowledgements iii

Contents v

List of Figures vii

List of Tables viii

List of Abbreviations ix

1 Introduction 1

1.1 History of UAVs 1

1.2 Literature Review 3

1.3 Future Work and Usages 6

1.4 Objective 7

2 Modem Design for Small UAVs 8

2.1 Introduction 8

2.2 RF Section 9

2.2.1 Transmitter 9
2.2.2 Receiver 12
2.3 Signal Processing Block 14
2.3.1 OFDM Based Transmitter 15
2.3.2 OFDM Based Receiver 18

3 FPGA Implementation **20**
3.1 Introduction 20
3.2 FFT and Butterfly Operation 21
3.3 Butterfly Operation 24
3.4 Single FFT Processor 25
3.4.1 Butterfly Processor 28
3.4.2 Dual Random Access Memory 29
3.4.3 Read Only Memory 29
3.4.4 State Machine 30
3.5 Available Resources 31
3.6 Performance Evaluation on Different FPGAs 32

4 Conclusion and Recommendations **34**

A Poster in AusCTW 2007 Regarding this Modem Design **36**

B Specification of FFT **49**

Bibliography **52**

List of Figures

1.1 Small UAV 3

2.1 Up Conversion 11
2.2 Down Conversion 13
2.3 Down Conversion 14
2.4 FPGA Based Transmitter and Receiver 15
2.5 Modulator Structure 16
2.6 OFDM Based Transmitter 16
2.7 OFDM Frame 17
2.8 Demodulator Structure 19
2.9 OFDM Based Receiver 19

3.1 Butterfly. 22
3.2 Signal Flow Graph for 8 Point FFT. 24
3.3 Schematic of Single Butterfly. 25
3.4 Different Butterfly Structures. 27
3.5 FFT Processor. 28
3.6 State Diagram of Control Processor. 31

List of Tables

2.1 Allowable modulation modes for the example adaptive modulation scheme. 10

2.2 Possible Modulation and Coding Modes for the Small UAV Modem. . 17

3.1 Comparison of number of computations in direct DFT and FFT. . . . 23

List of Abbreviations

UAVs	Unmanned Aerial Vehicles
DARPA	Defense Advanced Research Projects Agency
FAA	Federal Aviation Administration
NASA	National Aeronautics and Space Administration
DPI	Department of Primary Industries
FPGA	Field Programmable Gate Arrays
AHDL	Altera Hardware Descriptive Language
VHDL	Verilog Hardware Descriptive Language
DOD	Department of Defence
FFT	Fast Fourier Transforms
DFT	Discrete Fourier Transforms
IFFT	Inverse Fast Fourier Transforms
CP	Cyclic Prefix
OFDM	Orthogonal Frequency Division Multiplexing
LPM	Library of Parameterized Modules

Chapter 1

Introduction

1.1 History of UAVs

During World War I the first attempt to create an airborne counterpart of the naval torpedo took place in the United States. A plane without a human pilot was to be guided to a target and crashed into it in a power dive, exploding its charge.

The Hewitt - Sperry Automatic Airplane (a prototype) made a number of short test flights proving that the idea was sound in 1916-17. In 1917, the USA Army started a similar aerial torpedo, or flying bomb, project led by Lieut. Col. Bion J. Arnold for the Air Service and Charles Kettering for industry.

After that, various companies worked together and produced 20 complete pilotless aircrafts and a successful test flight was made Oct. 4, 1918. The further research and development were closed on these planes. However, the Navy's Bureau of Ordinance

decided to follow up one aspect of the over-all problem of the aerial torpedo and to develop a radio-controlled plane. An N-9 trainer seaplane was used as the basic "vehicle" and rebuilt with stabilization and radio control equipment developed by the Naval Research Laboratory and by Carl Norden.

In Sept. 15, 1924 a successful flight without a pilot aboard took place but the plane was damaged in landing and sank. Thus ended the career of the first of the drones, as pilotless planes not used for combat are now called. With the advances in technology and research in electronics the Navy started another drone program which was intended to provide realistic targets for antiaircraft gunnery practice but which directly influenced missile development. Lieutenant Commander (later Rear Adm.) D.S. Fahrney was in charge of the project. In this project the Stearman Hammond JH -1 plane was used and the radio system was developed by the Naval Research laboratory. This drone made its first successful flight in Nov. 15, 1937 and was later used for target practice by the antiaircraft batteries of USS Ranger. After the success of this project all the torpedos and dive bombers were converted into missiles. The converted dive bomber was crashed into a raft towed by a tug in Chesapeake Bay on April 19, 1942. The controlling pilot who "flew"the drone by television was 11 miles distant at the time. These tests proved that assault drones were practical, and various planes were converted and used in World War II.

Four missiles were developed in the United States: Little Joe, an antiaircraft missile propelled by solid fuel rockets; and three types of Gorgon, with pulse jet engine, turbojet engine, and liquid fuel rocket motor, respectively [1].

1.2 Literature Review

With the recent advances in miniature electronics a new avenue for unmanned aerial vehicles (UAVs) has emerged. Using today's low-cost, off-the-shelf miniature actuators, receivers and computers, an affordable and small UAV can be successfully designed and built within a 9-month period. Unmanned aircraft, known variously as "drones", "robot planes", "remotely piloted vehicles (RPVs)", and now "unmanned aerial vehicles (UAVs)", have been a feature of aviation for much of its history, though in limited or secondary roles. In the 21st century, the technology seems to be headed towards greatly expanded use [2].

Figure 1.1: Small UAV

These planes have been introduced by the Department of Defense (DOD) as powered, aerial vehicles that do not have any pilot inside to operate them. There are five types of UAVs:

1. **Target and Decoy**

   These UAVs are used to simulate an enemy aircraft or missile on ground and aerial gunnery.

2. **Reconnaissance**

   These UAVs provide battlefield intelligence.

3. **Combat**

   These UAVs are used to get the information about attack capability for high risk missions.

4. **Research and Development**

   These are used for further development and research of UAV technology.

5. **Civil and Commercial UAVs**

   Usually these types of UAVs are smaller and used for only commercial and civil purposes [3].

These UAVs can fly autonomously or controlled through remote and can carry payload but according to DOD ballistic, cruise missiles and artillery projectiles are not considered UAVs. They usually work with one ground control station. DOD currently has five types of UAVs: the Air Force's Predator (which is used to extend the eyes of submarines and in armed helicopters to improve targeting) and Global Hawk, the Navy and Marine Corps's pioneer, and the Army's Hunter and Shadow. Other key UAV developmental efforts include the Air Force and Navy's unmanned combat air vehicle (UCAV), Navy's vertical takeoff and landing UAV (VTUAV), and the Broad Area Maritime Surveillance UAV (BAMS), as well as the Marine Corps's Dragon Eye and Dragon Warrior [3].

In 1987 the small UAV was introduced when Dr Paul MacCready's AeroVironment company developed the Pointer, the first hand-launched, backpack-carried UAV. That pointer was combined with an electric motor, propeller and a radio network. It had some limitations; in particular, it was hard to maintain situational awareness.

Nowadays, thousands of small backpack UAVs are in service. In the 1990s small UAVs, there was a major problem that it could not get stable image from their cameras. Many UAVs follow the pointer model with the same few key features as well and they are operated up to 10km from their launching point. The largest project regarding the small UAV programme in the US is the US Army's Small UAV (SUAV). During the recent Iraq war, many small UAVs have been used by the US Army for defense purposes [4]. These small UAVs allow the military many possible mission types including:

- Operation from the battle field by small units
- Surveillance
- Communication relay
- Intelligence gathering
- Identification of chemical or biological warfare

The use of small UAVs for civil purposes is rapidly increasing such as:

- Firefighting
- Police surveillance

- Communications
- Environmental studies (collecting air samples)
- Technology development for satellite constellations control

1.3 Future Work and Usages

In the future, small UAVs will become part of homeland (border) security. In lots of countries, like India, Pakistan, and the USA, their homeland security departments already have plans to deploy UAVs to watch coastal areas and protect major oil and gas pipelines. The use of small UAVs in the recent US wars in Iraq and Afghanistan has shown many advantages. These days small UAVs are tremendously flexible devices, which can be used for a variety of civil purposes. For example, they can be used to provide entertaining videotaped scenes to movie-makers, news reporters and the tourism industry; to search for and rescue people in perilous locations or circumstances (collapses, spills and fires); and to monitor or deliver mail to important installations in either highly sensitive locations (borders, ports and power-plants) or remote or uninhabitable places (polar zones, deserts and off-shore oil rigs) [5]. As the use of UAVs in these fields is increasing, the Individual Unmanned Air Scout (IUAS) aircraft will become a more desirable aerial platform because of its safety and reliability in comparison to other aerial vehicles.

Currently, researchers are trying to make useful UAVs for farmers where they can use them to check the dam levels and for counting stocks. In Australia, Queensland's De-

partment of Primary Industries (DPI) and agricultural organization Kondinin Group are researching by having aims to explore how UAV technologies can help cut labour expenses within the beef industry. Australia is already using remote management technology with UAVs to take high resolution images of crops to predict grain protein levels. This really saves a lot of time and money for the farmers because they know everything about their crops without traveling there to have a look.

1.4 Objective

Present communication among the UAVs and the ground station is based on the radio systems in the ultra high frequency (UHF) band and low earth orbiting satellite links. Both of sources of UAVs communications are long range but have low bandwidth (50 Kbits/sec). In this project conducted jointly between School of Computer and Information Science (CIS) and ITR, UniSA, we design a modem for the small UAVs by considering the distance 100m to 10 km among the UAVs and the ground station as well. Our main goal is to provide the low cost modem design for small UAVs which could transmit at flexible data rates by using the adaptive resources (i.e. modulation technique and coding) and we are expecting to get the flexible data rate up to a few mega bits per second from this design. The Orthogonal Frequency Division Multiplexing (OFDM) technique will be used to get the flexible data rate. OFDM is usually implemented with the Fast Fourier Transform (FFT).

Chapter 2

Modem Design for Small UAVs

2.1 Introduction

The modem design for small UAVs is a quite challenging task with trade off among different parameters like data rate, hardware complexity, latency, power consumption and cost. In this chapter we focus on the design of transmitter and receiver. Our hardware design architecture needs to satisfy the market for a low cost and flexible data rate network. This design aims to offer low power consumption, reliability for control applications with low to moderate data rates, e.g. 62 kbits/s - 744 kbits/s. Advanced signal processing techniques such as FIR filtering, FFT and LDPC coding/decoding are advised for use in baseband signal processing. These are very complex schemes which require billions of multiply and accumulate operations per second [6].

The Orthogonal Frequency Division Multiplexing (OFDM) technique is recommended to readily achieve variable data rates and to provide a multipath resistant solution.

With this approach, we can send parallel data on different subcarriers by maintaining the orthogonality condition between the carriers. Usually OFDM systems are implemented with the Fast Fourier Transforms (FFT) and Inverse Fast Fourier Transforms (IFFT). These transforms are used to map the data onto orthogonal subcarriers [7]. These OFDM based systems commonly deal with Quadrature Amplitude Modulation (QAM) or Quadrature Phase Shift Key (QPSK) mapped symbols on each sub - carrier as shown in Table 2.1 [8].

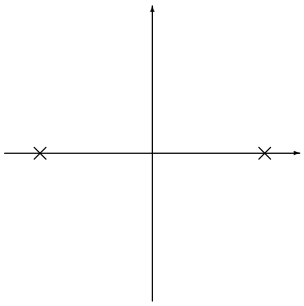
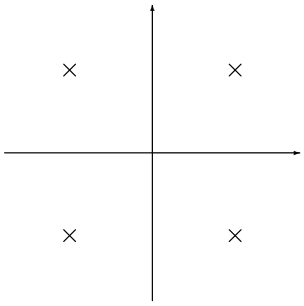
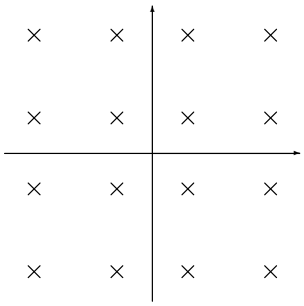
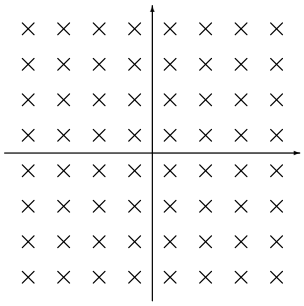
This chapter describes two main processing sections, one RF (analog signal) section where down and up conversion is done and secondly the digital domain section. In the digital domain section, Field Programmable Gate Arrays (FPGAs) are a likely option for low volume production and provide high flexibility in design reconfiguration of transmitter and receiver.

2.2 RF Section

2.2.1 Transmitter

Usually a simple transmitter consists of a power supply, an oscillator, a modulator, amplifier and antenna for transmitting radio frequency (RF). In this project we propose a Yagi antenna for the transmission of the radio frequency signal with 430 to 450 MHz Australian amateur band, because these antennas have high gain and are very directional. More specifically the Yagi antenna would only be used for ground stations while the monopole antenna would be used on the UAVs. Here we will de-

Table 2.1: Allowable modulation modes for the example adaptive modulation scheme.

Modulation scheme	Constellation	Mode index k	b_k (BPS)
None	No signal transmitted	0	0
BPSK		1	1
QPSK		2	2
16-QAM		3	4
64-QAM		4	6

scribe about the RF part of the transmitter, where the analog signal is up converted for the transmission with the required frequency, while the digital signal processing is done by the FPGAs, and will be discussed later.

Up Conversion

In the transmitter, for the up conversion the baseband digital signal is passed from the FPGA to the D/A converter for conversion to an analog signal using a sampling frequency of 4 MHz to obtain a signal at an intermediate frequency (IF) of 28 MHz. In the up converter section which consists of band pass filter (BPF), conventional analog quadrature mixer and oscillator, the continuous time signal (IF signal of 28 MHz) is multiplied with the oscillator (400 MHz) in the mixer and we get the signal back with the desired frequency (430 to 450 MHz) for the transmission. The unwanted signal is eliminated by using the band pass filter (bandpass sampling technique). This signal is amplified with the power amplifier (80 mW) and then it is transmitted. The complete up conversion process has been shown in the Figure 2.1 with all components.

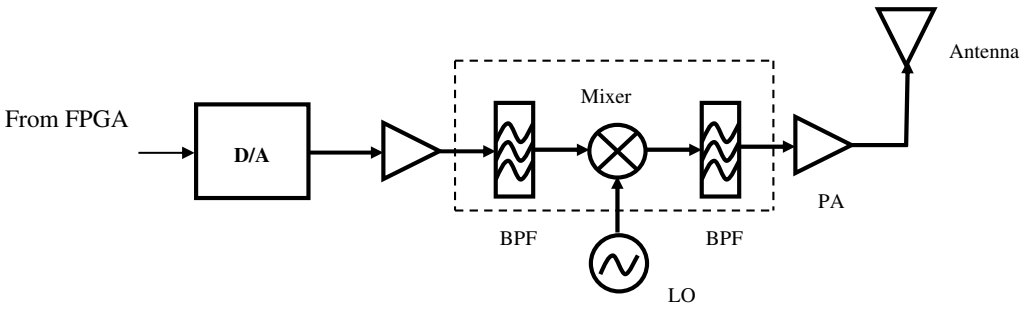


Figure 2.1: Up Conversion

2.2.2 Receiver

A simple receiver consists of demodulator, oscillator, quadrature mixer and antenna which is used to receive the signal from the transmitter. In the receiver we get the low IF signal which is then passed to the A/D converter and subsequently the signal processing section. There are a few reasons to use a low IF analog/digital interface [7]:

- It provides the greater flexibility in frequency translation.
- Cost saving.
- Avoids imperfections in quadrature mixers.

Down Conversion

The proposed down conversion architecture for the receiver is shown in Figure 2.2. The down conversion process consists of a Low Noise Amplifier (LNA), band pass filter, local oscillator and mixer. At the receiver end, the analog RF signal is received at the antenna and passed through the LNA, filtered by a band pass filter, and then multiplied by the local oscillator output (having frequency 400 MHz). After this down conversion, we get the IF signal and an unwanted frequency signal that is deleted by the band pass filter by using band pass sampling technique and translated to the baseband signal which is the requirement of the digital quadrature mixer. The IF signal is amplified with the power amplifier and then it enters into the analog to digital converter where this analog signal is converted into the digital signal by using

the sampling frequency (4 MHz). So after the conversion of analog signal to digital signal, it is forwarded into digital signal processing section (FPGA).

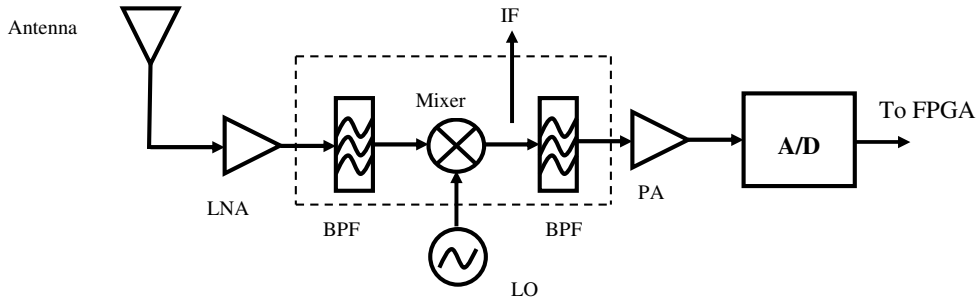


Figure 2.2: Down Conversion

Mathematical Description of IF Signal

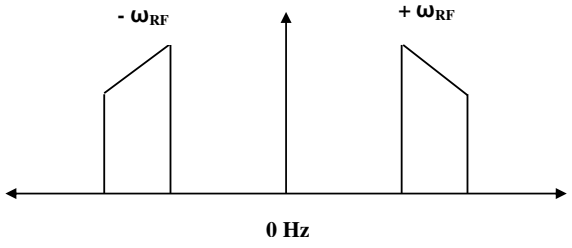
In the down conversion two signals are multiplied, one from the oscillator and other one which is coming from the antenna as shown in the equation (2.1).

$$V_{IF} = V_{LO}\cos(\bar{\omega}_{LO}t) \cdot V_{RF}\cos(\bar{\omega}_{RF}t) \tag{2.1}$$

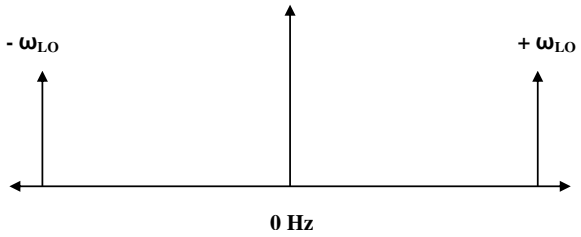
This equation gives two frequencies in terms of sum and difference as shown below

$$V_{IF} = \frac{V_{LO}.V_{RF}}{2}(\cos[(\bar{\omega}_{LO} - \bar{\omega}_{RF}) - \phi] + \cos[(\bar{\omega}_{LO} + \bar{\omega}_{RF}) + \phi]) \tag{2.2}$$

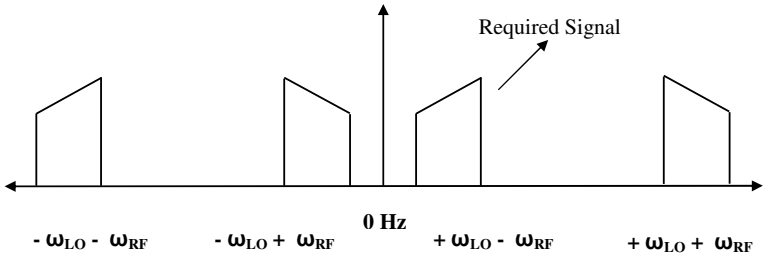
The bandpass filter (IF filter) will reject the unwanted frequency (RF + LO) and select the required difference frequency (LO - RF). Figure 2.3a shows the RF signal



2.3a. RF Signal



2.3b. Local Oscillator Signal



2.3c. Resultant Signal after the down conversion

Figure 2.3: Down Conversion

which is received by the receiver, Figure 2.3b shows the signal from local oscillator with high frequency and finally after multiplication of both these two signals, the required IF signal has been shown in Figure 2.3c.

2.3 Signal Processing Block

This section is very important because here we control the data rate by using the signal processing and modulation techniques. All signal processing will be done inside the FPGA. OFDM has been considered to transfer the data using different modulation

techniques e.g. Binary Phase Shift Key (BPSK), Quadrature Phase Shift Key (QPSK) and Quadrature Amplitude Modulation (QAM) technique etc. The Decimation In Time (DIT) and Decimation In Frequency (DIF) algorithms are used to calculate the Fast Fourier Transforms (FFT). Figure 2.4 shows the block diagram of the transmitter and receiver in the signal processing section.

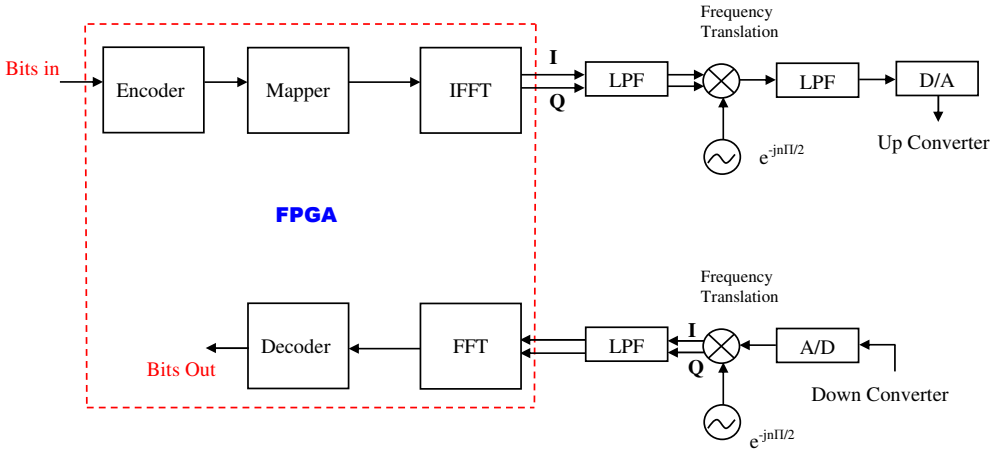


Figure 2.4: FPGA Based Transmitter and Receiver

2.3.1 OFDM Based Transmitter

Practically OFDM systems are implemented with the FFT and IFFT. In the transmitter, the quadrature mixer receives the signals in the form of multiplication of I and Q baseband signals with the same local oscillator (LO) and 90 degree phase shift is also provided on one path of the LO. Thus the signals which are separated by 90 degree are known as orthogonal to each other in the quadrature mixer. Thus we receive the composite low IF signal which is further processed by the digital to analog converter. The block diagram for the discrete modulator structure is shown in Figure 2.5. An OFDM based transmitter has been shown in the Figure 2.6 which consists

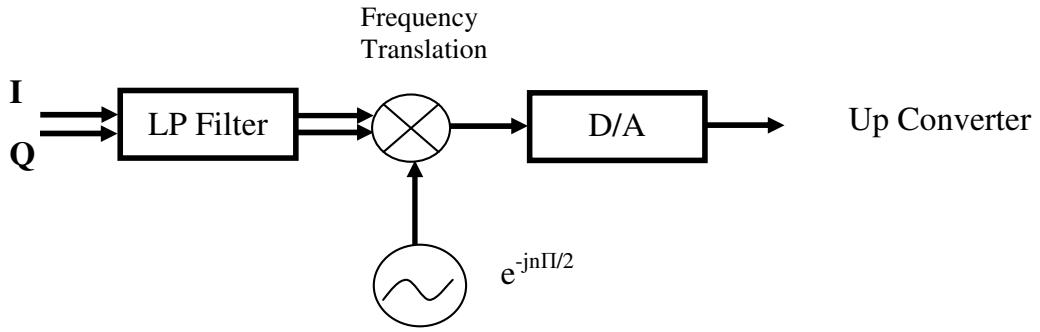


Figure 2.5: Modulator Structure

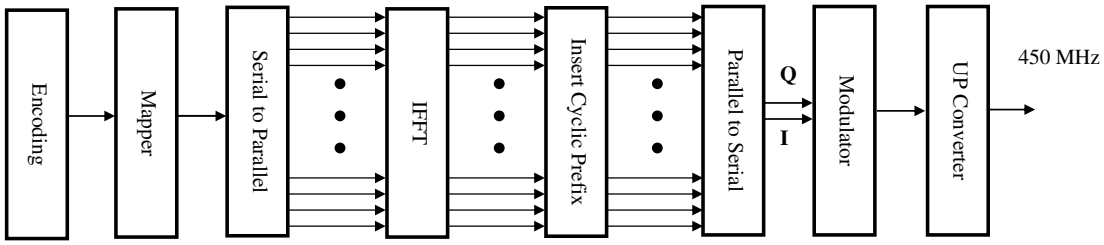


Figure 2.6: OFDM Based Transmitter

of four major parts, encoder, mapper, IFFT and cyclic prefix. The basic principle of OFDM is to divide the available spectrum into N orthogonal sub carriers. The data comes through encoder to the mapper block where it is mapped to signal constellations (e.g. QAM symbols). Next the IFFT is used to map these symbols on to orthogonal sub-carriers. Then in the Cyclic Prefix (CP) section, samples are added to the data to get rid of intersymbol interference (ISI) [9]. An OFDM frame is shown in Figure 2.7.

Data is modulated in the mapper section where the input data is converted into complex valued constellation points according to given constellation scheme as shown in the Table 2.1 [8], which shows the different possible constellation points for some examples of digital transmission techniques. We consider a variety of modulation and coding schemes for this project, which are shown along with data rates assuming a

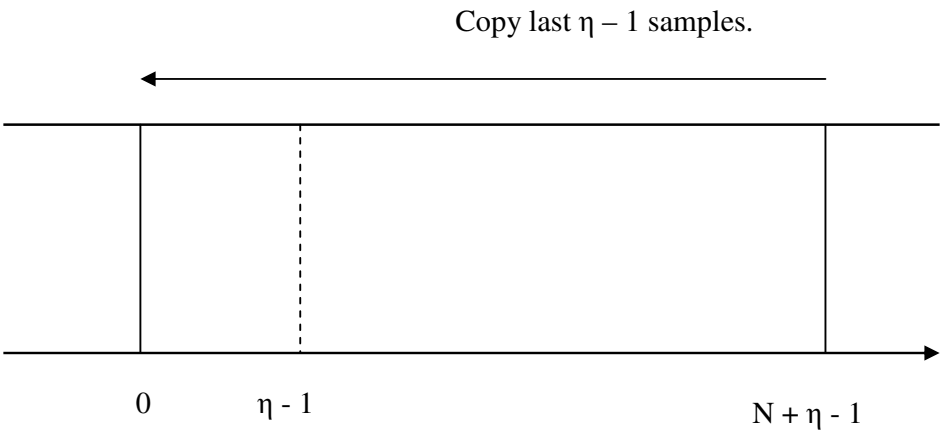


Figure 2.7: OFDM Frame

Table 2.2: Possible Modulation and Coding Modes for the Small UAV Modem.

Modulation	Code Rate, R	Bit Rate, R_b (kbits/sec)
BPSK	1/4	62.0606
BPSK	1/2	124.1212
BPSK	3/4	186.1818
QPSK	1/4	124.1212
QPSK	1/2	248.2424
QPSK	3/4	372.3636
8PSK	1/4	186.1818
8PSK	1/2	372.3636
8PSK	3/4	558.5455
16QAM	1/4	248.2424
16QAM	1/2	496.4848
16QAM	3/4	744.7273

symbol rate of 256 ksymbol/s in Table 2.2.

To combine all the different sub-carriers, a composite time domain signal is achieved by using the IFFT which converts the signal from frequency domain to time domain. The complete FFT structure and algorithm will be explained in Chapter 3. The

addition of CP depends upon the sub-carriers, i.e. for more sub-carriers used the less overhead introduced by the CP [10]. In the CP section, the data samples are copied from the IFFT and placed at the front of OFDM frame (Figure 2.7). The reason to do this is as follows. The convolution between the channel impulse response and data is circular and interference from the last symbol will affect only the first added samples in the beginning of the OFDM symbol. Therefore in the receiver this CP is discarded and circular convolution makes the equalization much easier with the receiver. The data rate decreases by the following factor due to the CP insertion:

$$R = \frac{N}{N + \eta}. \tag{2.3}$$

This shows that data rate will decrease if the number of samples in the CP increase, so therefore we have to really carefully choose the samples for the addition in CP. Typically, the CP length should be on the order of the channel impulse response length. Finally the data is converted back into serial form by using the parallel to serial converter and this discrete data is sent to modulator.

2.3.2 OFDM Based Receiver

In receiver the demodulator structure is the reverse of modulator which is shown in the Figure 2.8. The composite I/Q signal is mixed with the local oscillator at the carrier frequency in two paths, one is at the zero degree and the second one is at 90 degree phase shift. Thus this composite signal is broken into in phase I and quadrature Q components which are orthogonal to each other. We also decimate

using the low pass filter, the signal after the frequency translation, to obtain a single sample per symbol output signal.

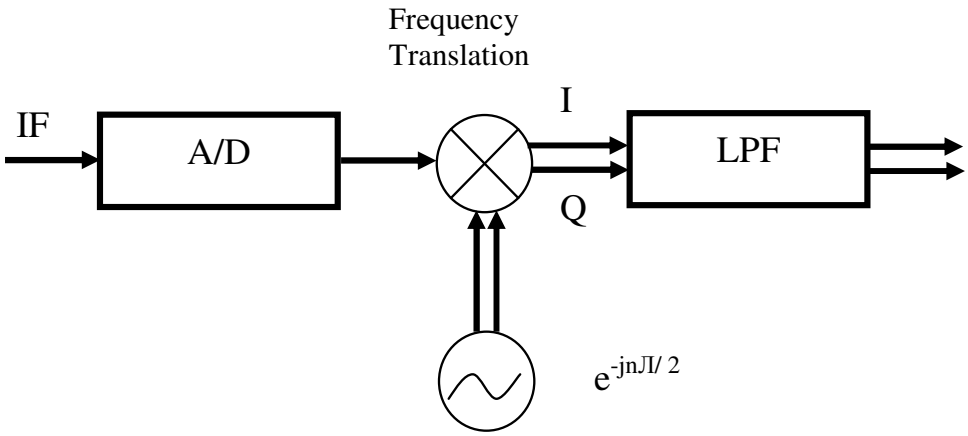


Figure 2.8: Demodulator Structure

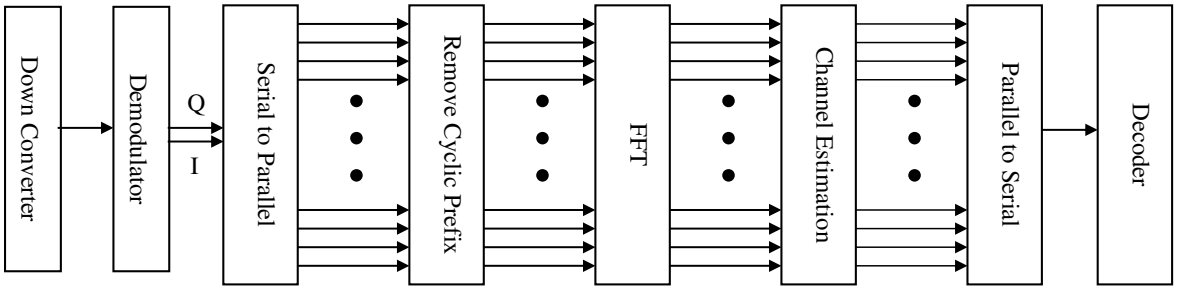


Figure 2.9: OFDM Based Receiver

The OFDM structure for the receiver is almost the reverse of the transmitter structure as shown in Figure 2.9. In the receiver the FFT block transfers the signal from the time domain to the frequency domain. In the absence of noise, and other imperfections, when the output of the FFT is plotted on the complex plane, we will observe the transmitted constellation.

Chapter 3

FPGA Implementation

3.1 Introduction

With the rapid progress in communication technologies, the DSP market is also growing. Once an algorithm has been chosen, the custom Application Specific Integrated Circuit (ASIC) chips can be implemented to get a low cost, size and power consumption system [11]. However, for prototyping and low volume production, FPGAs are typically used. In this chapter we focus on the design and processing of the FFT engine and also its implementation on different FPGAs (Cyclone I and II). The first section of the chapter introduces the theory and algorithm of FFT and in the second section, the implementation of butterfly operation and FFT has been described. The FFT is a very important part in multicarrier systems. The Cooley Tukey algorithm is used to calculate the Discrete Fourier Transforms (DFT). These DFTs are implemented on Field Programmable Gate Arrays (FPGAs). For real time processing

applications, dedicated FFTs are used to get fast computations, which increases the system's efficiency.

3.2 FFT and Butterfly Operation

The Fast Fourier Transform is an algorithm through which the Discrete Fourier Transform (DFT) is calculated with less number of computations as compare to the direct computations of DFTs. Table 3.1 shows the comparison of the number of complex multiplications and additions between the direct computation and FFT algorithm. In FFT complex multiplications and additions are considered. A Discrete Fourier Transform (DFT) for a sequence $x(0), x(1), x(2), ..., x(N-1)$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}, \tag{3.1}$$

where $k = 0,\ 1,\ 2,....,\ N-1,\ x(n)$ is the time domain sequence and $X(k)$ is the frequency domain representation of the sequence. If $W_N = e^{\frac{-j2\pi}{N}}$, which is also called twiddle factor, then equation (3.1) can also be written as

$$X(k) = \sum_{n=0}^{N-1} x(n) {W_N}^{nk}. \tag{3.2}$$

For the Inverse Discrete Fourier Transform (IDFT), we just need to invert the DFT as shown in equation (3.3). The IFFT is used to get the time domain signal from the

frequency domain signal.

$$x(n) = \frac{1}{N}\sum_{n=0}^{N-1} X(k)e^{\frac{j2\pi nk}{N}} \tag{3.3}$$

To calculate and implement the FFT, either the Decimation In Time (DIT) or

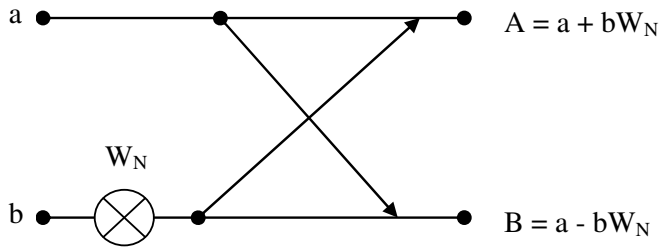


Figure 3.1: Butterfly.

Decimation In Frequency (DIF) algorithm is used. The DIT algorithm is referred to as the Cooley Tukey algorithm and the algorithm is DIF referred to as the Sande Tukey algorithm. A divide and conquer approach is used in these algorithms to make the system more efficient. The basic principle of this divide and conquer approach is N point DFTs computations are divided into two $N/2$ points and then from these $N/2$ points DFT two $N/4$ point DFT computations are obtained and it continues until we get the final transforms [12]. For more detail about these algorithms consider [13]. In this project, the DIT algorithm has been considered to implement the fast Fourier transforms for 64 points with radix-2. These kinds of DFTs are calculated by using radix - 2. There are three radix options ($r = 2, 4, and\, 16$) that are usually used to calculate the transforms. By increasing the radix we can decrease the computations in the processor, i.e. using high radix decomposition reduces the number of passes through the FFT processor and make them more efficient. In radix - 2 the total number of complex multiplications and additions are reduced to $\frac{N}{2}\log_2 N$ and $Nlog_2N$

respectively.

Table 3.1: Comparison of number of computations in direct DFT and FFT.

Number of Points N	Direct Additions $N(N-1)$	Direct Multiplications N^2	Radix-2 FFT additions $N \log_2 N$	Radix-2 FFT multiplications $\frac{N}{2} \log_2 N$
$4 = 2^2$	12	16	$4 \times \log_2 2^2 = 8$	$4/2 \times \log_2 2^2 = 4$
$8 = 2^3$	56	64	$8 \times \log_2 2^3 = 24$	$8/2 \times \log_2 2^3 = 12$
$16 = 2^4$	240	256	$16 \times \log_2 2^4 = 64$	$16/2 \times \log_2 2^4 = 32$
$32 = 2^5$	992	1024	$32 \times \log_2 2^5 = 160$	$32/2 \times \log_2 2^5 = 80$
$64 = 2^6$	4032	4,096	$64 \times \log_2 2^6 = 384$	$64/2 \times \log_2 2^6 = 192$
$128 = 2^7$	16,256	16,384	$128 \times \log_2 2^7 = 896$	$128/2 \times \log_2 2^7 = 448$

To calculate the FFT transforms for radix - 2, N must be that $N = 2m$ where m is number of decimations which can be calculated as $m = log2N$. Therefore for FFT with length of 64 points, 6 times decimations will be calculated. In other words there will be 6 stages of butterfly operations and each stage will process 32 butterflies operations for DIT radix- 2. The butterfly operation for the DIT - FFT is shown in Figure 3.1.

The DIT butterfly structure is different than DIF butterfly because in DIT, the twiddle factor is multiplied before the addition and subtraction is performed. In the DIT algorithm, the initial data is divided into even transform samples and odd transform samples and it continues until the initial transforms are reduced to set of

two point transforms of the initial data. The data is processed in bit reversed order at the input of butterfly. For example, if the binary address of the sequence of data is 110, then its bit reversed order would be 011. The block diagram and signal flow graph for an 8 point DIT FFT with radix-2 is shown in Figure 3.2, with a single butterfly illustrated in the dotted box.

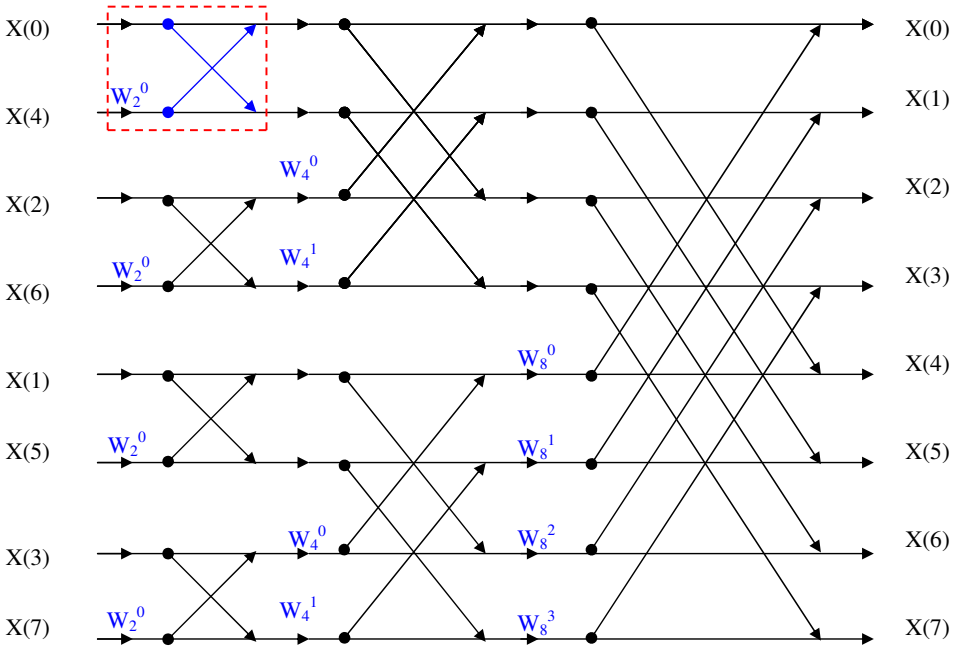


Figure 3.2: Signal Flow Graph for 8 Point FFT.

3.3 Butterfly Operation

In the butterfly operation, there are two inputs A and B (both A and B are complex numbers i.e. Ar, Ai, Br, Bi) and the outputs are C and D complex numbers (Cr, Ci, Dr, Di) as shown in Figure 3.3. The schematic of the butterfly consists of a complex multiplier, complex adder and complex subtractor. A library of parameterized modules (LPM) mega functions are used for multiplexers, additions and subtractions in

the AHDL model. The butterfly operation is processed in two stages. In the first stage, complex multiplication is performed between the pair of WN (Wi, Wr) coefficients and pair of inputs B (Br, Bi). In the second stage an addition is performed between the inputs A and the product of WN (Wr, Wi) and B (Br, Bi) input to get C (Cr, Ci) output. In the third stage the product of W_N and B is subtracted from the inputs A to get D (Dr, Di) output. In this model there is no checking of overflow or underflow.

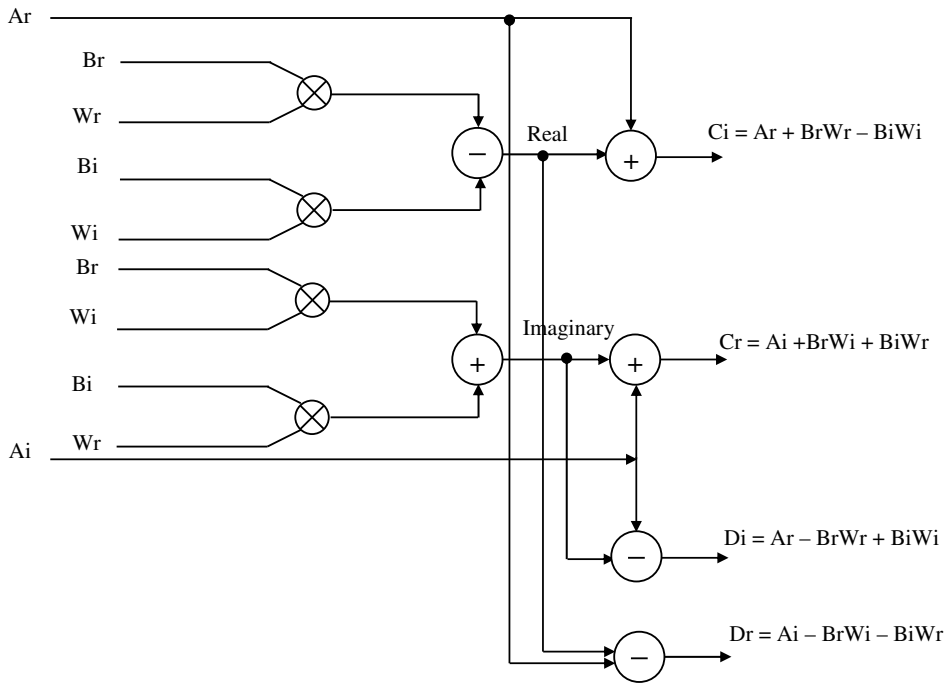


Figure 3.3: Schematic of Single Butterfly.

3.4 Single FFT Processor

To design an FFT we not only consider the speed by inserting pipelines but also we have to reduce the hardware resources as far as possible. Thus to increase the computational speed of DIT FFT in FPGAs, there are usually four different types

of architectures as shown in Figure 3.4. Figure 3.4a shows the single butterfly architecture which consists of only one random access memory (RAM). During this butterfly processing the outputs are stored back to the same memory locations used by butterfly inputs. Figure 3.4b shows the dual memory architecture in which the butterfly consists of two memories, with one used for the input and the other used to store the output. These architectures are used for low throughput with high clock [14]. Figure 3.4c shows the parallel architecture of the butterfly which increases the number of processing elements as well. Figure 3.4d shows the pipeline architecture for the butterfly process, which produces a non-stopping process on a clock frequency eual to the input data sampling rate. These two architectures are used to get high throughput of FFT. We focus on the dual memory structure as shown in Figure 3.5 with introducing some buffer elements as well. The butterfly processor is the combination of one complex multiplier, adder and subtractor. The hardware structure of the FFT processor in the FPGA has been shown in the Figure 3.5.

The FFT hardware structure consists of two RAMs, one for the inputs of the butterfly in which data is stored in the form of array x and y, while the second RAM is used to store the output from the butterfly operation. This makes the process fast; one Read Only Memory (ROM) is also used in which two constant arrays of Wr and Wi for look up tables are stored for use in the butterfly operation. The third important part is the controller through which it is confirmed that the right pair real and imaginary is going into the input of the butterfly and also from the ROM as well. Two buffers are also used at the front end of the butterfly, which read the inputs from RAM 1 and make sure that the right signal is processed in the butterfly operation. Two buffers

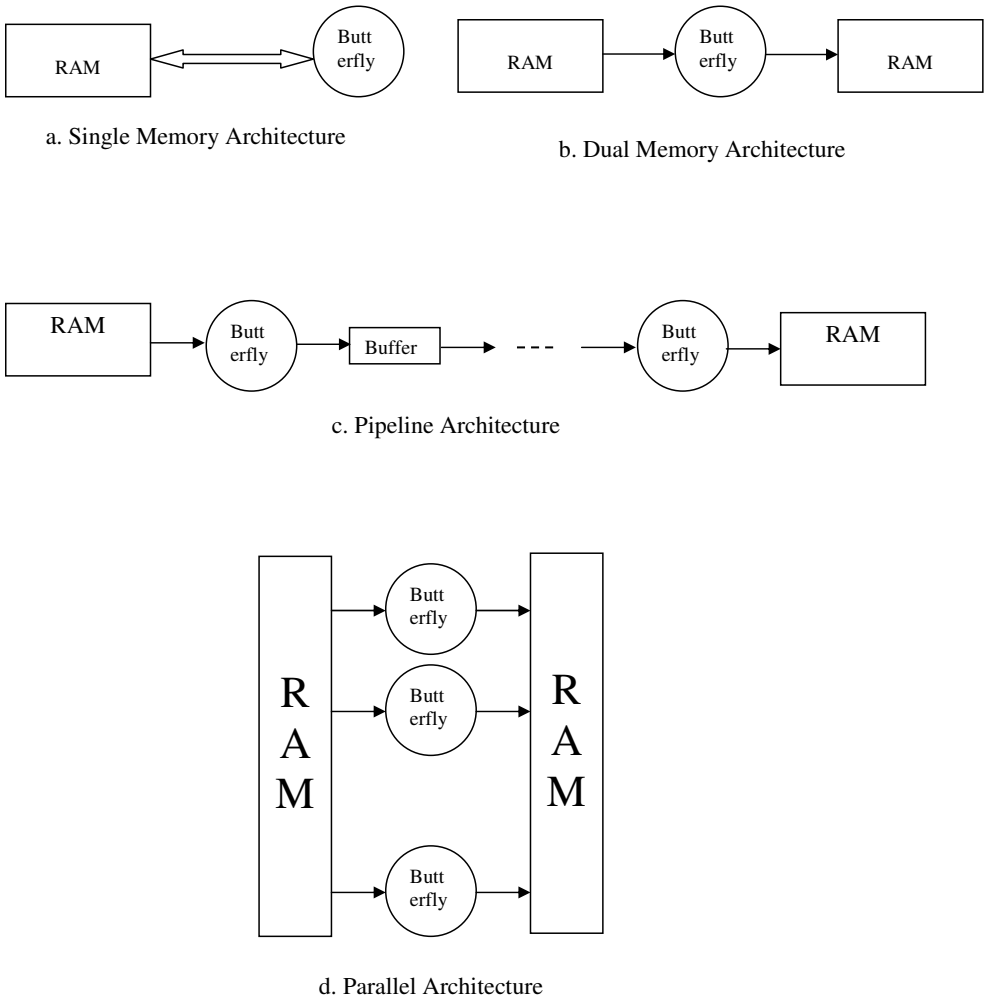


Figure 3.4: Different Butterfly Structures.

are also connected to store the output of the butterfly and to make sure that the output has been received after the butterfly operation and then it is written in the RAM 2. It is often advantageous to store the output from the butterfly into a buffer instead of using the single RAM circuit where the output from the butterfly is stored back to the same memory locations (in place algorithm). This makes the processing fast but it also requires more resources. For details consider [15]. By examining the FFT processor architechture in Figure 3.5, it can be observed that there are two main operations are involved. One is the butterfly operation and the second one is indexing (to access the right array of signals from the RAM). Our main focus remains

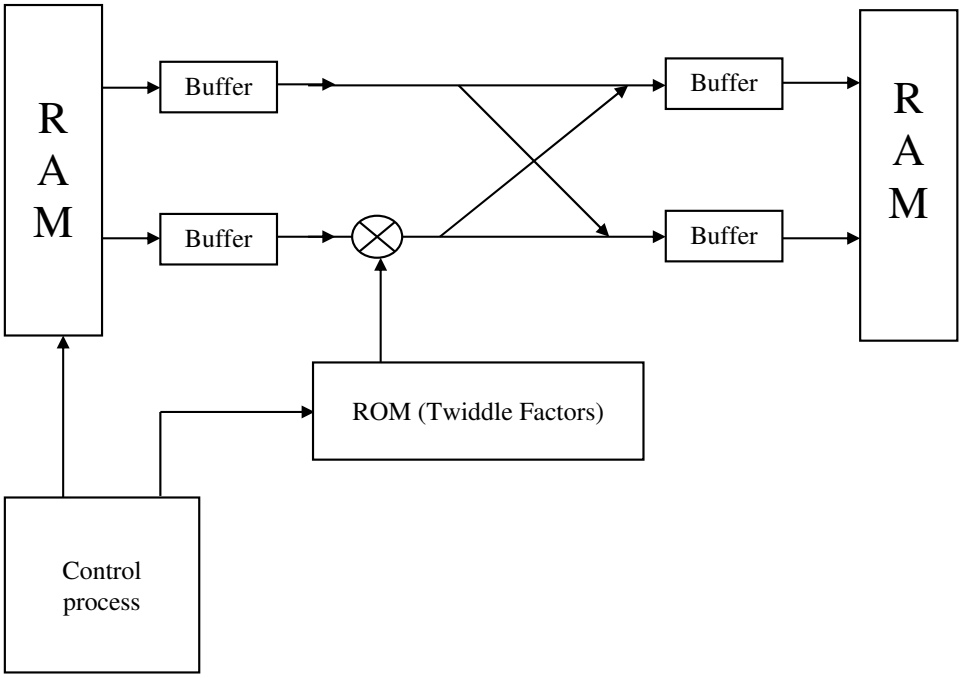


Figure 3.5: FFT Processor.

on the butterfly operation and the indexing operation in the memory. A sequence generator (state machine) has been considered to reduce the indexing complexity in the FFT structure. The function of each hardware component in the FFT structure is described below.

3.4.1 Butterfly Processor

The butterfly processor section performs the butterfly operation, with each 8 bit input data width. A complete butterfly operation requires one complex multiplier and a complex adder and a complex subtractor. This corresponds to four real multiplications, three real adders and three real subtractors. The four real multiplications are executed in parallel which enhances the processing speed [16]. The input size in the multipliers is 8×8 bits and the result is 16 bits while all the adders and subtractors

are operated with 16 bits input and 16 bits output. The final 16 bits result of the butterfly operation is truncated to the 8 most significant bits and is saved into the memory. A second option, which would provide higher precision for this operation is we can keep 16 bits results and then perform 16×16 operation in the rest of stages and the final result would contain 16 bits. In the butterfly operation when first cycle is finished then the first result bit of each real multiplication is ready, then the additions and subtractions are operated. Therefore, the butterfly operation is operated efficiently as all the operations are synchronous.

3.4.2 Dual Random Access Memory

In this double buffering FFT processor design, dual port RAM is used. One port is used to access the real 8 bits input and other one is used for imaginary 8 bits inputs. Each port is used as a unidirectional port to transmit the data in and data out for both real and imaginary parts. The output from the butterfly is stored in RAM2 and written back to RAM1 with the same memory locations. This is called the in place algorithm [17]. Read and write operations in RAM are controlled with the sequence generator or state machine.

3.4.3 Read Only Memory

All the twiddle factors W_N are stored in the ROM and output is specified by the address "addr" as an integer. Both the Wr and Wi numbers are read out at the same time. ROM is also used with 16 bit width, with 8 bit real and 8 bit imaginary

part. For the butterfly operation ROM with all coefficients is initialized with the sequence generator. For a 64 point FFT, 64 sets of coefficients are stored in ROM locations.

3.4.4 State Machine

All the hardware components of the FFT processor work concurrently. The controller is used to trigger the hardware. The control process is described with the help of a state machine because it gives lot of advantages regarding the debugging of the system. The state machine is also coded up in AHDL. To implement the state machine in AHDL requires the following [18]:

- Declaration.
- Control equations.
- State transitions by using case statements.

The transitions in the state diagram with the associated conditions have been shown in Figure 3.6. A transition in state machine occurs at the positive cycle of the clock and the actions always happen at the negative cycle of the clock [19]. "if statements" based on the clock conditions are used in the control process. When the clock = 1, the next state is determined by using the case statements. A change of state will never occur until the actions of current state are finished at the second half of the clock cycle. When the clock = 0, the current state action is activate and implemented in the case statement. At the end it changes the state to next state and renews the

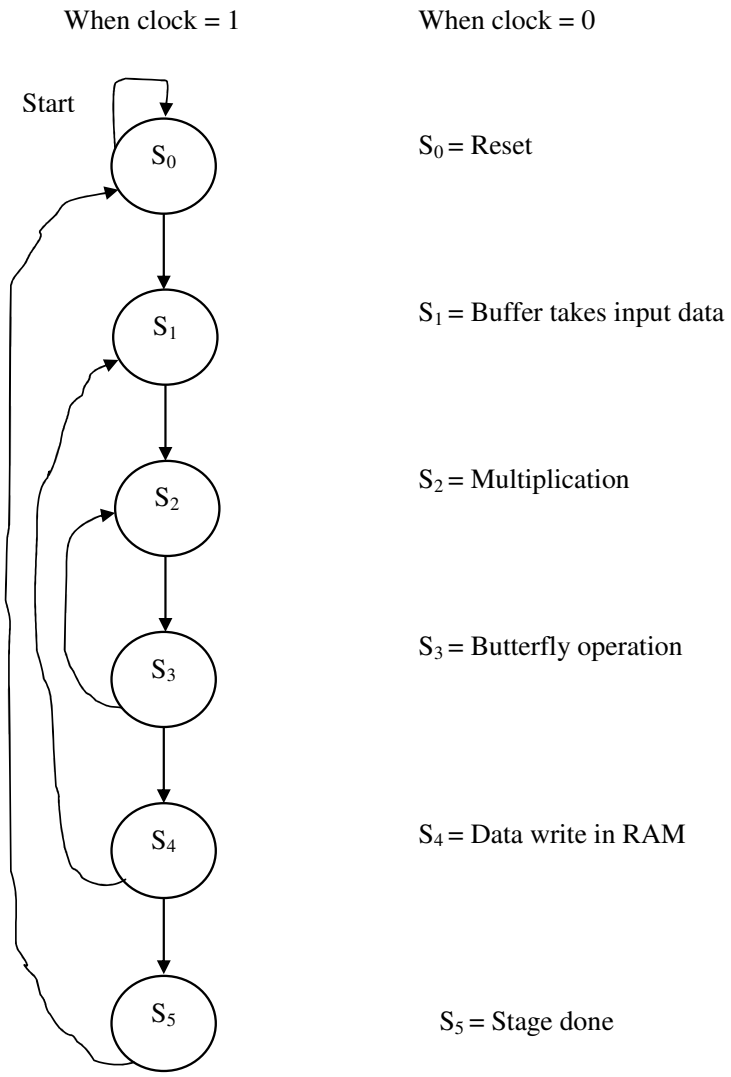


Figure 3.6: State Diagram of Control Processor.

cycle again.

3.5 Available Resources

Cyclone I

- 2,910 to 20,060 logic elements are available.

- 52 M4KRAM block ($128 \times 36bits$).
- 239,616 total RAM bits including M4KRAM.

Cyclone II

- 4,608 to 68,416 logic elements are available.
- 13 to 150 ($18 \times 18bits$) Multipliers.
- 119,808 to 1,152,000 RAM bits including M4K memory blocks.

3.6 Performance Evaluation on Different FPGAs

The butterfly architecture has been modeled in AHDL. To analyze the different FPGAs characteristics regarding the memory space, logic elements and other requirements, the butterfly model has been simulated with the Cyclone I and Cyclone II FPGAs at the clock rate of 100 MHz and the word length has been assumed as 8 bits. We get the following synthesis results:

Cyclone I

- Device : EP1C4F324C6
- Total logic elements: 526/4,000 (13%).
- Total pins : 178/249 (71%).

Cyclone II

- Device : EP2C20F484C6
- Total logic elements : 73/18,752 (<1%).
- Total registers : 71
- Total pins : 178/315 (57%).
- Embedded Multiplier 9 bit element : 4/52 (8%).

For $N = 64$, there are 6 stages and $192 = (64/2) \times log_2 64$ butterflies are computed. According to the simulation results one butterfly operation completes in two cycles, therefore $2 \times (\frac{64}{2}) \times log_2 64 + 6 = 390$ clock cycles are required for the 64 point FFT [20]. The total time taken by the FFT is 3.9 μs.

Chapter 4

Conclusion and Recommendations

In this master thesis report an OFDM based modem architecture for small UAVs has been proposed for achieving flexible data rates (62kbps to 744kbps) depending the on link quality, where the signal processing is done using the FPGAs. The 430 - 450 band has been considered for this small UAV project.

To meet with project requirements we recommend the use of low cost hardware components for up and down conversion which can be easily obtained from the market. We also recommend the OFDM technique because it facilitates the use of adaptive modulation and coding, and provides robustness in frequency selective fading environments. OFDM has been implemented with the FFT. We have proposed the symbol rate of 256 ksymbols/sec. We also proposed to adapt the data rate according to the link quality by using adaptive modulation and coding. To complete the modem

design we also need the decoder and encoder, which is the next task to be designed. Low Density Parity Check (LDPC) codes have been proposed for the coding.

In the third chapter of this report we described the FFT processor design and implemented the single butterfly operation on different FPGAs (Cyclone I and II) by using the Quartus II software. All the FFT design has been modeled in AHDL. We calculated the processing time for the FFT processor as 3.9μs from the presented design. From the results, it is noted that the Cyclone I (EP1C4F324C6) takes 526 logic elements which is the 13% of the total 4,000 logic elements while the Cyclone II (EP2C20F484C6) takes only 73 out of 18,752 which is the less than 1% of the total; it also use 71 registers and 4 embedded multipliers (9 bit) for a single butterfly operation. From these results, for the Cyclone II FPGA we note assume that only about 10% of the FPGA will be consumed by the FFT processor. We can assume that the whole modem design will consume less than 50% of Cyclone II resources. The project is ongoing and still many changes have to be done in the total FPGA design so that probably either of the Cyclone I and II will be useful for our modem design. It is noted that the use of a radix - 4 DIT algorithm to calculate the Fourier transforms could be advantageous since our FFT uses 64 points, which is a power of four. The use of radix - 4 can reduce the number of complex additions to $(3N/2)\log_2 N$ and the number of complex multiplications to $(3N/8)\log_2 N$.

Appendix A

Poster in AusCTW 2007

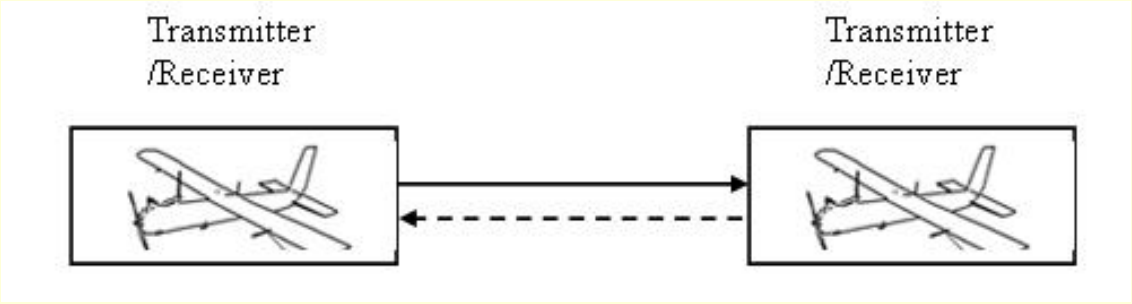
Regarding this Modem Design

We have presented our work as a poster in the Australian Communication Theory Workshop (AusCTW) 2007.

Modem Design for Small UAVs[1]

By: Shahid Mahmood
Supervisors: Prof. William G. Cowley
Dr. I. D. Holland

Institute for Telecommunications Research
University of South Australia



University of South Australia

[1]This work has been conducted under an occupational training scholarship at the Institute for Telecommunications Research (ITR), University of South Australia.

Outline

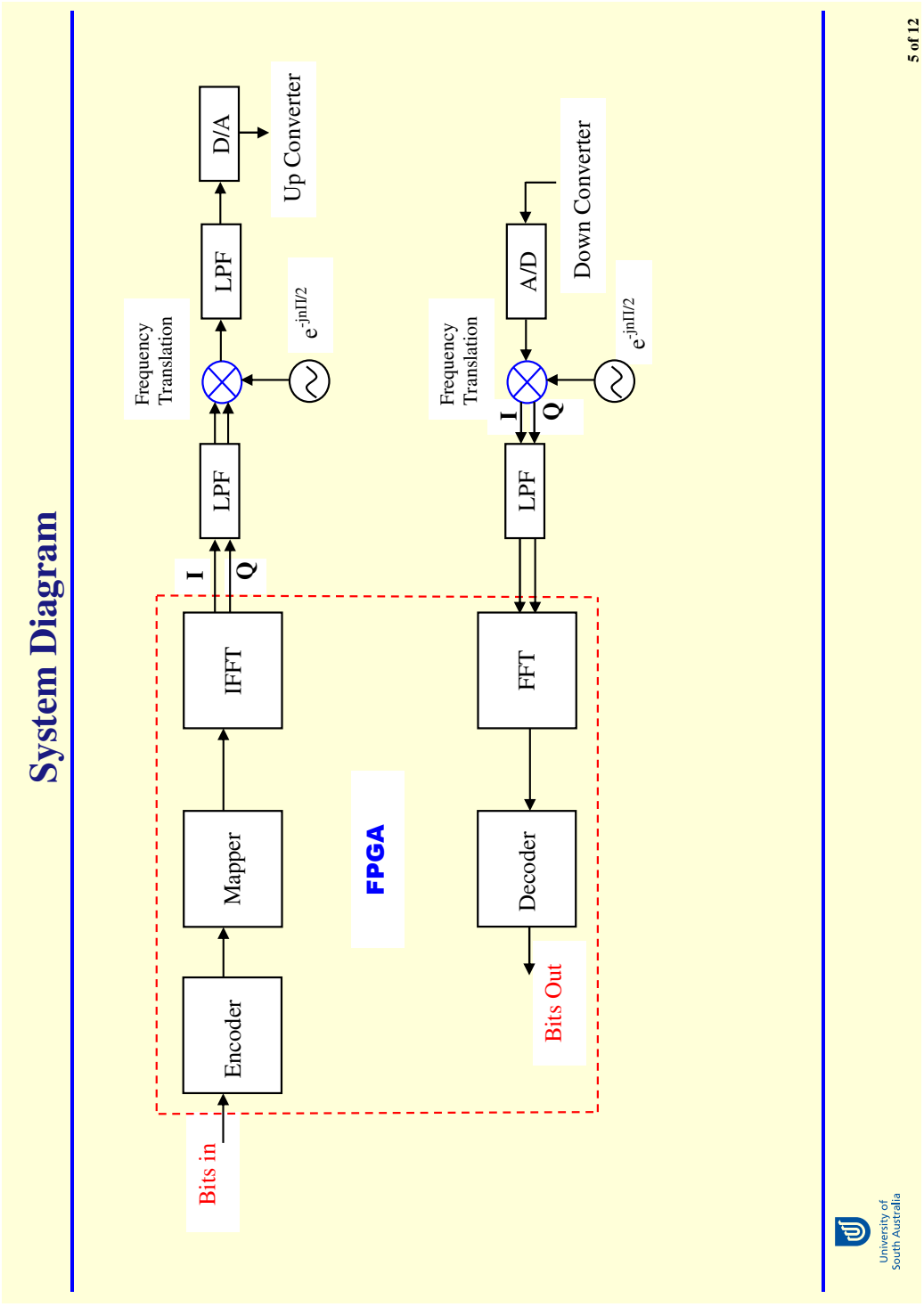
- Overview of the Small UAVs project.
- Objectives.
- Uses of small UAVs.
- Concept of adaptive modulation with OFDM.
- OFDM based transmitter and receiver model.
- Fast Fourier Transform (FFT).
- Available resources.
- Conclusions and future work.

Small UAVs Project Overview

- **Aim**
  - To build a complete modem inside a Field Programmable Gate Array (FPGA) using low-cost components for consumer and/or commercial applications by using the Australian amateur band (e.g. 430 - 450 MHz).
- **Participants**
  - Institute For Telecommunications Research (ITR), University of South Australia.
  - School of Computer and Information Science (CIS), University of South Australia.
  - Sponsored by Sir Ross and Keith Smith Foundation.

Uses of Small Unmanned Aerial Vehicles (UAVs)

- **Homeland Security.**
- **Traffic surveillance/monitoring systems which serves as Intelligence Transport Systems (ITS) [1].**
- **Forest fire surveillance.**
- **Hurricane monitoring.**
- **Weather measurements.**
- **In media industry, for example, for news reports, movie makers, etc.**


System Diagram
5 of 12
Bits in
Encoder
Mapper
IFFT
I
Q
LPF
Frequency Translation
$e^{-jn\Pi/2}$
LPF
D/A
Up Converter
FPGA
Frequency Translation
$e^{-jn\Pi/2}$
A/D
Down Converter
I
Q
LPF
FFT
Decoder
Bits Out
University of South Australia

Adaptive Modulation with OFDM

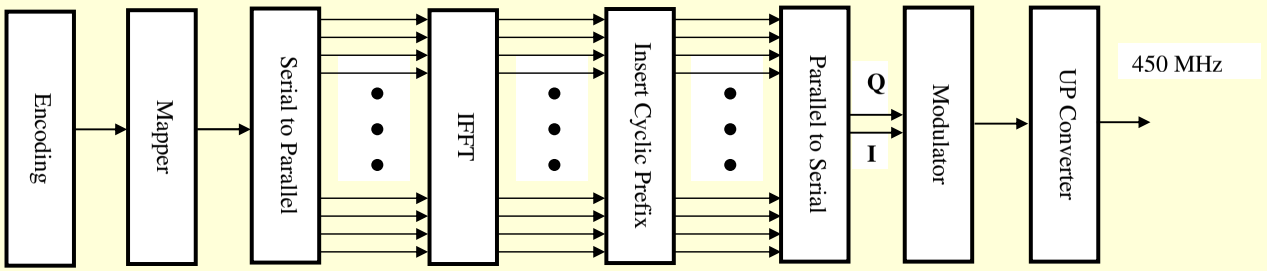
- Intersymbol interference (ISI) is removed with the sufficient cyclic prefix (CP) length.
- Provide flexible data rate depending on link quality.
- Possible data rates for BPSK and 16QAM modulation is 62.0606 and 744.7273 kbits/sec respectively, assuming the symbol rate of 256ksymbol/sec BPSK with LDPC coding with min. code rate 1/4 and 16QAM with LDPC coding with max rate is 3/4.
- Table of constellation points with different modulation techniques is shown below [3].

Modulation scheme	Constellation	Mode index k	b_k (BPS)
None	No signal transmitted	0	0
BPSK		1	1
QPSK		2	2
16-QAM		3	4
64-QAM		4	6

University of South Australia

OFDM Based Transmitter

- **Australian amateur band (430 to 450 MHz) may be used for transmissions.**
- **Bandpass sampling technique is considered for the up conversion of baseband signal (e.g. 28 MHz).**



University of South Australia

7 of 12

OFDM Based Receiver

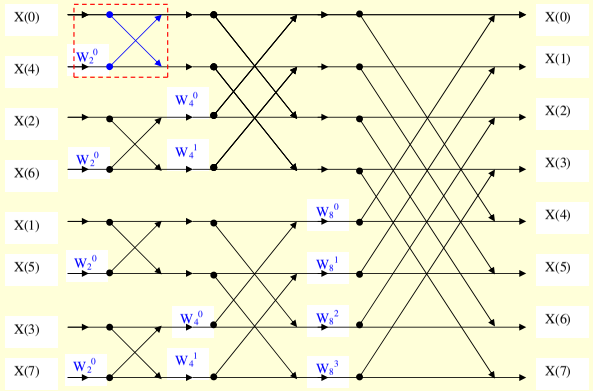
- Continuous time signal is down converted into Intermediate Frequency (IF) signal (e.g. 28 MHz).
- Band pass sampling technique is used.

8 of 12

University of South Australia

Architecture For Fast Fourier Transform(FFT)

- **Decimation in Time (DIT) with radix-2 algorithm is proposed to calculate the FFT.**
- **Pipeline model for butterfly operation has been considered.**
- **Single butterfly operation requires one complex multipliers and two complex adders and subtractors.**
- **Signal flow graph for 8 point FFT. (single butterfly is highlighted).**



University of South Australia

Available Resources

- **Cyclone I FPGA**
  - 2,910 to 20,060 Logic Elements.
  - 52 M4KRAM Blocks (128 $\times$ 36 bits).
  - 239,616 total RAM bits including M4KRAM [2].
- **Cyclone II FPGA**
  - 4,608 to 68,416 logic elements are available.
  - 13 to 150 18bit $\times$ 18bit Multipliers can be used.
  - 119,808 to 1,152,000 RAM bits including M4K memory blocks[2].

Conclusion and Preliminary simulation for single butterfly.

- **Preliminary Simulation Results for One Butterfly**
  - Timing simulations are based on a clock rate of 100 MHz.
  - One butterfly operation takes two clock cycles.
  - 526 logic elements are used by Cyclone I (EP1C4F324C6).
  - 73 logic elements and 4 multipliers (9bit $\times$ 9bit) are also used by Cyclone II (EP2C20F484C6).
- **Conclusions**
  - This is an ongoing project and many additional features would also be added in the FPGA total design.
  - OFDM with adaptive modulation and coding provides flexible data transmission rates to UAVs communication.
  - DIT with Radix-4 can reduce the number of passes and complex additions.

Thank You.

Questions?

References

[1] Raja Sengupta, Tim Macgee, Eric Frew. Vision-based road-following using a small autonomous aircraft., in Proc. *IEEE Aerospace Conference 2004*, March. 2004.

[2] Altera Product Catalog,October 2006.

[3] I. D. Holland, "Effective explicit and implicit link adaptation techniques for wireless communications", PhD. Thesis, Curtin University of Technology, Perth, Australia, Dec. 2004.

Appendix B

Specification of FFT

The performance of FFT has big impact on data width and twiddle factors because of memory resources and logic elements required by FPGAs are essentially linear to these values. So here are some general formulas to calculate the different parameters[20].

RAM (in bits) is given by: $2 \times datawidth \times points$ Twiddle factors in ROM (in bits) is given by: twiddlewidth $\times$ points/4 for Stratix FPGAs

twiddlewidth $\times$ points/4 for all other FPGAs

The performance of the core also dependent on:

- clock rate of the system.
- number of clock cycles required to compute the FFT.

Clock cycles are calculated by the number of stages and the number of clock cycles

per pass.

number of stages = ceiling $((\log_2(points))/2$ clock cycles per stage (for internal memory) =

points + ceiling$(\log_2(twiddlewidth))$+15

clock cycles per stage (for external memory) =

points + ceiling$(\log_2(twiddlewidth))$ + 19

The total number of clock cycles per transform is given by

total number of clock cycles per transform =

(number of passes× number of clock cycles per pass) + 6.

Bibliography

[1] P. T. Sleeper, *The 'Aerial Target' and 'Aerial Torpedo'in the USA*, Copyright 1999 CTIE, `http://www.ctie.monash.edu/hargrave/rpav_usa.html`, June 2003.

[2] G. Goebel, "Unmanned aerial vehicles", `http://www.vectorsite.net/twuav.htm`, January 07.

[3] "Unmanned aerial vehicles (uavs)", `http://en.wikipedia.org/wiki/UAV`, January 07.

[4] Cyber Aerospace, *UAV History*, Copyright 2004, Proxity Digital Network.Inc, December 06.

[5] B. Sweetman, *Mini UAVs - the next small thing.*, Copyrights 2006 Janes Information Group., October 2005.

[6] S. Naveen, *Reconfiqurable Modem Architecture*, January 2004.

[7] W. G. Cowley L. Berry, S. Bose, *Satellite Communications*, Number 0-7923-9333-3. Kluwer Academic Publishers, 1993.

[8] I.D. Holland, *Effective explicit and implicit link adaptation strategies for wireless communications.*, PhD thesis, Curtin University of Technology, Perth, Australia, Dec.2004.

[9] Anders Ollson Fredrik Kristensen, Peter Nilsson, "A generic transmitter for wireless ofdm system", *IEEE Proceedings on. Personal, Indoor and Mobile Radio Communications.*, vol. 3,pp.2234- 2238, 2003.

[10] Agilent, "Digital modulation in communications systems", *An Introduction*, 2006.

[11] J.H. Lee. K.L. Heo, S.M. Cho, "Application - specific dsp architecture for fast fourier transforms", *IEEE*, 2003.

[12] E.E. Fabris., "A bit serial fft processor", *UFRGS Federal University*, 2004.

[13] A.N. Kani, *Digital Signal Processing*, RBA, Chennai, India, 1998.

[14] H. Jiang and H. Luo., "Design of an efficient fft processor for ofdm systems", *IEEE Transactions on. Consumer Electronics*, vol. 51,pp.1099- 1103, 2005.

[15] L.R. Rabiner and B.Gold, *Theory and Applications for Digital Signal Processing*, Prentice Hall, 1975.

[16] K.K. Parhi., *VLSI Digital Signal Processing Systems.*, John Wiley and Sons, 1999.

[17] Orfanidis and Sophocles., *Introduction to Signal Processing.*, Prentice Hall, 1996.

[18] AUSPACE Limited, *Preliminary Design Documentation.*

[19] C.H. Lee., "Top down system design using vhdl.", *IEEE Proceedings on. ASIC Conference and Exhibit*, vol. 34,pp.256-265, 1993.

[20] Altera Cooperation, *FFT MegaCore Function User Guide.*