



UCGE Reports
Number 20278

Department of Geomatics Engineering

Development of a General Real-Time Multi-Channel IS-95
CDMA receiver for Mobile Position Location
(URL: <http://www.geomatics.ucalgary.ca/links/GradTheses.html>)

by

Nazila Salimi

December 2008



UNIVERSITY OF CALGARY

**Development of a General Real-Time Multi-Channel IS-95
CDMA receiver for Mobile Position Location**

by

Nazila Salimi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

December, 2008

© Nazila Salimi 2008

Abstract

Increasing demands for positioning has resulted in the emergence of numerous algorithms to cope with this field's various challenges (e.g. Multipath and NLOS phenomena). As part of a research group that focuses on positioning, it is of great interest to compare some of these algorithms and investigate their performance under different circumstances. Since each algorithm has different resource needs (processing power, RAM, etc.), it is necessary to provide a general and highly flexible platform for the signal processing unit. This thesis deals with the development of such a platform by employing a Field-Programmable Gate Array (FPGA) for a five-channel IS-95 CDMA signal. As such, the high computational tasks of the signal processing, Doppler removal and de-spreading modules, are implemented inside the FPGA. The final receiver is tested in static mode to evaluate its performance in acquisition and tracking of the signal while subjecting a negligible load on the PC. Due to the logistic limitations, the receiver cannot be tested in dynamic situations; however, it is shown that the receiver is able to track the signal code phase as long as the code phase decision making algorithm is executed within 10 seconds.

Acknowledgement

My special thanks to my parents who taught me how to live life to the fullest and pursue and accomplish my goals. If it was not for their encouragement, I could not achieve what I have now. I thank my brothers who have always been there when I require support and unconditional love. Thank my beloved boyfriend who supports me and helps me to overcome life's many difficulties.

Special thanks to my supervisor, Dr. Gérard Lachapelle, who instilled in me the strict discipline needed to succeed in my masters while remaining kind and understanding. I am also thankful to him for helping us in the PLAN group reach our full potential by providing us with excellent facilities in such a supportive environment.

Another special thanks to Dr. John Nielsen, who without his technical support this project could not be accomplished. He always had time for me, even when working from abroad, and for that I am truly grateful.

Thanks to my helpful friends Hendry Agus, Alfredo Lopez, and Dulini Siriwardena who helped me at different stages of this work. Special Thanks to Vahid Dehghanian, and Walid Abdel-Hamid whose keen insight and support made it possible for me to work effectively from abroad. I would also like to extend my gratitude to my PLAN group colleagues Tao Lin, Ali Broumandan, Surendran K. Shanmugam, and Ding Lu and wish them the best of luck in their studies.

Thanks to Cillian O'Driscoll, Dr. Mark Petovello and Rob Watson who kindly and patiently answered my questions.

Dedication

**To My Father who is always alive in my
heart and whose advice continues to
enlighten and inspire me each day.**

Table of Contents

Abstract.....	iii
Acknowledgement.....	iv
Table of Contents	vi
List of Figures.....	ix
List of Tables.....	xi
Abbreviations and Nomenclature.....	xii
CHAPTER 1: INTRODUCTION	1
1.1. Thesis Overview.....	1
1.2. Motivation.....	8
1.3. Thesis Objective and Contributions	10
1.4. Thesis Outline.....	11
CHAPTER 2: POSITIONING TECHNIQUES FOR IS-95 CDMA SYSTEMS	13
2.1. CDMA IS-95	13
2.2. Direct Sequence Spread Spectrum (DS-SS)	15
2.3. CDMA Forward Channel.....	17
2.4. Pilot Channel	21
2.4.1. Pilot Channel for Positioning	23
2.4.2. Pilot channel Demodulation	24
2.5. Radiolocation	26
2.5.1. Received Signal Strength (RSS) Method.....	27
2.5.2. Angle of Arrival (AOA) Method.....	28
2.5.3. Time of Arrival (TOA) Method	30
2.5.4. Time Difference of Arrival (TDOA) Method.....	31
2.6. Sources of Location Errors	32
2.7. Estimation Algorithms	35
2.8. Summary.....	37
CHAPTER THREE: SIGNAL PROCESSING IN CDMA PILOT CHANNEL RECEIVER	40
3.1. Signal Digitization.....	41
3.2. Acquisition.....	44
3.2.1. Doppler Removal	46

3.2.2. De-spreading Module.....	48
3.3. Tracking Loops.....	50
3.3.1. Frequency Tracking Loop.....	51
3.3.1.1. Numerically Controlled Oscillator.....	52
3.3.2. Code Tracking Loop.....	54
3.3.2.1. Early-Late gate DLL.....	56
3.3.2.2. Performance of the DLL.....	58
3.3.2.3. Non-Coherent DLL.....	59
3.3.2.4. General Coherent DLL.....	59
3.3.2.5. Tracking Loop Using ACF.....	62
3.4. Summary.....	63
CHAPTER 4: RECEIVER STRUCTURE.....	65
4.1. RF board.....	66
4.2. Signal Processing.....	69
4.2.1. Phase-I Receiver.....	69
4.2.2. Gage card-based receiver.....	71
4.2.3. Phase-II Receiver.....	72
4.2.3.1. ML310 Evaluation Development Board.....	73
4.2.3.2. Virtex-II PRO XC2VP30.....	74
4.2.3.3. New-version of Digital board.....	76
4.2.3.4. New Version of Firmware.....	77
4.2.4. Synchronization.....	78
4.2.5. Receiver Signal Processing Approach.....	81
4.3. Summary.....	86
CHAPTER 5: RECEIVER SIGNAL PROCESSING IMPLEMENTATION IN FPGA.....	87
5.1. Implementation Challenges.....	87
5.2. Overall Design.....	90
5.3. Acquisition.....	95
5.3.1. Doppler Removal Module.....	98
5.3.2. De-spreading Module.....	104
5.3.3. Synchronization Module.....	109
5.4. Tracking.....	110
5.4.1. Code tracking procedure.....	110
5.4.2. Frequency tracking procedure.....	112
5.5. Code and Frequency Tracking Evaluation.....	117
5.5.1. Code tracking test set up.....	117
5.5.2. Frequency tracking test set up.....	119
CHAPTER 6: CONCLUSIONS AND FUTURE WORK.....	122

6.1. Conclusions.....	122
6.2. Suggestions for Future Work.....	124
REFERENCE	127

List of Figures

Figure 1: MS receiver contains three main units for positioning.....	4
Figure 2: Comparison of different design choice (Parnell & Bryner 2004).....	5
Figure 3: Spread Spectrum Modulation in Time and Frequency domain	15
Figure 4: IS-95 CDMA forward channel (from Korowajczuk et al 2004).....	19
Figure 5: IS-95 CDMA forward channel structure	21
Figure 6: Pilot Channel Generation circuit.....	22
Figure 7: Linear Feedback Shift Register for generating Short PN code.....	23
Figure 8: Correlator demodulator – sliding correlate.....	24
Figure 9: Correlator demodulator – parallel correlator.....	24
Figure 10: Bank of Correlators.....	25
Figure 11: Correlator and its Matched filter equivalent circuit.....	26
Figure 12: Angle of Arrival method	28
Figure 13: Time of Arrival Method	30
Figure 14: hyperbolic system for TDOA method	32
Figure 15: Radiolocation in the presence of measurement errors.....	33
Figure 16: NLOS signal passes longer path than the LOS signal.....	34
Figure 17: LSE and MLE search space	37
Figure 18: Phase Ambiguity resulted from commensurate sampling rate (Maurizio et al 2004)	43
Figure 19: Two-dimensional search Acquisition procedure.....	45
Figure 20: Power attenuation due to Frequency mismatch (Watson 2005)	46
Figure 21: De-spreading circuit in frequency domain	50
Figure 22: NCO block diagram (Xilinx 2004)	52
Figure 23: Sine wave magnitude and phase (Analog Device 1996).....	53
Figure 24: General tracking loop structure (Meyr et al 1998)	54
Figure 25: Block diagram of optimum Delay Lock Loop.....	55
Figure 26: The Early-Late gate delay lock loop.....	57
Figure 27: DLL discriminator characteristic for four different Early-Late spacing	58
Figure 28: Non-coherent Early-Late gate delay lock loop	59
Figure 29: Generalized Coherent Delay Lock Loop (Wilde 1998).....	61
Figure 30: S-curve for GCDLL, $N_k=2$, $k_1 = k_{-1}=1$	62
Figure 31: Using ACF to model the S-curve.....	63
Figure 32: Overall Block Diagram of the Multi-Channel receiver.....	66
Figure 33: RF unit of the PLAN group receiver (Lopez 2006).....	67
Figure 34: Mixer Operation.....	68
Figure 35: RF board	68
Figure 36: PLAN receiver Digital board (Lopez 2006).....	70
Figure 37: Digital Board block diagram for one channel.....	70
Figure 38: Octopus CompuScope CS8280.....	72
Figure 39: Xilinx ML310 Evaluation development board	74

Figure 40: Virtex-II PRO Generic Architecture Overview (Xilinx 2004)	75
Figure 41: Block diagram of the new Digital board	77
Figure 42: Timing Synchronization Block diagram	79
Figure 43: Doppler frequency variation in static mode	80
Figure 44: Doppler frequency variation in static mode	81
Figure 45: Signal Processing Strategy categories	83
Figure 46: Signal processing flow chart of the PLAN receiver, PC-based tasks (dark boxes), FPGA-based tasks (light boxes)	85
Figure 47: Assembled system.....	92
Figure 48: To overcome metastability between two clock domain (a) Flip-Flop (b) MUX	94
Figure 49: System configuration related to code tracking (Red) and Frequency tracking (Blue)	95
Figure 50: Acquisition Result for 1 epoch (26.7 ms).....	97
Figure 51: Acquisition Result for 4 epochs (106.7 ms).....	97
Figure 52: Abrupt Doppler frequency changes	98
Figure 53: A sample of NCO output which shows a 10 MHz sine signal.....	100
Figure 54: Acquisition result for the strongest BS in the frequency range [-1800 1800] Hz	101
Figure 55: Correlation power of strongest BS for Doppler frequency range [-1800 1800] Hz	101
Figure 56: RF LO output (Lopez 2006).....	102
Figure 57: IF LO output (Lopez 2006).....	102
Figure 58: Acquisition result after replacing the LOs with signal generators.....	103
Figure 59: Result of the frequency offset after using the signal generators	104
Figure 60: Overall Correlator circuit	107
Figure 61: PN Generation for sequential Coefficient Offsets	108
Figure 62: Implemented Correlation method for two consecutive snap shots	108
Figure 63: Correlator peak drift in the lack of GPS synchronization (blue) and after TCXO drift compensation (red).....	110
Figure 64: Correlator output snap shot for Five BSs	111
Figure 65: Block Diagram of the Frequency Error Measurement related to TCXO frequency drift	112
Figure 66: Two Clock with timing error δ	113
Figure 67: Final TCXO frequency error compensator.....	117
Figure 68: The result of code tracking	118
Figure 69: Test set up using the Gage card post-processing algorithm for comparison.....	120
Figure 70: Result of frequency tracking	120

List of Tables

Table 1: XC2VP30 resources	76
Table 2: Summary of Sequential and Batch Processing characteristics	84
Table 3: Computational load to process 1 epoch of data in a CDMA IS-95 receiver for one BS	88
Table 4: Computational load of processing 1 epoch of data, $F_s= 2.4576$ MHz	89
Table 5: RAM size for 1 sec data and total amount of data in a 2-GB RAM, $F_s= 2.4576$ MHz	90
Table 6: Mean and Standard Deviation of the Gage card-based and FPGA-based receivers with and without TCXO frequency drift tracking.....	121
Table 7: Virtex-II PRO device (2vp30ff896-6) utilization, Synthesis Result.....	124

Abbreviations and Nomenclature

SYMBOL	DEFINITION
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
AJM	Anti-Jam
AOA	Angle Of Arrival
AWGN	Additive White Gaussian Noise
BB	Base Band
BS	Base Station
CDMA	Channel Division Multiple-Access
CIR	Channel Impulse Response
CLB	Configuration Logic Block
DCM	Digital Clock Manager
DLL	Delay Locked Loop
DS-SS	Direct Sequence Spread Spectrum
FF	Flip Flop
FFT	Fast Fourier Transform
FH-SS	Frequency Hopped Spread Spectrum
FLL	Frequency Lock Loop
FPGA	Field Programming Gate Array
IF	Intermediate Frequency
GNSS	Global Navigation Satellite Services

GPS	Global Positioning System
LLT	Logic Level Translator
LO	Local Oscillator
LOS	Line Of Sight
LPI	Low Probability of Intercept
LSE	Least Square Estimation
LUT	Look-Up Table
MLE	Maximum Likelihood Estimation
MS	Mobile Station
MTLL	Mean Time to Lose Lock
MUSIC	MULTiple Signal Classification
NCO	Numerically Controlled Oscillator
NI	National Instrument
NLOS	Non Line Of Sight
PC	Personal Computer
PDF	Probability Density Function
PLAN	Position, Location And Navigation
PLD	Programmable Logic Device
PLL	Phase Lock Loop
PN	Pseudo-Noise
PRN	Pseudo Random Noise
RAM	Random Access Memory

RF	Radio Frequency
RSS	Received Signal Strength
SDR	Software Digital Radio
SNR	Signal-to-Noise Ratio
TCXO	Temperature Controlled Crystal Oscillator
TDOA	Time Difference Of Arrival
TOA	Time Of Arrival
USB	Universal Serial Bus
UTC	Universal Time Coordinated
VGA	Variable Gain Amplifier

CHAPTER 1: INTRODUCTION

1.1. Thesis Overview

During the last decade, advances in wireless technology have made the practical implementation of various applications possible such as wireless mice and keyboards, satellite televisions, home and office security systems and cell phones. One of these developments is wireless location of mobile stations. Different services can take advantage of wireless location such as roadside assistance, fleet management, asset tracking, intelligent transportation system, and network resource management. Moreover, due to its high demand, the Emergency-911 (E-911), a safety service for cell phones in the United States, has been made compulsory by the U.S. Federal Communication Committee (FCC).

The main objective of a location system is to estimate the location of a Mobile Station (MS) based on the information collected about its position (Caffery & Stüber 1998). Different methods have been proposed for the positioning objective such as radiolocation, dead reckoning, and proximity systems (Caffery 2002). Among these methods this thesis

focuses on the radiolocation method that is widely used and has good position accuracy. Currently, two main positioning systems are using radiolocation. One type is the Global Navigation Satellite System (GNSS), a generic name to designate GPS, Galileo and other satellite-based systems. In these systems, only the downlink transmission is used for the positioning to provide unlimited capacity. Another type is the ground-based Cellular Base Station system. This system can also be divided into two categories: Network-based and MS-based. In the first category, which is also called the BS-based (Base Station) system, the MS transmitted signal is processed within three or more BSs. These BSs forward the information to a central location, where the position of the MS is calculated. In the MS-based systems, the MS calculates its position by using received signals from three or more BSs (Messier & Nielsen 1999).

In either system, the position is found using radiolocation which measures one parameter of the radio signal that travels between a set of transmitters and the MS. Depending on the estimated parameters, different methods have been suggested¹:

- Received Signal Strength (RSS) is based on the premise that the power loss of the received signal is inversely proportional to the distance squared that it traverses from the transmitter to the receiver.
- AOA (Angle Of Arrival) uses an antenna array to find out the direction of the arrived signal.

¹ Detail information on these methods is provided in Chapter 2. More information can also be found in (Lu 2007) and (Moghaddam 2007).

-
- TOA (Time Of Arrival) calculates the distance by measuring the time of arrival of the received signal and converting the time delay into a range.
 - TDOA (Time Difference Of Arrival) is similar to TOA, but uses a differential approach to remove the common errors (e.g. receiver clock bias) between different paths. Additionally, it is not necessary to synchronize the MS with the BS using TDOA.
 - A combination method such as AOA/TOA or AOA/TDOA, when more than one parameter estimation is supported by the MS receiver.

In general, a MS receiver is composed of three main units: The RF unit, the Signal Processing unit and the Position Solution unit. The RF unit is responsible for altering the unprocessed signal into a format that can be used by the signal processing unit. The desired information (e.g. TOA, AOA and TDOA) from this signal is then extracted by the Signal Processing unit. Lastly, the information gleaned from the signal is used by a position algorithm in the Position Solution unit to estimate the location (or/and velocity) of the receiver. Figure 1 illustrates the connectivity of these units inside a MS receiver.

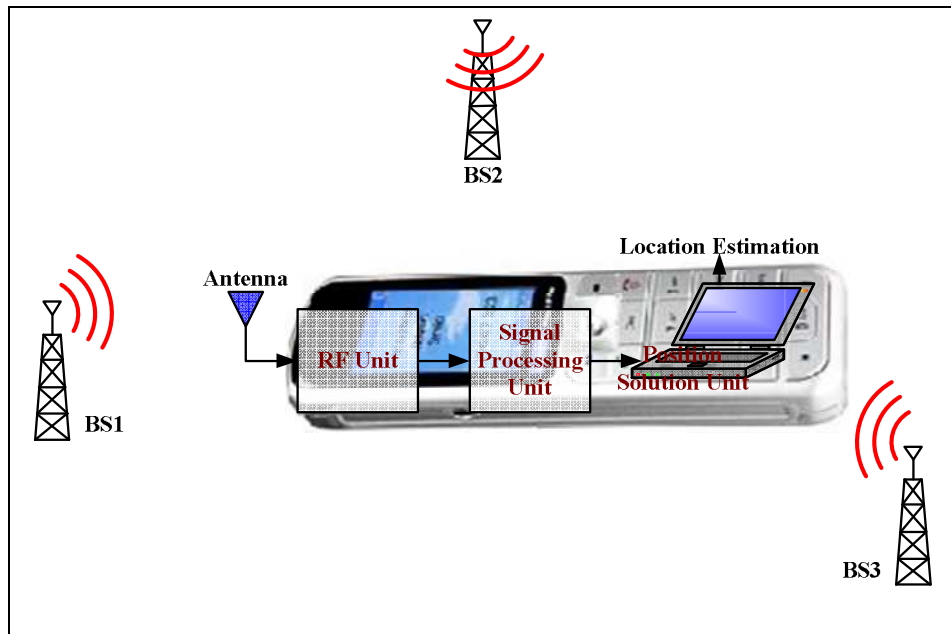


Figure 1: MS receiver contains three main units for positioning

The final position estimation accuracy and the features of the entire system depend on the functionality of the receiver's units. Among these units, the development basis for the RF and the position solution is clear. The RF unit has to be implemented in hardware, regardless of employing any of the alternatives such as super-heterodyne or homodyne approaches. Similarly, apart from the utilization of any of the proposed methods, the Position Solution unit has to be implemented in software.

In contrast, the signal processing unit has the capability to operate on different platforms (e.g. hardware, firmware¹, or software) that allows for different components of the same design to be implemented in a flexible manner. This feature enables designers to develop the signal processing unit based on their specific objectives. However, it should be noted

¹ In this thesis, the firmware is referred to the programs used for the FPGA and software is referred to the programs used for the PC, microprocessor.

that the final system performance can be heavily influenced by the signal processing partitioning; thus, this stage should be handled with care.

As stated previously, a variety of options can be considered for the signal processing design in a receiver, namely from Application Specific Integrated Circuit (ASIC) chips with high performance and low power consumption features, to pure software-based designs with low speed and high flexibility characteristics. Between these two extremes exist thousands of different Programmable Logic Devices (PLDs) and Field Programmable Gate Arrays (FPGAs). Since the selection of the signal processing design is a critical step, as it can have a substantial influence over the system performance, it is worthwhile to investigate the most important features of these different technologies, which are listed in Figure 2.

Technology	Performance /Cost	Time Ready for Market	Time to High Performance	Time to change functionality
ASIC	Very High	Very Long	Very Long	Impossible
FPGA	Low-Medium	Short	Short	Short
MicroProcessor	Low-Medium	Short	Not Attainable	Short

Figure 2: Comparison of different design choice (Parnell & Bryner 2004)

To achieve the highest performance, while consuming the lowest power¹, ASICs are the best choice. However, their long term fabrication process makes them less suited for certain applications. Specifically, for small markets and research projects where the system volume is low, the project budget is limited and the system needs to be made available within a short period of time. Moreover, the fixed functionality of ASICs, which cannot be

¹ Low power consumption is a key mobile equipment feature as it can extend the lifetime of the battery and reduce both the equipment and battery size.

modified, makes them less desirable for research purposes. Presently, the availability of low cost, high density FPGAs provides an attractive alternative to ASICs.

Consequently, FPGAs and microprocessors (i.e. general purpose microprocessors) are two practical options in most signal processing designs. They have many similar features as shown in Figure 2. The major advantage of the microprocessor over the FPGA is its capability to implement high complexity algorithms. However, for real-time applications, FPGAs' high level of parallelism and integrity result in higher performance than that of microprocessors.

In a CDMA receiver, Doppler removal and de-spreading modules are the most computationally demanding tasks. The high number of mathematical operations¹ makes it impossible for a microprocessor to perform these tasks in real-time. To reduce the microprocessor load, two different approaches have been proposed. One approach is to use more efficient algorithms and methods for the required computations, such as Single-Instruction Multiple-Data (SIMD) in specific MMX for x-86 processors (Charkhandeh et al 2006). Alternatively, a second approach to reduce the microprocessor load is to decrease the number of required operations (Petovello & Lachapelle 2006). Either way, the microprocessor is still the bottleneck for the system performance since its load remains high (e.g. the microprocessor has to complete other tasks such as multipath mitigation, position estimations, user applications etc).

¹ multiplications and additions

To overcome this problem, one can partition the signal processing unit and consider the implementation of high computational tasks on a FPGA. Hill (2004) investigates the implementation of three major components in such a receiver, namely Numerical Control Oscillator (NCO), Low Pass Filter (LPF), and Phase Lock Loop (PLL), and uses a Xilinx SPARTAN-II for this purpose.

Lück et al (2005) compare the baseband processing of a software receiver and FPGA-based receiver for an ideal case (no Multipath). They show that under this circumstance both receivers have almost identical results. However, unlike a FPGA-based receiver, the software receiver cannot completely fulfill the Nyquist rate criteria because of its insufficient processing power. Therefore, there is a reduction of 2.2 dB in the signal-to-noise-ratio due to the lower than Nyquist sampling rate.

Dovis et al (2005) implement a real-time hybrid FPGA/DSP platform Software Digital Radio (SDR). The high computational operations such as carrier removal, correlation array, the local carrier generator, and local code generator are implemented on the FPGA. However, the lower data rate math-intensive operations such as code and carrier discriminators and code and carrier loop filters are implemented on the DSP. They evaluated their design by comparing the results of the hybrid SDR and the simulation.

At Konkuk University (Korea), the correlator, code generator and NCO have been designed on the FPGA (Cho et al 2005). For the code generator, an on-chip FPGA RAM has been

used. The designers use the Simulink¹ to generate the VHDL code required for the FPGA. The interface between the FPGA and the PC is accomplished by a PCI connection, while verification has been done by comparing the results with the GP2021 receiver. An adaptive filter is also implemented in the FPGA in front of the correlator to degrade the narrowband interfaces.

From the above discussion, it becomes apparent that the FPGA provides a very convenient and highly flexible platform for research projects where several features of the FPGA are essential. First, as a field programmable device², it can be programmed “in the field” (Wolf 2004). In other words, as soon as the design is completed, it can be programmed to the FPGA and tested immediately. This is a vital element in research projects where the system needs to be constantly changed and updated. Second, an FPGA is an excellent prototyping vehicle. By using the FPGA in the final design, the jump from prototype to product is much smaller and easier to negotiate. Third, the same FPGA can be used in several designs. This reduces inventory costs of research projects. Lastly, the large number of the FPGA IOs and the provided tools by the FPGA vendors (such as ChipScope by Xilinx) offer a huge opportunity for testing the design.

1.2. Motivation

This thesis is a part of a larger project to estimate the position of an MS using the MS-based approach in a ground-based system. The pilot channel of the IS-95 CDMA is

¹ System generator software developed by Xilinx for those who do not have FPGA programming knowledge.

² Field Programmable Gate Array (FPGA)

used for this purpose. As a research project, it has to support different radiolocation methods¹; thus, a multi-channel receiver is designed. Other investigators have contributed to different parts of the main project. Lopez (2006) designed and implemented the RF unit of the 5-channel CDMA receiver. Lu (2007) and Moghaddam (2007) proposed methods to improve the accuracy of the positioning by mitigating multipath. Lu (2007) employed MUSIC to estimate the LOS AOA and attenuate the NLOS signals by using beam forming. Moghaddam (2007) assumed that, unlike multipath, the LOS signal is correlated in time and space to coherently combine the different spatial and time observations. The FPGA-based receiver has evolved through the above research projects to solve the problems encountered with the previous versions of the receiver such as real-time code and frequency tracking.

The focus of this work is to investigate the signal processing limitations of the current receiver and develop a practical and highly flexible solution through using an FPGA. Specifically, the large spreading factor of the IS-95 pilot signal ($=2^{15}$ chip/s) and the simultaneous support of five channels are two key challenges to real-time signal processing. For example, suppose a minimal system of five channels with an 8-bit quantization and a Nyquist sampling rate of 2.4576 MS/s. To acquire and track 10 BSs per channel, the system has to be capable of processing two Gigabits per second². It should be noted that part of the processor's power is required for non-signal processing tasks such as real-time interfacing with the RF unit and position calculation. Moreover, to detect weak

¹ Refer to section 1.1.

² 10 (base stations) \times 5 (channels) \times 2 (I/Q) \times $2 \times 1.2288 \times 10^6$ (Nyquist rate) \times 8 (bits) = 1.96608 Gb/s

signals and to support a dynamic platform, a large record of data is needed. As shown in Chapter 5, even for a 1 epoch of data record, the computational load is challenging for a regular PC-based receiver. Furthermore, it is of the utmost importance that the research project provides a flexible, high speed processing platform capable of supporting different positioning algorithms (e.g AOA which uses an antenna array) and future system growth.

As explained earlier, in other receivers the real-time signal processing is achieved by partitioning the signal processing unit to high computational tasks and low computational tasks. Similar partitioning has been used in this project; as such, the most computationally demanding tasks (Doppler removal and de-spreading modules) are implemented on an FPGA.

1.3. Thesis Objective and Contributions

The main objective of this work is to design and implement a flexible platform for the signal processing unit of a real-time multi-channel IS-95 CDMA pilot signal receiver. The resulting receiver, which is able to track several BSs' signals simultaneously, is an innovative receiver structure that enables subsequent research into analysis of simultaneous received CDMA signals. To the best of the writer's knowledge, this is the first serious attempt at the realization of such a platform. The major tasks of this thesis are:

1. Study the functionality of different partitioning for the signal processing unit and propose proper partitioning between firmware and software based on the functions of the receiver. In the final system, the correlator bank and the Doppler removal

components are implemented on an FPGA, while the acquisition and part of the tracking loop decision are implemented in the software.

2. Map the research requirements of an eventual multiple antenna receiver array, TOA, position estimation into an implementable receiver design and associated processing.
3. Study the different processing methodologies for the tracking procedure.
4. Design and implementation of the batch tracking procedure using correlation banks in firmware.
5. Design and implementation of the sequential tracking procedure for receiver clock drift in firmware.
6. Test the system with both simulated and real data in static mode.
7. Predict the system response in dynamic mode by investigating its limitations.

The contributions of this thesis are to design and implementation of the signal processing unit on an FPGA and PC and integrate them with the RF unit to develop a 5-channel CDMA instrumentation receiver.

1.4. Thesis Outline

The arrangement of this thesis is as follows:

Chapter 2 starts by introducing the CDMA IS-95, where the focus is on the pilot channel characteristics and demodulation. Next, different methods of radiolocation are briefly discussed. The chapter terminates with the clarification of two main estimation algorithms: Least-Squares and the Maximum Likelihood.

Chapter 3 includes the basic theoretical discussion of the signal processing modules in a CDMA receiver. It covers the major components and methods of the signal processing unit, such as ADC, acquisition, and tracking; thus, provides a fundamental basis for subsequent chapters.

Chapter 4 narrows the discussion of the receiver to that used in this thesis (i.e. the PLAN group receiver prototype). In this chapter, the main restrictions of the previously implemented designs are explained. It concludes with the introduction of the design proposed herein and a discussion of its practical aspects.

Chapter 5 covers the details of the design which consists of the methods used to develop the code and frequency tracking algorithms. The code tracking is implemented by extracting the CIR (Channel Impulse Response) using 50 samples correlation window. The frequency tracking consists of two separate parts, one related to the frequency error caused by the receiver motion and the other related to the frequency error caused by the receiver clock drift. These parts are implemented on the PC and inside the FPGA respectively. The chapter also discusses the practical challenges encountered during the receiver's conception and methods have been used to solve each. The chapter ends by showing the results of the developed algorithms.

Chapter 6 concludes the thesis and provides a summary for possible future work. The chapter ends by presenting the capability of the design to scale. An analysis is presented that shows the necessary design modification in order to use it for satellite-based systems.

CHAPTER 2: POSITIONING TECHNIQUES FOR IS-95 CDMA SYSTEMS

To build a general and highly flexible receiver, it is necessary to determine its system requirements; thus, it necessitates (i) a study of the IS-95 pilot channel characteristics and (ii) an investigation of different positioning techniques. The first section of this chapter gives a summary of the history and specifications of the CDMA IS-95 forward channel, followed by a review of the pilot channel signal, its characteristics and demodulation. The second section of this chapter reviews the main radiolocation positioning methods, their system requirements, and gives a brief discussion regarding the two most commonly used estimation algorithms, LSE and MLE.

2.1. CDMA IS-95

In 1989, the inadequate old analog standards, known as Advanced Mobile Phone Service (AMPS), obliged the Telecommunication Industry Association (TIA) to accept the Time Division Multiple Access (TDMA) technology as the radio interface standard. Nonetheless, from the start it was apparent that the new technology was not sufficient for the continued

growth of the wireless service. QUALCOMM Inc. then took the initiative and developed a Code Division Multiple Access (CDMA) system to meet TIA requirements. Eventually, in 1993 TIA accepted the Interims Standard-95 (IS-95) as the CDMA standard and named it CDMAone (Harte et al 1999).

CDMA like other access techniques¹ tries to share the available bandwidth efficiently by using Spread Spectrum Modulation. Spread Spectrum originated in the military, and as such, contains two specific features (Yacoub 2002): *(i)* the signal should not be detectable by the enemy and *(ii)* the signal should be impervious to enemy interference. Contrary to the other modulations, which try to minimize the required transmission bandwidth, Spread Spectrum employs a transmission bandwidth much larger than the signal bandwidth.

With Spread Spectrum, frequency efficiency is achieved if a large number of users share the available bandwidth. Two types of Spread Spectrum modulation are considered: Frequency Hopped Spread Spectrum (FH-SS) and Direct Sequence Spread Spectrum (DS-SS). In both, a pseudo-random pattern generator is used to avoid interference between signals from different users.

In FH-SS, the transmission frequency is periodically changed based on the pseudo-random pattern generator output. Thus, the resulting signal can be considered as a data modulation with time-varying, pseudorandom carrier frequency (Rappaport 2002). Alternatively, the

¹ The other two are Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA)

pseudo-random pattern generator in DS-SS is used to change the phase of the information signal pseudo-randomly. This is achieved by directly multiplying the information signal with the output of the pseudo-random pattern generator. The IS-95 CDMA, which is the focus of this thesis, uses the DS-SS modulation.

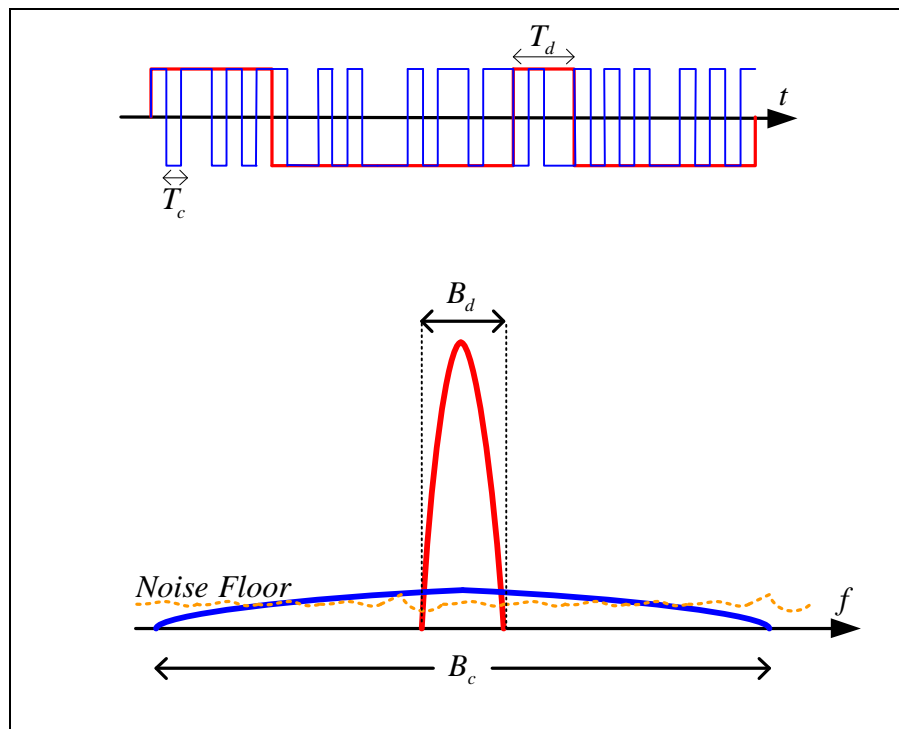


Figure 3: Spread Spectrum Modulation in Time and Frequency domain

2.2. Direct Sequence Spread Spectrum (DS-SS)

As previously stated, the frequency efficiency in DS-SS is achieved by spreading the information data $d(t)$ (with bit rate of B_d) with a much greater bandwidth random or pseudo-random pattern $pn(t)$ (with bit rate of B_c , where $B_c \gg B_d$). The resulting signal has a bandwidth of $B_c + B_d \approx B_c$. Figure 3 demonstrates this modulation in the time and

frequency domain. In the time domain (top), $T_d = \frac{1}{B_d}$ is the period of the information data and $T_c = \frac{1}{B_c}$ is the period of the pseudo-random pattern. In the frequency domain (bottom), the signal is buried under the noise floor which is shown by a dashed line. Since the pseudo-random pattern occupies a very large bandwidth, similar to the white noise, it is also called pseudo-noise (PN) sequence (or code).

Several properties of the DS-SS make it suitable for digital voice communications:

1. **No frequency planning:** Since in DS-SS systems all users can share the same spectrum at all time, the system does not have to assign frequencies to the users or have any frequency planning.
2. **Interference rejection:** This is also referred to as Anti-Jam (AJM) feature and can be considered from two aspects:
 - a. **Internal Interference rejection:** In DS-SS modulation, each user is assigned a unique PN sequence. These sequences are approximately orthogonal to each other. This means that each PN sequence has high auto-correlation and low cross-correlation with other PN sequences. Therefore, the interference between users inside a DS-SS system can be negligible as long as the number of users does not exceed the system capacity threshold.
 - b. **External Interference rejection:** Spreading the signal over a wide bandwidth makes the signal resistant against the external narrowband interference – this

type of interference only has an effect on a small portion of the spectrum (Rappaport 2002).

3. **High Security:** The noise-like appearance of the spreaded signal, while using low average transmitting power, can hide the signal in the background noise. In other words, the probability of detecting the signal by a third party is minimized; hence, this feature is also referred as Low Probability of Intercept (LPI) (Proakis 2001).
4. **Resistance to Multipath:** A PN sequence does not only have a poor correlation with other PN sequences, but also has a low correlation with its delayed versions. This implies that the multipath, which is a delay version of the original PN sequence, is considered as noise. This property is particularly useful for positioning systems where location estimation accuracy is dramatically reduced by multipath. However, it should be noted that multipath with a delay less than a chip period cannot be ignored as it is correlated to the original signal.

2.3. CDMA Forward Channel

The IS-95 CDMA forward channel is composed of 63 potential traffic channels, a synchronization channel, up to 7-paging channels and a pilot channel.

- Traffic channels transfer customer voice and data and can support different data rates of 1200, 2400, 4800, and 9600 bps (Rappaport 2002).
- Synchronization channel sends synchronization information to the MSs.
- Paging channels notify the MS of the incoming calls as well as send control signals.

- The Pilot channel is used as a coherent phase reference for demodulating other channels.

Figure 4 depicts the IS-95 CDMA forward channel in a transmitter. The traffic and paging channels undergo similar stages. At first, the data is encoded and is repeated, if required (to maintain a constant bit rate). After it is encoded, the data is interleaved to overcome the burst error problem in the channel in the Interleaver Block.

Next, the data, which at this point has a rate of 19.2 kbps, is scrambled by multiplying it with the Long PN code. This code is a $2^{42}-1$ chip sequence with chip rate of 1.2288 Mcps. To match the rate of the Long PN code with the output of the Interleaver Block, a decimator is used. The Long PN code is used as a unique identification for a call on both forward and reverse channels.

The next step is similar for all the channels and assures the orthogonality of different users and different channels in a cell. For this purpose, orthogonal codes known as Hadamard or Walsh codes are used. These Walsh codes are generated using a Walsh function matrix, defined as:

$$H_1 = 0$$

$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

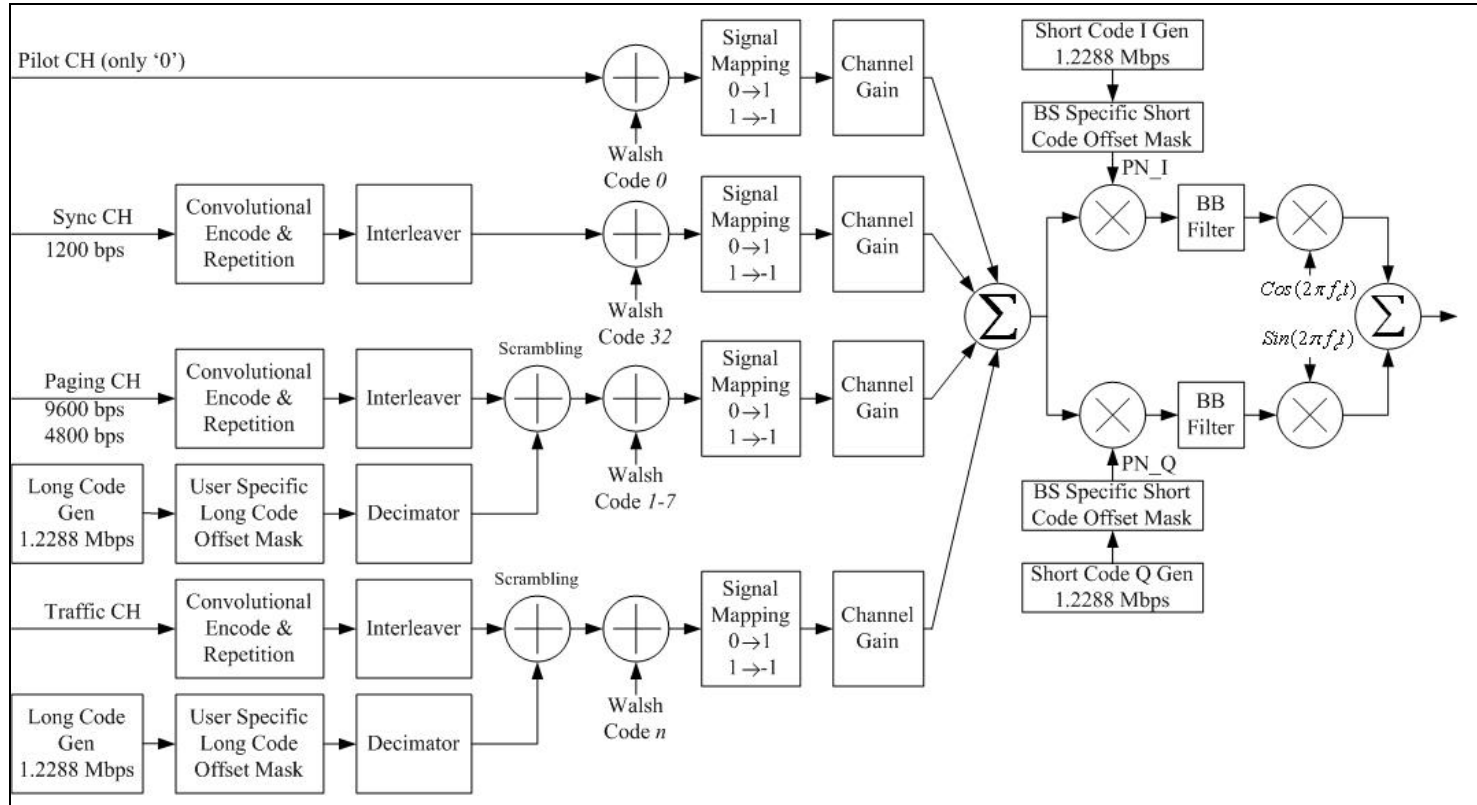


Figure 4: IS-95 CDMA forward channel (from Korowajczuk et al 2004)

$$H_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & \overline{H_N} \end{bmatrix}, \text{ where } N \text{ is a power of 2.}$$

In a forward channel, a 64×64 Walsh matrix is used. Each 64-bit row of the matrix is assigned to a channel. This indicates that 64 channels are available in each cell (BS). Some of the Walsh sequences are already assigned. Specifically, the first row (Walsh code 0) contains all '0's, and is assigned to the pilot channel. Also, the Walsh code 32 is assigned to the synchronization channel. In the instance of the paging, Walsh code 1 to 7 can be used. Consequently, given the dedication of the physical channels to the predetermined IS-95 forward channels, 55 to 61 channels can be used as traffic. Furthermore, the Walsh code period is 52.083 μs, which is equal to one scrambled data chip length; thus, each data chip is multiplied to one complete sequence of the Walsh code.

After the signal mapping and channel gain modules, the data from different channels are combined. Following this, the data is spread by multiplying it by two PN codes with the length of 2^{15} chips. To distinguish this code from the Long PN code, it is referred to as a Short PN code. Figure 5 depicts the final IS-95 CDMA forward channel structure.

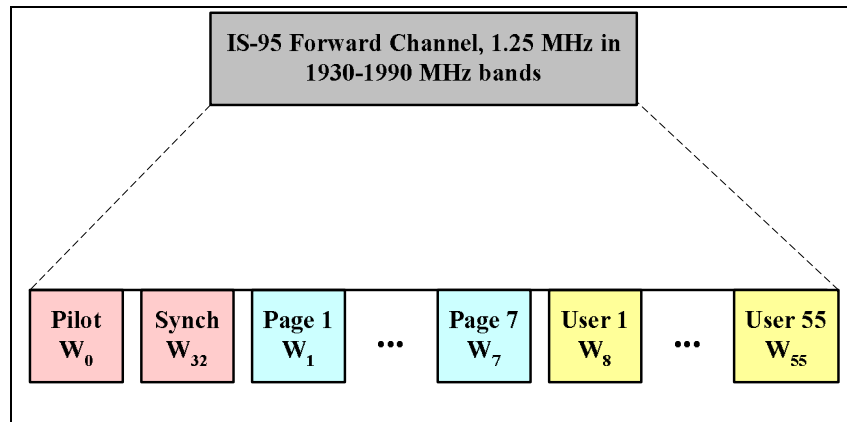


Figure 5: IS-95 CDMA forward channel structure

2.4. Pilot Channel

The Pilot channel generation circuit is repeated in Figure 6. As explained in the previous section, the initial pilot channel and its corresponding Walsh code (Walsh 0) consist of all zeros. This implies that after the spreading process, the final pilot channel is the short PN code. For this reason, in IS-95 CDMA, the Short PN code is also referred to as the pilot channel.

It is important to note that a unique Short PN sequence is assigned to all BSs. However, a BS specific mask is used after the Short PN code generator, as shown in Figure 4, to identify each BS in the cellular system. To create this BS specific mask, a time offset is applied to the PN sequence; thus, BSs synchronization is required. This time synchronization is accomplished by using a precise GPS receiver at each BS tower. To avoid any conflict between adjacent cells, the offsets are allocated in a space margin of 64 chips.

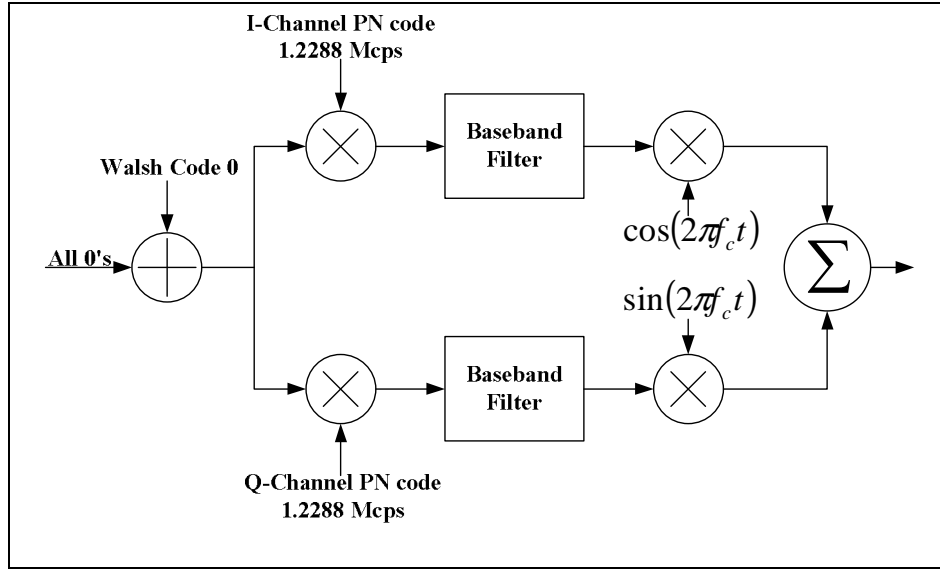


Figure 6: Pilot Channel Generation circuit

The I and Q polynomials related to the Short PN codes are:

$$P_I(x) = x^{15} + x^{13} + x^9 + x^8 + x^7 + x^5 + 1 \quad (2.1)$$

$$P_Q(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + 1$$

Using the polynomial characteristics, the equivalent linear recursion to (2.1) equations are

(Rappaport 2002):

$$i[n] = i[n-15] \oplus i[n-10] \oplus i[n-8] \oplus i[n-7] \oplus i[n-6] \oplus i[n-2] \quad (2.2)$$

$$q[n] = q[n-15] \oplus q[n-13] \oplus q[n-11] \oplus q[n-10] \\ \oplus q[n-9] \oplus q[n-5] \oplus q[n-4] \oplus q[n-3]$$

where, \oplus is the modulo-2 addition. Their corresponding Linear Feedback Shift Register (LFSR) circuits are depicted in Figure 7.

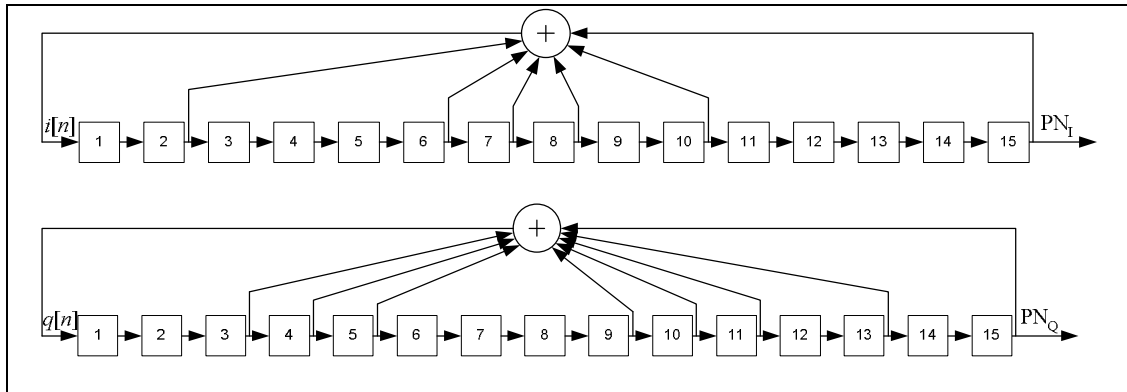


Figure 7: Linear Feedback Shift Register for generating Short PN code

Since the resulting codes of LFSR are $2^{15}-1$ length, a zero will be inserted at the end of each period. The new short code with the length of 2^{15} is called a modified short code and has a chip rate of 1.2288 Mcps (Etemad 2004). Each period of this code is called an epoch. To obtain different offsets of the PN sequence, a different initialization code is used for the LFSR.

2.4.1. Pilot Channel for Positioning

There are several reasons that make the pilot channel an appropriate candidate for positioning:

1. As an un-modulated (no data) signal, it is easy to acquire.
2. It is continuously broadcasting by all BSs. This implies that it is always available and no additional load is placed on the network if it is used for positioning.
3. Each BS sends the same pilot channel with a pre-defined code offset. This indicates that the direct result of channel acquisition is the TOA.
4. Since it is used for demodulation of other channels, it has the highest power. This increases the probability of finding a position solution.

2.4.2. Pilot channel Demodulation

Correlator architecture is considered to be optimized for demodulating the pilot channel corrupted in AWGN. The correlator consists of a multiplier and an integrator as shown in Figure 8. To detect the pilot channel, this demodulator should be used for different delayed versions of the locally generated pilot channel. This can be accomplished by integrating for different quantity of k in Figure 8.

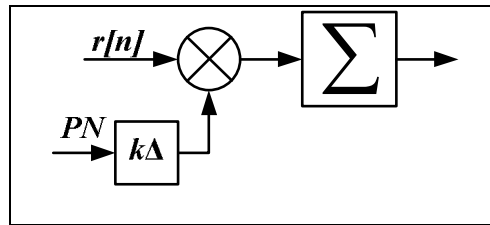


Figure 8: Correlator demodulator – sliding correlate

The integration should be obtained over a portion of the code’s period and the pilot channel code phase is detected by selecting the greatest output of the integrator. Assuming T_i is the time dedicated for one integration period, the correlator has to increase the delay k by one after each T_i . Since the received signal is periodic this structure also models the situation, where the received signal is fixed while the PN code is sliding over it chip-by-chip. Hence, this architecture is also known as a sliding correlator.

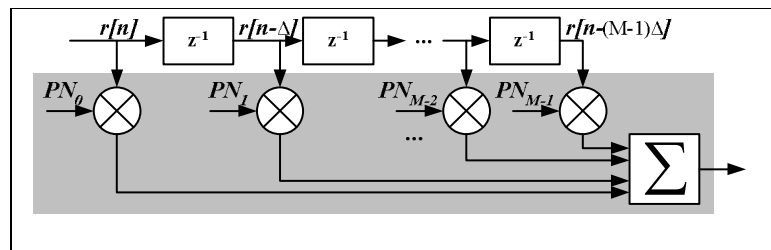


Figure 9: Correlator demodulator – parallel correlator

Although this structure is very resource efficient¹, it is very slow. To increase the speed of the demodulation, the architecture shown in Figure 9 can be used. Additionally, the correlator demodulator speed can be further enhanced if a bank of correlators is used as depicted in Figure 10.

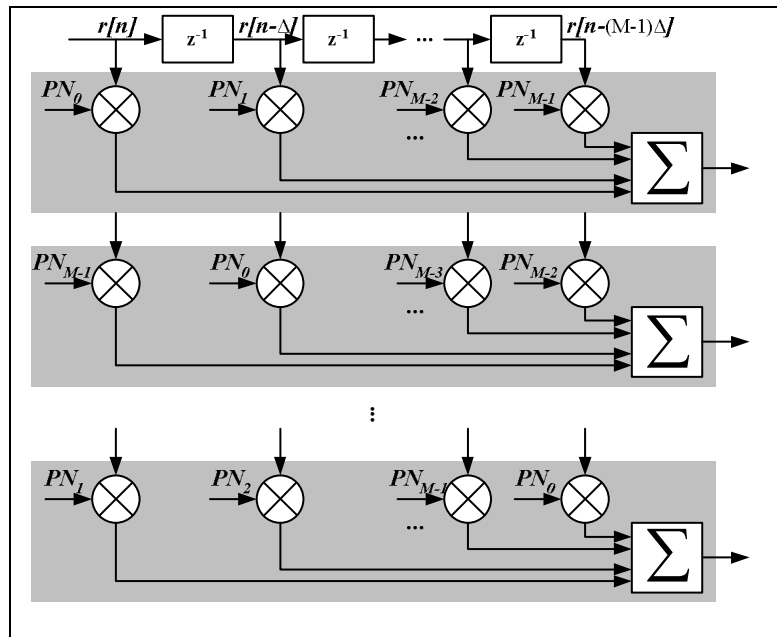


Figure 10: Bank of Correlators

The correlator's coefficients (PN_0, \dots, PN_{M-1}) are being changed at each clock cycle and the results of the multipliers are added together to produce the output of the correlator. This structure reveals that the bank of correlators can be replaced by a bank of Linear Filters. The impulse responses of these filters, which are known as the Match Filters, are the linear function of different shifted-version of the pilot code. Figure 11 depicts a correlator circuit with three sample integration time and its equivalent Matched Filter circuit.

¹ It only employs one multiplier and one adder.

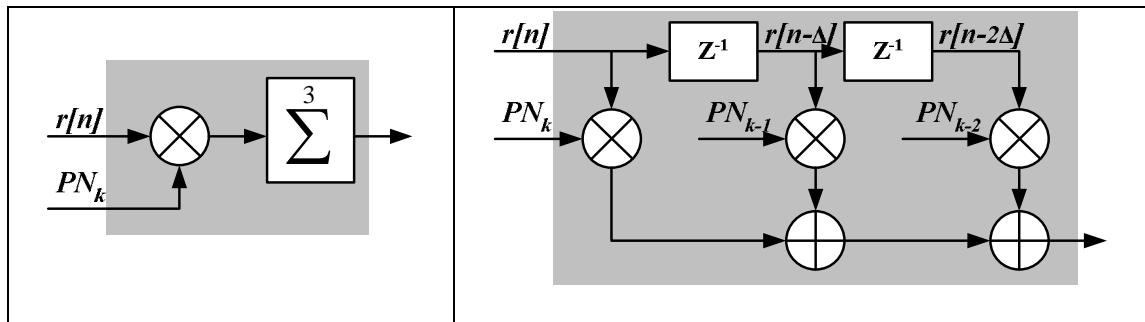


Figure 11: Correlator and its Matched filter equivalent circuit

2.5. Radiolocation

To locate the position of an object on the earth's surface, different techniques have been proposed such as dead reckoning, proximity systems and radiolocation (Caffery 2000). The main goal is to find the position of the object of interest with respect to a known position, also called as relative position estimation. Among different techniques, radiolocation is widely used for subscriber location estimation.

The main objective of the radiolocation is to find the location of a Mobile Station (MS) by measuring the distance between the MS and a set of BSs. For this purpose, different methods have been suggested: Received Signal Strength (RSS), Angle Of Arrival (AOA), Time Of Arrival (TOA), and Time Difference Of Arrival (TDOA). Under different conditions and in different circumstances any of these methods (or a combination) can give a more accurate result. Therefore, none of these methods can be assumed as the best. In any case, the position can be found via two steps:

1. Acquire the required information by measuring one of the received signal's parameters such as the signal power (in RSS), the signal angle of arrival (in AOA),

the signal time of arrival (in TOA), or the signal time difference of arrival (in TDOA).

2. Estimate the MS position based on the acquired information. This can be accomplished by using one of the well-known estimation algorithms such as Least-Squares (LS) or Maximum Likelihood (ML).

Generally, the ability to find the precise location of a MS greatly depends on the accuracy of the information that has to be obtained before using any location estimation algorithm (Caffery 2002). In the following sub-sections, the radiolocation methods will be briefly explained.

2.5.1. Received Signal Strength (RSS) Method

The RSS method is based on the path loss effect of the radio signal broadcast from the BS, where the strength of the signal is inversely proportional to the square of the distance between the transmitter and the receiver. The Friis Equation shows the relation of the signal power at the transmitter's and receiver's antenna (Cheng 1993):

$$P_{Rx} = \frac{G_{Tx} \cdot G_{Rx}}{L} \cdot \left(\frac{\lambda}{4\pi d} \right)^2 P_{Tx} \quad (2.3)$$

where

P_{Tx} : the transmitted power

G_{Tx} and G_{Rx} : the transmitter and receiver antenna gains, respectively

λ : the signal wavelength

d : the distance between transmitter and receiver

L : system loss factor (≥ 1 , for free-space this is 1).

The only unknown parameter d will be calculated after measuring the power of the signal at the receiver. This formula can be used only for Line Of Sight (LOS) and is not a realistic model for terrestrial propagation. The primary sources of error for RSS methods are multipath fading and shadowing, which cause low accuracy results at best and faulty results at worst (Caffery & Stüber 98).

2.5.2. Angle of Arrival (AOA) Method

In AOA, the direction of the received signal is estimated using directional antennas or antenna arrays. In most applications, the direction is calculated by measuring the phase difference between the antenna array elements. If the accuracy of the measurement is $\pm\alpha$ (i.e. α is the standard deviation), then the uncertainty in the resultant position will be limited to 2α (Pahlavan et al 2000). This indicates that the accuracy of the AOA is severely limited by the relative position of the MS and BSs (Figure 12).

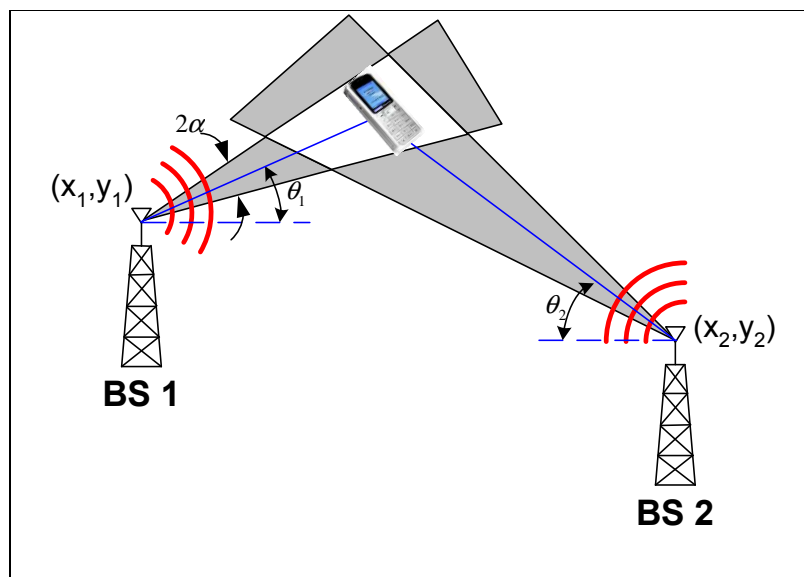


Figure 12: Angle of Arrival method

Compared to other methods, AOA has two important advantages. First, a minimum number of two BSs is sufficient for location estimation. Second, it does not require time synchronization between the BSs and MS.

Conversely, the main disadvantage of the AOA method is its sensitivity to the propagation environment, especially to the scattering near and around the MS and BS. This dramatically limits the AOA accuracy in indoor applications. In the absence of LOS signals the antenna array will lock on the reflected signal, which is unlikely to have the same direction as the LOS (Caffery 2002). Also, though the minimum number of two BSs is enough for this method, the estimation error can be reduced if more than two BSs are used. Furthermore, there are some practical problems related to the AOA method. For instance, the size of the antenna array has to be analogous to the signal wavelength. It is impractical to install such a huge antenna array on MSs (Gua 2004). Even in network-based positioning, it is still a challenge to install the antenna arrays in every BS tower (Sayed et al 2005).

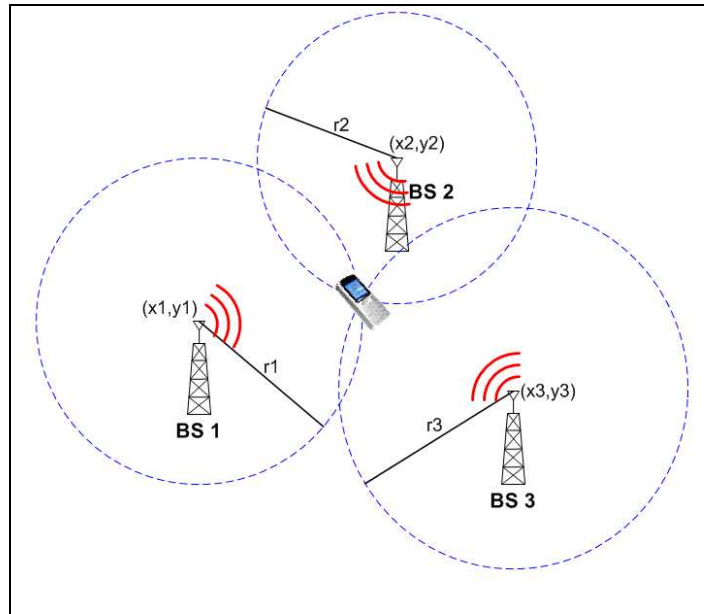


Figure 13: Time of Arrival Method

2.5.3. Time of Arrival (TOA) Method

In the TOA method the position of the MS is determined by measuring the time that it takes for the signal to travel between a set of BSs and the MS. From a geometric point of view, this can be seen as a set of circles around BSs, where the MS is located on the intersection of these circles as shown in Figure 13. The advantage of this method is in its robustness to multipath fading, while the main disadvantage is the necessity of MS to be time synchronized with the BSs. As an example, consider applying the TOA algorithm to the three BSs of Figure 13. The distance between MS and BS_i can be obtained by

$$r_i = (t_i - t_{0-i})c \quad (2.4)$$

where c is the speed of light, t_{0-i} is the time of transmit of the signal from BS_i and t_i is the time the signal is received at MS. Assuming (x_m, y_m) as the location of the MS, the distance from BS_i , located at the coordinate (x_i, y_i) , is calculated by:

$$r_i^2 = (x_i - x_m)^2 + (y_i - y_m)^2 = x_i^2 + y_i^2 + x_m^2 + y_m^2 - 2x_i x_m - 2y_i y_m \quad (2.5)$$

$i = 1, 2, 3$.

By calculating the subtraction of two pairs of equations (2.5):

$$r_2^2 - r_1^2 = (x_2^2 + y_2^2) - (x_1^2 + y_1^2) - 2(x_2 - x_1)x_m - 2(y_2 - y_1)y_m \quad (2.6)$$

and

$$r_3^2 - r_1^2 = (x_3^2 + y_3^2) - (x_1^2 + y_1^2) - 2(x_3 - x_1)x_m - 2(y_3 - y_1)y_m \quad (2.7)$$

Now if we assume that BS₁ is located at $x_1 = 0$, $y_1 = 0$, then after rearranging terms, one has:

$$\begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (x_2^2 + y_2^2) - r_2^2 + r_1^2 \\ (x_3^2 + y_3^2) - r_3^2 + r_1^2 \end{bmatrix}. \quad (2.8)$$

In general case, with n BSs, the location of the MS can be obtained by

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \frac{1}{2} \left(\begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}^T \cdot \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}^T \cdot \begin{bmatrix} (x_2^2 + y_2^2) - r_2^2 + r_1^2 \\ (x_3^2 + y_3^2) - r_3^2 + r_1^2 \\ \vdots \\ (x_n^2 + y_n^2) - r_n^2 + r_1^2 \end{bmatrix} \quad (2.9)$$

All of the variables in the right hand side of the equation (2.9) are known. Therefore, the unknown variables of the MS position (x_m, y_m) can be calculated.

2.5.4. Time Difference of Arrival (TDOA) Method

In the TDOA method, instead of using the absolute TOA, the ‘time difference of arrival’ is used. In this case, the circles around BSs in Figure 13 change to hyperbolic lines as seen in

Figure 14. In this figure, r_1 , r_2 , and r_3 are distances between MS and three base stations of BS1, BS2, and BS3, respectively. Hence the MS is located on the intersection of the

hyperbolic curves of $\begin{cases} r_1 - r_2 = \text{const} \\ r_3 - r_2 = \text{const} \end{cases}$. Compared to TOA, TDOA is more applicable

because it does not require MS-BS time synchronization.

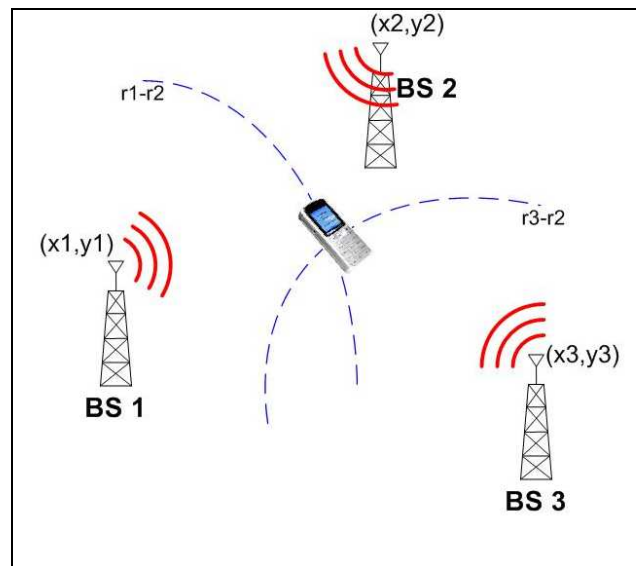


Figure 14: hyperbolic system for TDOA method

2.6. Sources of Location Errors

The result of the radiolocation methods is a series of observables that are used to find the distance between the receiver and transmitter. In an ideal case, the geometric interpretation of the measured parameters can be used. For instance, in the TOA case, this results in circles of possible position around each BS, while the intersection of multiple circles indicates the probable MS location. This type of algorithms can work perfectly if there are no measurement errors. However, as illustrated in Figure 15, the circles do not intersect at one unique point if the measurement error exists. In reality with multiple sources of noise

and interference, the geometric interpretation cannot be used. Therefore, statistical methods have to be employed to find the location.

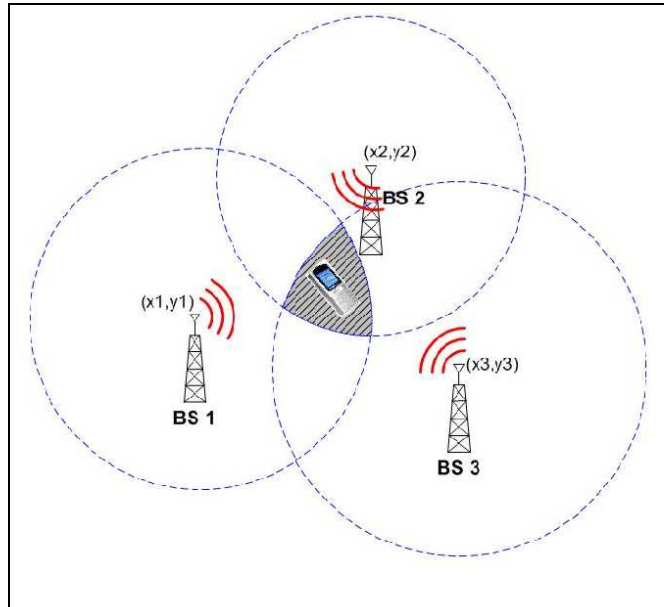


Figure 15: Radiolocation in the presence of measurement errors

Error sources can be classified in two groups; *(i)* errors related to the imperfect equipment and methods (e.g. AOA, TOA, and TDOA) used to measure the location parameters (Caffery 2002) and *(ii)* errors related to the propagation medium such as Multipath and Non-Line Of Sight (NLOS) (Caffery 2002). The later group of errors are the more serious – specifically those caused by multipath and NLOS.

Multipath refers to the phenomenon where the signal is reflected off of surrounding objects between the BS and MS. When multipath exists, several copies of the same signal arrive at the receiver. These signals have different amplitude, time delay and phase, which can be helpful in distinguishing the LOS signal from multipath. The first attempt towards distinguishing the LOS signal from multipath has been accomplished by the CDMA Spread

Spectrum modulation technique. As discussed earlier, by using this modulation, the multipath signals with more than one chip delay are not correlated with the LOS signal and can be omitted from the calculation. However, for multipath with delay less than a chip, some algorithms should be used to detect and mitigate its effect.

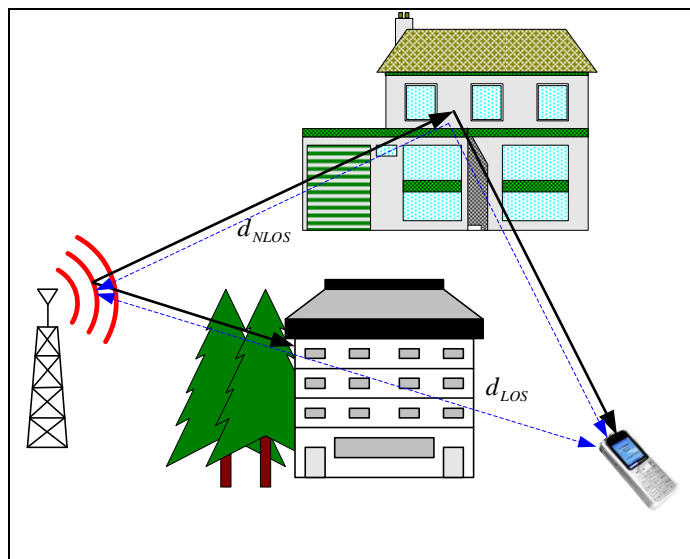


Figure 16: NLOS signal passes longer path than the LOS signal

The NLOS happens whenever the direct (LOS) signal from the BS to MS is obstructed as shown in Figure 16. Thus, the received signal at the MS is an indirect signal which passes through a longer distance than the LOS signal. The NLOS can severely affect the AOA result since it can arrive from a completely different direction. For the time-based methods, the signal experiences a larger delay due to the longer path $d_{NLOS} - d_{LOS}$. In most cases, the NLOS causes an error of up to 400-700 m¹ (Romdhani & Trad 2002, Silvetoinen & Rantalainen 1996).

¹ A very serendipitous event can happen if the NLOS signals from two BSs pass the same length, then the NLOS effect can be omitted by using the TDOA method (Caffery 2002).

2.7. Estimation Algorithms

The result of the mentioned techniques is an observable data set, for instance, $\{t_0, t_1, \dots, t_{N-1}\}$ for TOA. The estimator is responsible to produce a unique set of parameters based on this data set. Two widely adopted approaches to find such an estimator are Maximum Likelihood estimation (MLE) and Least-Squares Estimation (LSE).

The result of the MLE is the estimate of the unknown parameter, which is more likely to produce the measured data. The MLE can be employed in cases where the PDF is known. Then the unknown parameter is estimated by maximizing the PDF. The main assumption in the MLE is that there are a large number of observations (i.e. observed data): $N \rightarrow \infty$. In this case, it is guaranteed that the unknown parameter (vector) obtained by (Togneri 2005)

$$\hat{\theta} = \arg \max_{\theta} p(x; \theta) \quad (2.10)$$

is asymptotically distributed. This means that when $N \rightarrow \infty$, then the estimated value $\hat{\theta}$ converges to the real value θ .

Conversely, the Least-Squares Estimator (LSE) minimizes the squared differences between the measured data and the assumed, but unknown, noiseless data. This is accomplished by minimizing the LSE error criterion (or cost function) to measure the closeness of the measured data and noiseless data:

$$J(\theta) = \sum_{n=0}^{N-1} (x[n] - s[n; \theta])^2. \quad (2.11)$$

In the linear case, the observed data can be modeled as a linear function of the unknown parameter:

$$x[n] = s[n; \theta] + w[n] \quad n = 0, 1, \dots, N-1 \quad (2.12)$$

$$s(\theta) = H\theta$$

In equation (2.12), θ is a $p \times 1$ vector of unknown parameters, $s(\theta)$ is the unknown vector noiseless signal corresponding to the measurements and w is the noise vector of dimension $N \times 1$ with PDF $\mathbf{x}(0, C)$. H is a known $N \times p$ matrix ($N > p$). Both MLE and LSE result in (Kay 1993)

$$\hat{\theta} = (H^T C^{-1} H)^{-1} H^T C^{-1} x \quad (2.13)$$

where C is the error covariance matrix of the measurements. This matrix is generally assumed diagonal in which case a low error correlation in the equipment and propagation path is assumed. The cross-correlation error is difficult to evaluate and is therefore often neglected, resulting in a sub-optimal estimator.

Although the LS and ML methods give equivalent solutions in the linear case under the assumption of zero mean Gaussian noise, their results are different for the same data set when the measurements are not normally distributed. In this case, the MLE should be preferred over LSE. Nonetheless, it is important to be aware that the MLE is limited by its high computational intensity as it searches through all dimensions of the unknown vector; hence, it may not be appropriate for all applications.

Figure 17 shows a sample of the MLE (LSE) search space for the following observable data:

$$x[n] = As[n - n_\tau] + w[n] \quad n = 0, 1, \dots, N - 1 \quad (2.14)$$

where A and n_τ are the amplitude and delay (in terms of the data sample) of the noiseless signal $s[n]$ and $w[n]$ is AWGN. Thus, the result of MLE (or LSE) is the amplitude-delay values of the minimum of this bowl.

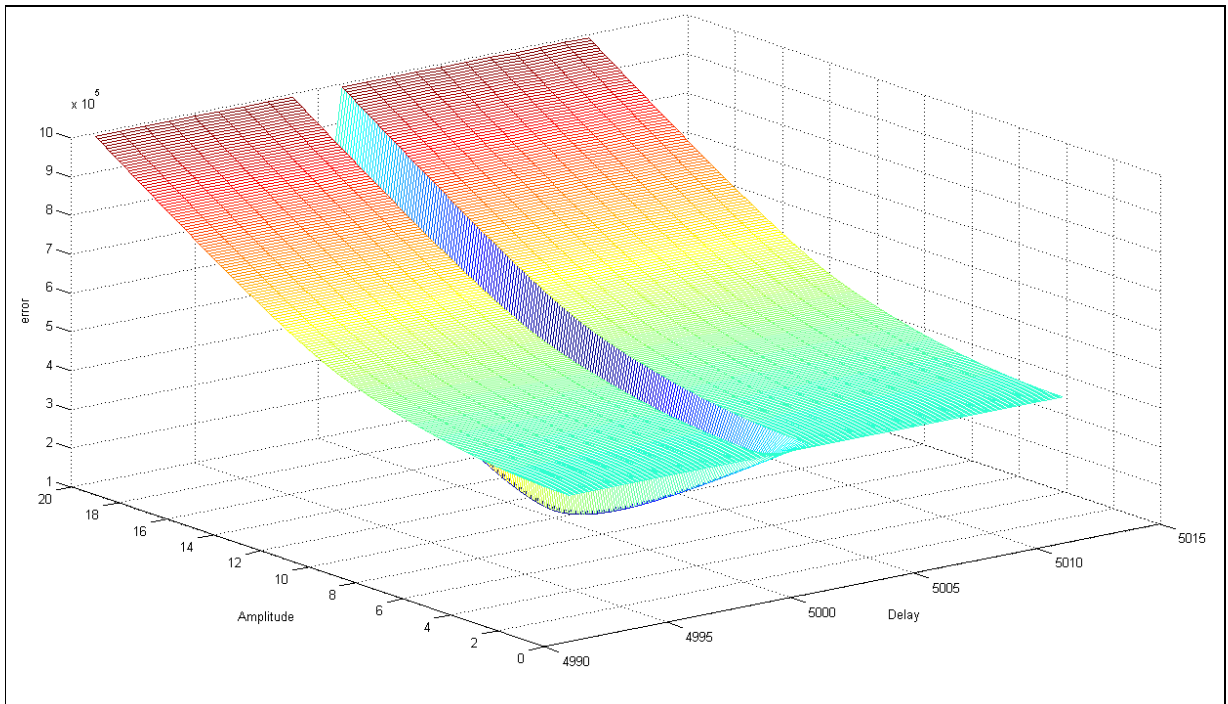


Figure 17: LSE and MLE search space

2.8. Summary

The structure of the pilot channel in CDMA IS-95 was introduced in this chapter. Moreover, different methods for radiolocation and their specifications and limitations were discussed. For instance, in most cases, at least three BSs are required to obtain the position

estimate. Also, to support the AOA, the system should be capable of processing several signals receiving from each branch of the antenna array.

To improve the result accuracy and to overcome a method's limitation, sometimes two methods are employed simultaneously in a system. As such, the system should have the required signal processing power and circuits of both methods. Also, it should be noted that in position applications, the signal is a nonlinear function of the unknown parameters and as explained earlier, the MLE is more likely to obtain more accurate results than the LSE. However, phenomena such as multipath increase the dimension of the unknown parameters. Thus, the corresponding MLE requires a massive search through all dimensions.

The focus of this thesis is to build a receiver

- for the IS-95 pilot channel with a large spreading factor of 2^{15} ,
- capable of supporting multiple BSs (required for different positioning methods) and multi-channels (for the methods using antenna array),
- capable of supporting (separately or simultaneously) different positioning algorithms regardless of their required processing power.

Therefore, to build a general and highly flexible receiver, which has the required foundation to support different positioning methods (separately or simultaneously, and regardless of their processing power), it is necessary to significantly decrease the load of the software and leave processing power for the positioning solution algorithm. This

suggests that it is appropriate to use an FPGA as an alternative. The next chapter explains the main modules inside such a receiver's signal processing unit.

CHAPTER THREE: SIGNAL PROCESSING IN CDMA PILOT CHANNEL RECEIVER

In general, signal processing refers to a set of procedures performed on the digitized BB (Base Band) signal. As explained in Chapter 2, in order to build a general and highly flexible positioning receiver, an FPGA can be used. This implies that it is necessary to partition the signal processing unit between the firmware (implemented on the FPGA) and software (implemented on the PC). As such, different signal processing algorithms should be studied and those which are suitable for firmware implementation should be chosen for the firmware design. This chapter covers the main signal processing algorithms:

- Digitization of the BB signal,
- Detection of the signal by establishing a two-dimensional search over time and frequency domains to obtain the signal's code phase and Doppler frequency, and
- Tracking the time variation of the code phase and Doppler frequency of the signal.

3.1. Signal Digitization

Two factors have to be considered when the signal digitization unit is designed: (i) at which stage the digitization should take place, and (ii) the selection of a proper sampling frequency. These factors are selected based on criterions which are discussed in the following paragraphs.

The signal digitization can be accomplished in three different stages. The first stage is called direct digitization. It involves sampling the signal at RF frequency. The second stage is to down-convert the input signal to an intermediate frequency (IF) before digitizing it (Tsui 2005). Lastly, the third stage is to digitize the signal at the Baseband (BB) frequency (Raquet 2006).

Digitizing in higher frequency is generally more advantageous as it does not require the mixer and local oscillator. A mixer, as a nonlinear device, can potentially pollute the signal by producing unwanted frequencies (Tsui 2005). Similarly, a local oscillator is capable of generating an impure frequency that can contribute distortion in the digitized signal (Tsui 2005). However, digitizing at high frequency requires the use of expensive amplifiers that are able to work at high frequency. It is also difficult to build an ADC that is capable of supporting high frequencies. Moreover, building a narrow-band filter at higher frequency is very complex and expensive.

Conversely, when digitizing at lower frequencies it is easier to create a narrow-band filter. Also, there are less costs associated with the amplifiers at low frequencies, but this can be offset by the use of a mixer and local oscillator – both of which can be expensive and can contribute to frequency errors (Tsui 2005).

In communication systems, the only sampling frequency criterion is related to the Nyquist theorem. Based on this theorem, the sampling frequency f_s has to be higher than twice of the bandwidth of the input signal Δf . Nevertheless, for positioning systems, another criterion has to be considered to choose a proper sampling frequency. In these systems the synchronization of the incoming signal with the locally generated signal can severely affect the accuracy of the position. The complete synchronization occurs if two signals have synchronized chip transitions. The synchronization accuracy, and subsequently the position accuracy, can be drastically decreased by the sampling process. To minimize the effect of sampling on the synchronization process, the sampling frequency should not be a multiple of the chip rate. Figure 18 shows the situation where the sampling rate is exactly twice of the “chip rate”. In this case, each analog chip is perfectly represented by two samples. Thus, the cross-correlation between the incoming signal and the local code is the same as long as the estimated phase lies between t_1 and t_2 (Fantino *et al* 2004). It is clear from the figure that by using the Nyquist boundary for the sampling rate, the time resolution¹ is half a chip.

¹ timing accuracy

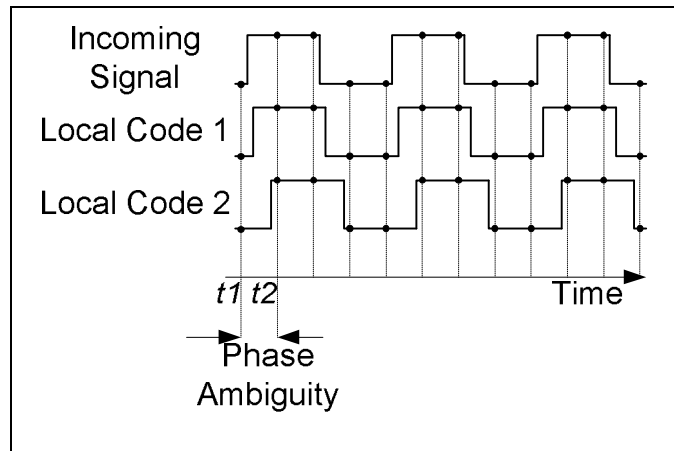


Figure 18: Phase Ambiguity resulted from commensurate sampling rate (Maurizio et al 2004)

For non-stationary applications, where incoming signal transitions are also affected by Doppler, the sampling frequency should not be a multiple of the “chip rate + code Doppler” (Tsui 2005). In such a system, choosing the sampling frequency even equal to the multiple of the chip rate does not deteriorate the synchronization. In other words, Doppler improves the synchronization (Akos 2006). A similar situation occurs for stationary applications via a free run sampling clock (Mileant *et al* 1995).

Assume that the analog signal is down-converted to the BB frequency. Thus, the input to the ADC can be written as (Raquet 2006)

$$I_B(t) = A PN_I(t) \cos(2\pi f_d t + \phi_0) \quad (3.1)$$

$$Q_B(t) = A PN_Q(t) \sin(2\pi f_d t + \phi_0)$$

where A is the amplitude of the signal at the BB, $PN_I(t)$ and $PN_Q(t)$ are the PN sequence of I and Q respectively, f_d is the Doppler frequency¹ and ϕ_0 is the phase of the received signal. After being sampled by the ADC, the signal can be represented by (Raquet 2006)

$$I_B^k = A PN_I^k \cos(2\pi f_d t_k + \phi_0) \quad (3.2)$$

$$Q_B^k = A PN_Q^k \sin(2\pi f_d t_k + \phi_0)$$

where I_B^k and Q_B^k are the sampled signal at time t_k .

3.2. Acquisition

The CDMA pilot channel signal acquisition involves a two-dimensional search procedure that roughly estimates the frequency and code offset of the received signal. As shown in Figure 19, the first step is to remove the Doppler frequency. This is accomplished by multiplying the BB signal with a replica carrier. If the frequencies of the two signals match perfectly, the output will be the incoming signal without Doppler. Thus, this process is also called Doppler removal or Doppler wipe off. Following this process is the de-spreading module where the signal is correlated with a replica code to determine the code phase.

¹ In this thesis, the Doppler frequency is defined as the combined frequency error generated by the clock drift and the transmitter-receiver relative movement.

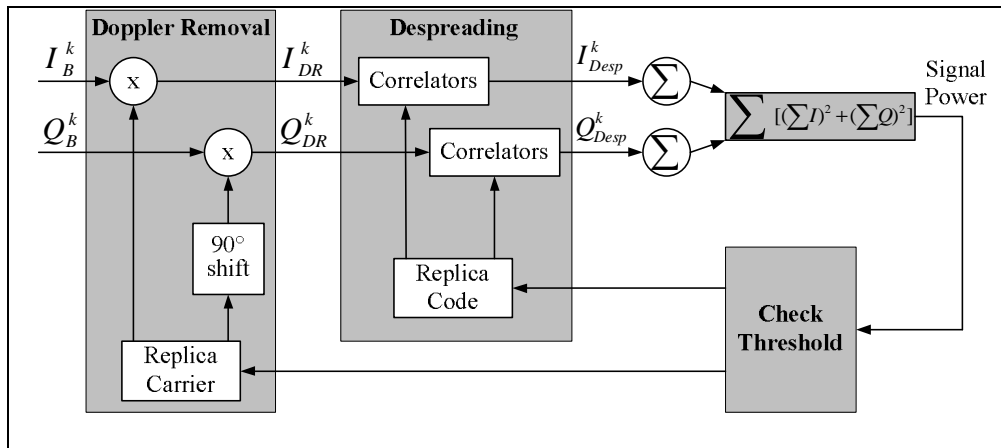


Figure 19: Two-dimensional search Acquisition procedure

The presence of the signal is detected by measuring the signal power. As such, the detection of the signal is completed in two steps: (i) the total I ($\sum I$) and total Q ($\sum Q$) are measured coherently over the integration time (T), (ii) these measured values are used to calculate the power of the signal ($\sum [(\sum I)^2 + (\sum Q)^2]$). If the power is greater than a predetermined threshold, then the signal is detected. Otherwise, the replica-carrier and code has to be updated and the process repeats for the new values.

The accuracy of the code offset is a function of the sampling rate. However, to increase the speed of the acquisition, the code offset bin size is usually chosen as half of a chip. The accuracy of the frequency offset (frequency uncertainty) is a function of coherent integration time¹.

¹ The time that the signal is integrated coherently ($\sum I$ and $\sum Q$ in Figure 19)

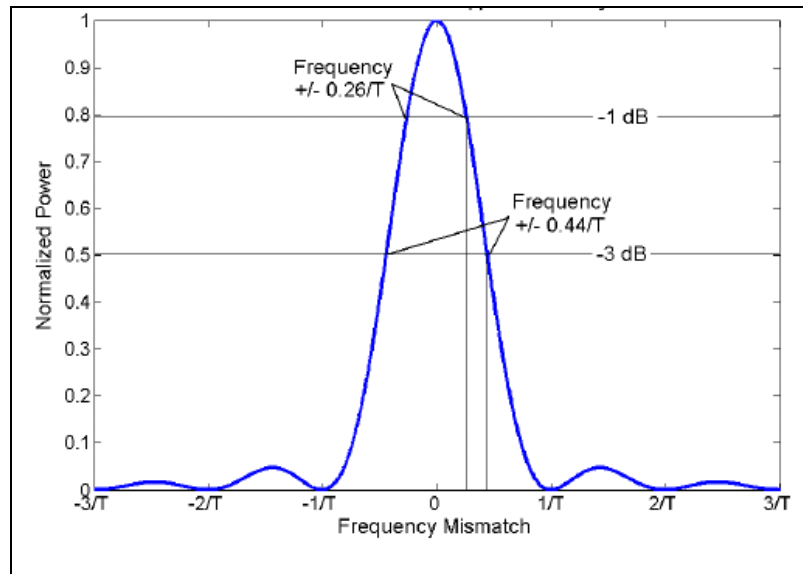


Figure 20: Power attenuation due to Frequency mismatch (Watson 2005)

The signal power is attenuated as a sinc-squared function of the frequency uncertainty as depicted in Figure 20 (Parkinson & Spilker 1996) (Watson 2005). The frequency error related to the 3 dB attenuation of the signal power is:

$$\Delta f = \frac{0.44}{T} \text{ Hz} \quad (3.3)$$

where T is the coherent integration time. Using a longer integration time can increase the chance of acquiring weak signals. However, as illustrated in equation (3.3), there is a trade off between the integration time and the speed of acquisition. A longer integration time produces a smaller frequency bin, resulting in a longer acquisition time.

3.2.1. Doppler Removal

As stated earlier, the Doppler removal module has to wipe off the Doppler frequency from the received signal. Without *a priori* knowledge of the Doppler shift, the search is

completed in a range of probable Doppler values. For the IS-95 Pilot Channel, this range should encompass the Doppler caused by the Clock offset at the receiver and the transmitter, and the Doppler shift caused by their relative motion. The Doppler search can start from the zero Doppler and then proceeds symmetrically one Doppler bin at a time until it covers all the range (Kaplan & Hegarty 2006). For each Doppler bin, all possible code offsets are searched before moving to the next Doppler bin.

At the receiver, after the down-conversion, the BB signal only contains the Doppler frequency as stated in equations (3.2). Thus, to extract the I and Q values from the BB signal, the latter has to be multiplied by the $\cos(2\pi f_d t)$ and $\sin(2\pi f_d t)$. The outputs of the

Doppler Removal module are

$$\begin{aligned}
 I_{DR}^k &= I_B^k \cos(2\pi f_{NCO}) - Q_B^k \sin(2\pi f_{NCO}) \\
 &= A PN_I^k \cos(2\pi f_d t_k + \phi_0) \cos(2\pi f_{NCO}) - A PN_Q^k \sin(2\pi f_d t_k + \phi_0) \sin(2\pi f_{NCO}) \\
 &= \frac{A PN_I^k}{2} [\cos(2\pi f_d t_k + \phi_0 + 2\pi f_{NCO}) + \cos(2\pi f_d t_k + \phi_0 - 2\pi f_{NCO})] \\
 &\quad + \frac{A PN_Q^k}{2} [\cos(2\pi f_d t_k + \phi_0 - 2\pi f_{NCO}) - \cos(2\pi f_d t_k + \phi_0 + 2\pi f_{NCO})]
 \end{aligned} \tag{3.4}$$

and

$$\begin{aligned}
 Q_{DR}^k &= -I_B^k \sin(2\pi f_{NCO}) + Q_B^k \cos(2\pi f_{NCO}) \\
 &= -A PN_I^k \cos(2\pi f_d t_k + \phi_0) \sin(2\pi f_{NCO}) + A PN_Q^k \sin(2\pi f_d t_k + \phi_0) \cos(2\pi f_{NCO}) \\
 &= -\frac{A PN_I^k}{2} [\sin(2\pi f_d t_k + \phi_0 + 2\pi f_{NCO}) - \sin(2\pi f_d t_k + \phi_0 - 2\pi f_{NCO})] \\
 &\quad + \frac{A PN_Q^k}{2} [\sin(2\pi f_d t_k + \phi_0 + 2\pi f_{NCO}) + \sin(2\pi f_d t_k + \phi_0 - 2\pi f_{NCO})]
 \end{aligned} \tag{3.5}$$

where f_d and f_{NCO} are the Doppler frequency and the replica carrier frequency, respectively.¹

3.2.2. De-spreading Module

The output of the Doppler removal module is correlated with the locally generated code. This process is also known as de-spreading, the reverse operation of spreading². As stated in Section 2.4.1, the direct result of pilot channel signal de-spreading is the TOA, which can be used to measure the range.

The de-spreading is performed by complex correlation of the complex received signal and the conjugate of the complex locally generated PN code as

$$\begin{aligned}
 R_{\tilde{r}, PN}(m) &= \frac{1}{N} \sum_{k=0}^{N-1} \tilde{r}[k] \cdot P\tilde{N}[k-m] \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} (I_{DR}^k + jQ_{DR}^k) (PN_I^{k-m} + jPN_Q^{k-m})^* \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} [(I_{DR}^k PN_I^{k-m} + Q_{DR}^k PN_Q^{k-m}) + j(Q_{DR}^k PN_I^{k-m} - I_{DR}^k PN_Q^{k-m})]
 \end{aligned} \tag{3.6}$$

where \tilde{r} is the complex signal at the input of the de-spreading module, $\tilde{r} = I_{DR}^k + jQ_{DR}^k$, and $P\tilde{N}$ is the complex local PN code. Equation (3.6) represents the de-spreading process in the time domain. Since the IS-95 pilot channel has 2^{15} (=32768) chips, the time domain correlation can be very time consuming – especially considering that this calculation has to be done for each Doppler bin.

¹ The Doppler removal module is also part of the tracking loop. Thus, further explanation is obtained in the frequency tracking section

² The spreading refers to the multiplying a narrow band signal with a wide bandwidth PN sequence. Refer to section 2.2.

The time-domain correlation is equivalent to the frequency domain multiplication. Thus, a more efficient method is to de-spread the signal in the frequency domain (i.e. FFT method).

Transferring two complex signal $\tilde{r}[k]$ and $P\tilde{N}[k]$ to the frequency domain results in

$$\tilde{r}[k] = \sum_{f=0}^{N-1} R(f) \cdot \exp(2\pi f k / N) \quad (3.7)$$

$$P\tilde{N}[k] = \sum_{f'=0}^{N-1} PN(f') \cdot \exp(2\pi f' k / N)$$

where $R(f)$ and $PN(f')$ are the Discrete Fourier Transform (DFT) of the $\tilde{r}[k]$ and $P\tilde{N}[k]$. By substituting Equations (3.7) into (3.6), the frequency domain de-spreading is obtained as

$$\begin{aligned} R_{\tilde{r}, P\tilde{N}}(m) &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{f=0}^{N-1} R(f) \cdot \exp(j2\pi f k / N) \right) \left(\sum_{f'=0}^{N-1} P\tilde{N}^*(f') \cdot \exp(-j2\pi f' (k-m) / N) \right) \quad (3.8) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left\{ \left(\sum_{f=0}^{N-1} R(f) \cdot \exp(j2\pi f k / N) \right) \right. \\ &\quad \left. \times \left(\sum_{f'=0}^{N-1} P\tilde{N}^*(f') \cdot \exp(-j2\pi f' k / N) \cdot \exp(j2\pi f' m / N) \right) \right\} \\ &= \frac{1}{N} \sum_{f=0}^{N-1} \sum_{f'=0}^{N-1} R(f) \cdot P\tilde{N}^*(f') \cdot \exp(j2\pi f' m / N) \cdot \left[\sum_{k=0}^{N-1} \exp(j2\pi (f-f') k / N) \right] \\ &= \frac{1}{N} \sum_{f=0}^{N-1} \sum_{f'=0}^{N-1} R(f) \cdot P\tilde{N}^*(f') \cdot \exp(j2\pi f' m / N) \cdot [N\delta(f-f')] \\ &= \sum_{f'=0}^{N-1} R(f') \cdot P\tilde{N}^*(f') \cdot \exp(j2\pi f' m / N) \end{aligned}$$

The last term of Equation (3.8) implies that the de-spreading of the signal is equivalent to calculating the Inverse Discrete Fourier Transform (IDFT) of the product of the DFT of the

received signal and the locally generated PN code. The equivalent circuit of Equation (3.8) is depicted in Figure 21.

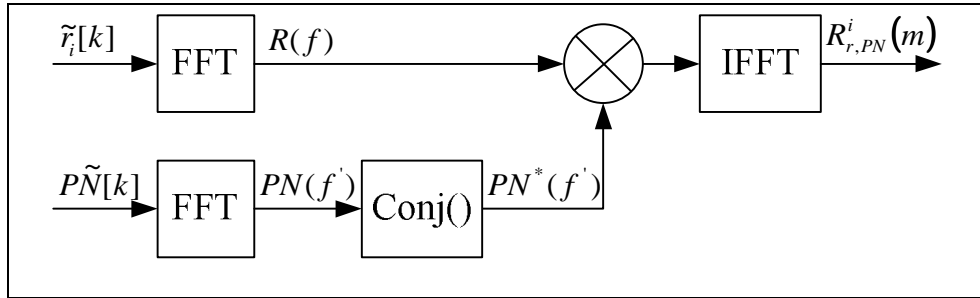


Figure 21: De-spreading circuit in frequency domain

It is essential to realize that the FFT method complexity compared to the traditional correlation method crucially depends on the length of the two involved signals (number of samples, N). For signals with less than 60 samples, the correlation complexity is proportional to the signal length (Smith 1997). However, for larger number of samples it is in the order of N^2 , $O(N^2)$. The FFT method complexity remains as $O(N\log_2 N)$ for any signal length (Smith 1997).

3.3. Tracking Loops

The code and frequency offsets obtained through the acquisition procedure are time variant. Even for stationary applications, the clock drift in the transmitter and receiver continuously changes these offsets. While using highly stable clocks can slow down the changes, in single transmitter-receiver applications clock drift effects cannot be eliminated. In non-stationary applications the variations can be much higher and less predictable. To overcome this problem, two loops have to be employed to track the code and frequency changes.

3.3.1. Frequency Tracking Loop

To keep track of the Doppler frequency changes, a replica carrier signal is generated in the receiver. The frequency of the replica carrier is first initialized by the result of the acquisition procedure and then a loop is required to track the received signal frequency changes. Phase Locked Loops (PLL) and Frequency Locked Loops (FLL) are two main methods for this objective. In a PLL, the phase error between the replica and received signals is used to adjust the replica carrier, while in a FLL, the frequency error between them is employed to tune the replica carrier. Hence, there is no concern about the phase mismatch between the replica and the received signal in the FLL. This indicates that a constant phase offset might exist between them.

In spread spectrum receivers, and in specific in IS-95 pilot channel receivers, one has to decide to use FLL or PLL based on the specific application. To track the phase variation precisely and to obtain low noise measurements, a combination of a PLL and a narrow bandwidth filter loop has to be employed. In contrast, to tolerate high dynamics, a combination of an FLL and a wider bandwidth filter loop has to be used (Cheng et al 2007).

The main limitation of the PLL is its sensitivity to the data transition. However, the IS-95 pilot channel is data less, thus both PLL and FLL can be used in the carrier tracking loop. In either case, the Frequency Tracking loop uses a Numerically Controlled Oscillator (NCO).

3.3.1.1. Numerically Controlled Oscillator

The NCO is a key component of a digital communication system and can be utilized in a variety of applications such as up/down conversion and modulator/demodulator (Xilinx LogiCore 2008). The NCO is a desirable component in a tracking loop since it can generate a complex or real values of sine and cosine. The block diagram of an NCO is shown in Figure 22.

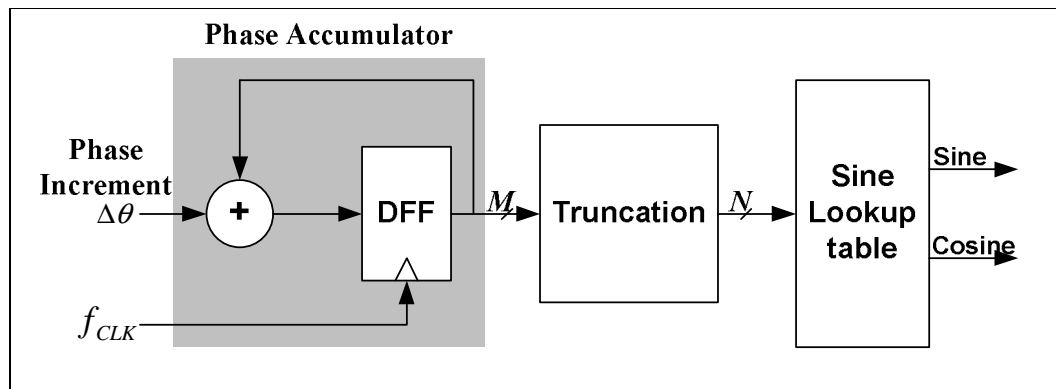


Figure 22: NCO block diagram (Xilinx 2004)

The symmetric nature of the sine and cosine makes it possible to use a lookup table (LUT) for their generation. The LUT is filled with the uniformly spaced samples of a sine quadrant cycle (range of 0° to 90°). Different frequencies can be generated by controlling the address to this LUT – the Phase Increment shown in Figure 22. In the following paragraph, the relation between the phase increment and the output frequency is extracted.

Figure 23 demonstrates the dependency of the sine wave's magnitude and phase to the time interval δt . Although the magnitude is a nonlinear function of the time interval, the phase changes linearly as:

$$\Delta\theta = \omega \delta t \quad (3.9)$$

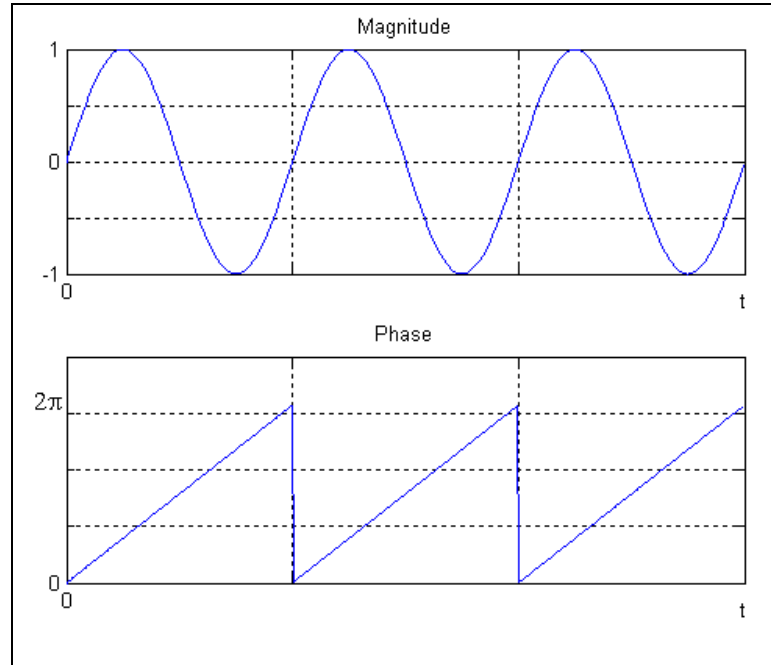


Figure 23: Sine wave magnitude and phase (Analog Device 1996)

The time interval can be interpreted as the inverse of the clock frequency $\delta t = \frac{1}{f_{CLK}}$. After substituting the angular rate¹ and the time interval into the Equation (3.9), the former can be written as

$$\Delta\theta = \frac{2\pi f}{f_{CLK}}. \quad (3.10)$$

Thus, the output frequency is calculated as

$$f = \frac{f_{CLK}\Delta\theta}{2\pi} \text{ Hz}. \quad (3.11)$$

¹ $\omega = 2\pi f$

In the digital implementation, the phase range of $[0 \ 2\pi]$ can be represented by the N -bit resolution phase accumulator (+truncation module in Figure 22). Hence the Equation (3.11) can be re-written as

$$f = \frac{f_{CLK}\Delta\theta}{2^N} \text{ Hz.} \quad (3.12)$$

3.3.2. Code Tracking Loop

As explained in Chapter 2, the receiver position can be found by determining the time delay difference between the received signal (pilot channel) and the locally generated signal. The more precise this delay measure is, the more accurate the resulting position will be. To measure this delay, a Delay Locked Loop (DLL) circuit, which is able to estimate the delay of the received signal, is used. Another duty of the DLL is to track the time variation of this delay, which is why the DLL is known as the tracking loop.

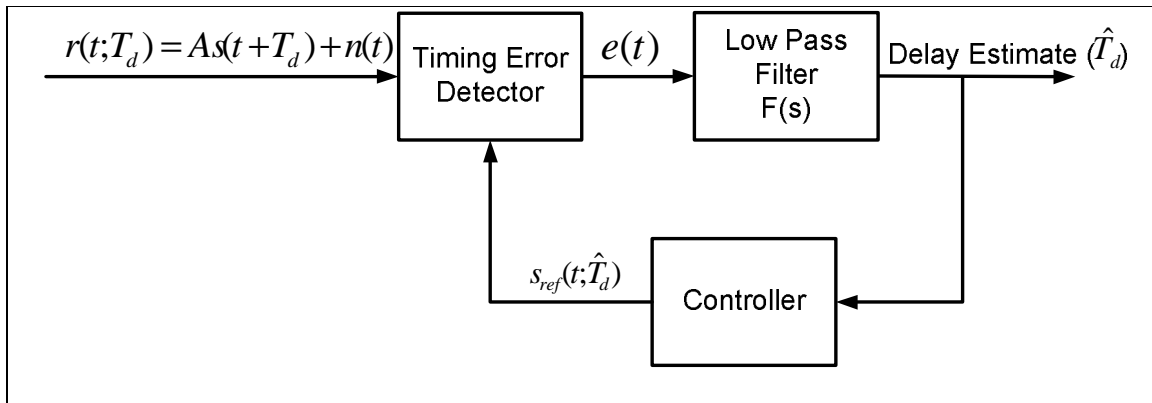


Figure 24: General tracking loop structure (Meyr et al 1998)

In general, a tracking loop, which is shown in Figure 24, is a non-linear circuit that generates a periodic reference signal and attempts to synchronize it with the incoming signal. The loop consists of three main modules:

1. A Timing Error Detector module, which is a nonlinear circuit that compares the reference signal, $s_{ref}(t; \hat{T}_d)$ with the incoming signal $r(t; T_d)$ and generates the error signal $e(t)$.
2. A Low Pass Filter module with frequency response $F(s)$, which is used to filter the error signal.
3. A Controller module, which adjusts the reference signal based on its input so that the error signal between the incoming and reference signal decreases.

Spilker & Magill (1961) showed that the Optimum DLL can be designed by finding the correlation of the received signal and the derivative of the locally generated signal. The equivalent circuit of the optimum DLL is shown in Figure 25.

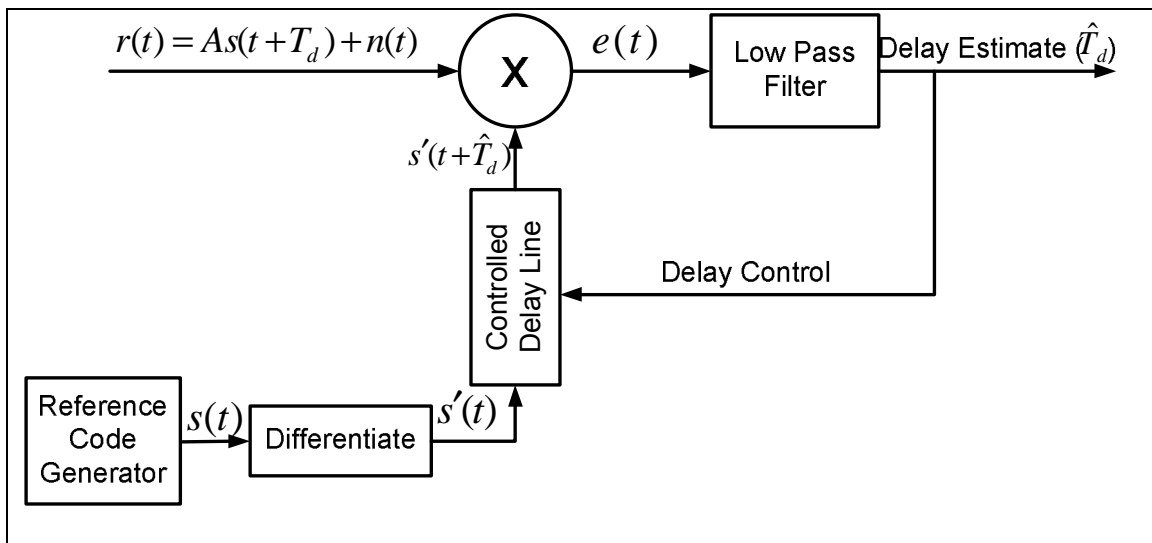


Figure 25: Block diagram of optimum Delay Lock Loop

In general, a DLL is characterized by its well-known S-curve or DLL discriminator characteristics. The S-curve is the expected value of the error signal (the multiplier output). Assuming that the LPF can perfectly filter undesired terms, then the DLL output can be written as (Peterson et al 1995)

$$y_{LPF}(t) = AS'(\varepsilon_a)\varepsilon(t) \quad (3.13)$$

where A is the amplitude of the received signal, $\varepsilon(t)$, is the code phase error and is defined as

$$\varepsilon(t) = \frac{T_d(t) - \hat{T}_d(t)}{T_c} \quad (3.14)$$

and $S'(\varepsilon_a)$ is the derivative of the S-curve at its stable point of ε_a . The linear dependency of the tracking loop output and the code phase error in Equation (3.13) indicates that it can be used to adjust for this error.

3.3.2.1. Early-Late gate DLL

The conventional implementation of optimum DLL uses a structure that approximates the derivative of the reference signal. Remember that the derivative of a function $f(t)$ is defined as

$$f'(t) = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t-h)}{2h}. \quad (3.15)$$

When the signal is linear or does not have intensive fluctuation, the following linear form of the derivative with a good approximation can be used:

$$f'(t) \cong \frac{f(t+h) - f(t-h)}{2h}. \quad (3.16)$$

Equation (3.16) demonstrates that the approximation of the optimum DLL is obtained by substituting the derivative of the locally generated signal with two branches. One branch is shifted forward ($t-h$), while the other one is shifted backward ($t+h$) of the locally generated signal. This structure, shown in Figure 26, is called an Early-Late gate DLL.

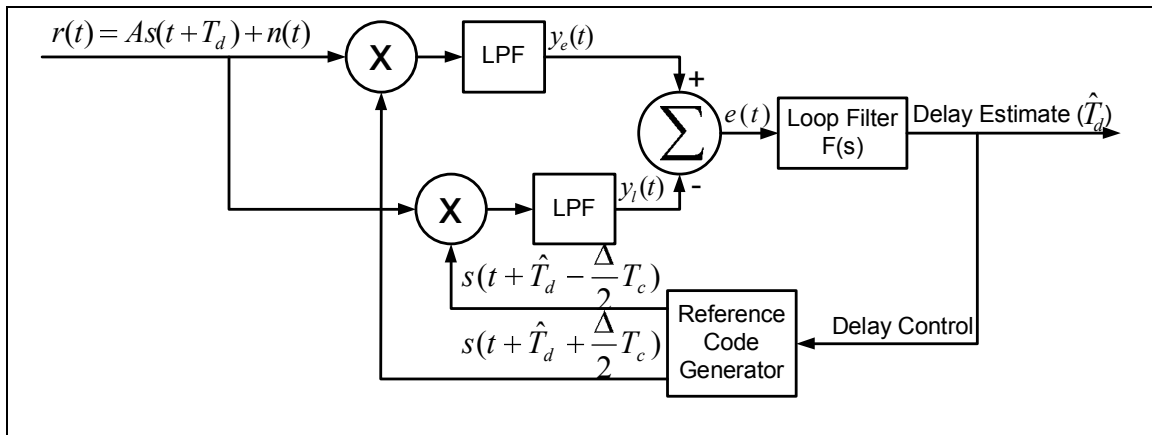


Figure 26: The Early-Late gate delay lock loop

In practice, the parameter h is defined as a function of the chip period $h = \frac{\Delta}{2}T_c$, as shown in Figure 26. The new parameter $\Delta.T_c$ is known as the Early-Late spacing, which denotes the spacing between the early and late branches. Figure 27 shows the effect of the Early-Late spacing on the DLL discriminator characteristic. The larger the $\Delta.T_c$, the greater the linear tracking range. However, for a smaller $\Delta.T_c$, the slope of the S-curve increases. This indicates that a small amount of phase error can generate a greater error in the output of the DLL.

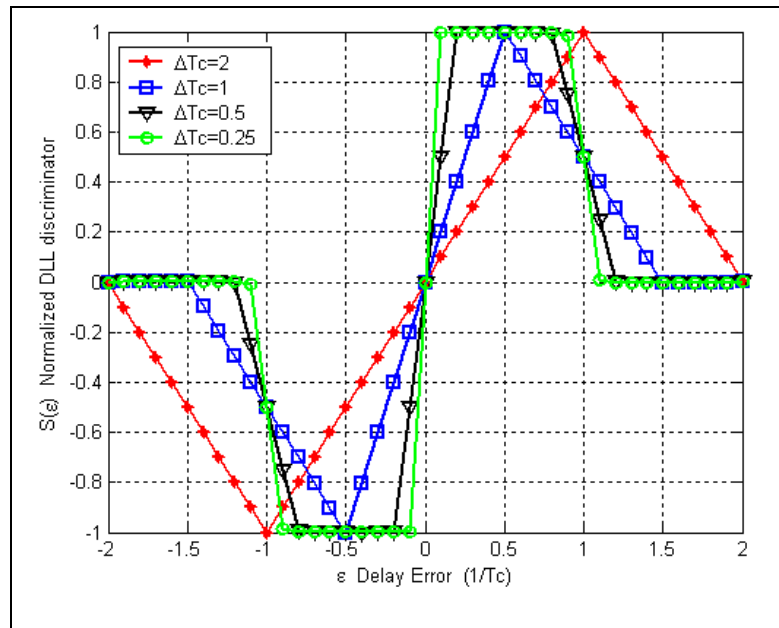


Figure 27: DLL discriminator characteristic for four different Early-Late spacing

3.3.2.2. Performance of the DLL

The performance of a DLL is evaluated using two indicators: RMS tracking jitter¹, and mean time to lose lock (MTLL). To satisfy the normal operation condition of a DLL, the tracking jitter should be low (Wilde & Bernhard 1995). This ensures an acceptable bit error and delay error for data transmission and ranging systems respectively (Wilde & Bernhard 1995). The MTLL specifies the mean time that the DLL stays in its linear tracking range² (Wilde 1996). From the aspect of the signal-to-noise ratio (SNR), the tracking jitter is a criterion for the high SNR, while the MTLL is important when the SNR is in the range of low to moderate (Wilde & Bernhard 1995).

¹ “The variation of the delay error around the stable tracking point \mathcal{E}_a ” (Wilde 1998).

² The region where the S-curve depends linearly (almost linearly) on the error \mathcal{E} ,

3.3.2.3. Non-Coherent DLL

The structure of the DLL that has been discussed up to this point belongs to a category of tracking loops called coherent DLL. A coherent DLL assumes that its input is the pure spreading signal, $s(t)$. This assumption fails in most applications, as it does not account for the carrier frequency, phase and data modulation (Holmes 1982, Peterson et al 1995). The carrier frequency can be ignored if there is a carrier tracking loop before the code tracking loop. To resolve the data modulation problem, a non-coherent DLL, as shown in Figure 28, can be implemented. The square module in the figure removes the effect of the data modulation. However, in the Non-Coherent DLL, the noise floor increases as the noise is also squared.

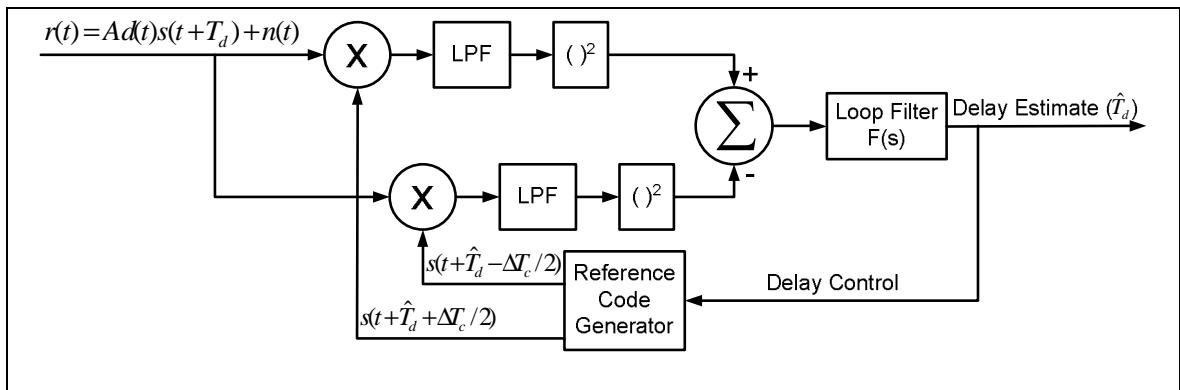


Figure 28: Non-coherent Early-Late gate delay lock loop

3.3.2.4. General Coherent DLL

The Conventional DLL (CDLL), which has been reviewed in the literature (Holmes 1982, Simon et al 1994, Peterson et al 1995, Parkinson & Spilker 1996) uses two correlators to generate the error signal, as illustrated in previous sections. The CDLL suffers from the narrow linear tracking range. This indicates that the DLL can handle small code timing errors before losing lock, implying that each time the loop loses its lock (because of noise

or any other reason), the system has to go through the time consuming acquisition procedure (Wilde & Bernhard 1995).

One approach to increase the linear tracking range, TR_{lin} , is to use more correlators. Increasing the number of correlators enhance the complexity of the implementation and the tracking jitter of the loop. In the following paragraph, the General Coherent Delay Lock Loop (GCDLL) with $2 N_k$ correlators will be explained.

Assume that the output of the i^{th} correlator is weighted by coefficient k_i as shown in Figure 29. In the GCDLL structure, the error signal $e(t)$ is generated by subtracting the weighted output of the multipliers in the Early section from the Late section (Wilde 1998). In the simplest case, the GCDLL characteristic, $S(\varepsilon)$, is assumed to be symmetrical around the origin. This suggests that the spacing of the correlators from the origin and their absolute weights are pair wise identical (Wilde 1998).

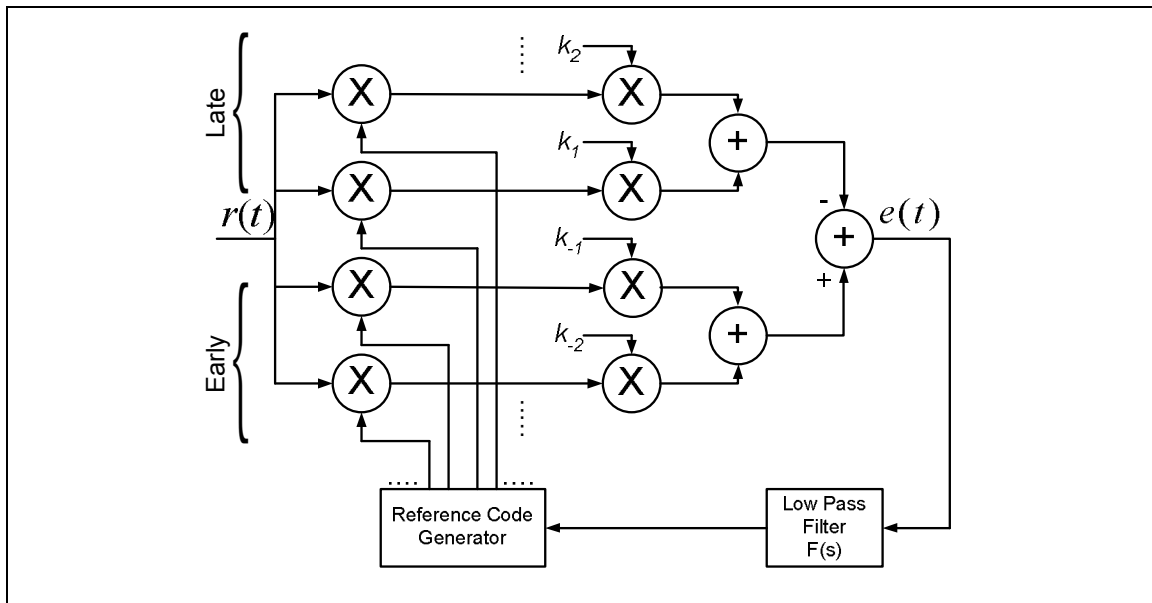


Figure 29: Generalized Coherent Delay Lock Loop (Wilde 1998)

Figure 30 shows the GCDLL discriminator characteristic with four correlators ($N_k=2$) for different outer correlator weights (different k_2). In order to ease the comparison between the GCDLL and the CDLL, the inner correlators are weighted as $k_1 = k_{-1} = 1$. The increase of the linear tracking jitter occurs when $k_2 = 1$ and 2. The tracking jitter is the variance of the phase error $\varepsilon(t)$ and is a proportional to the sum of correlators' weights square as (Wilde 1998)

$$\sigma_{\varepsilon(t)}^2 \propto \sum_{i=-N_k}^{N_k} k_i^2 \tag{3.17}$$

where k_i and N_k are the correlators' weights and the number of correlators respectively.

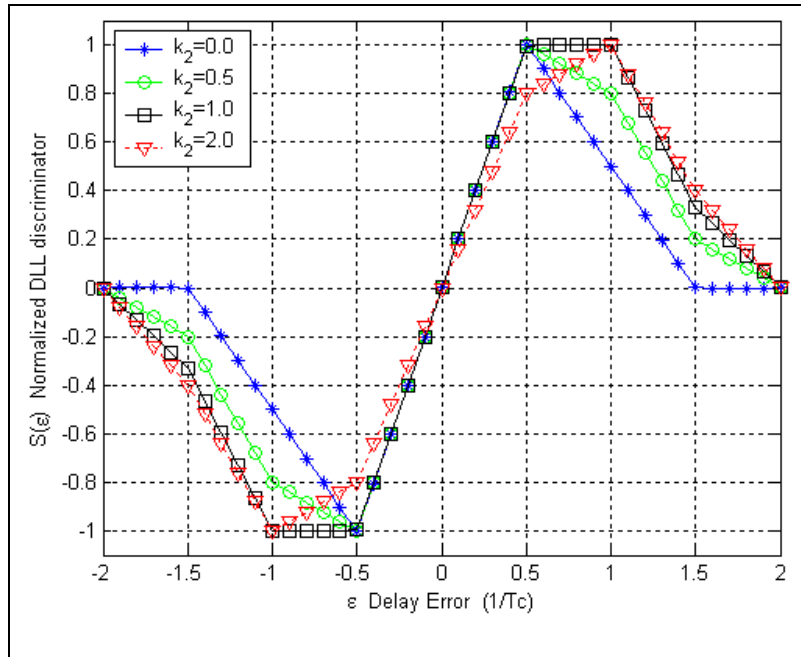


Figure 30: S-curve for GCDLL, $N_k=2$, $k_1 = k_{-1}=1$

3.3.2.5. Tracking Loop Using ACF

Remember that the general idea of the DLL is to generate an error signal, map it to the positive slope of the S-curve and then, by using the closed loop, try to decrease the error signal to zero. Having this in mind, Wilde (1995) suggested using the autocorrelation function instead of generating S-curves. The idealized autocorrelation function of the spreading signal has a positive slope on the left side of the peak which can be used as the

S-curve. It only requires a shift of $\frac{T_c}{2}$ to the right and $\frac{T_c}{2}$ to the bottom so that it passes the

origin. The S-curve for two Early-Late spacing of $\Delta T_c = 2$ and $\Delta T_c = 1$ and the shifted version of the autocorrelation function are shown in Figure 31. Although the slope of the resulting S-curve (from autocorrelation) is the same as the S-curve for $\Delta T_c = 2$, the linear tracking range is halved. Also the ACF S-curve is asymmetrical around the origin.

However, this shows that the autocorrelation function with a very good accuracy can model the S-curve and therefore be used for tracking purposes.

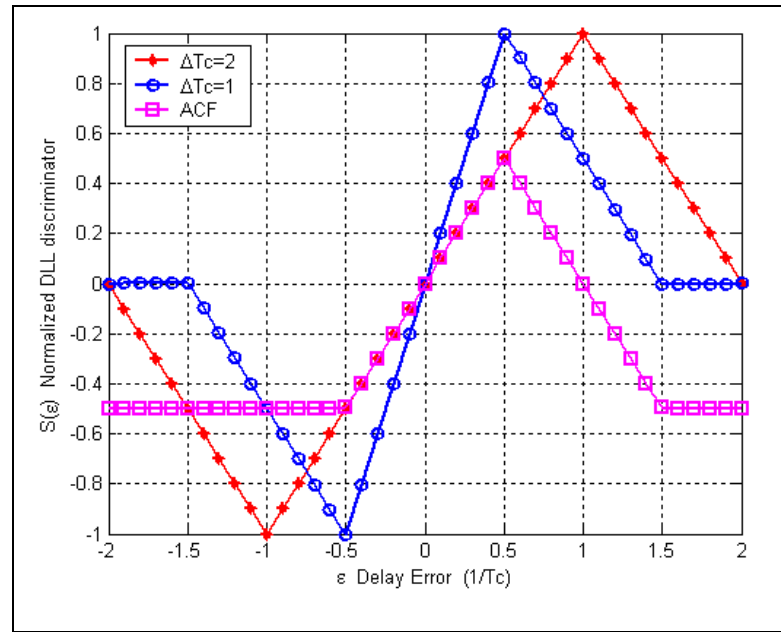


Figure 31: Using ACF to model the S-curve

3.4. Summary

In this chapter, the signal processing procedures including the sampling, detection and tracking of the IS-95 pilot channel were described. It also discussed why the FFT is more advantageous than the correlation when the number of samples involved is large. In other words, in the acquisition procedure, where the signal code phase uncertainty is large (around 1 epoch), the FFT should be used. However, for code tracking, where the uncertainty is around 1 chip, using the correlation method is more reasonable.

Moreover, since the pilot channel is an un-modulated signal, both PLL and FLL can be used for the Doppler frequency tracking. The last section of the chapter was dedicated to

the main methods of the code phase tracking algorithm. It described that the code tracking could be more robust if more correlators were used (e.g. by using the auto-correlation function of the signal). However, increasing the number of correlators increases the complexity of the system – hardware complexity and execution time. These complexities can be ignored if a general-purpose hardware such as a FPGA is employed. In the next chapter, a detailed explanation of the PLAN receiver structure is given.

CHAPTER 4: RECEIVER STRUCTURE

The previous chapter discussed the theoretical concepts of the signal processing in a CDMA pilot channel receiver. This chapter deals with the structure of the specific receiver used to test and verify the signal processing unit developed and implemented herein. As discussed earlier, the main objective of this thesis is to design and implement a real-time signal processing unit for a multi-channel CDMA IS-95 pilot channel receiver. In this work, a multi-channel receiver is essential as it is able to support AOA involving multiple antenna elements. The objective is not to achieve high position accuracy, but rather to build a prototype system comprised of a real-time code and frequency tracking. In such a system, the position accuracy can be improved by using other methods such as hybrid techniques like TOA/AOA and TDOA/AOA (Lu 2006) or space-time super resolution algorithms such as MUSIC (Moghaddam 2007).

The general block diagram of such a multi-channel system is shown in Figure 32. The signals received by the five antennas are passed through five identical paths on the RF board. The RF board accomplished the following tasks: *(i)* filtering, *(ii)* amplification, *(iii)*

down-conversion, and (iv) demodulation. The resultant BB signal is then passed to the signal processing unit. As discussed in Chapter 3, the first stage contains an ADC to digitize analog BB signal. Following this, the digitized signal is passed to the acquisition module, where one of the algorithms mentioned in Chapter 3 (e.g. FFT) is used to acquire the signal. As mentioned in Chapter 2, the direct result of this acquisition is (roughly) the TOA measurement of the signal. Finally, the position of the MS is calculated using one of the estimation methods of Chapter 2 (e.g. LSE) in the Position Solution unit.

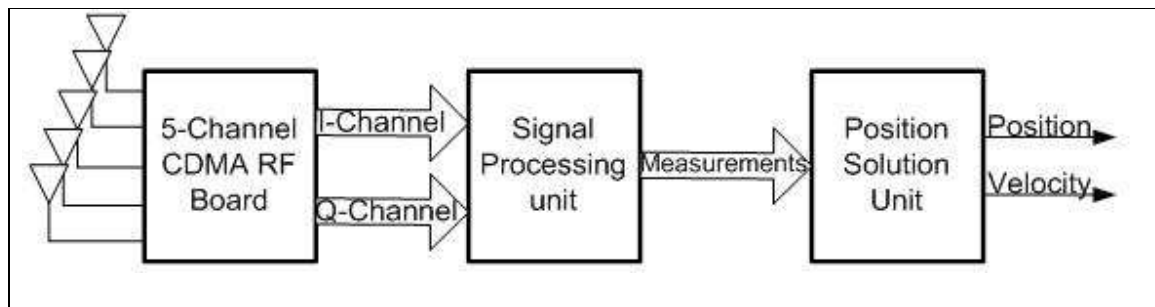


Figure 32: Overall Block Diagram of the Multi-Channel receiver

In order to test and verify the methods and algorithms required in each stage of signal processing, a receiver has been developed in the PLAN group of the Department of Geomatics Engineering. In the following sections, the system prototype of this receiver is explained in detail.

4.1. RF board¹

A Superheterodyne receiver was selected with a two-stage conversion of RF-to-IF and IF-to-BB. Compared to other receiver architectures, such as homodyne, where there is only one conversion stage RF-to-BB, the Superheterodyne receiver has more gain distribution

¹ Details can be found in (Lopez 2006).

and employs inexpensive amplifiers and filters¹ (Lopez 2006). The final architecture of the PLAN group receiver for one channel is depicted in Figure 33, and a brief explanation of its components is provided in the following paragraphs.

The first component is the RF filter, which has the most influence on the total Noise Figure of the receiver. Therefore, in addition to selecting the received band, it also has to have a small noise figure. The next component is a low noise figure amplifier. Since it is not an ideal component, it will also provide some gain to the frequencies around the desired band. This necessitates the use of another filter (similar to the RF filter) after the amplifier.

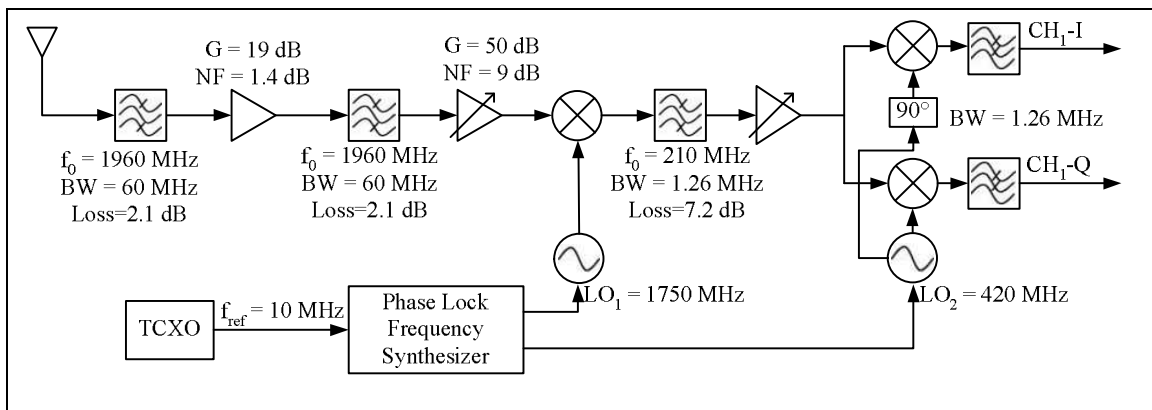


Figure 33: RF unit of the PLAN group receiver (Lopez 2006)

Since the CDMA signal power after amplification can vary by more than 80 dB, an AGC (Automatic Gain Control) is required to prevent the overloading of the successive stages (Lopez 2006). The next component is a mixer to convert the RF frequency to an Intermediate Frequency (IF). To generate the required frequency for the mixer, a 10 MHz

¹ Refer also to section 3.1

Temperature Controlled Crystal Oscillator (TCXO) is used¹. The mixer generates undesired frequencies, which have to be filtered as shown in Figure 34.

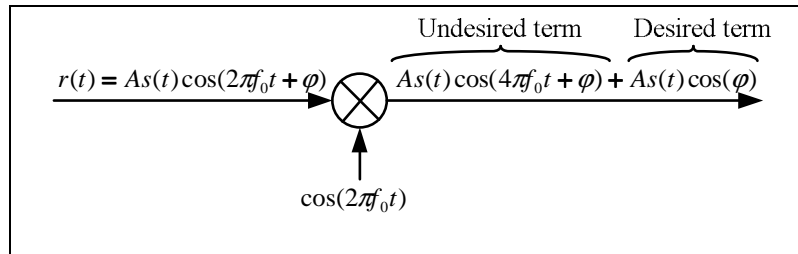


Figure 34: Mixer Operation

The next component consists of a Variable Gain Amplifier (VGA). The last components in the RF board are the demodulator followed by a filter. The demodulator extracts the demodulated signal as in-phase (I) and quadrature phase (Q) from the IF signal and converts the signal to the BB. The RF board is shown in Figure 35.



Figure 35: RF board

¹ More details will be given in section 4.2.4.

4.2. Signal Processing

For the receiver, several options can be used to develop the signal processing unit. In the following sections, these options and their features are discussed. Lastly, the system employed herein, which is called the Phase-II receiver, is explained.

4.2.1. Phase-I Receiver

A straightforward approach is to develop a software receiver that implements the entire signal processing in software. For such a receiver, the five digitized I and Q pairs are multiplexed and sent to a PC. Figure 36 shows the Digital board used for this purpose; it includes several stages: *(i)* a VGA stage to avoid saturation of the ADCs, *(ii)* an ADC stage to convert the analog signal to a 8-bit digital signal, *(iii)* an ALTERA development board to generate the required signals for the ADCs and VGAs¹, as well as to develop a 5-to-1 Parallel-to-Serial circuit that serializes the 8-bit ADC outputs before sending them to the PC and *(iv)* a TCXO which is responsible for generating the required frequencies and clocks for the RF, IF and BB stages.

¹ For instance, the sampling clock for the ADCs.



Figure 36: PLAN receiver Digital board (Lopez 2006)

The block diagram of the Digital board for channel 1 is shown in Figure 37. The data transmission to the PC is carried out using a National Instrument Data Acquisition Card (NI-DAC). The rest of the Signal Processing unit as well as the Position Solution unit are implemented on the PC.

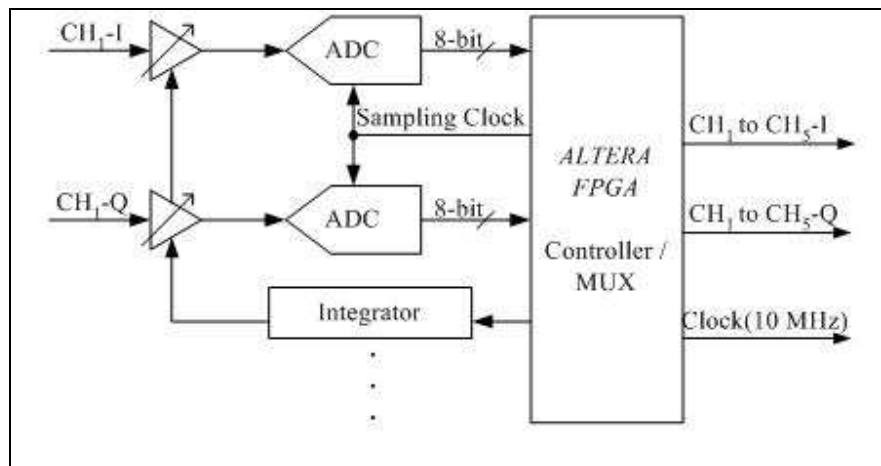


Figure 37: Digital Board block diagram for one channel

There are inherent limitations associated with the software processing; thus, some of these limitations are carried over to the Phase-I receiver. In general, software is slow; therefore,

system designers try to pre-process the high rate data in hardware or firmware. In the case of the Phase-I receiver, it has to support five channels simultaneously, consequently the output sample rate of the Digital board is 12 Mega complex samples per second. This can quickly fill the available RAM storage; furthermore, post-processing of excessively large arrays on the PC is a cumbersome procedure. On the other hand, capturing weak signals and dynamic scenarios requires a large amount of data collection. Therefore, the Phase-I receiver has limited data collection and analysis scenarios.

4.2.2. Gage card-based receiver

An alternative option is to use a multi-channel digitizer to sample five I and Q pairs with a frequency greater than twice the chip rate. As such, the receiver uses the Octopus CompuScope CS8280, a product designed by the GAGE company, that is capable of digitizing up to eight channels with a sampling frequency of 10 MHz per channel. Its on-board memory can store up to 128 MB samples with resolution of 12 bits. Figure 38 shows this single-slot PCI board.

Since this board has eight inputs, it can support up to four CDMA channels, each with two separate physical channels for I and Q. Moreover, the 128 MB on-board memory should be divided between these channels. Assuming 12-bit precision for each sample¹, and a minimum sampling frequency of 2.4576 MHz, a maximum of 17 s of signal can be captured in the single channel mode. This decreases to 4 s of signal in the 4-channel mode.

¹ Total 24 bits for each sample of I and Q.

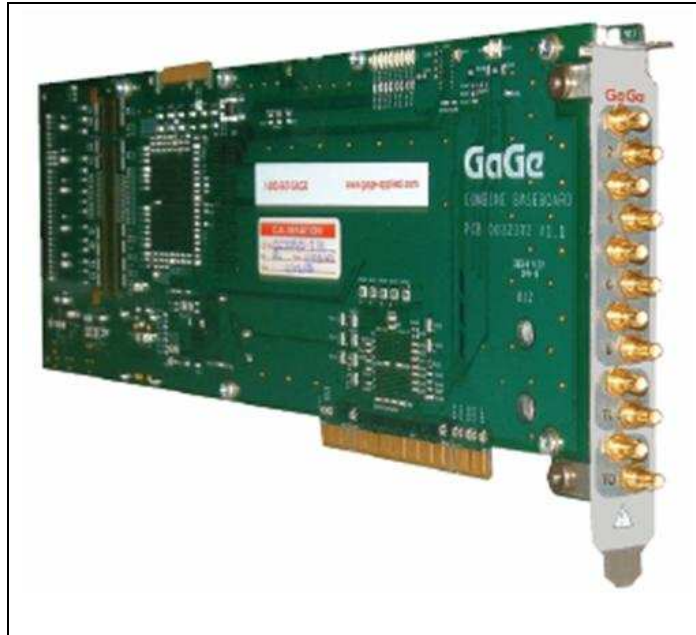


Figure 38: Octopus CompuScope CS8280

The above analysis reveals that the real-time frequency tracking is not applicable on the PC or on the Gage card. Therefore, the correlation peak attenuation, resulting from the uncompensated frequency error, severely limits the maximum coherent integration time (Moghadam 2007).

4.2.3. Phase-II Receiver

As mentioned in Section 4.2.1, pre-processing is essential to obtain low rate data suitable for the software processing. However, pre-processing is not feasible given the inherent limitation of the GAGE card (i.e. its features) and the ALTERA FPGA (i.e. its lack of resources). To increase the speed of the design, an evaluation board is employed. As such, the ML310 Evaluation Development board from Xilinx was selected. There were three deciding factors that contributed to this choice. First, this board uses a platform FPGA (Virtex-II PRO) with many resources and features. For instance, it has a PowerPC CPU,

which can be used to develop the software algorithms currently running on the PC. Second, the board has useful connectors such as the external clock connector, and high speed connectors. These connectors provide several options for the connectivity and compatibility to the other boards like the RF board, Local Synthesizer board and the PC. Third, this board is economically priced for such a project.

4.2.3.1. ML310 Evaluation Development Board

Figure 39 shows the Xilinx ML310 board, where the components used in the receiver are labeled:

1. Virtex-II PRO FPGA: it accommodates the high computationally expensive tasks of signal processing unit.
2. 2.5 V external clock connector: it is used to connect a designed TCXO board for generating the required mixer, demodulator and FPGA frequencies. This board is used instead of the 100 MHz on-board oscillator because it can provide a more stable clock.
3. High-speed connectors of the board: These connectors, known as *Personality Module* (PM), are connected to the high-speed I/O signals on the FPGA. The PM units can also support differential pairs and shielding ground as well as power and ground connections. These connectors are used for transferring the digitized data from the Digital board to the ML310. The rate of the data passing through PM connectors is 12.288 MHz – the range of the PM connectors.
4. 33MHz/32-bit, 3.3 volt PCI bus: it is connected directly to the FPGA and is used by a USBEE-ZX device to transfer data between the FPGA and PC.

- Header J13: outputs some of the FPGA I/Os and used as test points. During the system operation, the user can use these test points to select and investigate any of the internal FPGA signals.

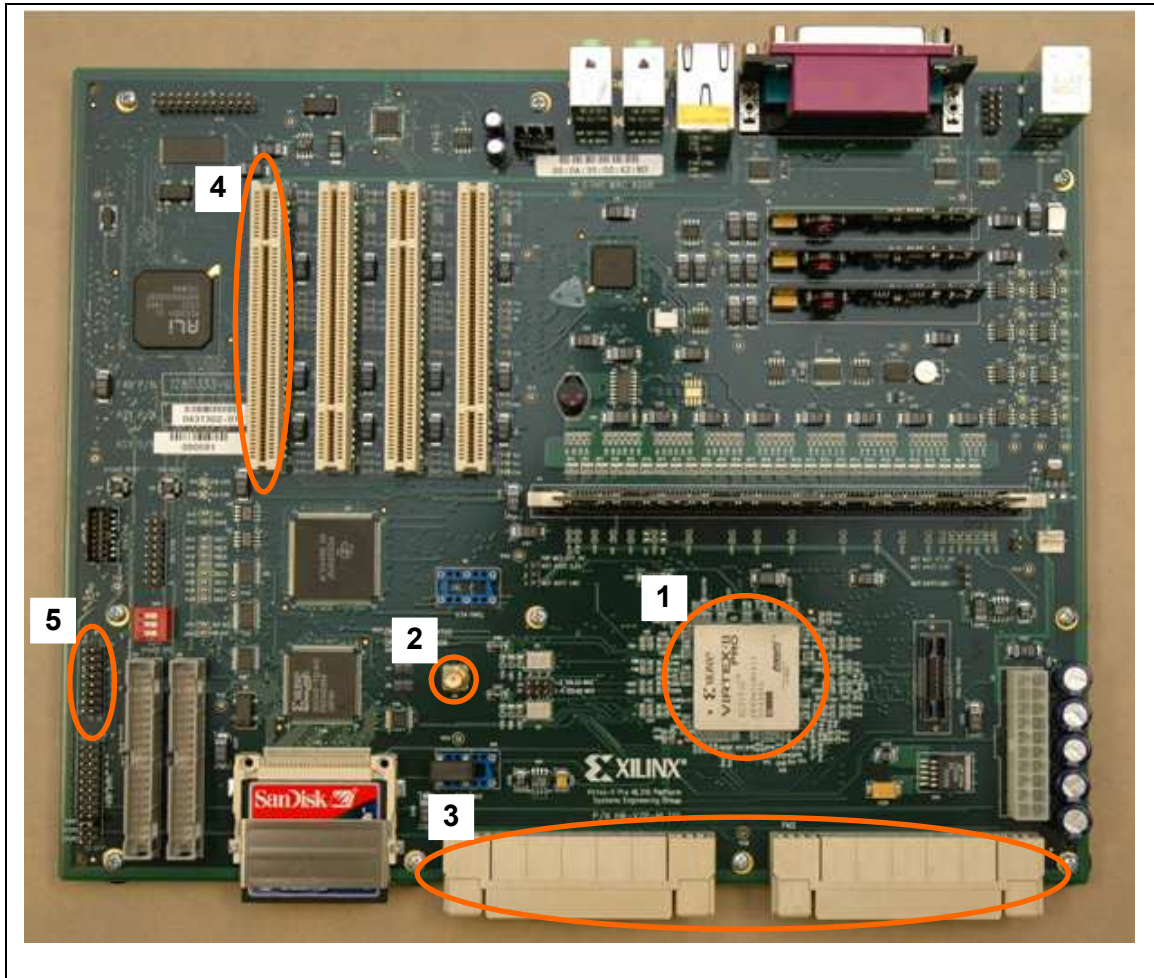


Figure 39: Xilinx ML310 Evaluation development board

4.2.3.2. Virtex-II PRO XC2VP30

The Virtex-II PRO families are platform FPGAs for designs that are based on IP cores and customized modules. The family contains multi-gigabit transceivers and PowerPC CPUs.

They are user-programmable gate arrays consists of configurable elements and embedded blocks. The generic architecture of the Virtex-II PRO family is shown in Figure 40.

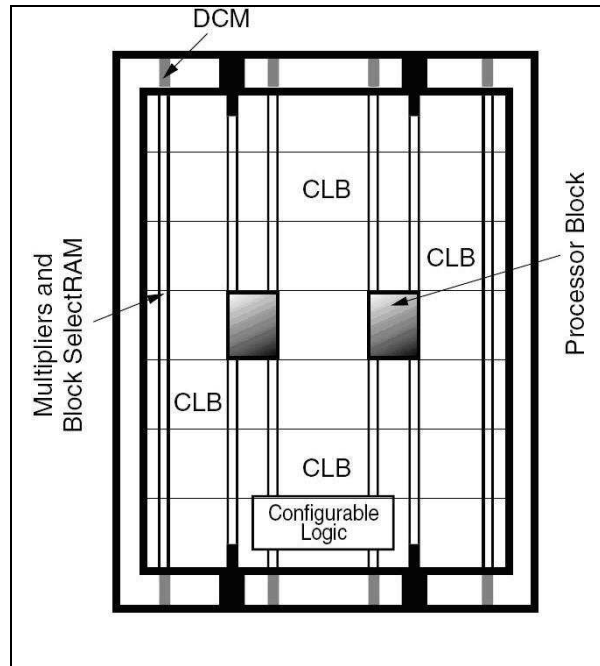


Figure 40: Virtex-II PRO Generic Architecture Overview (Xilinx 2004)

Table 1 summarizes the important specifications of the XC2VP30 from the Virtex-II family, which are:

- **Configuration Logic Blocks (CLBs):** These are the functional elements for combinatorial and synchronous logics including storage elements. Each CLB includes four slices and two tri-state buffers. The individual slices contain two function generators that are configurable as 4-input LUTs, 16-bit shift registers, or 16-bit distributed SelectRAM+ memory. Also, they include two storage elements, an arithmetic logic gate, large Multiplexers, a wide function capability, a fast carry look-ahead chain, and a horizontal cascade chain (OR gate).

- **SelectRAM+ Memory** Blocks: These provide 18 Kb Dual-Port RAMs, programmable from 16K×1 bit to 512×36 bit RAM.
- **Embedded Multiplier** Blocks: These are 18-bit×18-bit 2's-complement signed dedicated multipliers.

Table 1: XC2VP30 resources

Logic Cells (= 4-input LUT+1-FF+Carry Logic)		30,816
CLB (1=4 slices = max 128 bits)	Slices	13,696
	Max Distributed RAM (Kb)	428
18×18 Bit Multiplier Blocks		136
Block SelectRAM+	18 Kb Blocks	136
	Max Block RAM (kb)	2,448
DCMs		8
Maximum User I/O Pads		644

- **Digital Clock Manager (DCM)** Blocks: provide self-calibrating, fully digital solutions for clock distribution delay compensation, clock multiplication and division, and coarse- and fine-grained clock phase shifting.

4.2.3.3. New-version of Digital board

Since the voltage of the PM connectors is not compatible with the output voltage of the Digital board, a Logic Level Translator (LLT) is added to the Digital board. The function of the LLT is to convert a CMOS signal from a voltage level of 5 V to a voltage level of 2.5 V. Figure 41 shows the block diagram of the new Digital board.

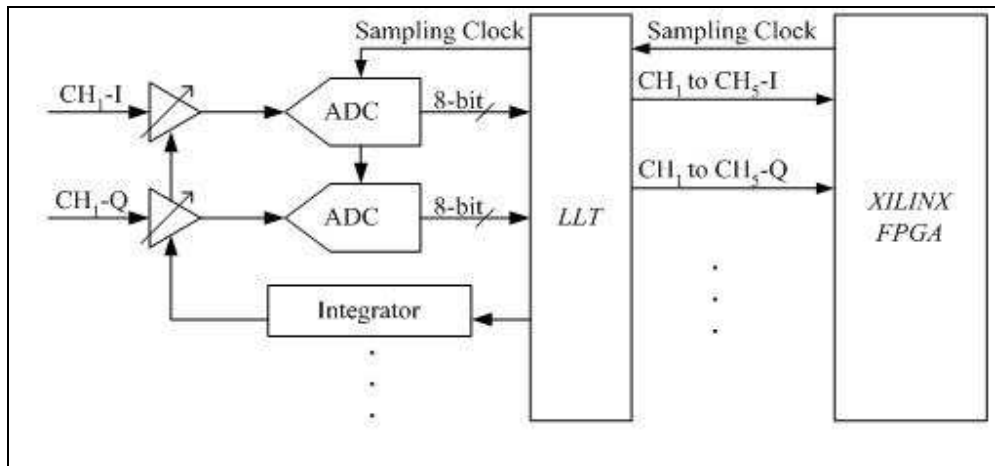


Figure 41: Block diagram of the new Digital board

4.2.3.4. New Version of Firmware

The Virtex-II PRO FPGA's abundant resources allow for the development of a new structure for the signal processing unit. In the new structure, the following tasks are assigned to the firmware:

- Generates the sampling clock for the ADC on the Digital board.
- Generate the control signal for the VGAs on the Digital board.
- Provide the required circuit for the synchronization to GPS time.
- In the acquisition procedure:
 - Select a channel based on the user's settings.
 - Provide the frequency de-rotation to compensate for the Doppler frequency.
 - Transfer the raw data¹ to PC for the acquisition procedure.
- In the frequency/code tracking procedure:
 - Provide the required circuit for the clock drift compensation.

¹ This is a data with rate of 4.9152 MHz.

- Select five channel/BS combinations for tracking procedure based on the user's settings.
 - Calculate the Channel Impulse Response (CIR) for the selected combinations.
 - Transfer the processed data¹ to the PC for further processing of the tracking algorithm.
- Provides the required interface to the PC in order to transfer the raw data (in acquisition) and processed data (in tracking).

4.2.4. Synchronization

As explained in Chapter 2, all BSs are synchronized to the GPS time. Since the receiver has to support all position methods, such as TOA, where BS-MS synchronization is required, a NovAtel GPS receiver is used for timing. It generates a 1 Hz clock² synchronized with the global standard time of Universal Time Coordinated (UTC). Since a much higher frequency is required for the RF and signal processing units, a 10 MHz TCXO is employed. This oscillator generates the required frequencies for the mixer, demodulator and FPGA global clock.

Inside the FPGA, all modules are synchronized with this free running TCXO. The synchronization is accomplished by generating a 2 s clock from the 1 PPS signal of the GPS receiver; thus, an ambiguity remains as to the even or odd second. In the last data

¹ This is a data with rate of 2.4576 MHz.

² It is also called 1 Part Per Second (PPS) clock.

packing stage before the data transmission to the PC, a flag is inserted every 1 epoch of the PN sequence. This flag is reset by the GPS synchronized 2 s clock, the smallest clock interval that can be used for synchronization, as there are an integer number of epochs in the 2 s period. In other words, by using the 2 s clock, the flag reset does not occur in the middle of the epoch. Therefore, depending on the frequency error ratio between the TCXO and the GPS pulse, the last epoch in the 2 s interval can be shorter or longer than the other epochs. This is illustrated in the synchronization block diagram of Figure 42. In this block diagram, the last epoch of data becomes shorter than other epochs, which indicates that the TCXO oscillator is operating slower than its nominal frequency.

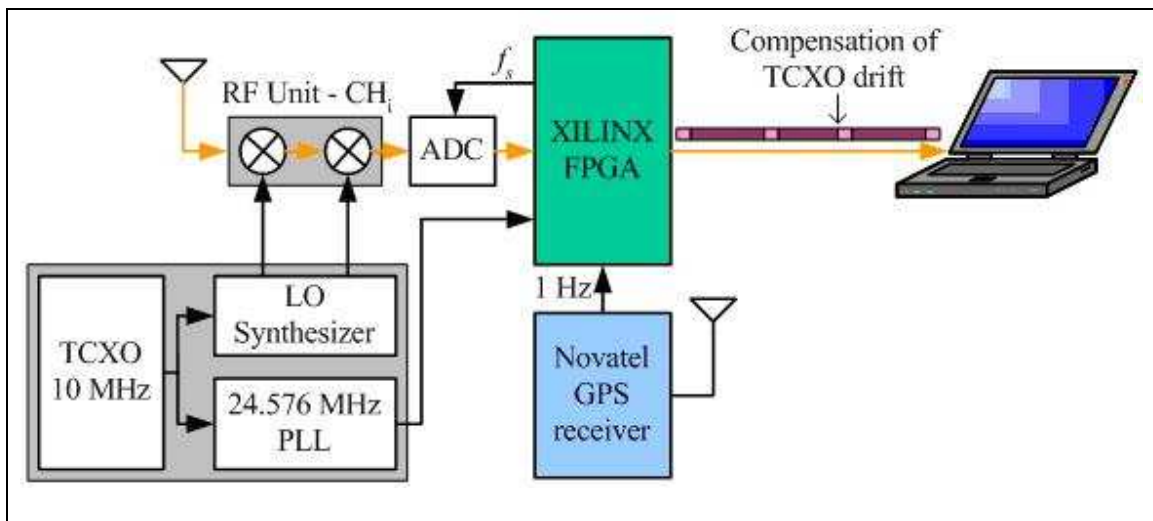


Figure 42: Timing Synchronization Block diagram

The TCXO accuracy is 0.5 ppm. For the IS-95 with the approximate 2 GHz RF carrier frequency – this is equivalent to a 1 KHz error. Figure 43 shows the frequency offset of the received signal during three hours in static mode and demonstrates the long term stability of the TCXO during this period.

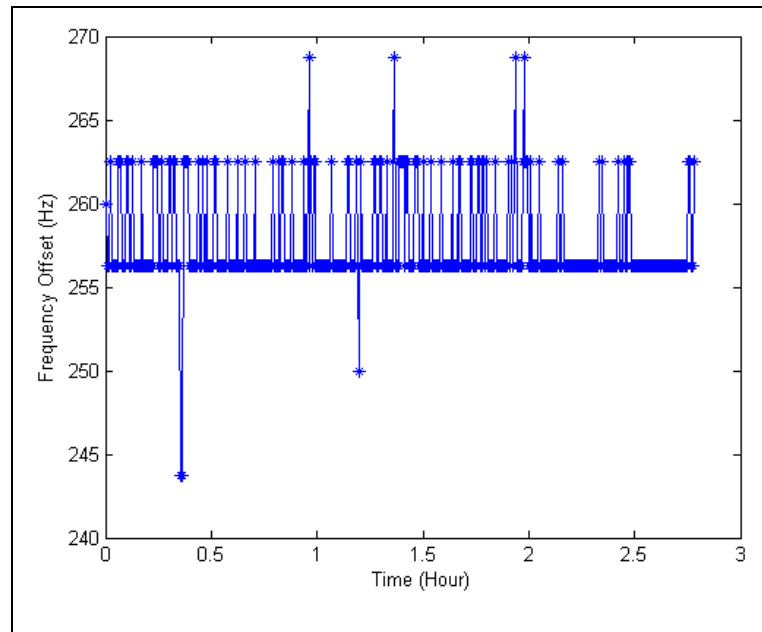


Figure 43: Doppler frequency variation in static mode

The frequency changes observed in Figure 43 are primarily due to small changes in temperature¹. To measure this frequency drift, which is a measure of the stability of the TCXO clock, a statistical metric, namely the Allan variance, is used. Figure 44 depicts the Allan variance of the TCXO. This implies that the TCXO is stable for periods less than 60 seconds, however, for longer time intervals, the lack of stability will cause significant errors.

¹ Other causes can be power supply fluctuation, shock, vibration and aging.

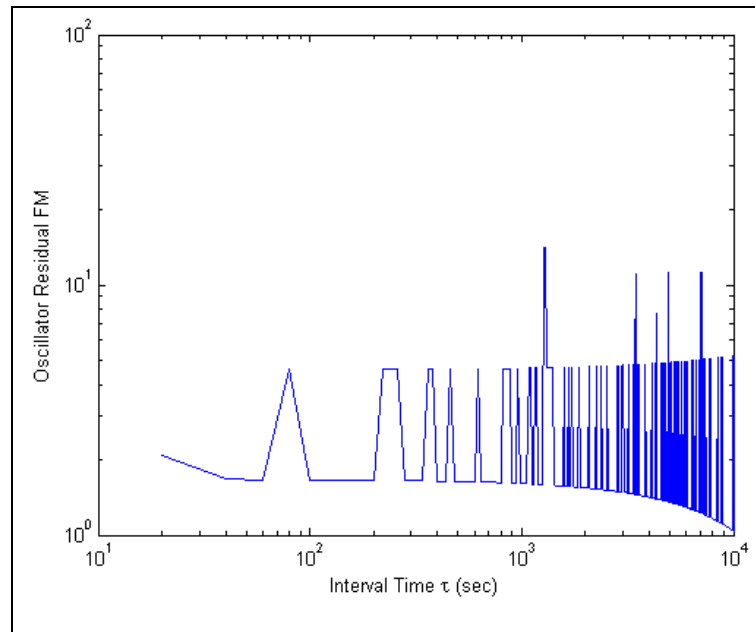


Figure 44: Doppler frequency variation in static mode

4.2.5. Receiver Signal Processing Approach

For the signal processing implementation, two strategies can be applied: (i) sequential processing and (ii) batch processing. In the sequential approach, the signal is processed based on a sample-by-sample basis. In the batch approach, the signal is processed as a batch of data. Van Graas *et al* (2005) define these terms for a GPS receiver and they are applicable for this project. In addition, they compare the performance of the sequential and batch strategies from the signal processing point of view. These two strategies can be considered for both the acquisition and tracking procedures.

In the sequential processing approach, the acquisition and tracking loop are separated. The signal is acquired by searching through the well-known two-dimensional search in sequential steps. One implementation manner is to use a correlator to sequentially search all

code phases for each Doppler frequency. Thus, the sequential acquisition process can be understood as correlation in the time domain. In the sequential tracking process, a correlator is employed (the same correlator can be used for both acquisition and tracking) to correlate the incoming signal with the replica signal sequentially. The code phase of the replica signal is instantiated by the result from the acquisition. The output of the correlator is then used to calculate the tracking error by using a proper discriminator. The output of the discriminator is first filtered by the loop filter and then is used to adjust the replica code.

Conversely, in the batch processing approach, the acquisition and tracking procedures are indistinguishable. In this approach, there are batches of replica signals created over the code phase and Doppler frequency. Using a batch correlator produces a three-dimensional signal image which shows the power of the signal for different code phases and Doppler frequencies. The batch correlator computes the correlation coefficients for all code phases. Thus, it can be realized as the FFT implementation discussed in Chapter 3. In the batch processing approach there is not any closed loop for the tracking procedure.

In accordance with the above discussion, the sequential processing only requires dealing with a few numbers of samples (of the received signal and replica code). Thus, it requires less memory and computational resources. However, as the information of the sequential approach is based on a small number of samples, its resistance to the external errors such as noise spike and strong interference is limited. This feature is known as the observability of the signal. On the other hand, in the batch processing, the number of computations is much

higher than in the sequential processing. This difference comes from the fact that in total the number of required computations is approximately the same for both approaches. However, in sequential processing this computation is spread over time. The main advantage of the batch processing over the sequential processing is its better signal observability. This is caused by the fact that the signal is investigated as a whole and thus external errors can be detected. Therefore, if the memory and the computational resources are not an issue, batch processing can be used to give systems a better signal observability.

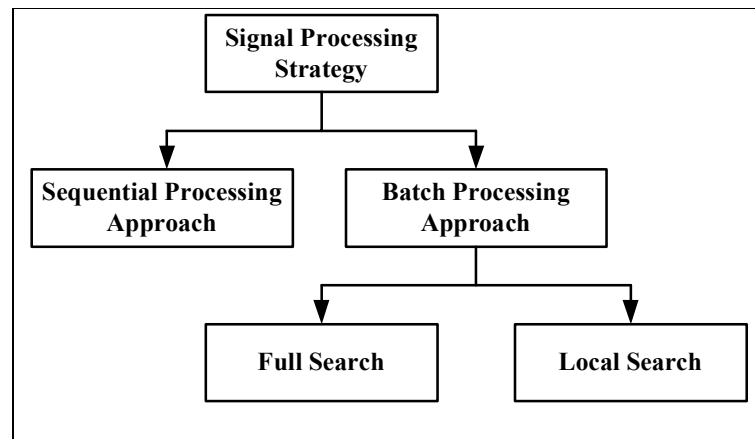


Figure 45: Signal Processing Strategy categories

The batch processing can also be divided into two sub categories: (i) Full search and (ii) Local Search. In the Full search, each data batch is processed independently and there is no feedback from the previous result to the current processing. Conversely, in the Local Search, the result of the previous processing (which can be the first Full Search, or another Local Search) is used to reduce the search space of the current processing. Based on this knowledge, it is reasonable to assume that in the absence of high dynamic (e.g aircraft), it is more efficient to use the Local Search. Figure 45 shows different categories of the signal processing strategies. The features of sequential and batch processing are listed in Table 2.

Table 2: Summary of Sequential and Batch Processing characteristics

	Memory (# Samples)	Signal Observability	Capability of Parallel Processing	Tracking Robustness
Sequential	Two < < Seven	Low	not required	Low/Medium
Batch	Batches	High	required	High

The powerful Virtex-II PRO FPGA used in this thesis offers ample of resources and is capable of parallelism in the receiver; thus, makes it possible to implement the Doppler removal and de-spreading modules (i.e. the batch processing of the code phase tracking). The signal processing begins on the PC by searching through all possible code phases and a wide range of frequency offsets. Then, it is followed by the initialization of the frequency de-rotation (Doppler removal) and replica code generator modules using the acquired frequency offset and code phase, respectively. Afterwards, the code tracking is carried out using a bunch of correlators inside the FPGA to obtain an overall view of the signal code phase. The outputs of the correlators are sent to the PC to provide the required information regarding the next replica code generator update¹. Since the code tracking algorithm is implemented in firmware (parallelism) and provides a total image of the signal and its multipath, it is considered a batch processing design.

Conversely, the Doppler removal module is updated by two separate sequential processing modules: (i) a module inside the FPGA to compensate the frequency error resultant from

¹ More details are given in Chapter 5.

TCXO clock drift, and (ii) a module on the PC to calculate the resultant frequency error from the motion of the receiver. Figure 46 shows the flowchart of the signal processing in the PLAN receiver.

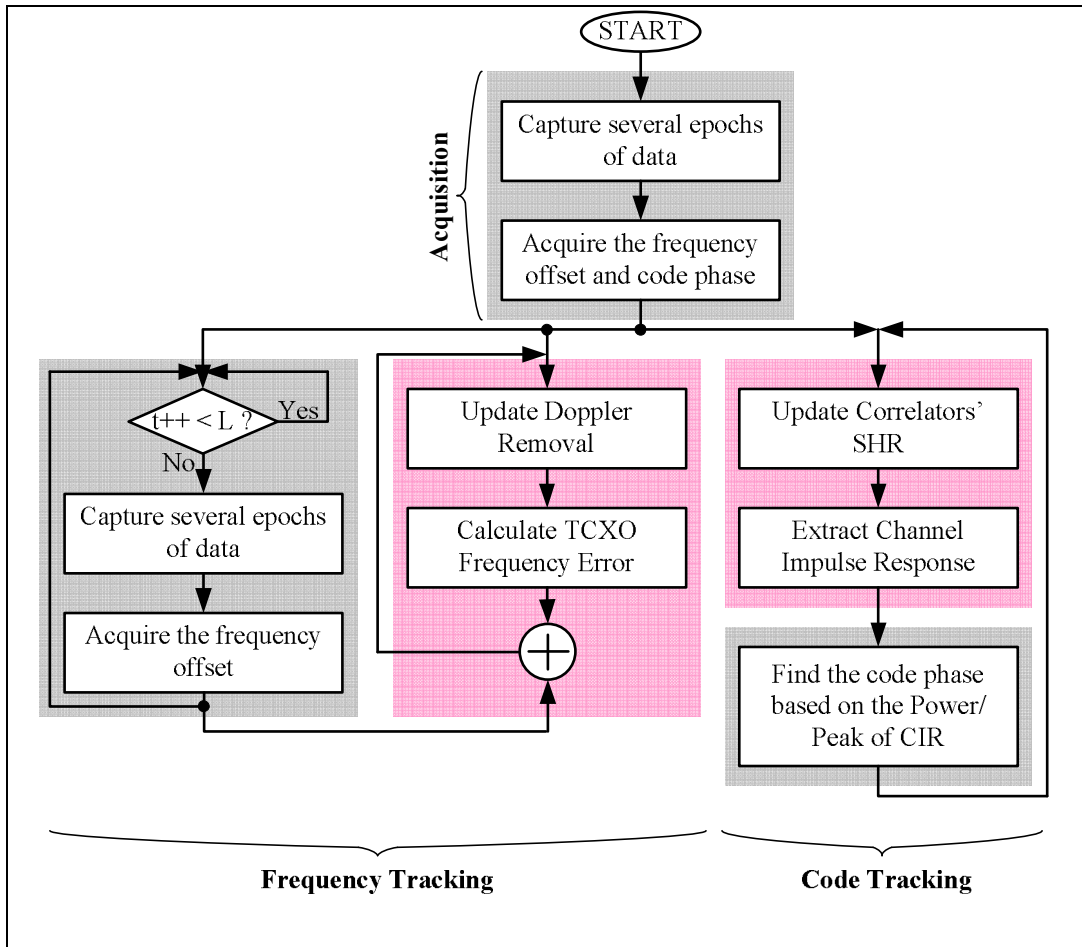


Figure 46: Signal processing flow chart of the PLAN receiver, PC-based tasks (dark boxes), FPGA-based tasks (light boxes)

As it will be explained in Chapter 5, the high computational task of the code tracking is accomplished inside the FPGA and thus it is not a bottleneck for the real-time receiver. However, the frequency tracking procedure is done in the software, and is consequently the most time consuming procedure of the signal processing. The repetition of this search is indicated by the parameter L in Figure 46. Therefore, it is more desirable to compensate for

the frequency error or part of it in firmware to increase parameter L and decrease the load of the PC related to the search repetition. Three sources cause the frequency error in the receiver: (i) the clock drift in transmitter, (ii) the clock drift in the receiver, and (iii) the motion of the receiver.

$$f_{error} = f_{Tx_CLKDRIFT} + f_{Rx_CLKDRIFT} + f_{Rx_MOTION} \quad (4.1)$$

Assuming that the clocks used in the BSs are precise, the frequency error related to this term can be ignored. The frequency error related to the receiver motion can be calculated as

$$f_{Rx_MOTION} = f_c \cdot \frac{v}{c} \quad (4.2)$$

where f_c is the carrier frequency, v is the speed of the receiver and c is the speed of the light. Since the maximum speed of a car is typically 100 Km/h, the frequency error results from this speed is 185 Hz. In Chapter 5, a method is explained to compensate the TCXO frequency drift inside the FPGA. This method can reduce the frequency of the two-dimensional searches and lessening the load of the PC.

4.3. Summary

This chapter covered the history of the PLAN receiver, the limitations of the previous designs and the evaluation of certain criteria that led to the selection a FPGA (Xilinx ML310). The next chapter will give a detailed account of the FPGA design and its results.

CHAPTER 5: RECEIVER SIGNAL PROCESSING IMPLEMENTATION IN FPGA

5.1. Implementation Challenges

The efficient management of resources (processor, memory etc) among different tasks is the main challenge faced by any real time positioning system (satellite-based or ground-based). As mentioned before, the major procedures for such systems are: signal acquisition, the code and frequency tracking and the position estimation. Among these procedures, the Doppler removal and de-spreading modules¹ are the most computationally demanding tasks. In Chapter 2 it was mentioned that most position algorithms require at least three BSs to find the position. However, more BSs are used in real systems to reduce the effect of random noise in the position estimation. In order to take advantage of a large number of BSs, the system has to be able to support the additional data provided by those BSs. For instance, assume a multi-channel receiver for the IS-95 pilot channel capable of supporting 10 BSs per channel with an 8-bit quantization and a minimum Nyquist sampling rate of 2.4576 MHz. To acquire and track 10 BSs per channel, the system has to be able to process

¹ Refer to Chapter 3.

two Giga bits per second¹. This is a substantial amount of data for a regular processor, especially if it has to perform other operations such as executing a position algorithm. Table 3 shows the computational load (Number of operation) of the Doppler removal and de-spreading modules to process 1 ms of data for different sampling rates. The results illustrate the computational load increment as the sampling rate increases. For a minimum sampling rate of 2.4576 MHz, Table 4 depicts the computational load needed to support different number of BSs.

Table 3: Computational load to process 1 epoch of data in a CDMA IS-95 receiver for one BS

	Doppler Removal Unit	Doppler Removal & Correlation Unit	Correlation Unit
Sampling Rate	Sin & Cos generation ²	Multiplication ³	Addition ⁴
2.4576 MHz	131072	524288	262144
2.5 MHz	133333	533333	266667
5 MHz	266667	1066667	533333

Heckler & Garrison (2006) shows that a real-time complex correlation cannot be performed on software GPS receiver using normal integer arithmetic. This is true for the 5-channel IS-95 CDMA receiver as well. In order to have a real-time receiver, 30 correlations (=5-CH×3×2-I/Q) have to be performed every epoch, assuming 3 correlators for Early, Prompt and Late. To track the signals receiving by 5 channels, 1125 correlations per second

¹ 10 (base stations) × 5 (channels) × 2 (I/Q) × 2 × 1.2288 × 10⁶ (Nyquist rate) × 8 (bits) = 1.96608 Gb/s

² (#data in 1ms) × (#CH-BS) × 2 (Sin/Cos)

³ (#data in 1ms) × (#CH-BS) × [4 (Doppler Removal) + 4 (Correlation) * 3 (Early/Prompt/Late)]

⁴ (#data in 1ms) × (#CH-BS) × [2 (Doppler Removal) + 3 (Correlation) * 3 (Early/Prompt/Late)]

have to be completed. To make this analysis more realistic, it is assumed that only 50% of the CPU time is available. This means that in total 2250 correlations have to be completed per second. A regular desktop computer with a 2.0 GHz processor is able to work at rate of 2×10^9 clock cycles per second which indicates that this processor can allocate 88889 clock cycles per correlation.

However, a Pentium 4 processor can perform an addition in 1 clock cycle and a multiplication in 14 clock cycles. Assuming minimum sampling rate of 2.4576 MHz, it can be calculated that 7602176 clock cycles is needed which is much larger than 88889 the limited number of clock cycles that the processor can provide.

Table 4: Computational load of processing 1 epoch of data, $F_s = 2.4576$ MHz

	Doppler Removal Unit	Doppler Removal & Correlation Unit	Correlation Unit
# of BSs	Sin & Cos generation	Multiplication	Addition
2	262144	1048576	524288
5	655360	2621440	1310720
10	1310720	5242880	2621440

Another problem arises when dealing with weak signals and dynamic platforms, which require large data records (more than 1 epoch of data) and thus, a large RAM. The minimum size of the RAM needed to support different numbers of BSs and the equivalent data length (in second) in a 2-GB RAM are shown in Table 5. Here, the sampling rate is 2.4576 MHz. Although the size of the RAM cannot be a problem in current technology, the

computational load of processing large amount of data is the main challenge encountered by a real-time software receiver.

Table 5: RAM size for 1 sec data and total amount of data in a 2-GB RAM, $F_s = 2.4576$ MHz

# of Base Stations Per Channel	Minimum RAM size (MB)	Number of seconds in a 2 GB RAM (s)
1	24.576	80
5	122.88	16
10	245.76	8

One method to cope with the stated challenges is to divide the signal processing tasks into two groups, namely (i) the high computational load/low complex algorithms and (ii) the intermediate or low computational load/high complex algorithms. The first group is suitable for the firmware implementation, while the second one can be better developed in software.

In this chapter, the signal processing procedure implementation of the PLAN receiver is discussed in detail. The result of each procedure in the static mode is demonstrated and the behavior and the limitation of the system in dynamic mode are explained.

5.2. Overall Design

To decrease the load of the PC in the PLAN receiver, the Doppler removal and de-spreading procedures are implemented in an FPGA. Even though the FPGA has a higher computational capacity than the PC, it still has limited resources – there are a finite number

of logic cells, multipliers and RAMs. More resources can be gained through using a larger FPGA; however, there is a tradeoff between the size of the circuit and the maximum frequency that can be achieved. For a specific FPGA and design, the larger the design, the lower the maximum achievable frequency will be. In this respect, a lower maximum frequency obtains a lower sampling frequency that results in a coarser autocorrelation function and thus, poorer position accuracy.

The goal of the FPGA processing in this project is not the actual computation of the position estimation, but rather to provide a flexible platform capable of supporting different positioning algorithms (or their combination) and to decrease the load placed on the PC so that it can use a high resolution position algorithm such as those proposed by Lu (2007) and Moghaddam (2007). Therefore, the system used in this project is designed as a prototype system with the lowest possible sampling frequency (i.e. 2.4576 MHz) and the smallest number of BS per channel (i.e. 1 BS per channel). Figure 47, as described in the following paragraphs, illustrates the top level of the implemented system configuration.

The required clocks in the system, as mentioned in Chapter 4, are generated by employing a 10 MHz TCXO. To produce the clock for the digital segment of the system (after ADC module), a PLL is used to generate a 24.576 MHz clock. This PLL derives a Digital Clock Manager (DCM) module inside the FPGA. The DCM is used to produce clocks needed for modules inside the FPGA with precise phase relative to each other.

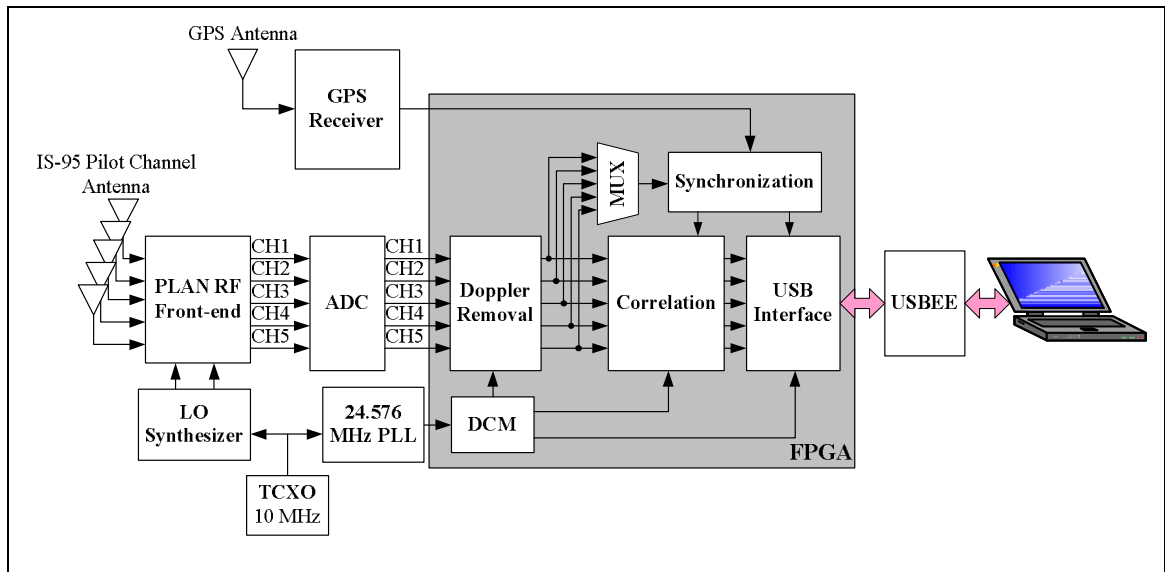


Figure 47: Assembled system

Before being sent to the FPGA, the I and Q signals from five channels are sampled by a 2.4576 MHz clock. Inside the FPGA, the sampled I and Q signals are multiplied by the sine and cosine signals using five complex multipliers. After wiping off the Doppler frequency¹, one of the channels is selected and sent to the Synchronization module, which is responsible for the synchronization of the received signal with GPS time.

In the synchronization module, a 2 s clock synchronized with the 1 Hz GPS clock, is generated. The signal in two different paths have to become synchronized with this clock: (i) the signal sent to the PC, and (ii) the signal in de-spreading module. The first synchronization is accomplished by chopping the signal into data packets of 65536 complex samples. Each data packet starts with a Flag (i.e. a 16-bit known sequence of '0's and '1's). Every 2 s, the position of the Flag in the data stream is modified to compensate

¹ In this thesis, the Doppler frequency is defined as the combined frequency error generated by the (TCXO) clock drift and the receiver motion.

for the TCXO clock drift. These data packets are then transmitted to the PC for further processing. The second synchronization is accomplished by resetting the correlators in the de-spreading module by the 2 s clock.

A USBEE-ZX is used to transfer the data between the FPGA and the PC. It is comprised of ten wires – 1-byte of data, a clock and a trig to indicate the transfer direction (i.e. ‘1’ for ‘from PC to FPGA’ and ‘0’ otherwise). All transmissions are initiated from the PC by driving the clock line. To develop the proper communication, two USB interfaces are designed on the PC and FPGA respectively. The PC sends 1-byte command to inform the FPGA about the type of the data it should send to or receive from the PC. The USB interface on the FPGA side consists of a state machine which receives and responds in accordance to the received PC command. Since PC uses a separate clock, asynchronous to the FPGA clock, there is a prone to metastability failure.

Different methods such as using a flip-flop or a MUX between two asynchronous clock domains can be used to overcome metastability. As shown in Figure 48 (a), the flip-flop method uses an extra flip-flop to register the data by the second clock (CLKB). Hence, the downstream circuit (the circuit with input DB2) always receives the stable signal DB2. The same situation will happen if the circuit in Figure 48 (b) is used.

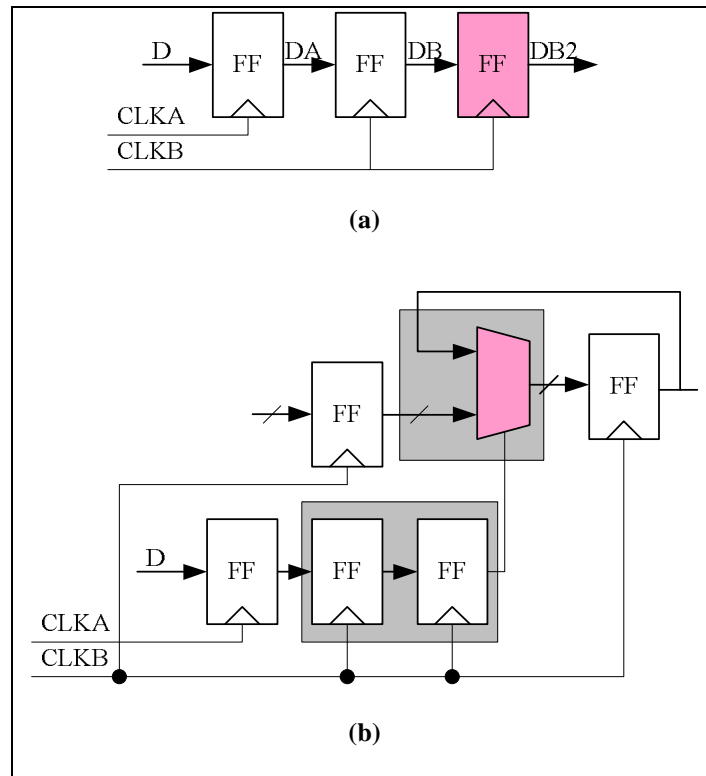


Figure 48: To overcome metastability between two clock domain (a) Flip-Flop (b) MUX

As explained in Chapter 4, the two-dimensional acquisition procedure initiates the code and frequency tracking. The main part of the code tracking is implemented inside the FPGA while the code tracking decision making is a Matlab code running on the PC. The frequency tracking is divided to two segments, where one is related to the TCXO clock drift and is designed inside the FPGA and the other is related to the receiver motion and is executed on the PC every L seconds. The results of these two parts are added together at the input of the NCO inside the FPGA. In order to increase the frequency tracking procedure speed, the frequency error (or Doppler) range related to receiver motion is shrunk in the successive repetitions of the search and thus is referred to as the Local Search¹. However,

¹ Refer to Section 4.2.5

in the case of signal loss, the acquisition is executed with the default (wide) Doppler range.

The paths related to frequency and code tracking are illustrated in Figure 49.

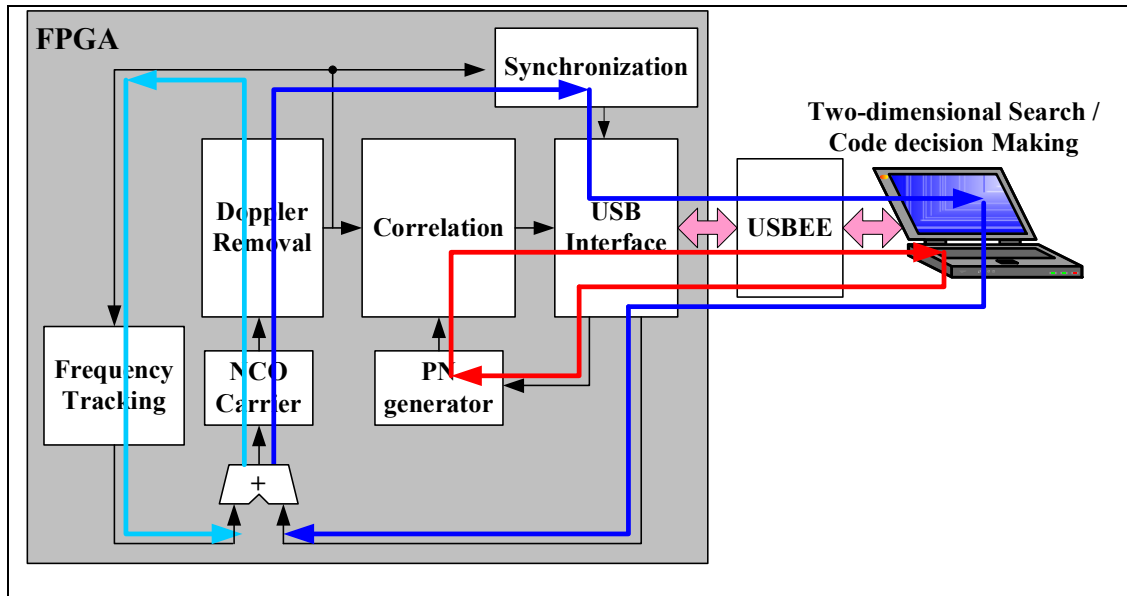


Figure 49: System configuration related to code tracking (Red) and Frequency tracking (Blue)

The following sections provide a description of each module implementation. Section 5.3 goes over the design of the acquisition procedure. The code and frequency tracking are discussed in the section 5.4. The chapter then concludes by explaining the test setup and its results.

5.3. Acquisition

The acquisition procedure is a Matlab code that employs a complex FFT for efficiency. This search procedure has to cover all the code and frequency uncertainty space. The total code uncertainty space depends on the position and time uncertainty (Kaplan & Hegarty 2006):

$$\sigma_{CodePhase}^2 = 4\sigma_{Pos}^2 + \sigma_{CP_Time}^2 \tag{5.1}$$

where σ_{Pos}^2 indicates the uncertainty of the position as a circle with radius of half of the chip (≈ 120 m), and $\sigma_{CP_Time}^2$ indicates the uncertainty of half-chip in the 2^{16} (=65536) half-chip in one epoch of pilot channel. The Doppler uncertainty is proportional to the time, position, user motion (velocity), oscillator uncertainties (Kaplan & Hegarty 2006):

$$\sigma_{Doppler}^2 = \sigma_{Dopp_Time}^2 + \sigma_{Dopp_pos}^2 + \sigma_{Dopp_Vel}^2 + \sigma_{Dopp_Osc}^2 \quad (5.2)$$

Among these factors, the oscillator uncertainty is the dominant term. In the case of the PLAN receiver this term is 1 KHz/ppm. However, in practice and in the static mode, the experimental results show that the frequency range of [-300, 300] Hz is sufficient for the first execution of the acquisition. The user can select the integration time by choosing the number of epochs that participate in the acquisition procedure. Figure 50 and Figure 51 show the acquisition results for different integration times. As expected, increasing the integration time enhances the chance of weak signal detection. For instance, more BSs are detected in Figure 51 that are not clearly observed in Figure 50.

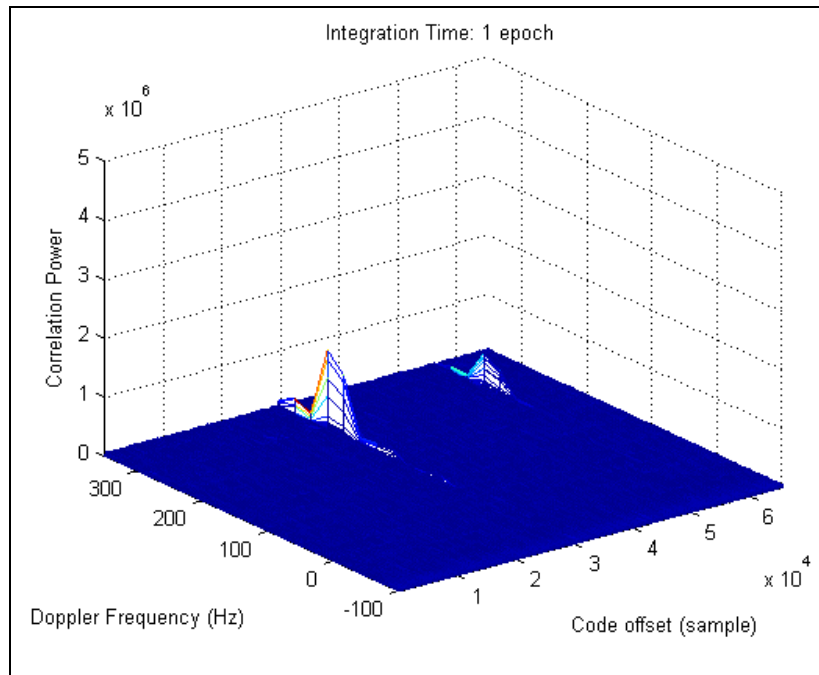


Figure 50: Acquisition Result for 1 epoch (26.7 ms)

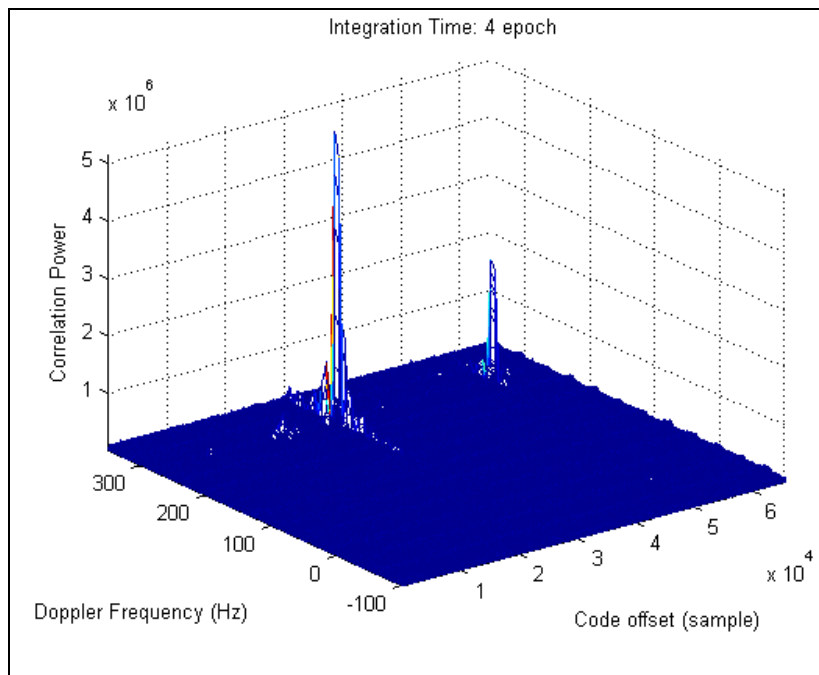


Figure 51: Acquisition Result for 4 epochs (106.7 ms)

5.3.1. Doppler Removal Module

After the first repetition of the acquisition procedure, the Doppler range is narrowed. Since the tests are set up in the static mode, it is anticipated that the Doppler frequency changes are in a small range around the mean. However, some unexpected abrupt decreases in the correlation power are detected that occasionally result in signal loss. To resolve this, the time interval between the repetitions of the acquisition algorithm is decreased; unfortunately, intermittent signal loss still persists. To investigate the problem further, the Doppler search range is fixed to $[-300, 300]$ Hz for consecutive repetition of acquisition. Figure 52 depicts the sudden changes in the Doppler frequency over a 20-hour period.

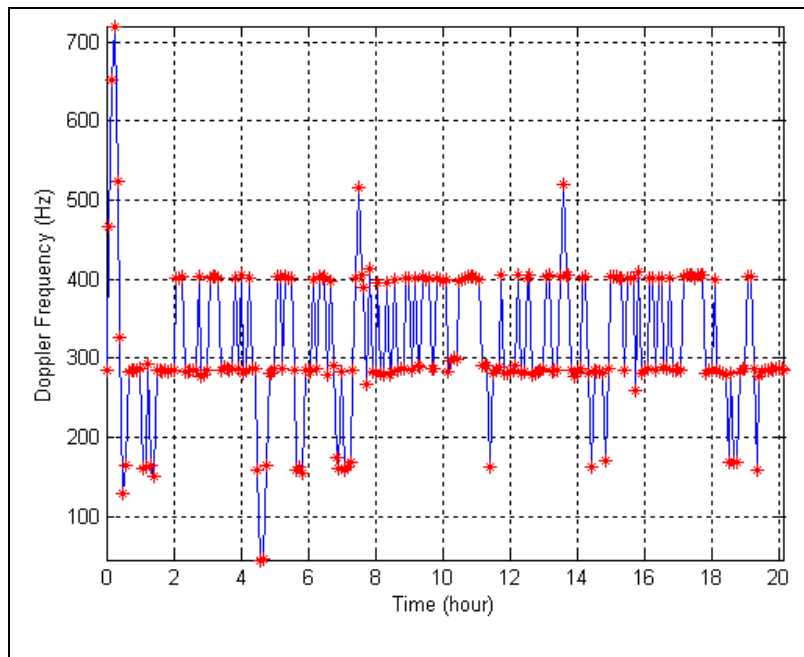


Figure 52: Abrupt Doppler frequency changes

It is assumed that the phenomenon is caused by one of the following reasons:

1. The missing of some I and Q data packets during the time of data transfer between the FPGA and the PC.

2. The malfunction of the Doppler removal module as a result of an incorrect carrier NCO frequency (update/generate).
3. An unknown problem regarding the real data transmitted by BSs¹.

To study the first assumption, a counter is used in the FPGA and its output is sent to the PC instead of the selected channel. A program on the PC is used to check the received data from the FPGA by selecting the first sample as a correct number and comparing the rest of the data with a counter output on the PC. The program is repeated for numerous times to reveal that the transmission cannot be the case. To examine the second assumption, a feature is added to the GUI to read the input phase of the NCO as well as to transfer the NCO output to the PC. Tests show that the NCO is receiving the correct update values and it generates the correct frequency based on its input phase. Figure 53 shows a sample of the NCO output after sending a request for a 10 Hz sine wave. The NCO output is transferred to the PC using a 4.9152 MHz clock. However, the NCO itself is clocked by 2.4576 MHz; therefore, the frequency of the sine in Figure 53 can be calculated by

$$f_{NCO} = \left(\frac{B - A}{4.9152 \text{ MHz} / 2} \right)^{-1} \quad (5.1)$$

where A and B are the x-coordinates of two points on the sine separated by a distance of one period. The calculation of the NCO output frequency, using Equation (5.1) shows that the NCO is properly generating the desired frequency (10 Hz). Repeated testing confirms that the NCO is not responsible for this problem, but some other unaccounted factor.

¹ This is just an implausible assumption since it would bring down the whole network, which is not the case.

The last assumption is investigated using an external PN generator which generates a maximum length PRN15 sequence. The output of this simulator is connected directly to the input of the RF unit. Similar peaks are observed from this examination that reveals the problem is not related to the real data received from BSs, but the receiver itself.

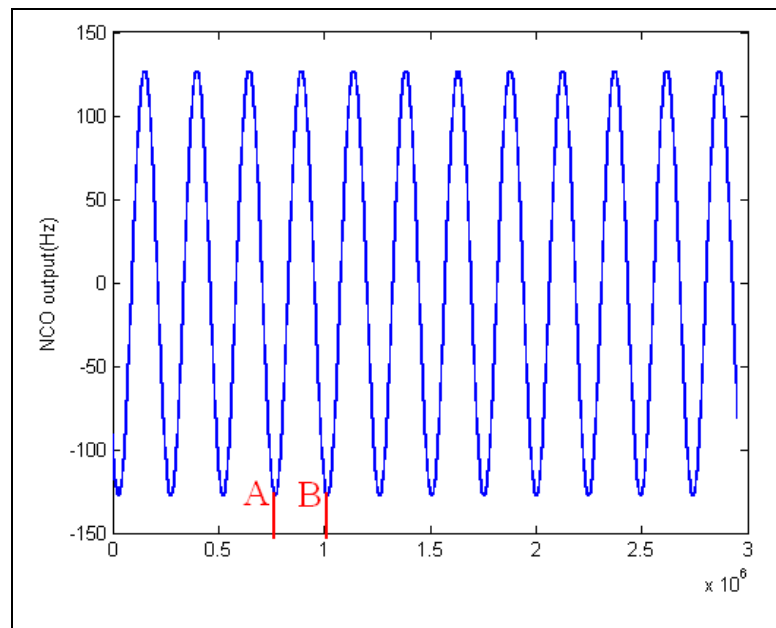


Figure 53: A sample of NCO output which shows a 10 MHz sine signal.

To further analyze the problem, the Doppler range is expanded to $[-1800 \ 1800]$ in the acquisition procedure. Figure 54 shows that after the Doppler range expansion, there are several peaks with approximately the same amplitude in the acquisition result. Figure 55 depicts the two dimensional view of these peaks. Experimental results show that the amplitude of peaks changes over time; therefore, the Doppler frequency procedure locks randomly to one of them at each repetition. The time varying feature of this problem makes it difficult to be captured and solved in a timely manner. However, it demonstrates that the periodic signal loss is related to the time varying module in the system, i.e. the clock unit.

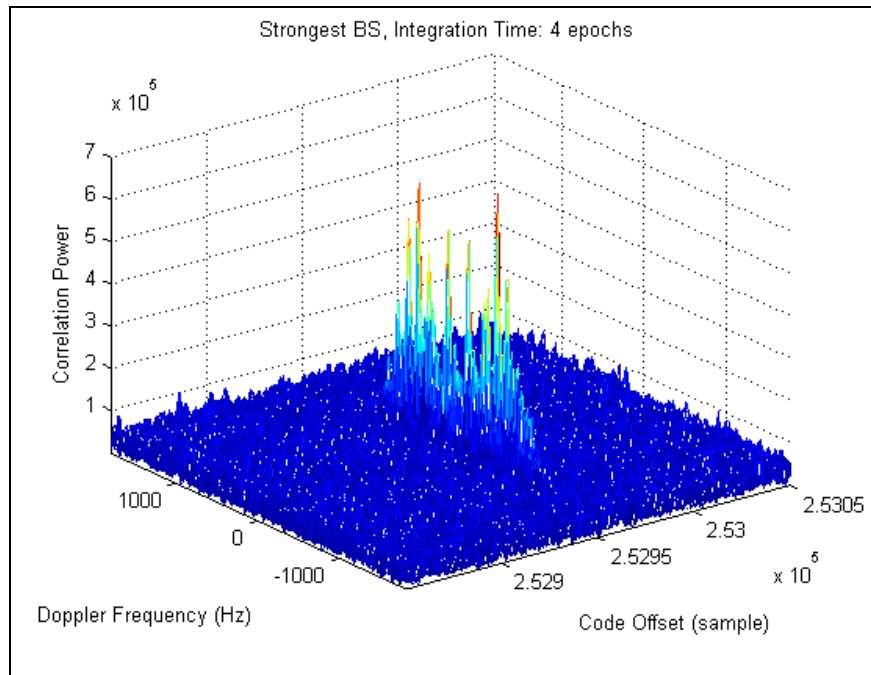


Figure 54: Acquisition result for the strongest BS in the frequency range [-1800 1800] Hz

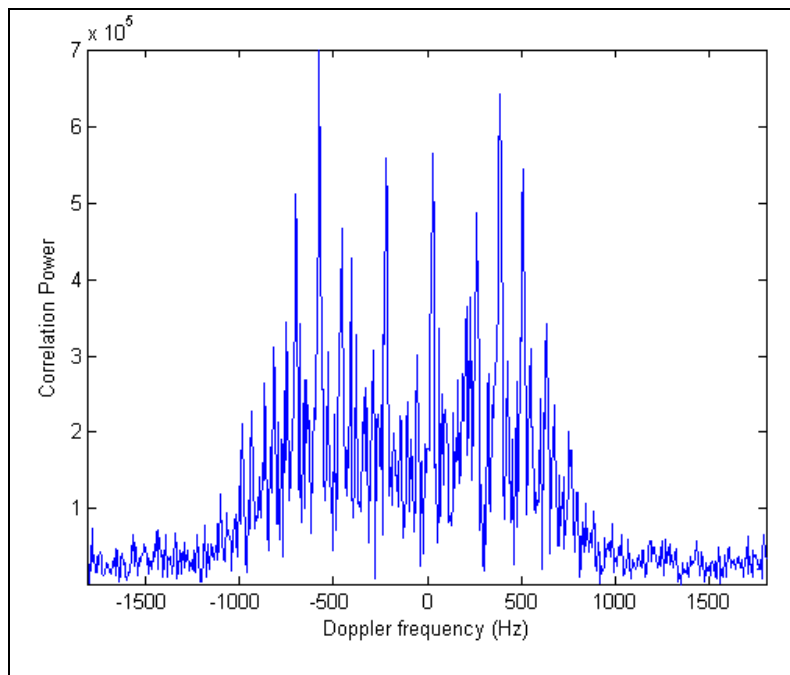


Figure 55: Correlation power of strongest BS for Doppler frequency range [-1800 1800] Hz

To verify the system clock unit, the output of the local oscillator synthesizer is examined using a spectrum analyzer. Figure 56 and Figure 57 show the output of the RF and IF LOs respectively, and indicate the generation of the desired frequencies.

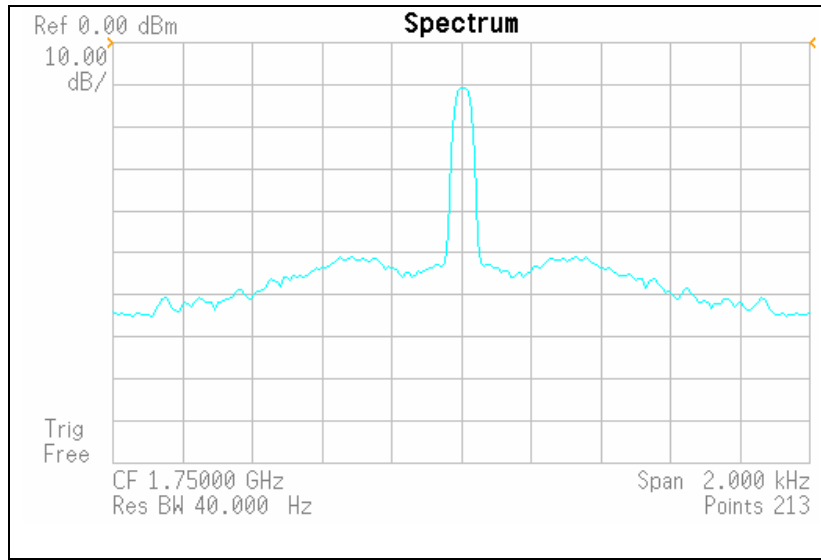


Figure 56: RF LO output (Lopez 2006)

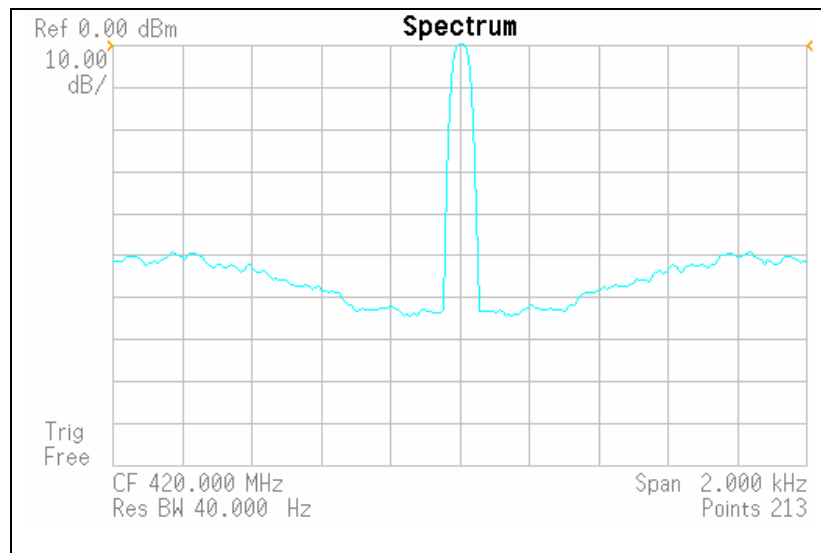


Figure 57: IF LO output (Lopez 2006)

However, it should be noted that the peaks in Figure 55 are only several hundred hertz apart from each other; therefore, even if they exist, they cannot be observed in the scale of LOs' output. To determine if the RF and IF LOs are the cause of the peaks, they are replaced with two signal generators that have properly adjusted amplitudes and frequencies. The resultant acquisition after this replacement is shown in Figure 58. Comparison of the Figure 58 with Figure 54 reveals that undesired frequencies are introduced by the LOs. Therefore, instead of using the LO synthesizer for the RF and IF down-conversion, two signal generators are used henceforth.

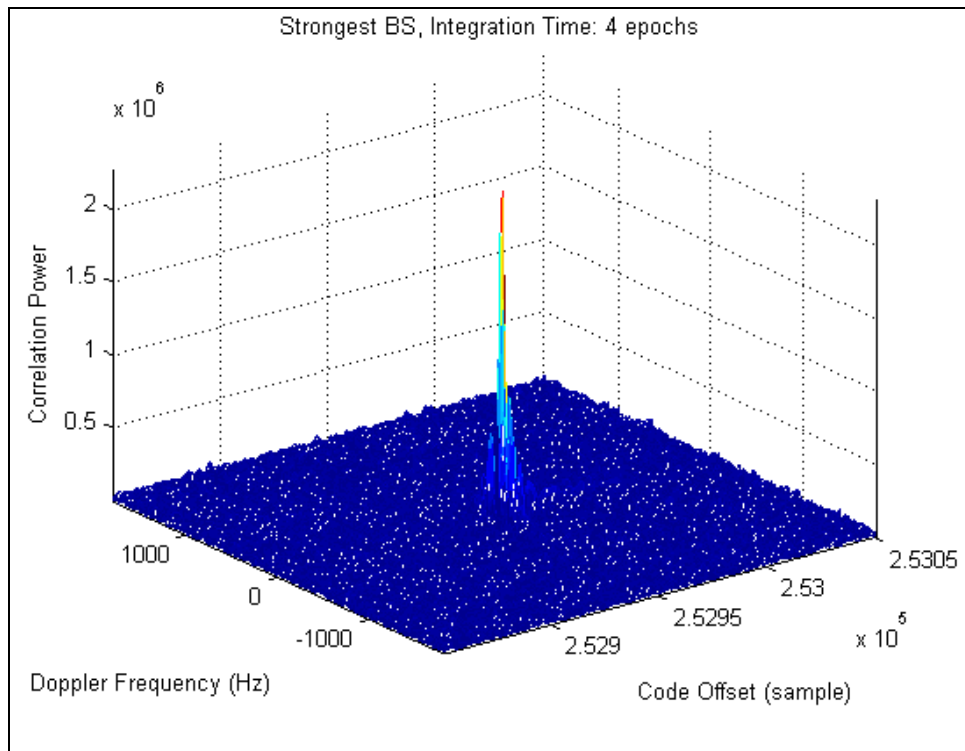


Figure 58: Acquisition result after replacing the LOs with signal generators

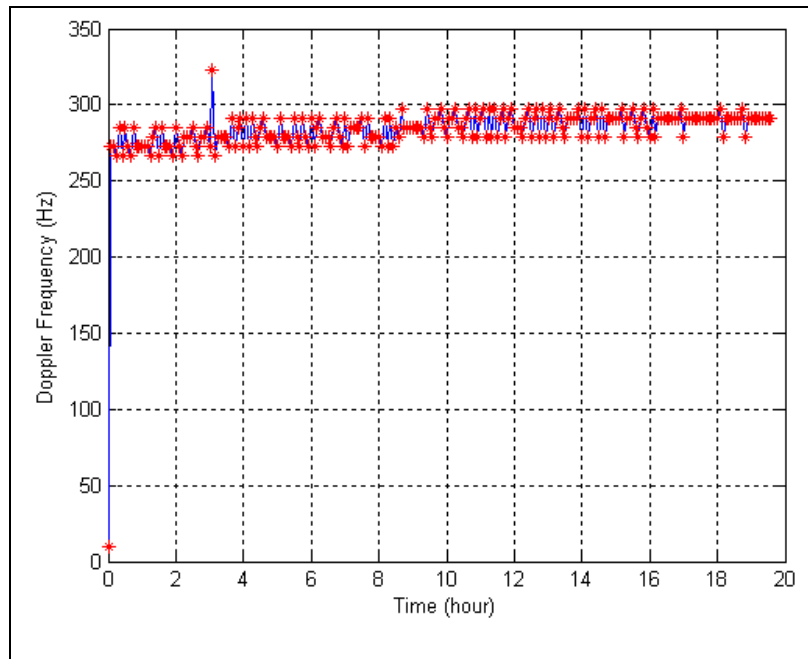


Figure 59: Result of the frequency offset after using the signal generators

After this change, the frequency error during 20 hours is depicted in Figure 59. The mean and standard deviation of the values in this figure are 284.78 and 8.96 Hz respectively. The discrete values in this figure are because of using FFT method with coarse frequency resolution of 6.25 Hz. Since the receiver is in static mode, the frequency changes can be caused by TCXO clock drift resultant from temperature change, power supply fluctuation, shock and vibration.

5.3.2. De-spreading Module

As mentioned in Chapter 3, employing more correlators increases the tracking jitter and improves the MTLL, all of which make the tracking loop more robust against the noise spike. Moreover, it provides a better observation of the signal and its multipath. This information can be used to better detect and investigate the channel and the multipath

signals. The high parallel computational capability of the FPGA suggests that more correlators can be used in the receiver.

The size of the correlation window has to be large enough to (i) monitor all observable multipath components of the signal and (ii) avoid signal loss in the interval of the code tracking decision procedure execution. An investigation of the multipath suggests that, in the worse case, it cannot occur beyond the $2 \mu\text{s}$ windows on either side of the main correlation peak. It should be noted that the code tracking decision procedure execution (correlation update) is mostly sensitive to the frequency tracking procedure executed on the PC since the latter takes much longer than the code tracking decision execution. For instance, assume there are three correlators, where the two outer ones are separated by half of a chip (equal to 122 m) from the middle one. If the receiver speed changes from 0 to 60 km/h (or vice versa) in 10 s while the frequency tracking execution time is more than 10 s, then the receiver will lose the correlation peak. Following this logic, assume that the receiver, which uses a 2 GHz processor, can execute the frequency tracking procedure at intervals of 40 s or less. Also, an ordinary car cannot have an acceleration that exceeds 4 m/s^2 ; therefore, in order to avoid signal loss in the repetition of the frequency tracking procedure, the correlation window should be at least $20 \mu\text{s}$ (to cover the 6 km interval), the equivalent of 25 chips¹. The receiver developed herein satisfies this requirement by employing 50 correlators in the de-spreading module. However, this receiver cannot support the acceleration of high dynamic platforms (e.g., 28 m/s^2).

¹ $4 \times (40)^2 = 6400 \text{ m}$, on either side of the correlation peak.

As explained in Section 3.3.2, a Multiplier-Adder combination is traditionally used for the de-spreading process. Using this approach, the IS-95 pilot channel requires two sets of correlators for I and Q signals for each channel. This implies that 250 multipliers have to be implemented for the 5-channel PLAN receiver – a number beyond what the Virtex-II PRO is capable of providing (e.g. 136 multipliers¹). The shortage of the required multipliers can be solved by replacing the correlators with accumulators. This substitution is feasible since the generated PN code is a sequence of ‘1’s and ‘-1’s, and the correlation is equal to simple addition or subtraction of the previous result with the new sample. The intermediate results of the accumulators are written with a 24-bit precision in the internal SRAMs inside the FPGA. In the event of overflowing, an algorithm is developed on the PC to detect the overflow and adjust the result.

Using the SRAM to store the intermediate result requires reading from and writing to the same location of the RAM in one clock cycle. The Virtex-II PRO SRAM does not permit the simultaneous read and write operations². Furthermore, whenever the PC sends a ‘read correlation’ command, the RAM output has to be available to the USB interface. Since this should not make any interruption in the despreading process which also needs read and write to the RAM, two SRAMs are employed for each accumulator. At each snap shot, the despreading process reads the previous result from one RAM and writes the new result into the other. It is important to note that the USB interface reads the results from the same RAM that the despreading module reads from. This assures that there is no metastability

¹ Refer to Table 4.2

² The Dual port RAMs are capable of simultaneously reading and writing from/to different addresses.

issue due to the asynchronous FPGA/PC clocks since the PC is always reading from the RAM that is not being written on. After 50 clock cycles, the read and write operations are reversed and consequently the USB interface starts reading from the other RAM. Figure 60 demonstrates the overall correlator circuit based on an accumulator and two SRAMs.

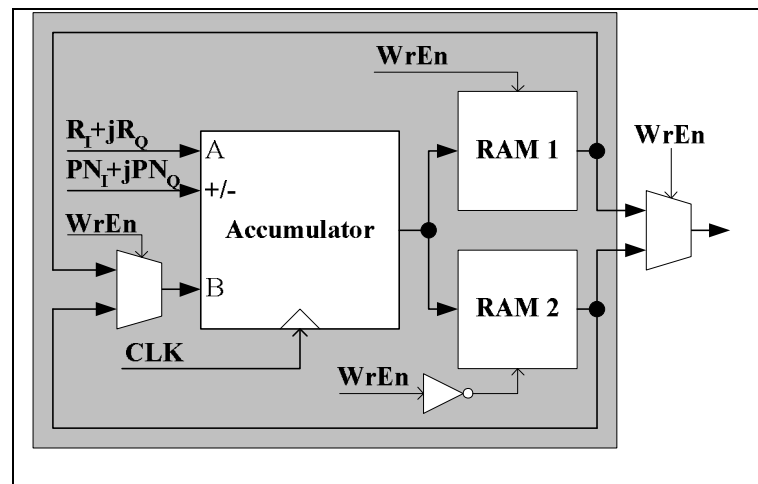


Figure 60: Overall Correlator circuit

Another FPGA implementation challenge is related to the large PN sequence length factor. Even for the lowest sampling frequency of 2.4576 MHz, 65536 bits are required for each PN_I and PN_Q . Storing only one copy of the PN sequence in an internal or external RAM is not an option since several PN generators require access to different addresses in the RAM at the same time. Thus, such a design creates a bottleneck for accessing RAM. To avoid this problem using a 5-channel system, at least five copies of the PN sequence are required. Since this is larger than the RAM capacity of the Virtex-II PRO, a 15-bit LFSR, as discussed in Section 2.4, is implemented to generate the PN sequence. The LFSRs are initiated using the measured code phases in the acquisition procedure or the code tracking decision making on the PC. Moreover, supporting 50 correlators requires generating 50 samples of the local PN sequence. Because these samples are sequential, one sample can be

generated by the LFSR, while the subsequent samples are simply the delayed versions of this sample. The final designed configuration is depicted in Figure 61.

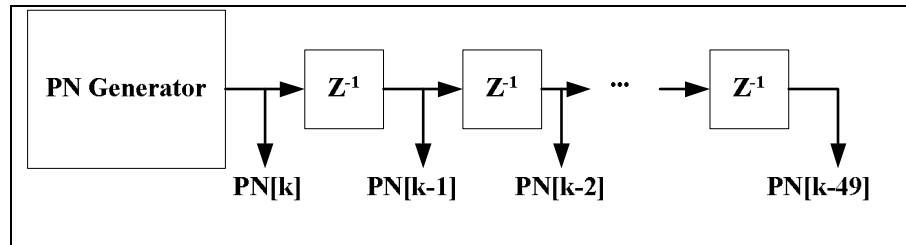


Figure 61: PN Generation for sequential Coefficient Offsets

Each incoming signal has to be correlated with the output of the PN generator and 49 outputs of the delay modules. Consequently, these outputs have to be calculated during the time interval of one sample; thus, the accumulator and PN generator modules need to be clocked by 50 times of the incoming sample frequency. The implemented Correlation method for two consecutive incoming samples is shown in Figure 62.

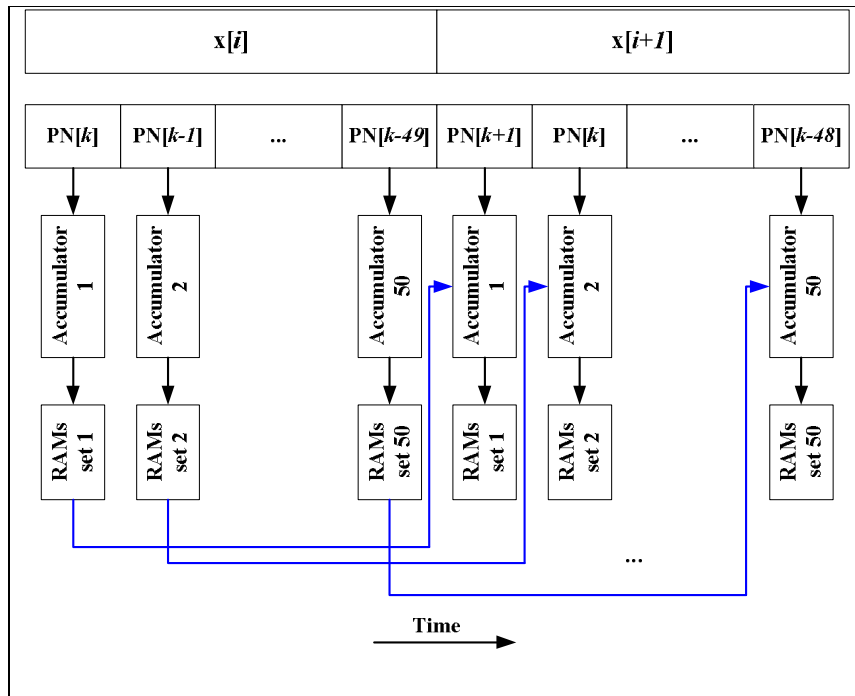


Figure 62: Implemented Correlation method for two consecutive snap shots

5.3.3. Synchronization Module

As discussed earlier, the synchronization of the raw data¹ is accomplished by inserting a Flag at the beginning of every epoch of data. The Flag is reset by the 2 s clock synchronized with the GPS time. The synchronization of the processed data² is also done by employing the 2 s clock as it is explained in the following paragraphs.

As explained in Section 5.2, a DCM is used to generate the required clocks for the digital part of the system. As such, two clocks are produced: (i) 24.576 MHz (main clock) and (ii) 122.88 MHz for the de-spreading module. The main clock generates the divided clocks via producing several clock enables. These clock enables can then be used, instead of gated (combinatorial) clocks, to avoid the clock skew in the system.

The input clock of the DCM is produced by a PLL, synchronized with the 10 MHz TCXO. As explained in Section 4.3.4, the accuracy of the TCXO is 0.5 ppm, which indicates that in the worse case a 0.6144 chip shift occurs every second. The blue plot in Figure 63 illustrates the correlator peak drift that results in the loss of the peak after approximately 45 seconds. To compensate the TCXO drift in the de-spreading module, a signal synchronized with the 1 PPS GPS clock is generated to reset the correlators every 2 seconds. The red plot in Figure 63 shows the signal code phase after using the 2 s reset for TCXO drift in the de-spreading module.

¹ The data which is processed on the PC and is used for the acquisition and calculation of frequency error induced by receiver motion.

² The data, which is processed inside the FPGA, is the output of the de-spreading module.

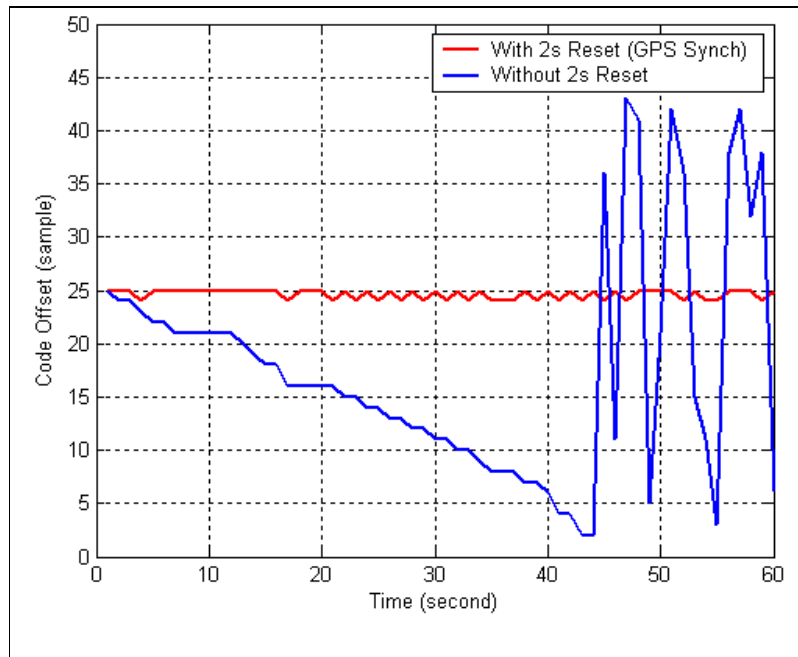


Figure 63: Correlator peak drift in the lack of GPS synchronization (blue) and after TCXO drift compensation (red)

5.4. Tracking

Part of the frequency tracking is designed inside the FPGA to compensate for the TCXO frequency drift. The other part of the frequency tracking which is related to the receiver motion and the code tracking decision making procedure are designed on the PC. In the following subsections, the code tracking and TCXO frequency error tracking are explained in detail.

5.4.1. Code tracking procedure

Traditionally, the code phase tracking is carried out by dividing the correlation window into two equal size windows: Early and Late. The power in both windows is calculated and the correlation window is moved to reduce the power difference to less than a threshold.

Alternatively, a simpler method of “correlation peak tracking” can be implemented if a large number of correlators is being used.

In the PLAN receiver, the second method is more desirable since it is simpler and requires fewer computations. To implement this method, the correlation outputs are sent to the PC, where an algorithm adjusts the results for the probable overflow and detects the peaks. Based on the detected peaks, the local PN generators are instantiated so that the generated PN sequence and consequently the correlation window moves properly to adjust the peaks in the middle. Figure 64 illustrates a sample of the correlators’ output.

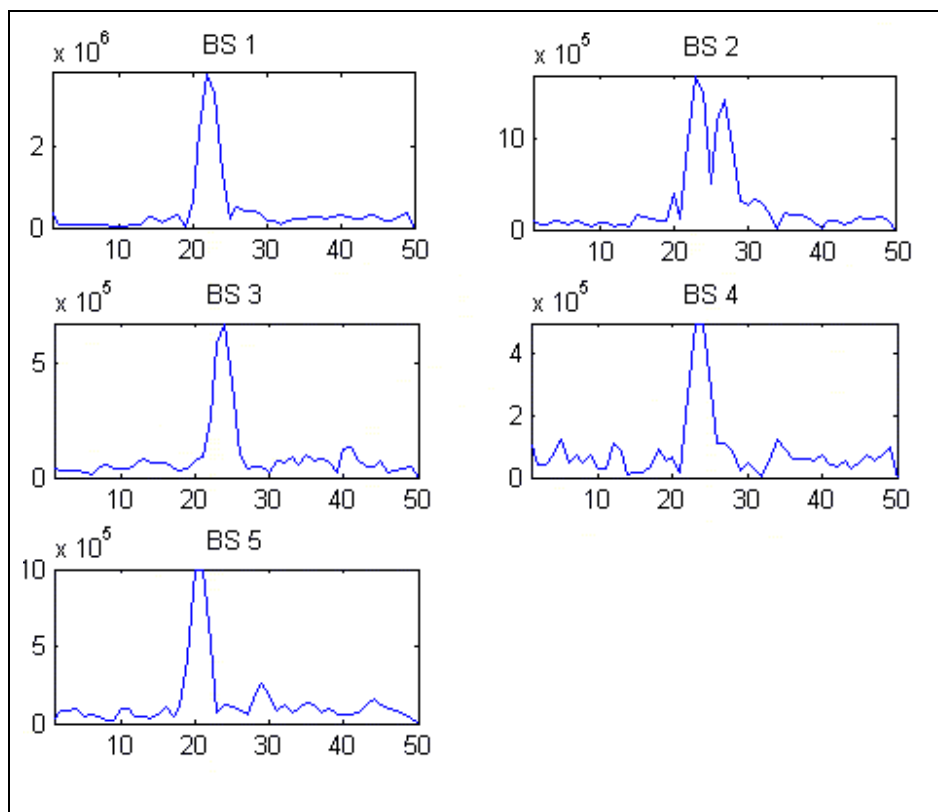


Figure 64: Correlator output snap shot for Five BSs

5.4.2. Frequency tracking procedure

The carrier NCO is updated by the addition of the frequency error caused by the TCXO frequency drift and the receiver motion. The procedure to compensate the receiver motion is the two-dimensional acquisition algorithm with a shrunk frequency error range. Figure 65 demonstrates the block diagram of the circuit inside the FPGA for the TCXO frequency drift compensation. The main concept of the method employed is to count the number of TCXO cycles within a certain number (k) of the GPS pulses.

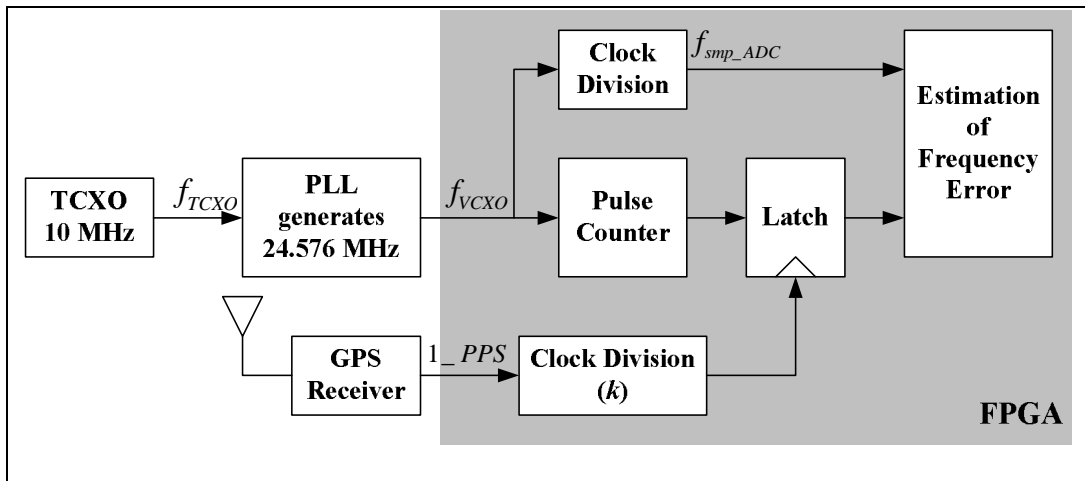


Figure 65: Block Diagram of the Frequency Error Measurement related to TCXO frequency drift

To understand the theory behind this design, suppose two clocks that have respective periods of T_1 and $T_1 + \delta$, where δ represents the timing error between two clocks. If these clocks are superimposed, as depicted in Figure 66, then the second clock will drift by δ each time a new cycle begins. Therefore, if the time stamp acquired by the first clock sampling the second clock is available, then the difference of two consecutive time stamps will be equal to offset δ .

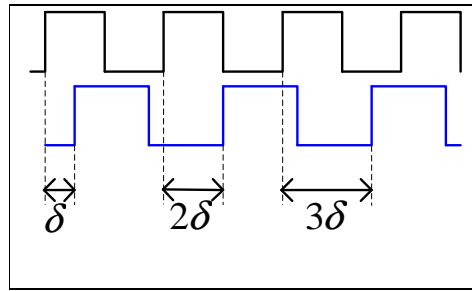


Figure 66: Two Clock with timing error δ

To apply this approach in the PLAN receiver, a counter is used that is clocked by the main clock of 24.576 MHz to generate a ramp function. The counter output resets every time it reaches its maximum value. Suppose that the 1 PPS clock is used to sample this ramp function. Then the difference between two consecutive sampled values d_s divided by the VCXO¹ frequency is equal to δ :

$$\delta = \frac{d_s}{f_{VCXO}} \quad (5.3)$$

If the counter is M-bits, it can count from 0 to 2^M-1 over the $\frac{2^M}{f_{VCXO}}$ second period. In this

case, the counter overflows $\left\lfloor \frac{f_{VCXO}}{2^M} \right\rfloor$ times within 1 PPS GPS clock. Ambiguity may arise

with an overflow counter when two values are sampled before and after the counter overflows. For instance, assume that two consecutive sampled values are 2^M and 2. δ may seem to slip by “ 2^M-2 ” of the VCXO cycles where in reality it only slips by two cycles.

This ambiguity can be resolved by using the following function:

¹ The VCXO frequency (f_{VCXO}) refers to the output of the PLL in Figure 65.

$$\text{Calculate } \begin{cases} d_{s1} = d_s + 2^M \\ d_{s2} = d_s \\ d_{s3} = d_s - 2^M \end{cases} \quad (5.4)$$

$$d_s = \text{Min}\{\text{Mag}(d_{s1}), \text{Mag}(d_{s2}), \text{Mag}(d_{s3})\}$$

where, d_{s1} is related to the case when the counter leads the PPS pulse, while d_{s3} is related to the case when the counter lags the PPS pulse. Consider the case where the 1 PPS pulse is also divided by k , as depicted in Figure 65. This can further enhance the resolution of the measured VCXO frequency error as will be explained in the following paragraphs. The frequency of the VCXO can be expressed as

$$f_{VCXO} = 24.576 \text{MHz} + f_{err_VCXO} \quad (5.5)$$

Or

$$T_{VCXO} = \frac{1}{24.576 \text{MHz} + f_{err_VCXO}}.$$

Since the 1 PPS clock is divided by k , the new period related to the GPS pulse can be assumed as $T'_{GPS} = k$ seconds. This implies that in one period of the new GPS pulse, the M-

bit counter can overflow $\left\lfloor \frac{f_{VCXO}}{2^M} \right\rfloor \cdot k$ times. The time duration for this number of overflows is

calculated as

$$T'_{VCXO} = \frac{2^M \cdot \left\lfloor \frac{f_{VCXO}}{2^M} \right\rfloor \cdot k}{24.576 \text{MHz} + f_{err_VCXO}}. \quad (5.6)$$

The timing error δ can be calculated by

$$\begin{aligned}\delta &= T'_{GPS} - T'_{VCXO} \\ &= k - \frac{2^M \cdot \left\lfloor \frac{f_{VCXO}}{2^M} \right\rfloor \cdot k}{24.576 \text{MHz} + f_{err_VCXO}}.\end{aligned}\quad (5.7)$$

Substituting (5.3) in (5.7) and solving for f_{err_VCXO} will result in

$$f_{err_VCXO} = \frac{d_s}{k} - \left\{ 24.576 \text{MHz} - 2^M \cdot \left\lfloor \frac{f_{VCXO}}{2^M} \right\rfloor \right\}.\quad (5.8)$$

Selecting $M = 16$ eliminates the term inside the bracket and equation (5.8) is simplified to

$$f_{err_VCXO} = \frac{d_s}{k} = \frac{\text{Min}\{d_s\}}{k}.\quad (5.9)$$

Equation (5.9) implies that the step or resolution size of the frequency error can be increased by the factor $\frac{1}{k}$. However, the error frequency in the I and Q baseband samples is

related to the frequency error in the LO frequency ($= -f_{err_LO}$). The relation between this frequency error and the VCXO frequency error in Equation (5.9) is

$$f_{err_LO} = f_{err_VCXO} \frac{f_{LO}}{f_{VCXO}}.\quad (5.10)$$

Thus, the correction frequency of f_{err_LO} should be applied in the de-rotation procedure. In other words, the input samples have to be multiplied by $\exp(j2\pi f_{err_LO} t)$. Nonetheless, time is quantized by the sampling period of

$$\Delta t = \frac{1}{f_{smp_ADC}} = \frac{1}{R_{smp_TCXO} (f_{TCXO} + f_{err_TCXO})}\quad (5.11)$$

where, R_{smp_TCXO} is the ideal ratio of the desired sampling frequency to the ideal TCXO frequency f_{TCXO} (= 10 MHz). The required phase increment or decrement of the de-rotation procedure is calculated as

$$\Delta\phi_{NCO} = 2\pi f_{err_LO} \Delta t = \frac{2\pi f_{err_LO}}{R_{smp_TCXO} (f_{TCXO} + f_{err_TCXO})}. \quad (5.12)$$

Equation (5.12) can be simplified as Equation (5.13) as follows (This simplification can cause the phase to be incorrect by at most 0.5 ppm):

$$\Delta\phi_{NCO} = \frac{2\pi f_{err_LO}}{R_{smp_TCXO} \times f_{TCXO}}. \quad (5.13)$$

Substituting (5.10) in (5.13) and using the GPS clock division of 16 ($k=16$) results in

$$\begin{aligned} \Delta\phi_{NCO} &= \frac{2\pi}{f_{smp_ADC}} \cdot \frac{f_{LO}}{f_{VCXO}} \cdot f_{err_VCXO} \\ &= \frac{2\pi}{f_{smp_ADC}} \cdot \frac{f_{LO}}{f_{VCXO}} \cdot \frac{Min\{d_s\}}{16} \end{aligned} \quad (5.14)$$

For digital implementation of the NCO and based on discussion in 3.3.1.1, this equation can be rewritten as

$$\begin{aligned} \Delta\phi_{NCO} &= \frac{2^{31}}{f_{smp_ADC}} \cdot \frac{f_{LO}}{f_{VCXO}} \cdot \frac{Min\{d_s\}}{16} \\ &= 4327.7778 \times Min\{d_s\} \end{aligned} \quad (5.15)$$

The final designed phase error circuit is shown in Figure 67.

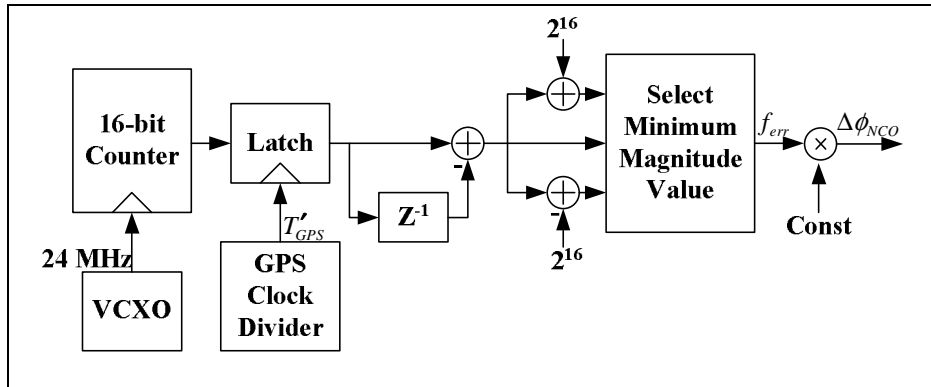


Figure 67: Final TCXO frequency error compensator

5.5. Code and Frequency Tracking Evaluation

To evaluate the code and frequency tracking algorithms introduced in the previous section, the system should ideally be tested under real dynamic situations. In the cellular systems the dynamic scenarios are limited to the terrestrial vehicles such as cars, motor bikes and pedestrians. Due to logistic limitations, the receiver could not be tested in such a situation. Therefore, an attempt was made to predict the behavior of the receiver under dynamic conditions. In the following sections, the code and frequency tracking algorithm results are explained separately. The results are valid because these two algorithms are executed on different sites – frequency tracking inside the FPGA and code tracking on the PC – thus they cannot impede one another. Consequently, two separate setup tests were performed to independently evaluate their performance.

5.5.1. Code tracking test set up

As mentioned in Section 5.3.3, a signal is generated to reset the de-spreading module every 2 s to compensate the TCXO clock drift. To evaluate the functionality of the code tracking algorithm running on the PC, the reset signal was disabled. This causes the correlation peak

to drift 0.6144 chip every second in the worse case where TCXO clock drift is equal to 0.5 ppm. Figure 68 illustrates the correlation peak drift after disabling the 2 s reset signal with different delay for the code tracking execution. The results demonstrate that the code tracking algorithm is able to track the code phase changes results from the TCXO drift.

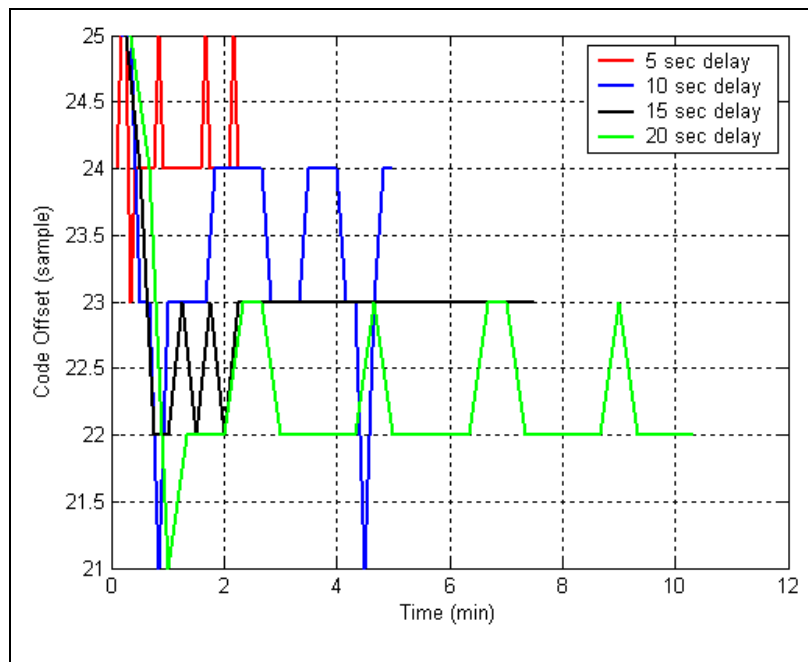


Figure 68: The result of code tracking

In reality, the code tracking has to be able to track the signal under dynamic conditions. As such, two important dynamic scenarios are discussed: (i) when the car experiences extreme acceleration or deceleration, and (ii) when the car maintains a constant high speed. The first and most difficult scenario to simulate involves acceleration. Keeping in mind that the fastest car acceleration does not exceed 28 m/s^2 , it would take approximately 10 seconds to lose the correlation peak (a 12.5 chip shift equal to half of the correlation window). For the second scenario, consider a car that travels at 100 km/h. Under these settings, it would require 109 s before the correlation peak could drift more than half the correlation window size. Considering these two scenarios, it is possible to conclude that the code tracking

algorithm, which repeats within a 10 s time period, will be able to track code phase changes under dynamic conditions.

5.5.2. Frequency tracking test set up

To evaluate the performance of the frequency tracking, a test setup was designed to compare the measured Doppler frequency of the FPGA-based receiver with the Gage card-based receiver. As explained in Chapter 4, the Gage card receiver employs independent algorithms for signal (post-) processing. To avoid any discrepancies that may arise from using different RF front-end units, a common PLAN RF front-end was used for both the FPGA-based and Gage card-based receivers. The output of the common RF front-end is split by a switch and is connected to the FPGA and the Gage card respectively. Two common signal generators are used to produce the required RF and IF frequencies. To generate a clock for the digital part of the receivers, two different TCXO-PLL boards are used. This is necessary since the Gage-card receiver requires a minimum sampling frequency of 2.5 MHz, while the FPGA-based receiver can work with a 2.4576 MHz clock. Figure 69 illustrates the final test setup for the two receivers.

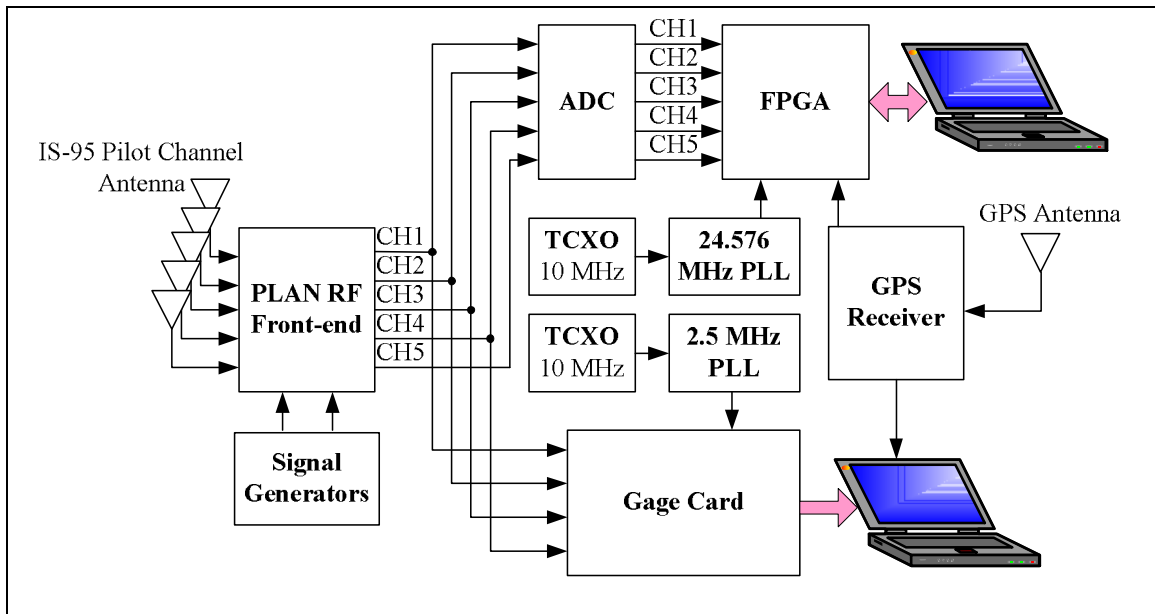


Figure 69: Test set up using the Gage card post-processing algorithm for comparison

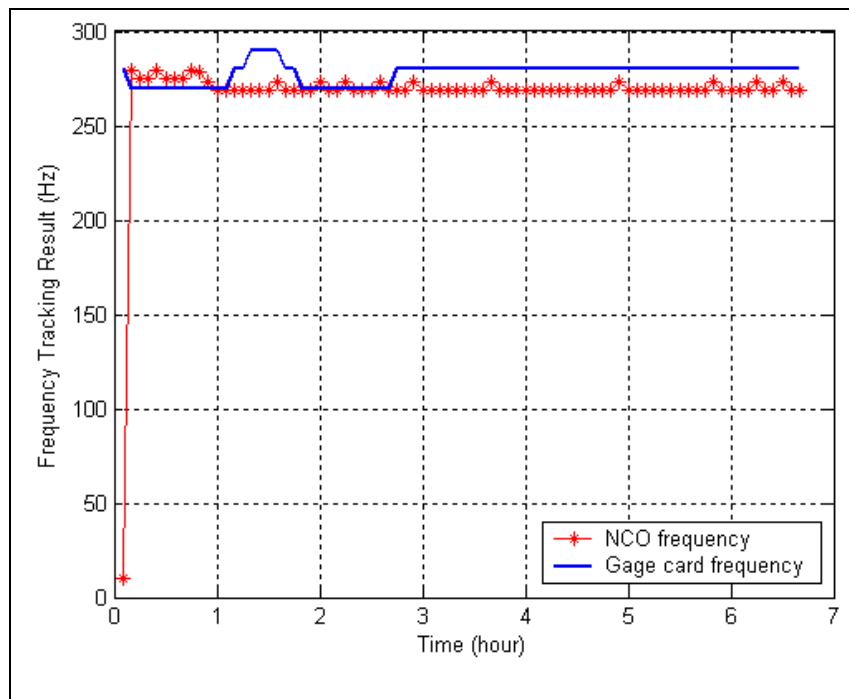


Figure 70: Result of frequency tracking

The result of frequency tracking for both the FPGA-based and Gage card-based receivers are shown in Figure 70. The Gage card-based receiver frequency error is measured using an FFT algorithm with frequency resolution of 6.25 Hz. However, the frequency error in the FPGA-based receiver is a direct read of the NCO input inside the FPGA. The mean and standard deviation of the Gage card-based receiver and the FPGA-based receiver in both cases of with and without TCXO frequency drift tracking is shown in Table 6.

Table 6: Mean and Standard Deviation of the Gage card-based and FPGA-based receivers with and without TCXO frequency drift tracking

Receiver Type		Mean (Hz)	STD (Hz)
Gage card-based		277.6250	5.3353
FPGA-based	<i>Without TCXO drift tracking</i>	284.7872	8.9675
	<i>With TCXO drift tracking</i>	269.8012	2.3764

Even though it seems that using the TCXO frequency drift method improves the frequency tracking result, this cannot be claimed. This is because the frequency resolution in FPGA-based receiver with TCXO frequency drift is less than a hertz which is much finer than the two other cases.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

This thesis focused on the implementation and development of a general and highly flexible real-time Signal Processing unit in a Multi-channel CDMA receiver for positioning. Several steps were used to achieve this end. The first step was to insure that general and highly flexible features of the receiver were met; thus, different positioning algorithms and their system requirements were presented. For instance, to support the AOA method, an antenna array should be used. This implies that the general receiver has to be able to simultaneously process the received signals from each of these antennas.

The second step is to provide the receiver with a real-time capability through decreasing the load of the microprocessor. This is accomplished by pre-processing the high rate data on a more efficient platform before sending the data to the microprocessor for further processing. This requires partitioning the signal processing unit into two groups of low and high computational tasks and implementing each on the software and hardware platform respectively. The FPGA is selected for the hardware platform because *(i)* it is easy to

program and debug, (ii) it can provide massive amount of resources, and (iii) it is a flexible platform, which allows the design to be modified to meet future needs.

Since the highest computational tasks are the Doppler removal and de-spreading modules related to the frequency and code tracking algorithm, the third step is to study different tracking algorithms to find a proper solution for firmware implementation. This study recommends that the FPGA contain 50 accumulators for the de-spreading and a NCO and several complex multipliers for the Doppler removal module.

The fourth step is to construct the primary design of the receiver's signal processing unit. This includes the development of the Doppler removal and correlators inside the FPGA and the acquisition procedure and the tracking decision making on the PC. The design quality is evaluated in terms of scalability, the potential ability of the receiver to accommodate more BSs, and optimality, how efficient the resources are used (internal FPGA logic cells, RAM etc). In this regard, the firmware design is highly scalable due to its modularity, but is not optimal. The primary design was initially optimal, however several subsequent changes were made that decreased the optimality degree of the design.

The fifth step is to design the TCXO frequency drift compensation method inside the FPGA to facilitate the real-time processing of the frequency tracking algorithm. Finally, the last step entails testing the developed system under static scenarios and predicting its behavior under dynamic scenarios.

The result of the code and frequency tracking shows that the system is capable of tracking the signal in static mode. It also demonstrates that the system is able to track the code phase in any dynamic situation if the decision making algorithm is executed at intervals of 10 s or less. Due to logistic limitations, the receiver could not be tested under actual dynamic situations; hence, the frequency tracking could not be evaluated under this condition.

6.2. Suggestions for Future Work

As mentioned before, the employed FPGA, Virtex-II PRO, is a very powerful FPGA with ample resources. This gives the opportunity for future expansion of the system. The synthesis result of the current design which supports five channel-BS combinations is depicted in Table 7.

Table 7: Virtex-II PRO device (2vp30ff896-6) utilization, Synthesis Result

Device resource (Number of)	Used	Total	Percentage
Slices	5446	13696	39%
Slice Flip Flops	7082	27392	25%
4 input LUTs	8781	27392	32%
I/O	77	556	13%
BRAMs	20	136	14%
MULT18X18s	19	136	13%
DCM	1	8	12%

To further expand the system and support more BSs, two limitation factors should be considered: (i) the FPGA resources, such as Number of Slices, size of RAMs, and number of Multipliers, and (ii) maximum frequency.

(i) FPGA Resources:

1. Number of Slices: By assuming a linear relationship between the number of slices and the number of supported BSs, the FPGA can provide the additional slices needed for seven more BSs.
2. Number of RAMs: The number of required RAMs depends on the number of accumulators, where there are two RAMs per accumulator and two accumulators per BS. Thus, the system can provide the necessary number of RAMs to support an extra 29 BSs.
3. Number of Multipliers: Four real multipliers are used for the 32-bit×32-bit multiplication in Figure 67. These multipliers remain the same despite the number of supported BSs. However, for each complex multiplication inside the Doppler removal module, an additional three real multipliers are being used, which indicates that the FPGA can supply additional multipliers needed for 39 more BSs¹.

Other FPGA resources are approximately the same for different number of BSs. Based on the above discussion, twelve BSs could be supported in total.

(ii) Maximum Frequency:

The current maximum frequency is 176.678 MHz, which is far greater than the required frequency of 122.88 MHz – 50 times the sampling rate of 2.4576 MHz. This implies that

¹ $4+3\times(5+N_{BS})=136$.

the system can support more BSs. However, the relation between the maximum achievable frequency and the maximum number of BSs is not clear; therefore, no assumption can be made about the effect of this limitation. It should be noted that since the design is not optimal, the maximum achievable frequency can be further increased by improving the design optimality (e.g. use pipelining for all segments of the design that are combinatorial – non-clocked).

While improvements can be made for the maximum frequency and FPGA resources, it is important to remember that this design has two key advantages that make its continued development worthwhile. First, the current design uses the Verilog codes, implying that the current FPGA, Virtex-II PRO, can be replaced with other FPGA devices without design modification as long as the new FPGA has sufficient resources. Second, the design's modularity allows it to be adapted to other CDMA receivers (e.g. GNSS receivers). For instance, the following changes are required in order to use the firmware in a GNSS receiver:

1. Replace the pilot channel LFSR with the desired PRN LFSR,
2. For each supported PRN, an equivalent Doppler removal module should be used,

The second change indicates that the final GNSS receiver cannot support as many BSs as the IS-95 receiver, and thus can be considered as a restriction.

REFERENCE

Analog Device (1996) “AD9830 datasheet”, URL: http://www.analog.com/UploadedFiles/Data_Sheets/AD9830.pdf, Last Access: May 11, 2008.

Akos, D. M., M. Pini (2006) Effect of Sampling Frequency on GNSS Receiver Performance, *Journal of Navigation*, Vol. 53, No. 2, Summer 2006

Caffery, J.J. (2002) *Wireless Location in CDMA Cellular Radio Systems*, Springer Netherlands, Volume 535, <http://www.springerlink.com/content/978-0-306-47329-6/>, last access: March 4th, 2007

Caffery, J.J., and G. L. Stüber (1998a) “Overview of Radiolocation in CDMA Cellular Systems”, *IEEE Communications Magazine*, April 1998

Caffery, J. and G. Stüber (1998b), Subscriber Location in CDMA Cellular Networks, *IEEE Trans. on Vehicular Technology*, Vol. 47, pp. 406–416.

Charkhandeh, S., M.G. Petovello, R. Watson, and G. Lachapelle (2006) “Implementation and Testing of a Real-Time Software-Based GPS Receiver for x86 Processors” in

Proceedings of ION NTM 2006, 18-20 January, Monterey, California, U.S. Institute of Navigation

Cheng, N., Y. Ren, Y. Wan (2007) New Carrier Tracking Technique for High-Dynamic Spread Spectrum Signals, IEEE International Conference on Wireless Communications, Networking and Mobile Computing.

Cheng, D.K. (1993) *Fundamentals of Engineering Electromagnetics*, Addison-Wesley, USA, pp. 455-459

Cho, H. S., S. H. Im, G. I. Jee (2005) "A FPGA-based Software GPS Receiver Implementation Using Simulink and Xilinx System Generator", Konkuk University, *ION GNSS 18th International Technical Meeting of the Satellite Division*, Long Beach, CA

Dovis, F., M. Spelat, P. Mulassano, and C. Leone (2005) "On the Tracking Performance of a Galileo/GPS Receiver Based on Hybrid FPGA/DSP Board", *ION GNSS 18th International Technical Meeting of the Satellite Division*, Long Beach, CA

Etemad, K. (2004) *CDMA2000 Evolution: System Concepts and Design Principles*, John Wiley & Sons, Hoboken, New Jersey

Fantino, M., F. Dovis, L. Presti (2004) Design of a Reconfigurable Low-Complexity Tracking Loop for Galileo Signals, IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications, ISSSSTA 2004, Sydney Australia

Gua Y.J. (2004) *Advances in Mobile Radio Access Networks*, Artech House

Harte, L., R. Kikta, and D. McLaughlin (1999) *CDMA IS-95 for Cellular and PCS*, McGrawHill

Heckler, G.W., J.L. Garrison (2006) *SIMD correlator library for GNSS software receivers*, GPS Solut

Holmes, J.K. (1982) *Coherent Spread Spectrum Systems*, Original from University of Michigan, John Wiley & Sons

Hill, J. (2004) "Navigation Signal Processing with FPGAs", University of Hartford, *ION NTM*, San Diego, CA

Kaplan, E. D., C. J. Hegarty (2006) *Understanding GPS Principles and Applications*, Artech House, INC., Second Edition

Korowajczuk, L., B. de S.A. Xavier, A.M. Fartes Filho, L.Z. Ribeiro, C. Korowajczuk, L.A. DaSilva (2004) *Designing Cdma2000 Systems*, John Wiley & Sons, England

Lachapelle, G. (2000) *Hydrography*. ENGO 545 Lecture Notes, Department of Geomatics Engineering, University of Calgary.

Lopez, A. (2006) *Design and Implementation of a 5-Channel CDMA Receiver for Mobile Position Location*, MSc Thesis, Department of Electrical and Computer Engineering, University of Calgary, Canada, (Available at <http://plan.geomatics.ucalgary.ca>)

Lu, D. (2007) *Multipath Mitigation in TOA Estimation Based on AOA*, PhD Thesis, Department of Electrical and Computer Engineering, University of Calgary, Canada, (Available at <http://plan.geomatics.ucalgary.ca>)

Lück, T., M. Bodenbach, J. Winkel, T. Pany, D. Sanroma, B. Eissfeller, and F. Föster (2005) "Trade-off between pure software based and FPGA based base band processing for a real time kinematics GNSS receiver", *ION GNSS 18th International Technical Meeting of the Satellite Division*, Long Beach, CA

Messier, G.G. and J.S. Nielsen (1999) "An Analysis of TOA-Based Location for IS-95 Mobiles," *IEEE VTC-Fall, IEEE VTS 50th*, Vol.2, 19-22 Sept., pp. 1067-1071

Meyr, H., M. Moeneclaey, S.A. Fechtel (1998) *Digital Communication Receivers*, pp. 90

Mileant, A., S. Million, S. Hinedi, U. Cheng (1995) *The Performance of the All-Digital Data Transition Tracking Loop using Nonlinear Analysis*, IEEE

Moghaddam, A. R. (2007) *Enhanced Cellular Network Positioning using Space-Time Diversity*, MSc Thesis, Department of Geomatics Engineering, University of Calgary, Canada, (Available at <http://plan.geomatics.ucalgary.ca>)

Pahlavan, K., X. Li, M. Ylianttila, R. Chana, and M. Latva-aho (2000) *An Overview of Wireless Indoor Geolocation Techniques and Systems*, Springer-Verlag Berlin Heidelberg

Parkinson, B.W., J.J. Spilker (1996) *Global Positioning System: Theory and Applications* Volum I, pp.258, Washington, DC

Parnell, K., and R. Bryner (2004) *Comparing and Contrasting FPGA and Microprocessor System Design and Development*,
http://www.xilinx.com/support/documentation/white_papers/wp213.pdf, last accessed March 23, 2008

Peterson, R.L, R.E. Ziemer, and D.E. Borth (1995) *Introduction to Spread Spectrum Communications*,

Petovello M.G., and G. Lachapelle (2006) "An Efficient New Method of Doppler Removal and Correlation with Application to Software-Based GNSS Receivers", *Proceedings of ION GNSS 19th International Technical Meeting of Satellite Division*

Proakis, J. (2001) *Digital Communication* 4th Edition, McGraw-Hill

Rappaport, T.S. (2002) *Wireless Communications, Principle and Practice*, Prentice Hall

Raquet, J. (2006) *Advanced GNSS Receiver Technology*, ENGO 699.45 Course Notes, Department of Geomatics Engineering, University of Calgary, Canada, pp. 86 & 109

Romdhani, L., and A. Trad (2002) *Mobile Location Estimation Approaches*, University of Nice Sophia-Antipolis, <http://www-sop.inria.fr/planete/atrad/mobile-location-report.ps>, Last Access: April 27, 2008.

Sayed, A.H., A. Tarighat, and N. Khajehnouri (2005) Network-Based Wireless Location: Challenges faced in developing techniques for accurate wireless location information, *IEEE Signal Processing Magazine*, July 2005

Simon, M.K., K. Omura, R.A. Scholtz, and B.K. Levitt (1994) *Spread Spectrum Communications Handbook*, McGraw-Hill School Education Group

Smith., S.W. (1997) *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego, CA, California Technical Publishing

Spilker, J.J., D.T. Magill, (1961), The Delay-Lock Discriminator-An Optimum Tracking Device, *Proc. IRE*, vol. 49, pp. 1403-1416.

Togneri, R. (2005) Estimation Theory for Engineers, URL: http://www.ee.uwa.edu.au/~roberto/teach/Estimation_Theory.pdf, Last Access: April 26, 2008.

Tsui, J.B.Y. (2005) Fundamentals of Global Positioning System Receivers: A software approach, John Wiley & Sons, Hoboken, New Jersey, ch6.

Van Graas, F., A. Soloviev, M.U. de Haag, S. Gunawardena, and M. Braasch (2005) Comparison of Two Approaches for GNSS Receiver Algorithms: Batch Processing and Sequential Processing Considerations, *Ohio University*, ION GNSS 18th International Technical Meeting of the Satellite Division, 13-16 September 2005, Long Beach, CA

Watson, R. (2005) High sensitivity GPS L1 Signal Analysis for Indoor Channel Modeling, MSc Thesis, Department Geomatics Engineering, The University of Calgary, URL: <http://plan.geomatics.ucalgary.ca/papers/05.20215.robwatson.pdf>, Last access: May 04, 2008.

Wilde, A. (1998) The Generalized Delay-Locked-Loop, *Wireless Personal Communications*,

Wilde, A. (1996) On the Performance of Extended Tracking Range Delay-Locked Loops, *Satellite Systems for Mobile Communications and Navigation*, 13-15th May 1996,

Wilde, A., and U.P. Bernhard (1995) Mean Time to Lose Lock for a Second-Order Extended Tracking Range Delay-Lock-Loop,

Wilde, A., (1995) Reduced complexity delay-locked loop, 9th November 1995 Vol. 31 No. 23, pp. 1979-1980

Wolf, W. (2004) *FPGA-Based System Design*, Prentice Hall Professional Technical Reference, Upper Saddle River, New Jersey 07458, Chapter 3 is available at http://www.phptr.com/content/images/0131424610/samplechapter/0131424610_ch03.pdf

Last Access: April 1, 2007

Xilinx LogiCore (2008) DDS Compiler v2.1, available at: http://www.xilinx.com/support/documentation/ip_documentation/dds_ds558.pdf, last access: May 11, 2008

Xilinx (2004) Virtex-II PRO and Virtex-II Pro X Platform FPGAs: Complete Datasheet, <http://cas.ee.ic.ac.uk/people/lah100/ds083.pdf> Last access: March 16, 2008.

Yacoub M.D. (2002) *Wireless Technology: Protocols, Standards, and Techniques*, CRC Press.