DESIGN AND IMPLEMENTATION OF GPS L1 SIGNAL TRACKING AND DATA PROCESSING FOR POSITION

A Project Report

submitted by

DAMARLA BALAJI

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **IMPLEMENTATION OF GPS L1 SIGNAL TRACKING ON FPGA AND DATA PROCESSING FOR POSITION**, submitted by **DAMARLA BALAJI**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr V. Kamakoti
Research Guide
Professor
Department of Computer Science
and Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 8th May 2017

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude towards several people who enabled me to reach this far with their timely guidance, support and motivation.

First and foremost, I offer my earnest gratitude to my guide, Dr. V. Kamakoti whose knowledge and dedication has inspired me to work efficiently on the project and I thank him for motivating me, allowing me freedom and flexibility while working on the project.

I would like to thank my co-guide Dr.Nitin Chandrachoodan and faculty advisor Dr.Shreepad Karmalkar, who have guided me through out the program.

My special thanks and deepest gratitude to Neel Gala who has been very supportive. He has enriched the project experience with his active participation and invaluable suggestions.

My thanks goes to my fellow labmate Zaid for his help and support.

ABSTRACT

KEYWORDS: GPS Acquisition, Tracking, Navsolver, FPGA, Virtex 7, Phase Locked Loop (PLL), Delay Locked Loop (DLL)

Hardware implementation of GPS receiver is carried out to study the underlying principles and algorithms of the GPS system. It requires minimum hardware such as Antenna, AD9361 RF Agile Transceiver, Virtex 7(VC707) FPGA Board along with a Computer. The main aim of this project is to develop an IP of GPS L1 receiver in the FPGA.

A GPS receiver has to perform three main steps to decode the navigation data: *Acquisition, Tracking and Navigation Data Processing.* The GPS receiver need data from atleast four GPS satellites to find its location.Acquisition deals with finding Satellite ID, Code Phase and Doppler Frequency of incoming GPS I,Q sample stream. Sampling rate is set at 4 MHz satisfying Nyquist criterion. The FPGA prototyping of the L1(1575 MHz) signal Tracking stage and Processing of Navigation(NAV) Data are explained in this thesis.

Tracking module is a digital implementation of a Costas Loop and Delay Locked Loop and it involves three real time correlations with Early, Late and Prompt replicas of C/A code.Stripping off C/A code and Doppler and keeping track of phase gives NAV BITS which have very low frequency (50 HZ) and they have to be accumulated in frames. With the collected data using Tringulation method and iterative approach user position is calculated.The Navigation Data Processing is done through software. The rapid FPGA prototyping, design, verification and Processing of Navigation data for finding the position are the main focus of the thesis.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

BPSK	Binary Phase Shift Keying
CORDIC	COordinate Rotation DIgital Computer
GPS	Global Positioning System
CA Code	Coarse Acquisition Code
PRN	Pseudo Random Noise
HDL	Hardware description language
HOW	Hand Over Word
NCO	Numerically Controlled Oscillator

CHAPTER 1

Introduction

GPS is the world's first Global Navigation Satellite System (GNSS) developed and deployed by the US department of defense. GPS became fully functional in 1995 and it consists of a cluster of 32 satellites orbiting around the earth in the medium earth and geostationary orbits. GPS satellites send data using the CDMA transmission scheme in the L1 and L2 band of frequencies. GPS data is BPSK modulated and it uses Gold codes for spread spectrum modulation. All the GPS satellites have a precision atomic clock which are internally synchronised between them and it plays a major role in the accuracy of position measurements. These satellites send their position information and time of transmission as GPS data. A GPS receiver on earth requires data from atleast four satellites to calculate its position. The underlying principle behind the position calculation is the method of trilateration.

A GPS receiver has to perform three fundamental steps in order to compute the position information:*Acquisition, Tracking and Navigation data processing.* In this project, a Standard Positioning Service (SPS) of the Global Positioning System is implemented on Hardware FPGA for Acquisition and Tracking stages using verilog HDL.

CHAPTER 2

GPS SIGNAL

2.1 Signals and Data

The GPS signals are transmitted on two radio frequencies in the UHF band. The UHF band covers the frequency band from 500 MHz to 3 GHz. These frequencies are referred to as L1 and L2. These are derived from a common frequency,$f_0 = 10.23MHz$:

$$f_{L1} = 154f_0 = 1575.42MHz,\ f_{L2} = 120f_0 = 1227.60MHz.$$

The signals are composed of the following three parts:
Carrier The carrier wave with frequency f_{L1} or f_{L2},

Navigation data The navigation data contain information regarding satellite orbits.This information is uploaded to all satellites from the ground stations in the GPS Control Segment. The navigation data have a bit rate of 50 bps.

Spreading sequence Each satellite has two unique spreading sequences or codes. The first one is the coarse acquisition code (C/A), and the other one is the encrypted precision code (P(Y)). The C/A code is a sequence of 1023 chips.(A chip corresponds to a bit.) The code is repeated each ms giving a chipping rate of 1.023 MHz. The P code is a longer code(=2.35x10_4 chips) with a chipping rate of 10.23 MHz. It repeats itself each week starting at the beginning of the GPS week which is at Saturday/Sunday midnight.

2.2 GPS Signal Scheme

The block diagram of GPS signal generation is given below. At the far left, the main clock signal is supplied to the remaining blocks. The clock signal has a

frequency of 10.23 MHz. When multiplied by 154 and 120, it generates the L1 and L2 carrier signals.At the very bottom the data generator generates the navigation data.

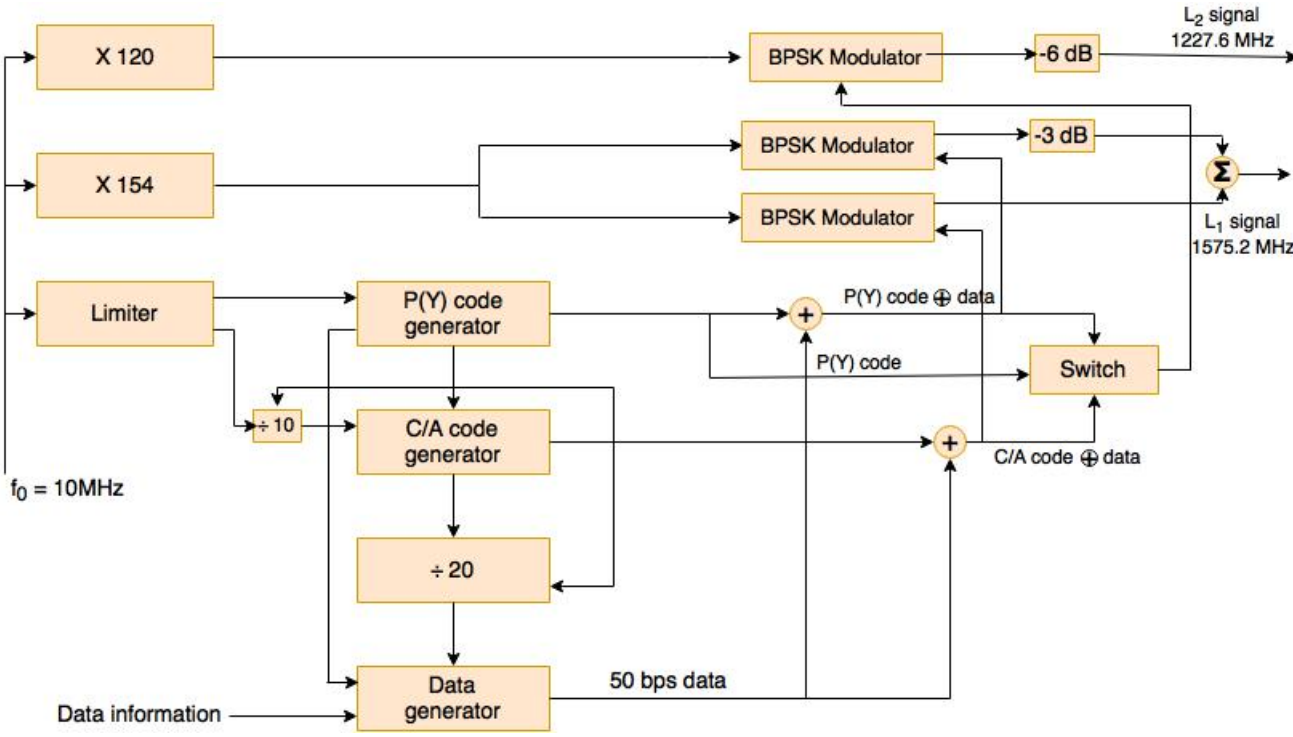

Figure 2.1: Generation of GPS signals at the Satellites

After code generation, the codes are combined with the navigation data through modulo-2 adders. Ordinary multiplication is used here on polar non-return-to-zero representation, i.e., 1's and -1's. The corresponding properties of the multiplication with two binary non-return-to-zero sequences are shown in following table.

The C/A code $\otimes$ data and the P(Y) code $\otimes$ data signals are supplied to the two modulators for the L1 frequency. Here the signals are modulated onto the carrier signal using the binary phase shift keying (BPSK) method. The so-called standard positioning service (SPS) is based on C/A code signals alone.

2.3 C/A Code

C/A codes are often referred to as Gold codes, as Robert Gold described them. They are also referred to as pseudo-random noise(PRN) sequences.

2.3.1 Gold Sequence Generation

The generation of the Gold codes is sketched below. The C/A code generator contains two shift registers known as G_1 and G_2. These shift registers each have 10 cells generating sequences of length 1023. The two resulting 1023 chip-long sequences are modulo-2 added to generate a 1023 chip-long C/A code, only if the polynomial is able to generate code of maximum length.

Every 1023rd period, the shift registers are reset with all ones, making the code start over. The G_1 register always has a feedback configuration with the polynomial

$$f(x) = 1 + x^3 + x^{10},$$

meaning that state 3 and state 10 are fed back to the input. In the same way, the G_2 register has the polynomial

$$f(x) = 1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{10}.$$

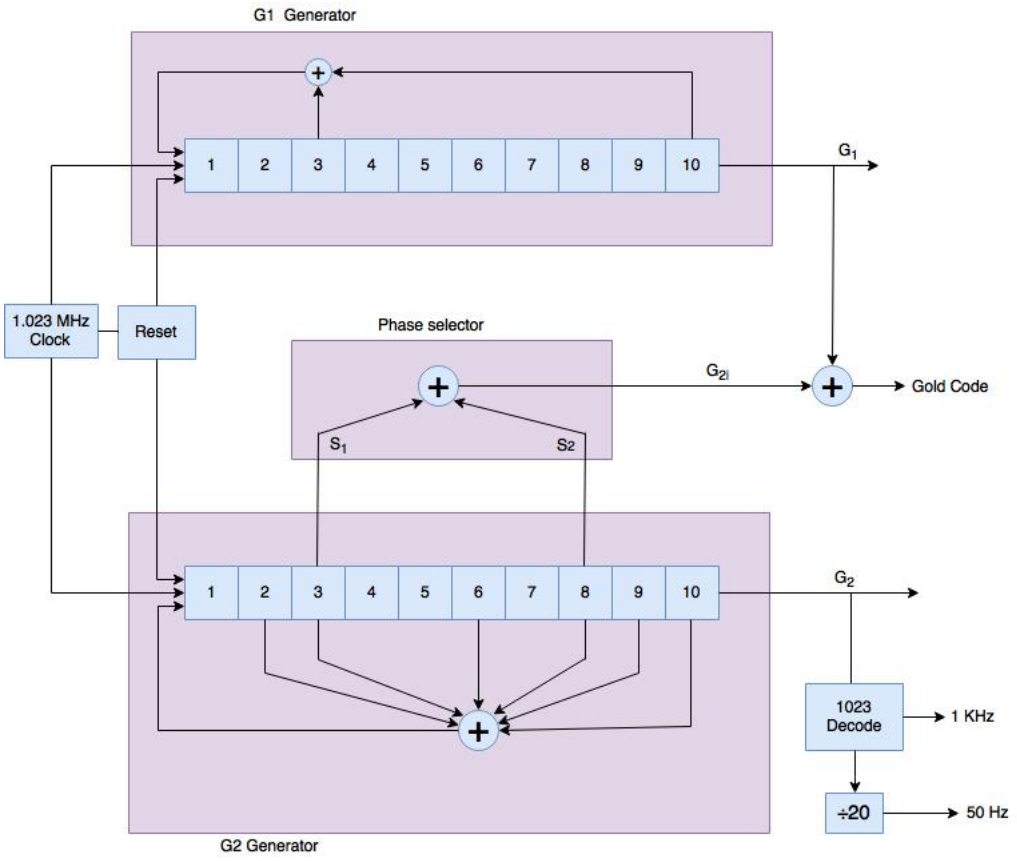

Figure 2.2: C/A code generator. The code generator contains two shift registers, G1 and G2. The output from G2 depends on the phase selector. The different configurations of the phase selector makes the different C/A codes.

To make different C/A codes for the satellites, the output of the two shift registers are combined in a very special manner. The G_1 register always supplies its output. The selection of states for the modulo-2 adder is called the phase selection. The following table shows the combination of the phase selections for each C/A code. It also shows the first 10 chips of each code in octal representation.

2.4 Navigation Data

The navigation data is transmitted on the L1 frequency with the earlier mentioned bit rate of 50 bps. This section describes the structure and contents of the navigation data. The following figure shows the overall structure of an entire navigation message.

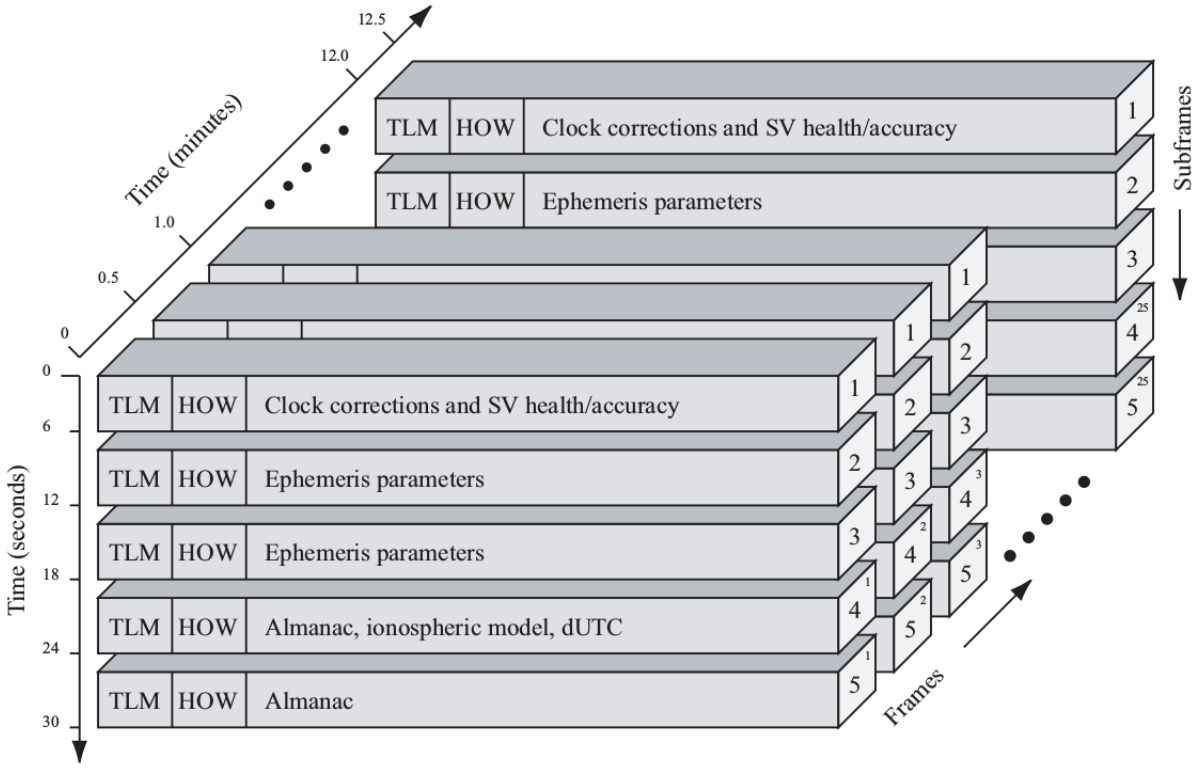

Figure 2.3: GPS navigation data structure.

The basic format of the navigation data is a 1500-bit-long frame containing 5 subframes, each having length 300 bits. One subframe contains 10 words, each word having length 30 bits. Subframes 1, 2 and 3 are repeated in each frame. The last subframes 4 and 5 have 25 versions (with the same structure but different data) referred to as page 1 to 25. With the bit rate of 50 bps, the transmission of a subframe lasts 6 s, one frame lasts 30 s, and one entire navigation message lasts 12.5 minutes.

2.4.1 Telemetry and Handover Words

The subframes of 10 words always begin with two special words, the telemetry (TLM) and handover word (HOW) pair.

TLM is the first word of each subframe and it is thus repeated every 6 s. It contains an 8-bit preamble followed by 16 reserved bits and parity. The preamble should be used for frame synchronization.

HOW contains a 17-bit truncated version of the *time of week* (TOW), followed by two flags supplying information to the user of antispoofing, etc. The next three bits indicate the subframe ID to show in which of the five subframes in the current frame this HOW is located.

2.4.2 Data in Navigation Message

In addition to the TLM and HOW words, each subframe contains eight words of data. This will only be a cursory description of the data in the different words and not a complete description of all bits.

Subframe 1 - Satellite Clock and Health Data The first subframe contains first of all clock information. That is information needed to compute at what time the navigation message is transmitted from the satellite. Additionally, subframe 1 contains health data indicating whether or not the data should be trusted.

Subframes 2 and 3 - Satellite Ephemeris Data Subframes 2 and 3 contain the satellite ephemeris data. The ephemeris data relate to the satellite orbit and are needed to compute a satellite position.

Subframes 4 and 5 - Support Data These last two subframes repeat every 12.5 minutes, giving a total of 50 subframes. Subframes 4 and 5 contain almanac data. The almanac data are the ephemerides and clock data with reduced precision. Additionally, each satellite transmits almanac data for all GPS satellites while it only transmits ephemeris data for itself. The remainder of subframes 4 and 5 contain various data, e.g., UTC parameters, health indicators and ionospheric parameters.

CHAPTER 3

Carrier and Code Tracking

3.1 Introduction

3.1.1 Acquisition

The purpose of acquisition is to determine visible satellites and coarse values of carrier frequency and code phase of the satellite signals. The satellites are differentiated by the 32 different PRN sequences. The Acquisition is usually performed on a block of data. The longer the data record used the higher the signal-to-noise ratio that can be achieved. There are two factors that can limit the length of the data record. The first one is whether there is a navigation data transition in the data. The second one is the Doppler effect on the C/A code.

Since the C/A code is 1 ms long, it is reasonable to perform the acquisition on atleast 1ms of data. Even if only one millisecond of data is used for acquisition, there is a possibility that a navigation data phase transition may occur in the data set. If there is a data transition in this set of data, the next 1 ms of data will not have a data transition. Therefore, in order to guarantee there is no data transition in the data, one should take twao consecutive data sets to perform acquisition. This data length is up to a maximum of 10 ms. If one takes two consecutive 10 ms of data to perform acquisition, it is guaranteed that in one data set there is no transition.

The second limit of data length is from the Doppler effect on the C/A code. If a perfect correlation peak is 1, the correlation peak decreases to 0.5 when a C/A code is off by half a chip. This corresponds to 6 bB decrease in amplitude. Assume that the maximum allowed C/A code misalignment is half a chip for effective correlation. The chip frequency is 1.023 Mhz and the maximum Doppler shift expected on the C/A code is 6.4 Hz. It takes about 78 ms (1/(2 x 6.4)) for two frequencies different by 6.4 Hz to change by half a chip. This data length limit

is much longer than the 10 ms; therefore, 10 ms of data should be considered as the longest data used for acquisition.

The Doppler frequency range that needs to be searched is +10 kHz or -10kHz. It is important to determine the frequency steps needed to cover this 20 kHz range. The frequency step is closely related to the length of the data used in acquisition. If the data record is 1 ms, a 1 kHz signal will change 1 cycle in the 1 ms. In order to keep the maximum frequency separation at 0.5 cycle in 1 ms, the frequency step should be 1 kHz. Under this condition, the furthest frequency separation between the input signal and the correlating signal is 500 Hz. Here, we have used conventional Time domain approach for correlation in Acquisition.

GPS signal is in L1 band(1575 MHz). An antenna is used to receive signal. Band Pass Filter is used to filter out of band components and this is passed through RF Chain to get I,Q samples at desired Data Rate. The RF Front End Receiver block diagram is

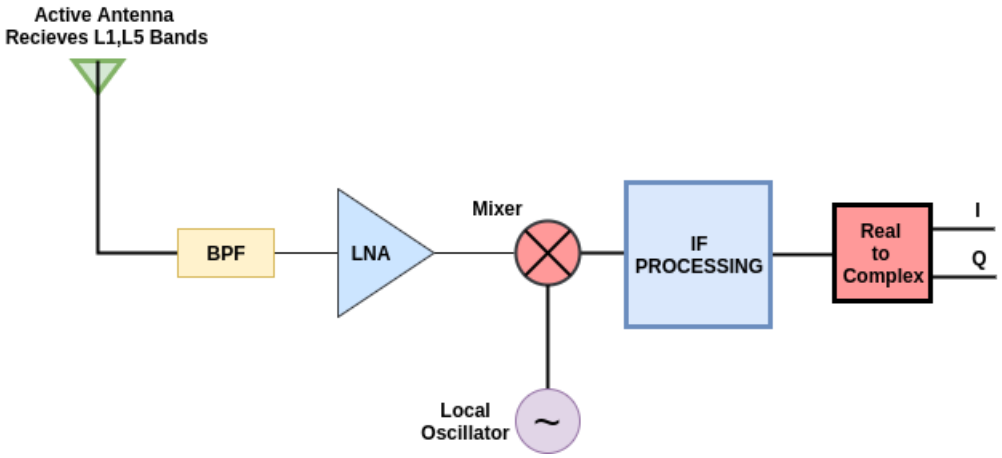

Figure 3.1: Block Diagram of RF Front End Receiver

The two major steps in navigation signal demodulation are signal acquisition and tracking. It takes 30 seconds to fully receive a single frame of data transmitted by a GPS satellite. The major issue in data decoding is the time varying Doppler shift associated with the received signal. Since the satellite is moving in space at a large speed(∼2.5 Km/s) relative to the receiver, the transmitted data undergoes a frequency shift due to Doppler effect. Both the carrier and code frequency suffer

from Doppler shift independently and the worst part is that the frequency shifts vary over time.

The first task of the receiver is to identify the four GPS satellites which are in the line of sight and transmit a healthy signal. This is done by correlating the received signal with the PRN code corresponding to each of the satellite. The four satellites that give the highest correlation will be selected for positioning service. The first step is known as acquisition in which the receiver finds the four satellites along with the exact alignment of the PRN code and initial Doppler shift of the carrier. The two most common algorithms used for data acquisition are serial search acquisition carried out in the time domain and parallel search acquisition carried out in the frequency domain.

Once the acquisition is carried out, the receiver need to do a follow up which is the tracking of satellite data. Receiver need to acquire the data over a complete frame from all the four satellites. The code phase and initial Doppler shift of a selected satellite are known from the acquisition stage. Now that the receiver need to keep track of the varying Doppler frequency in the carrier and code of these four satellites over the entire frame for a duration of 30 seconds. PLL's and DLL's are used for tracking the carrier and code frequencies respectively. Once the Doppler is removed, despreading is carried out to extract the navigation bits. The third task of the GPS receiver is the navigation data decoding which is done after tracking to get the receiver location.

3.2 Carrier Tracking

To demodulate the navigation data successfully an exact carrier wave replica has to be generated. To track a carrier wave signal, phase lock loops (PLL) or frequency lock loops (FLL) are often used.

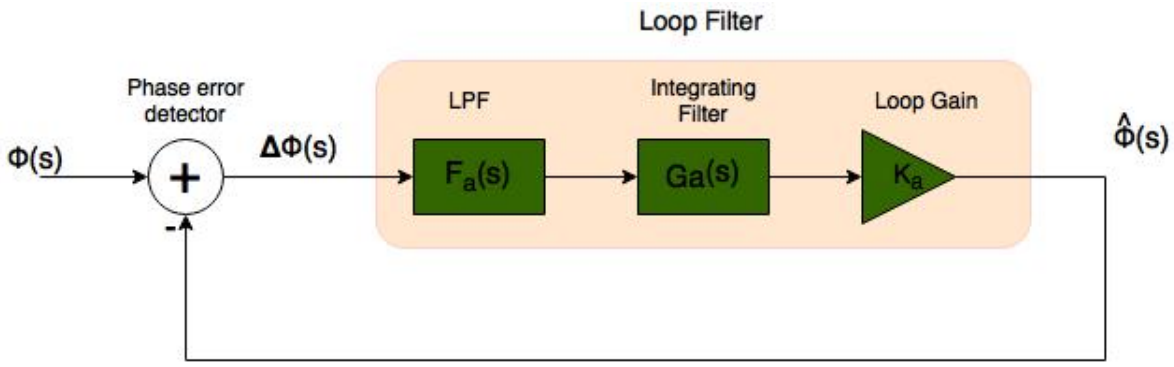

Figure 3.2: PLL Architecture

The above figure shows a basic block diagram for a phase lock loop. The two first multiplications wipe off the carrier and the PRN code of the input signal. To wipe off the PRN code, the I_p output from the early-late code tracking described above is used. The loop discriminator block is used to find the phase error on the local carrier wave replica. The output of the discriminator, which is the phase error (or a function of the phase error), is then filtered and used as a feedback to the numerically controlled oscillator (NCO), which adjusts the frequency of the local carrier wave. In this way the local carrier wave could be an almost precise replica of the input signal carrier wave.

The problem with using an ordinary PLL is that it is sensitive to 180^0 phase shifts. Due to navigation bit transitions, a PLL used in a GPS receiver has to be insensitive to 180^0 phase shifts.

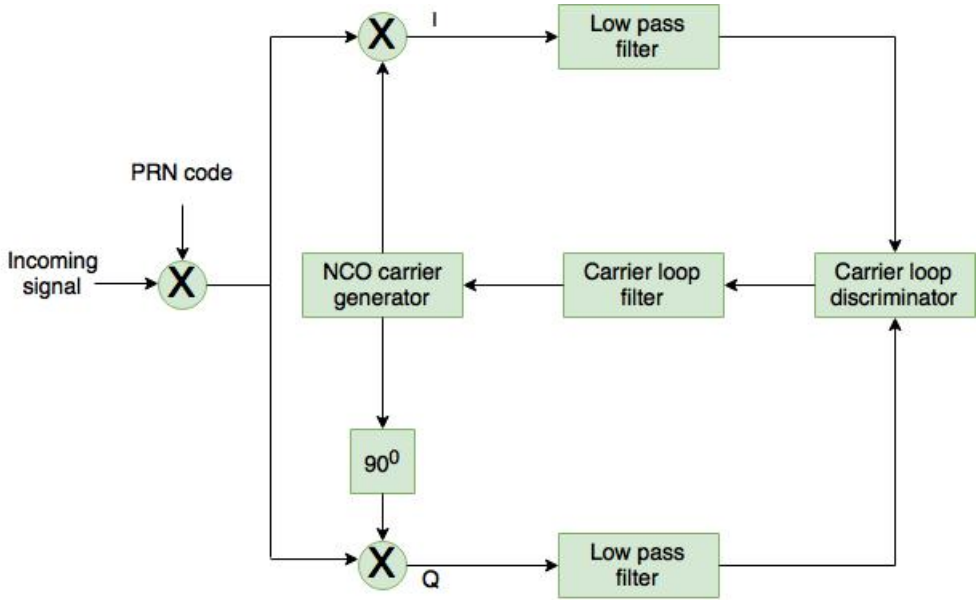

Figure 3.3: Costas loop used to track the carrier wave

The above figure shows a Costas loop. One property of this loop is that it is insensitive for 180^0 phase shifts and hereby a Costal loop is insensitive for phase transitions due to navigation bits. This is the reason for using this carrier tracking loop in GPS receivers. The Costas loop contains two multiplications. The first multiplication is the product between the input signal and the local carrier wave and the second multiplication is between a 90^0 phase-shifted carrier wave and the input signal. The goal of the Costas loop is to try to keep all energy in the I (in-phase) arm. To keep the energy in the I arm, some kind of feedback to the oscillator is needed. If it is assumed that the code replica in figure is perfectly aligned, the multiplication in the I arm yields the following sum:

$$D^k(n)cos(\omega_{IF}n)cos(\omega_{IF}n+\varphi) = \frac{1}{2}D^k(n)cos(\varphi) + \frac{1}{2}D^k(n)cos(2\omega_{IF}n+\varphi),$$

where φ is the phase difference between the phase of the input signal and the phase of the local replica of the carrier phase. The multiplication in the quadrature arm gives the following:

$$D^k(n)cos(\omega_{IF}n)sin(\omega_{IF}n+\varphi) = \frac{1}{2}D^k(n)sin(\varphi) + \frac{1}{2}D^k(n)sin(2\omega_{IF}n+\varphi).$$

If the two signals are lowpass filtered after the multiplication, the two terms with the double intermediate frequency are eliminated and the following two signals remain:

$$I^k = \frac{1}{2}D^k(n)cos(\varphi),$$

$$Q^k = \frac{1}{2}D^k(n)sin(\varphi).$$

To find a term to feed back to the carrier phase oscillator, it can be seen that the phase error of the local carrier phase replica can be found as

$$\frac{Q^k}{I^k} = \frac{\frac{1}{2}D^k(n)sin(\varphi)}{\frac{1}{2}D^k(n)cos(\varphi)} = tan(\varphi),$$

$$\varphi = tan^{-1}(\frac{Q^k}{I^k}).$$

From above equation, it can be seen that the phase error is minimized when the correlation in the quadrature-phase arm is zero and the correlation value in the in-phase arm is maximum. The arctan discriminator in above equation is the most precise of the Costas discriminators, but it is also the most time-consuming.

3.3 Code Tracking

The goal for a code tracking loop is to keep track of the code phase of a specific code in the signal. The output of such a code tracking loop is a perfectly aligned replica of the code. The code tracking loop in the GPS receiver is a delay lock loop(DLL) called an early-late tracking loop. The idea behind the DLL is to correlate the input signal with three replicas of the code as seen in below figure.

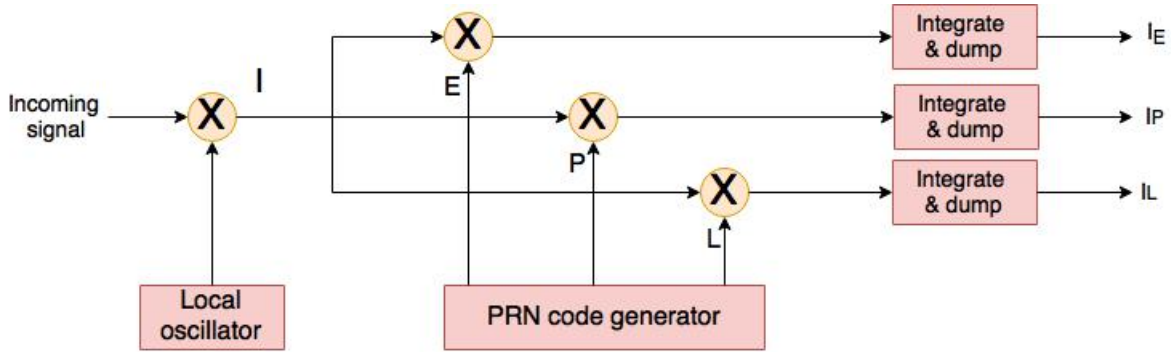

Figure 3.4: Basic code tracking loop block diagram

The first step is converting the input signal to baseband, by multiplying the incoming signal with a perfectly aligned local replica of the carrier wave. Afterwards the signal is multiplied with three code replicas. The three replicas are nominally generated with a spacing of $+\frac{1}{2} or -\frac{1}{2}$ chip. After this second multiplication, the three outputs are integrated and dumped. The output of these integrations is a numerical value indicating how much the specific code replica correlates with the code in the incoming signal.

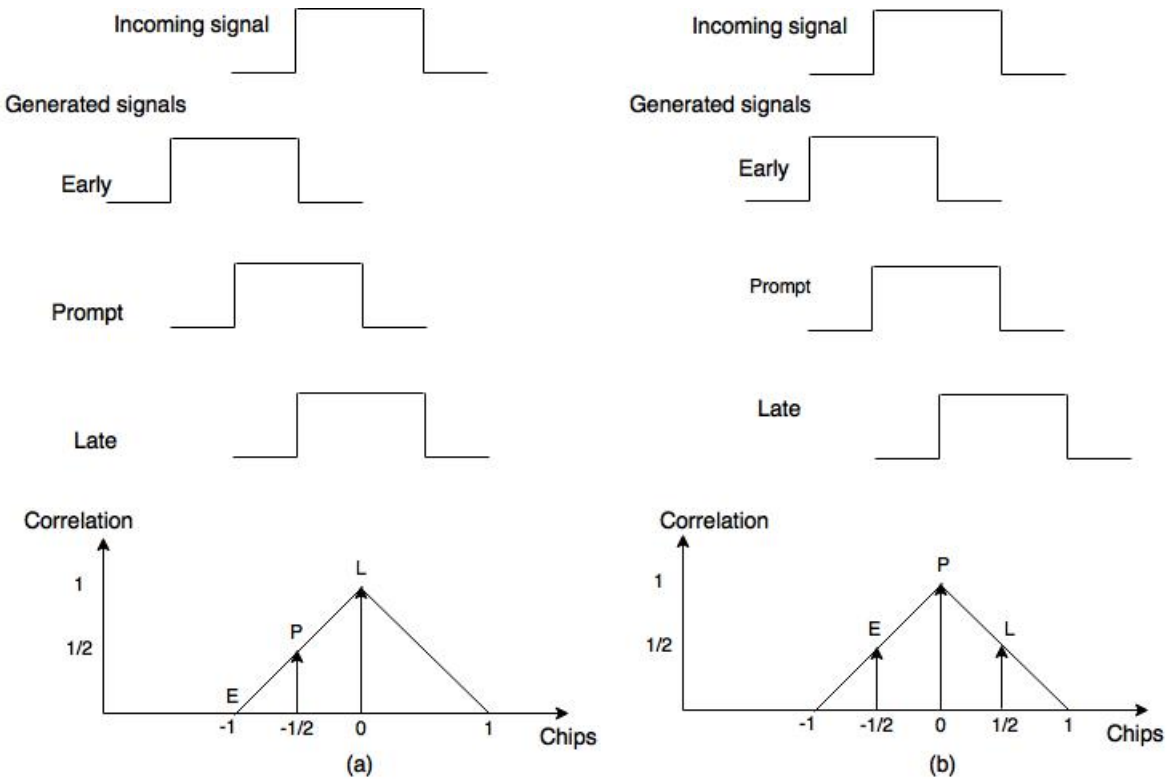

Figure 3.5: Code tracking

The three correlation outputs I_E, I_P, I_L are then compared to see which one provides the highest correlation. In above figure a the late code has the highest correlation, so the code phase must be decreased. In figure b above the highest peak is located at the prompt replica, and the early and late replicas have equal correlation. In this case, the code phase is properly tracked.

The DLL with three correlators is optimal when the local carrier wave is located in phase and frequency. But when there is a phase error on the local carrier wave, the signal will be more noisy, making it more difficult for the DLL to keep lock on the code. So instead the DLL in a GPS receiver is often designed with six correlators.

This design has the advantage that it is independent of the phase on the local carrier wave. If the local carrier wave is in phase with the input signal, all the energy will be in the in-phase arm. If the code tracking loop performance has to be independent of the performance of the phase lock loop, the tracking loop has to use both the in-phase and quadrature arms to track the code.

The implemented tracking loop discriminator is the normalized early minus late power. This discriminator is described as

$$D = \frac{\sqrt{(I_E^2 + Q_E^2)} - \sqrt{(I_L^2 + Q_L^2)}}{\sqrt{(I_E^2 + Q_E^2)} + \sqrt{(I_L^2 + Q_L^2)}},$$

where I_E, Q_E, I_L and Q_L are output from four of the six correlators. The normalized early minus late power discriminator is chosen because it is independent of the performance of the PLL as it uses both the in-phase and quadrature arms. The normalization of the discriminator causes that the discriminator can be used with signals with different signal-to-noise ratios and different signal strengths.

The tracking loop generates three local code replicas. In this section, the chip space between the early and prompt replicas is half a chip. The DLL can be modeled as a linear PLL and thus the performance of the loop can be predicted based on this model. In other words the loop filter design is the same, just the parameter values are different.

CHAPTER 4

Implementation of Tracking Algorithm

4.1 CORDIC Algorithms and Architectures

Digital signal processing (DSP) algorithms exhibit an increasing need for the efficient implementation of complex arithmetic operations. The computation of trignometric functions, coordinate transformations or rotations of complex valued phasors is almost naturally involved with modern DSP algorithms. The COordinate Rotation DIgital Computer algorithm (CORDIC) offers the opportunity to calculate all the desired functions in a rather simple and elegant way.

The CORDIC algorithm was first introduced by Volder for the computation of trignometric functions, multiplication, division and datatype conversion, and later on generalized to hyperbolic functions by WALTHER.Two basic CORDIC modes are known leading to the computation of different functions, *the rotation mode* and the *vectoring mode.*

For both modes the algorithm can be realized as an iterative sequence of additions/subtractions and shift operations, which are rotations by a fixed rotation angle but with variable rotation direction. Due to the simplicity of the involved operations the CORDIC algorithm is very well suited for VLSI implementation.

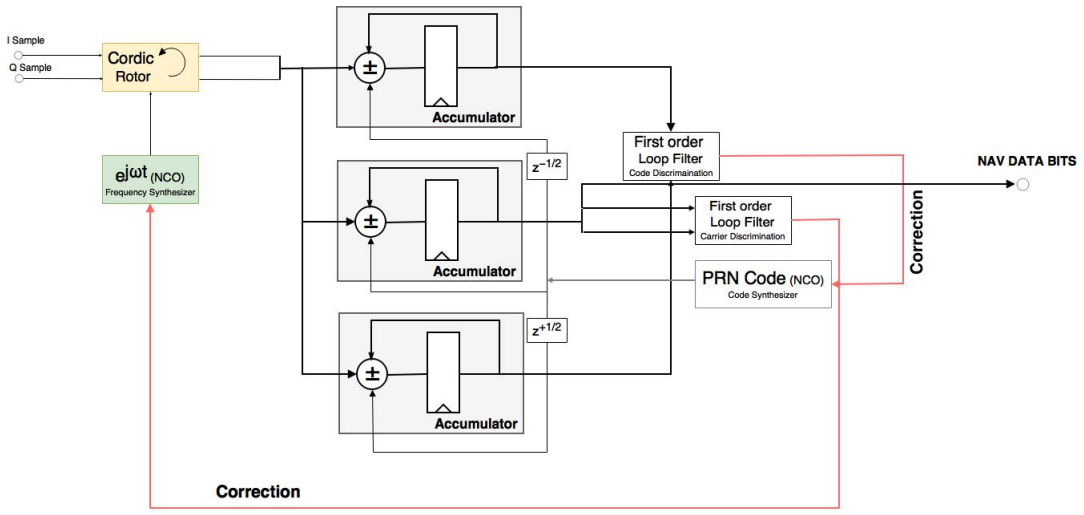

Figure 4.1: Block Diagram of Tracking Algorithm

4.1.1 The CORDIC Algorithm

In the most general form one CORDIC iteration can be written as

$$x_{i+1} = x_i - m\mu_i y_i \delta_{m,i}$$

$$y_{i+1} = y_i + \mu_i x_i \delta_{m,i}$$

$$z_{i+1} = z_i - \mu_i \alpha_{m,i}$$

Although it may not be immediately obvious this basic CORDIC iteration describes a rotation (together with a scaling) of an intermediate plane vector $v_i = (x_i, y_i)^T$ to $v_{i+1} = (x_{i+1}, y_{i+1})^T$. The third iteration variable z_i keeps track of the rotation angle $\alpha_{m,i}$. The variable $m\epsilon 1, 0, -1$ specifies a circular,linear or hyperbolic coordinate system. The rotation direction is steered by the variable $\mu_i \epsilon 1, -1$. In order to avoid multiplications $\delta_{m,i}$ is defined to be
$\delta_{m,i} = 2^{-s_{m,i}}; Radix2 number system$
For obvious reasons we restrict consideration to radix 2 number systems below: $\delta_{m,i} = 2^{-s_{m,i}}$. The shift sequence $s_{m,i}$ is generally a nondecreasing integer sequence. Hence, a CORDIC iteration can be implemented using only shift and add/subtract operations.

The x_{i+1},y_{i+1} equations of the system of equations can be written as a matrix-vector product as

$$v_{i+1} = \begin{pmatrix} 1 & -m\mu_i\delta_{m,i} \\ \mu_i\delta_{m,i} & 1 \end{pmatrix} \tag{4.1}$$

In order to verify that the matrix-vector product in above equation describes indeed a vector rotation and to quantify the involved scaling we consider now a general normalized rotation and to quantify the involved scaling we consider now a general normalized plane rotation matrix for the three coordinate systems. For m = 1, 0, -1 and an angle $\mu_i\alpha_{m,i}$ with μ_i determining the rotation direction and $\alpha_{m,i}$ representing an unsigned angle, this matrix is given by

$$R_{m,i} = \begin{pmatrix} cos(\sqrt{m}\alpha_{m,i}) & -\mu\sqrt{m}sin(\sqrt{m}\alpha_{m,i}) \\ \frac{\mu_i}{\sqrt{m}}sin(\sqrt{m}\alpha_{m,i}) & cos(\sqrt{m}\alpha_{m,i}) \end{pmatrix} \tag{4.2}$$

which can be easily verified by setting m to 1, 0, -1.

4.1.2 Rotation Mode of CORDIC algorithm in Circular Coordiante System

In our project we have implemented Rotation mode in circular coordinate system for finding rotation of input vector by an angle θ of cos and sin.In Rotation mode, the values of alpha and mu are

$$\alpha_{m,i} = \frac{1}{\sqrt{m}}tan^{-1}(\sqrt{m}2^{-s_{m,i}})$$

,

$$\mu_i = sign(z_i)$$

In circular coordinate system, the values are m=1 and $\alpha = tan^{-1}(2^{-i})$. Then equations are

$$x_{i+1} = x_i + \mu_i y_i 2^{-i}$$

$$y_{i+1} = y_i - \mu_i x_i 2^{-i}$$

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

where $\mu_i = sign(z_i)$. The pre-rotation angle must be between $-\Pi/2 to + \Pi/2$. So, we will subtract $\pi/2$ if angle is in second quadrant and add $\pi/2$ if angle is in third quadrant.

The operation of cordic algorithms is shown below

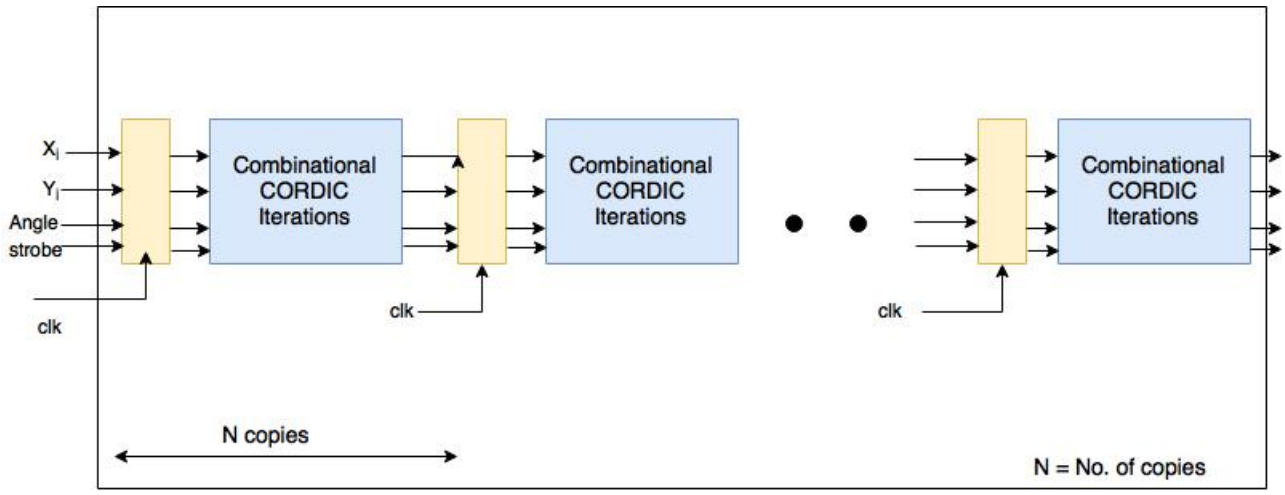

Figure 4.2: CORDIC ALGORITHM IMPLEMENTATION

The combinational cordic iterations contains the following basic structure

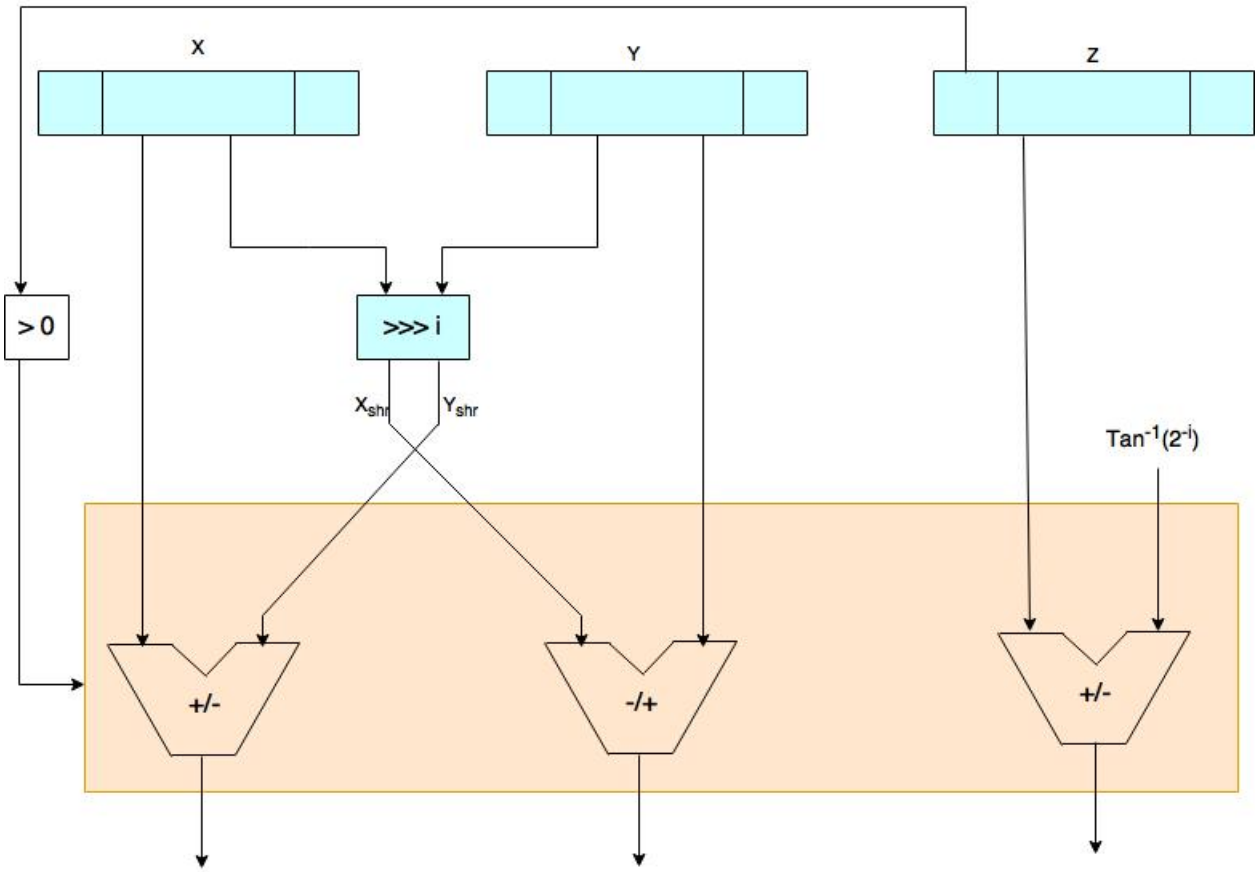

Figure 4.3: Basic structure of a processing element for one CORDIC iteration in Rotation mode of circular coordinate system

4.1.3 Vector Mode of CORDIC algorithm in Circular Coordiante System

In our project we have implemented Vector mode in circular coordinate system for finding absolute value or Magnitude of given vector and angle of the vector. In vector mode, the values of alpha and mu are

$$\alpha_{m,i} = \frac{1}{\sqrt{m}} tan^{-1}(\sqrt{m}\frac{y}{x})$$

,

$$\mu_i = sign(y_i)$$

.

In circular coordinate system, the values are m=1 and $\alpha = tan^{-1}(2^{-i})$. Then

equations are

$$x_{i+1} = x_i - \mu_i y_i 2^{-i}$$

$$y_{i+1} = y_i + \mu_i x_i 2^{-i}$$

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

where $\mu_i = sign(y_i)$. The pre-rotation angle must be between $-\Pi/2to + \Pi/2$. So, we will subtract $\pi/2$ if angle is in second quadrant and add $\pi/2$ if angle is in third quadrant.

The operation of cordic algorithms is shown below

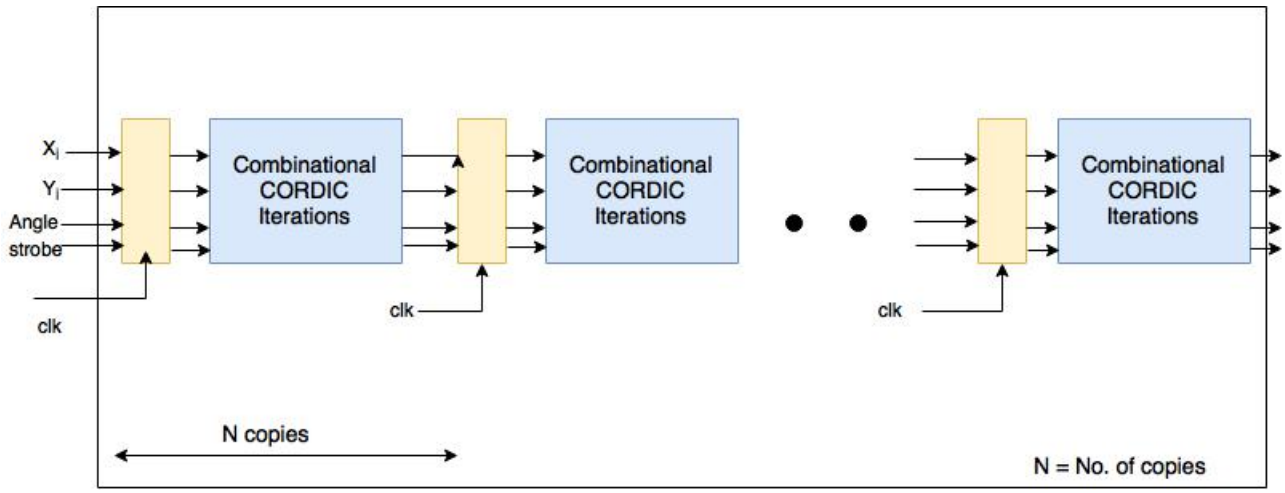

Figure 4.4: CORDIC ALGORITHM IMPLEMENTATION

The combinational cordic iterations contains the following basic structure

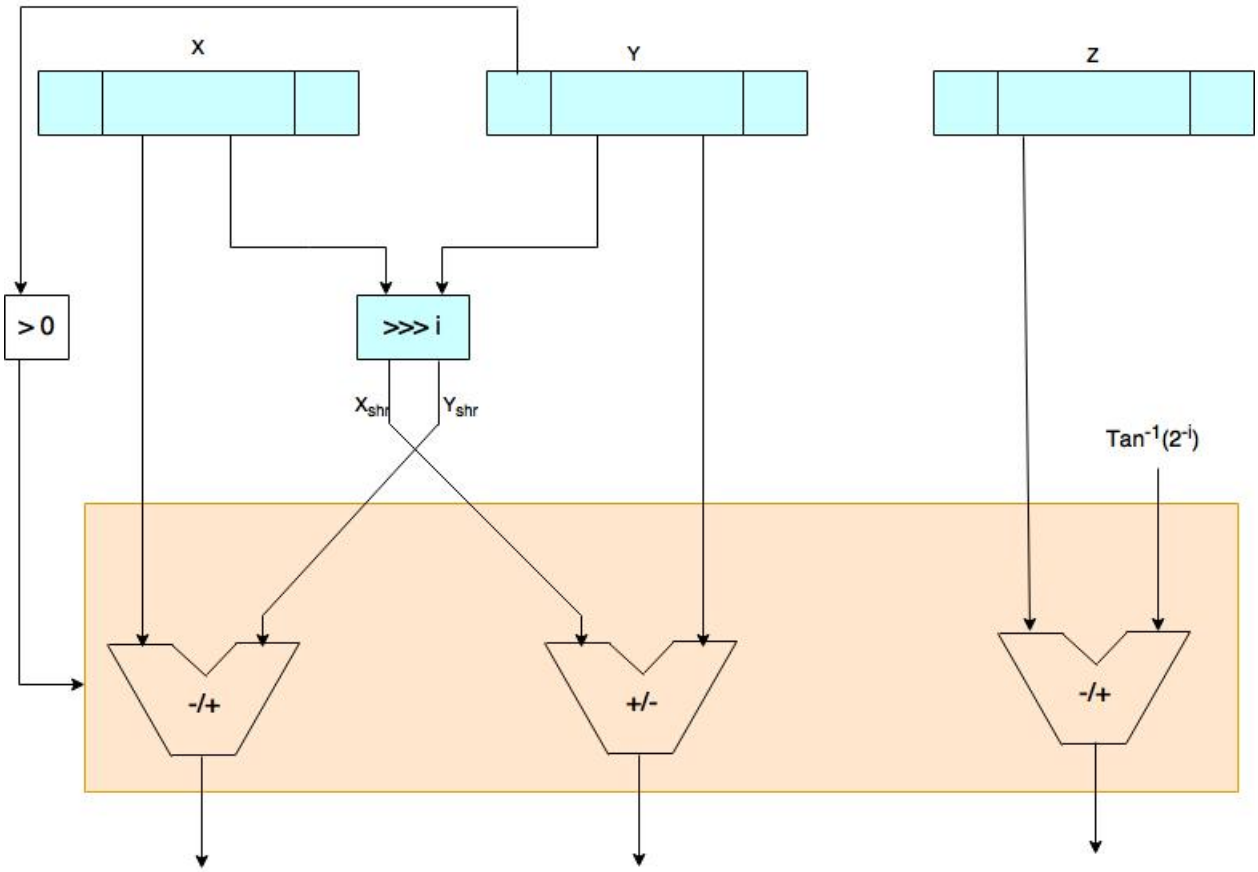

Figure 4.5: Basic structure of a processing element for one CORDIC iteration in vector mode of circular coordinate system

4.1.4 Rotation Mode of CORDIC algorithm in Linear Coordiante System

In our project we have implemented Rotation mode in Linear coordinate system for finding product of two numbers. In Rotation mode, the values of alpha and mu are

$$\alpha_{m,i} = \frac{1}{\sqrt{m}} tan^{-1}(\sqrt{m} 2^{-s_{m,i}})$$

,

$$\mu_i = sign(z_i)$$

.

In Linear coordinate system, the values are m=0 and alpha is found by taking limit as m tends to zero and solve it using L-hospital rule. we get $\alpha = (2^{-i})$. Then

equations are

$$x_{i+1} = x_i$$

$$y_{i+1} = y_i - \mu_i x_i 2^{-i}$$

$$z_{i+1} = z_i + \mu_i(2^{-i})$$

where $\mu_i = sign(z_i)$. This works for the inputs of values between -1 and +1 with inputs as x and z, output as y.

The operation of cordic algorithms is shown below

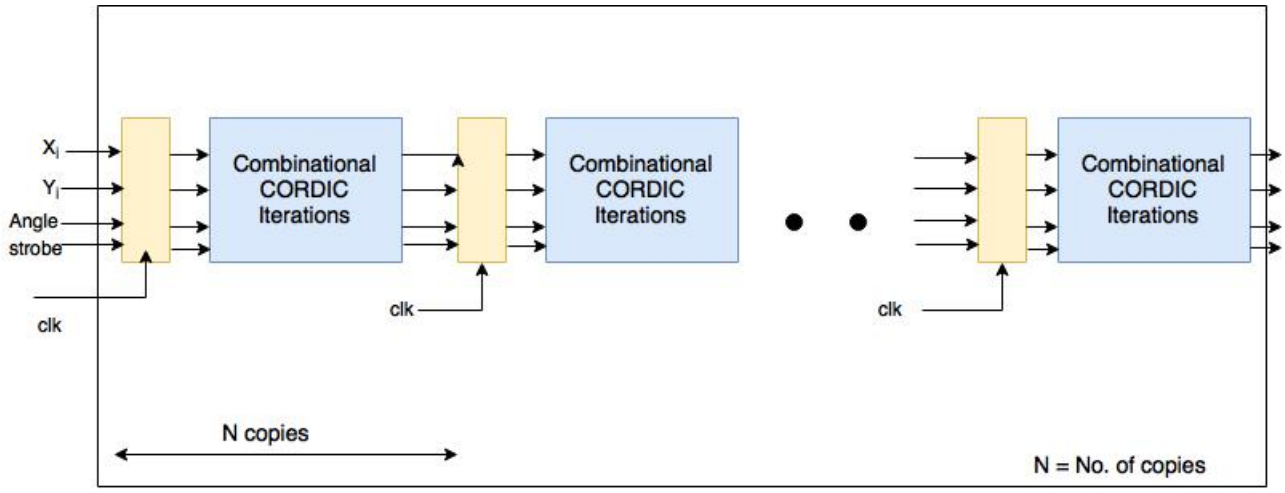

Figure 4.6: CORDIC ALGORITHM IMPLEMENTATION

The combinational cordic iterations contains the following basic structure

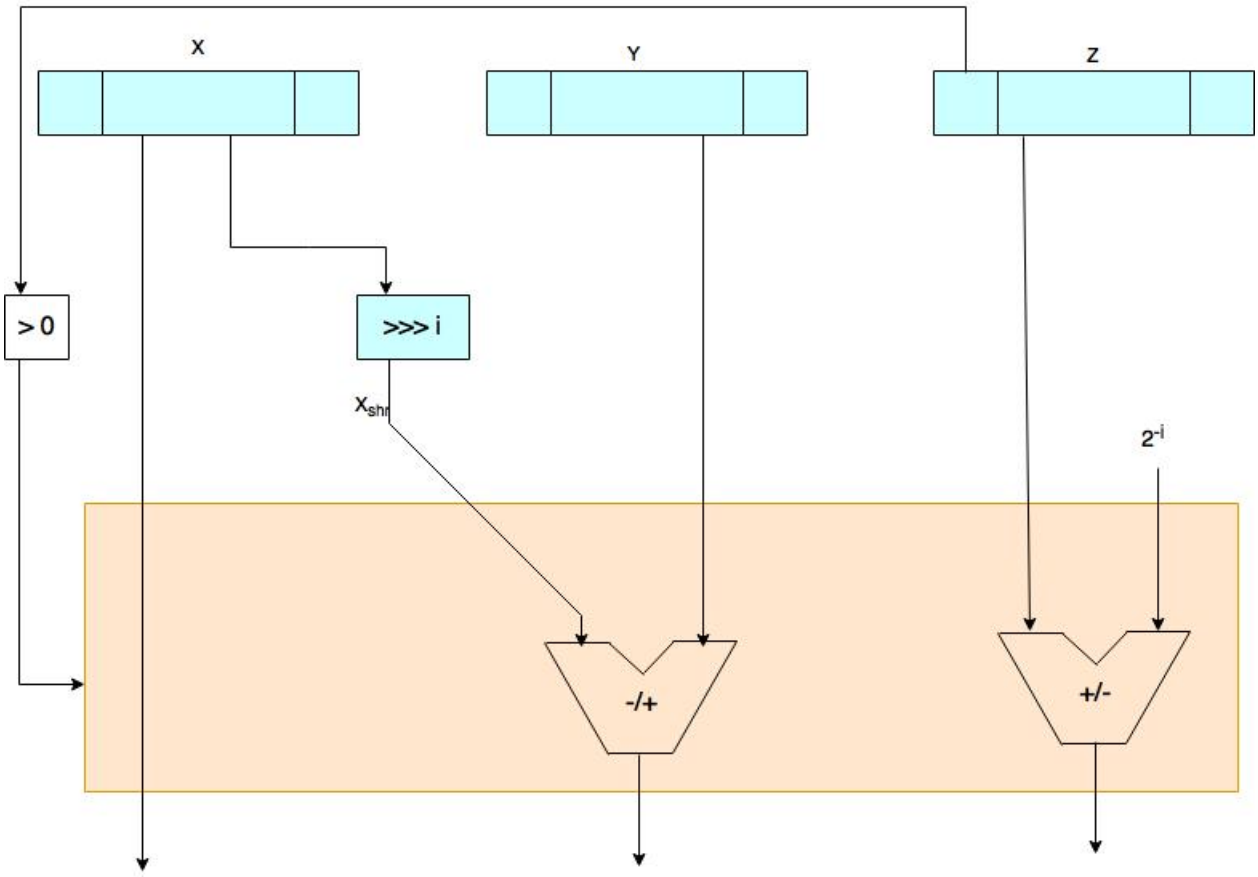

Figure 4.7: Basic structure of a processing element for one CORDIC iteration in Rotation mode of Linear coordinate system

4.1.5 Vector Mode of CORDIC algorithm in Linear Coordiante System

In our project we have implemented Vector mode in Linear coordinate system for finding division of two values. In vector mode, the values of alpha and mu are

$$\alpha_{m,i} = \frac{1}{\sqrt{m}} tan^{-1}(\sqrt{m}\frac{y}{x})$$

,

$$\mu_i = sign(y_i)$$

.

In Linear coordinate system, the value of m is 0 and alphas is found by taking

limit as m tends to zero and using L-hospital's rule we get division of two values i.e $\alpha = y/x$. Then equations are

$$x_{i+1} = x_i$$

$$y_{i+1} = y_i + \mu_i x_i 2^{-i}$$

$$z_{i+1} = z_i - \mu_i(2^{-i})$$

where $\mu_i = sign(y_i)$. Here we have to do pre-rotation of Inputs by making all inputs as positive since division works only in range zero to one.
The operation of cordic algorithms is shown below

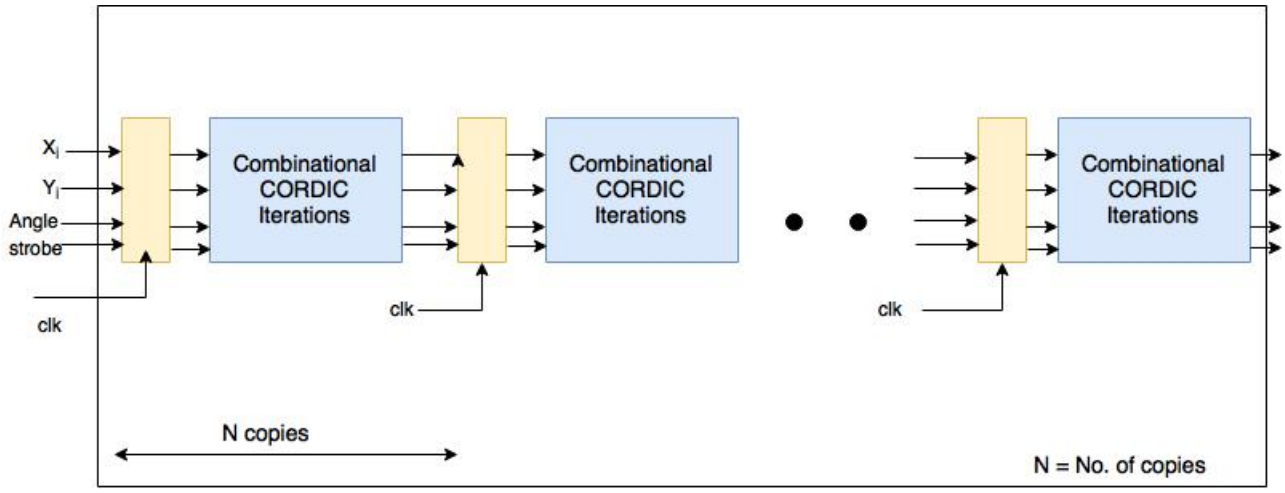

Figure 4.8: CORDIC ALGORITHM IMPLEMENTATION

The combinational cordic iterations contains the following basic structure

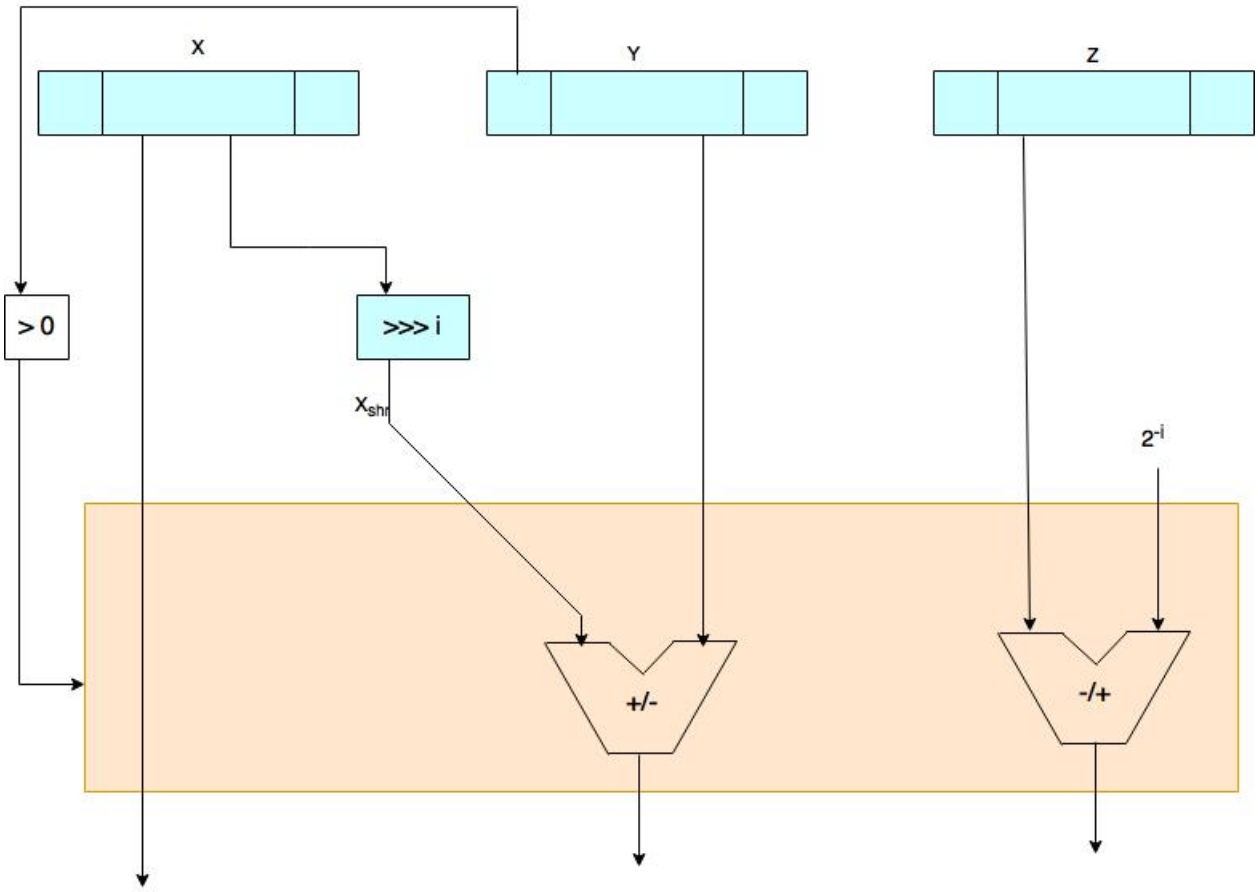

Figure 4.9: Basic structure of a processing element for one CORDIC iteration in vector mode of Linear coordinate system

4.1.6 Computational Accuracy

Due to the n fixed rotation angles, a given rotation angle z can only be approximated, resulting in an angle approximation error. Even if all other error sources are neglected, the accuracy of the outputs of the nth iteration is pricipally limited by the magnitude of the last rotation angle. So, the number of iterations should be chosen as $s_{m,n-1}$ = W for a desired output accuracy of W bits. This leads to n = W + 1 iterations.

A second error source is given by the finite precision of the involved variables which has to be taken into account for fixed point as well as floating point implementations. The format of the internal CORDIC variables is shown in below figure.

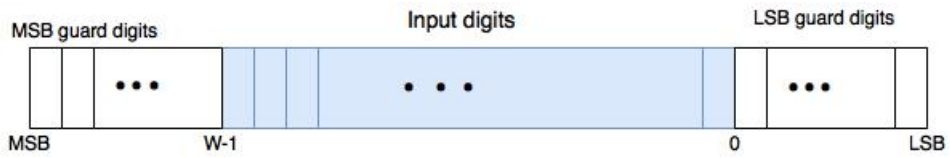

Figure 4.10: Format of the internal CORDIC variables

4.2 Generation of C/A code NCO

In this inputs are Phase per sample and strobe signals. Phase accumulator used to accumulate phase. shift signal starts the generation of LFSR bank consisting of two registers G1 and G2. Phase selector selects the Early and Late signals based on the msb of cumulative phase signal.

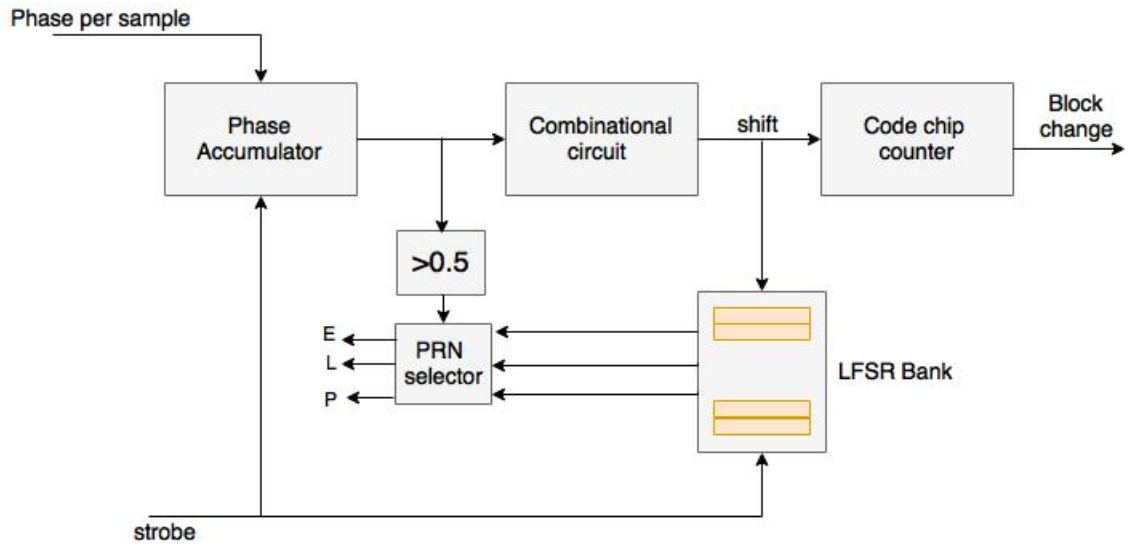

Figure 4.11: C/A CODE NCO

CHAPTER 5

Data Processing for Positioning

The position of the receiver can be known only we know the Arrival time of Data and Transmitted time.

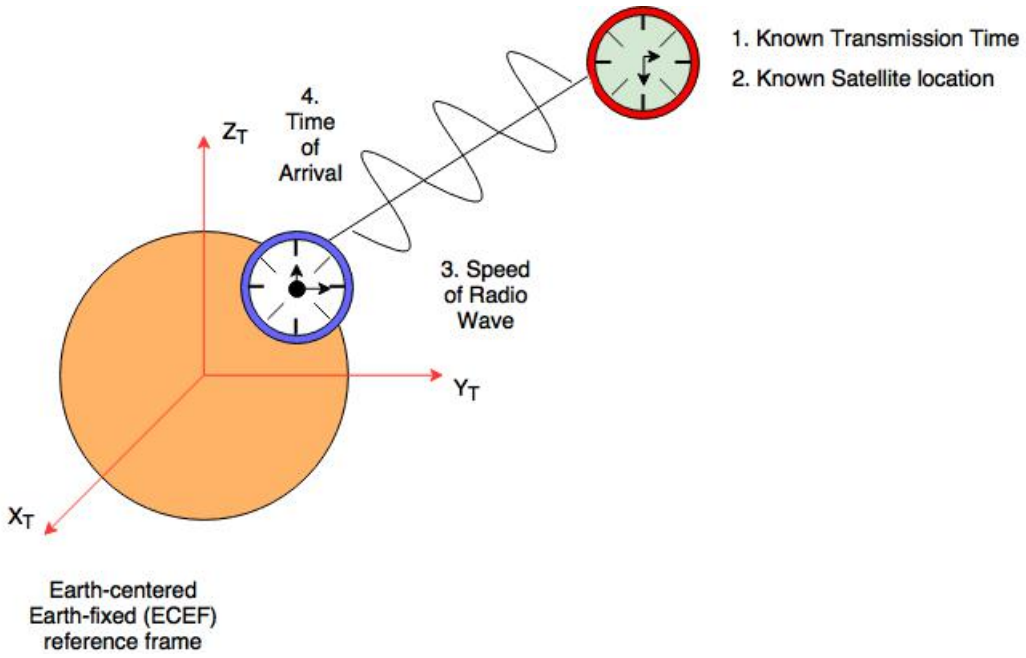

Figure 5.1: Data Transmission between GPS system and Receiver on Earth

5.1 Navigation Data Recovery

The output from the tracking loop is the value of the in-phase arm of the tracking block truncated to the values 1 and -1. Theoretically we could obtain a bit value every ms. However, we deal with noisy and weak signals, so a mean value for 20 ms is computed and truncated to -1 or 1. One navigation bit durates 20 ms.

The bit rate of the navigation data is 50 bps. The sample rate of the output from the tracking block is 1000 sps corresponding to a value each ms. Before the navigation data can be decoded, the signal from the tracking block must be

converted from 1000 sps to 50 bps. That is, 20 consecutive values must be replaced by only 1 value. This conversion procedure is referred to as *bit synchronization*.

5.2 Navigation Data Decoding

The navigation data decoding follows a scheme defined in the GPS Interface Control Document, ICD-GPS-200 (1991).

5.2.1 Location of Preamble

The first problem in the GPS navigation data decoding is to determine the location of the beginning of a subframe. The beginning of a subframe is marked by an 8-bit-long preamble. The pattern of the preamble is 10001011. Because of the Costas loop's ability to track the signal with a 180_0 phase shift, this preamble can occur in an inverted version 01110100. Naturally, these two bit patterns can occur anywhere in the received data so an additional check must be carried out to authenticate the preamble. The authentication procedure checks if the same preamble is repeated every 6 s corresponding to the time between transmission of two consecutive subframes.

The preamble search is implemented through a correlation. The first input to the correlation function is the incoming sequence of navigation data bits. This sequence is represented with -1's and 1's. The second input to the correlation function is the 8-bit preamble also represented with -1's and 1's. When using values -1 and 1 instead of 0 and 1, the output of the correlation function is 8 when the preamble is located and -8 when an inverted preamble is located.

The correlation function should give only six maximum correlation values as the sequence should contain six subframes. In addition to the large number of maximum correlation values, it also gives a minimum correlation value of -8 several times. The method for distinguishing which of the maximum correlation values that really is a beginning of a subframe includes the determination of the delay between consecutive maximum correlation values. Only if the delay between the maximum correlation values is exactly 6 s and the parity checks do not fail is the

beginning of a subframe indicated.

When the correct preambles are located, the data from each subframe can be extracted. If the correlation shows that the preamble is inverted, the entire navigation sequence must be inverted.

Due to the Doppler effect the length of the navigation bit can deviate from the exact value of 20 ms. Over a short time this length difference even may accumulate to a significant value. Therefore, a better solution is to look for a preamble in the original 1000-sps output from the tracking. The algorithm remains the same, but each bit in the reference preamble pattern is converted to 20 values (samples). Now the correlation peak will have a maximum value of 8 x 20 = 160 instead of 8. Simultaneously, this modified algorithm also finds bit transition time.

5.2.2 Extracting the Navigation Data

Every correct preamble marks the beginning of a navigation data subframe. Each of the subframes contains 300 bits divided into 10 30-bit words. The structure of the first two words of a subframe is shown in Figure below *Parity Check* Besides 24 bits of data, every 30-bit word contains a 6-bit parity. The parity is used to check for misinterpreted bits in the navigation data. The parity is computed through the equations in Table. Here $\otimes$ denotes the modulo-2 or exclusive OR operation.

D_1-D_{24} are the 24 data bits in a word, while D_{25}-D_{30} are the 6 parity bits in a word. The two bits denoted D^*_{29} and D^*_{30} are the last two parity bits from the previous word. When the navigation data are received, a parity check must be performed to test if the received data re interpreted correctly. *Time of Transmission* When the parity check has been performed successfully, the contents of the navigation data sequence can be decoded. The decoding is following the scheme from, ICD-GPS-200 (1991), which gives details of every word similar to what is shown in above The first important issue is to determine the time when the current subframe was transmitted from the GPS satellite.

The second word of every subframe is the so-called HOW that includes a truncated version of the TOW. This number is referred to as the Z-count. The Z-count is the number of seconds passed since the last GPS week rollover in units

of 1.5 s. The rollover happens at midnight between Saturday and Sunday. The maximum value of the Z-count is 403,199 as one week contains 604,800 s and 604,800 s/1.5 = 403,200 s. The Z-count value in the HOW is a truncated version containing only the 17 most significant bits (MSB). This truncation makes the Z-count increase in 6-s steps corresponding to the time between transmission of two consecutive navigation subframes.

The truncated Z-count value in the HOW corresponds to the time of transmission of the next navigation data subframe. To gest the time of transmission of the current subframe, the truncated Z-count should be multiplied by 6 and 6 s should be subtracted from the result.

Remaining Parameters The remaining parameters of the navigation data are also decoded according to ICD-GPS-200 (1991),e.g., the ephemeris parameters are decoded according to above Table. In the two's-complement, the sign bit (+ or -) occupies the MSB. The unit semicircle is multiplied with π to be converted into the unit radian.

The parameter IODE is short for Issue of Data Ephemeris. IODE is an eight-bit number that uniquely identifies the data set.

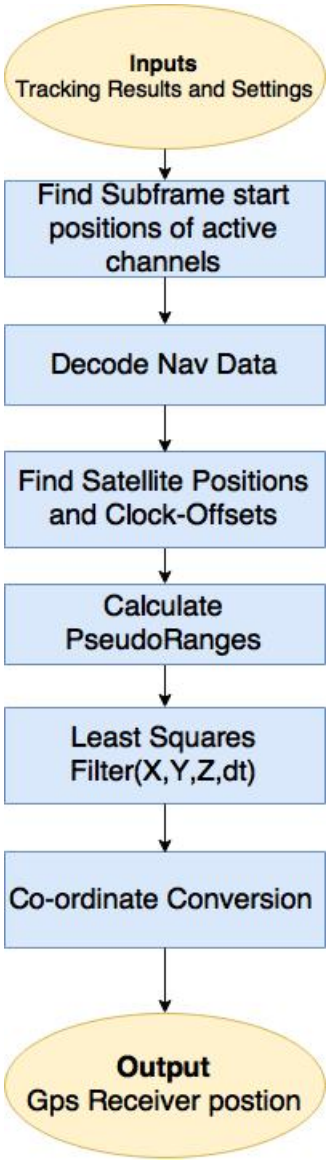

Figure 5.2: Flow Diagram of Receiver Postion Calculation

5.3 Computation of Satellite Position

This section connects Earth-centered and Earth-fixed (ECEF) coordinates X,Y,Z to a satellite position described in space by Keplerian orbit elements. The six Keplerian orbit elements(a,e,ω,Ω,i and μ constitute an important description of the orbit.

The X-axis points toward the intersection betweeen equator and the Greenwich meridian. For our purpose this direction can be considered fixed. The Z-axis coin-

cides with the spin of the Earth. The Y-axis is orthogonal to these two directions and forms a right-handed coordinate system.

The orbit plane intersects the Earth equator plane in the *nodal line*. The direction in which the satellite moves from south to north is called the *ascending node* K. The angle between the equator plane and the orbit plane is the inclination i. The angle at the Earth's center C between the X-axis and the ascending node K is called Ω; it is a right ascension. The angle at C between K and the perigee P is called *argument of perigee* ω; it increases counterclockwise viewed from the positive Z-axis.

The following figure shows a coordinate system in the orbital plane with origin at the Earth's center C. The ξ axis points to the perigee and the η-axis toward the descending node. The ζ is perpendicular to the orbit plane.

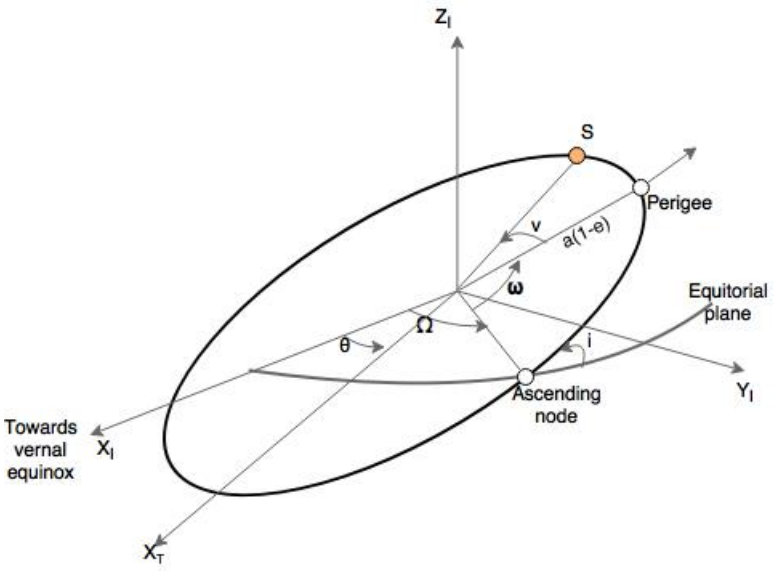

Figure 5.3: The Keplerian orbit elements

From above figure, we read the eccentric anomaly E and the true anomaly f. Also, immediately we have

$$\xi = rcosf = acosE - ae = a(cosE - e)$$

$$\eta = rsinf = bsinE = bsinE = a\sqrt{1-e^2}sinE$$

. Hence the position vector r of the satellite with respect to the center of the Earth C is

$$\mathbf{r} = \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} a(cosE - e) \\ a\sqrt{1-e^2}sinE \\ 0 \end{bmatrix} \tag{5.1}$$

Simple trignometry leads to the following expression for the norm:

$$\|r\| = a(1 - 2cosE)$$

In general, E varies with time t while a and e are nearly constant. (There are long and short periodic perturbations to e, only short for a). Recall that $\|r\|$ is the geometric distance between satellite S and the Earth center C=(0,0).

For later reference we introduce the mean motion n, which is the mean angular satellite velociy. If the period of one revolution of the satellite is T , we have

$$n = \frac{T}{2\pi} = \sqrt{\frac{GM}{a^3}}$$

The product GM has the value $3.986005X10^{14}m^3/s^2$. This value shall be used for computation of satellite positions (based on broadcast ephemeris), although more recent values of GM are available.

Let t_0 be the time the satellite passes perigee, so that $\mu(t) = n(t - t_0)$. Kepler's famous equation relates the mean anomaly μ and the eccentric anomaly

$$E = \mu + esinE$$

.

From Equation (1) we finally get

$$f = arctan(\frac{\eta}{\xi}) = arctan(\frac{\sqrt{1-e^2}sinE}{cosE - e})$$

.

By this we have connected the true anomaly *f*, the eccentric anomaly E and the mean anomaly μ . These relations are basic for every calculation of a satellite position.

It is important to realize that the orbital plane remains fairly stable in relation to the geocentric X, Y, Z-system. In other words, seen from space, the orbital plane remains fairly fixed in relation to the equator. The Greenwich meridian plane rotates around the Earth spin axis in accordance with Greenwich apparent sidereal time (GAST), that is, with a speed of approximately 24h/day. A GPS satellite performs two revolutions a day in its orbit having a speed of 3.87 km/s.

In the orbital plane the Cartesian coordinates of satellite S are given as

$$\begin{bmatrix} r_j^k cos f_j^k \\ r_j^k sin f_j^k \\ 0 \end{bmatrix} \tag{5.2}$$

where $\overset{k}{j} = \left\| r(t_j \right\|$ comes from equation 2 with a,e and E evaluated for $t = t_j$;

This vector is rotated into the X, Y, Z-coordinate system by the following sequence of 3D rotations:

$$R_3(-\Omega_j^k) R_1(-i_j^k) R_3(-\omega_j^k)$$

.

The matrix that rotates the XY-plane by φ and leaves the Z-direction alone, is

$$R_3(\varphi) = \begin{bmatrix} \cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

and similarly for a rotation about the X-axis:

$$R_1(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi & sin\varphi \\ 0 & -sin\varphi & cos\varphi \end{bmatrix} \tag{5.4}$$

.

Finally, the geocentric coordinates of satellite k at time t_j are given as

$$\begin{bmatrix} X^k(t_j) \\ Y^k(t_j) \\ Z^k(t_j) \end{bmatrix} = R_3(-\Omega_j^k)R_1(-i_j^k)R_3(-\omega_j^k) \begin{bmatrix} r_j^k cos f_j^k \\ r_j^k sin f_j^k \\ 0 \end{bmatrix} \tag{5.5}$$

.

However, GPS satellites do not follow the presented normal orbit theory. We have to use time-dependent, more accurate orbit values. They come to us as the so called broad ephemerides; see section 2. We insert those values in a procedure given below and finally we get a set of variables to be inserted into equation 8.

Obviously, the vector is time-dependent, and one speaks about the ephemeris of the satellite. These are the parameters values at a specific time. Each satellite transmits its unique ephemeris data.

The parameters chosen for description of the actual orbit of a GPS satellite and its perturbations are similar to the Keplerian orbital elements. The broadcast ephemerides are calculated using the immediate previous part of the orbit and they predict the following part of the orbit. The broadcast ephemerides are accurate to 1-2m. For geodetic applications, better accuracy is neede. One possibility is to obtain post-processed *precise ephemerides*, which are accurate at the dm-level.

An ephemeris is intended for use from the epoch t_{oe} of reference counted in seconds of the GPS week. It is nominally at the center of the interval over which the ephemeris is useful. The broadcast ephemerides are intended for use during this period. However, they describe the orbit to within the specified accuracy for 2 hours afterward. The broadcast ephemerides include the parameters in Table 2. The coefficients $C_\omega, C_r and C_i$ correct argument of perigee, orbit radius and orbit inclination due to inevitable perturbations of the theoretical orbit caused

by variations in the Earth's gravity field, albedo and sun pressure and attraction from sun and moon.

Given the transmit time t (in GPS time), the following procedure gives the necessary variables to use in equation 8: Time elapsed since t_{oe} $t_j = t - t_{oe}$

Mean anomaly at time t_j $\mu_j = \mu_0 + (\sqrt{GM/a^3} + \Delta n)t_j$

Iterative solution for E_j $E_j = \mu_j + esinE_j$

True anomaly $f_j = arctan\frac{\sqrt{1-e^2 sinE_j}}{cosE_j - e}$

Longitude for ascending node $\Omega_0 = \Omega_0 + (\dot{\Omega} - \omega_e)t_j - \omega_e t_{oe}$
where $\omega_e = 7.2921151467.10^{-5} rad/s$ Argument of perigee $\omega_j = \omega + f_j + C_{\omega c}cos2(\omega + f_j) + C_{\omega s}sin2(\omega + f_j)$

Radial distance $r_j = a(1 - ecosE_j) + C_{rc}cos2(\omega + f_j) + C_{rs}sin2(\omega + f_j)$

Inclination $i_j = i_0 + \dot{i}t_j + C_{ic}cos2(\omega + f_j) + C_{is}sin2(\omega + f_j)$

The mean Earth rotation is denoted ω_e. This algorithm is coded as satpos function. The function calculates the position of any GPS satellite at any time. It is fundamental to every position calculation.
After knowing the position of transmitted satellites, we use Triangulation method to find the position of receiver. The position of the receiver can be known only we know the bias time between the transmitted data and receiver.

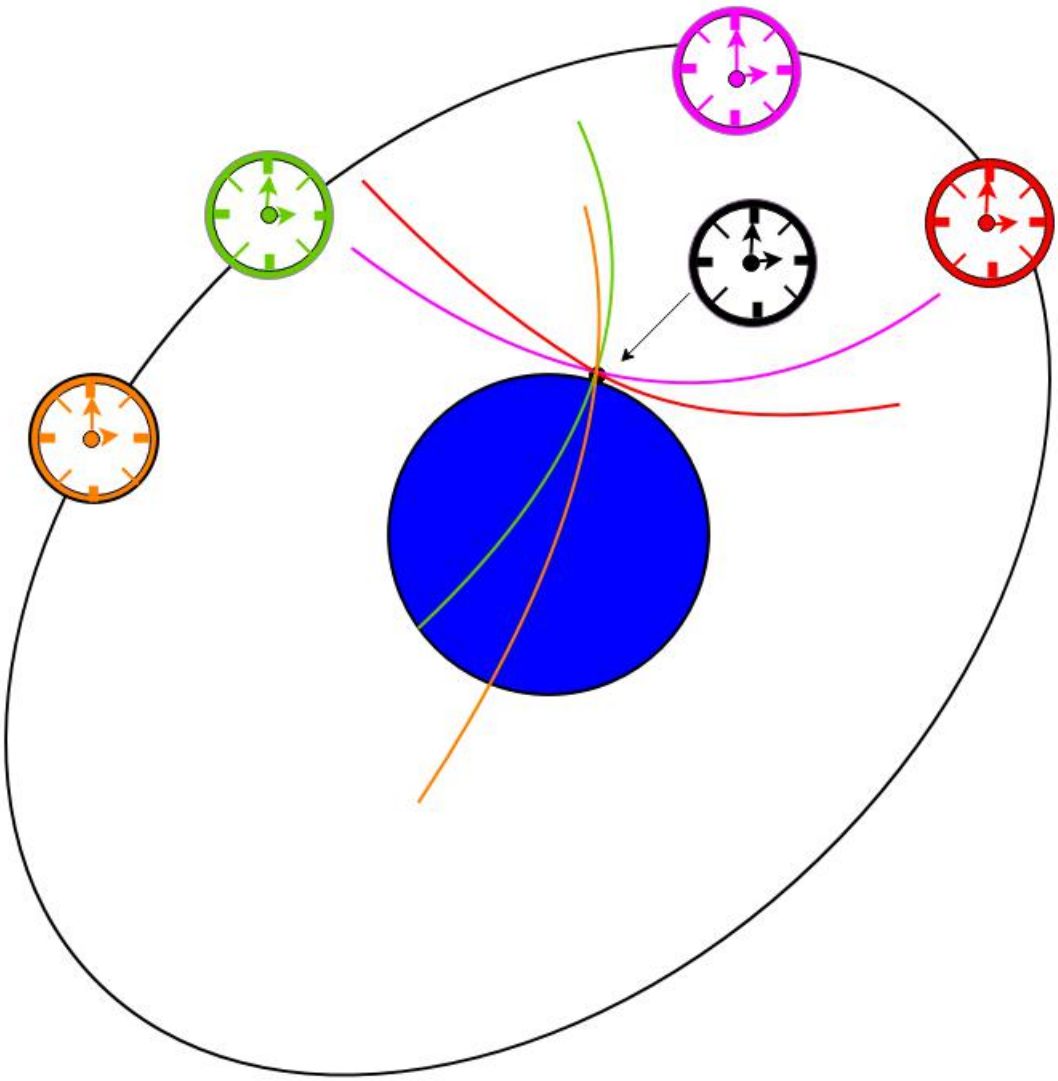

Figure 5.4: Triangulation diagram for finding position of Receiver

5.4 Pseudorange Estimation

Pseudorange estimations can be divided into two sets of computations. The first computational method is to find the initial set of pseudoranges and the second computational method is to keep track of the pseudoranges after the first set is estimated. Two computational methods are described below.

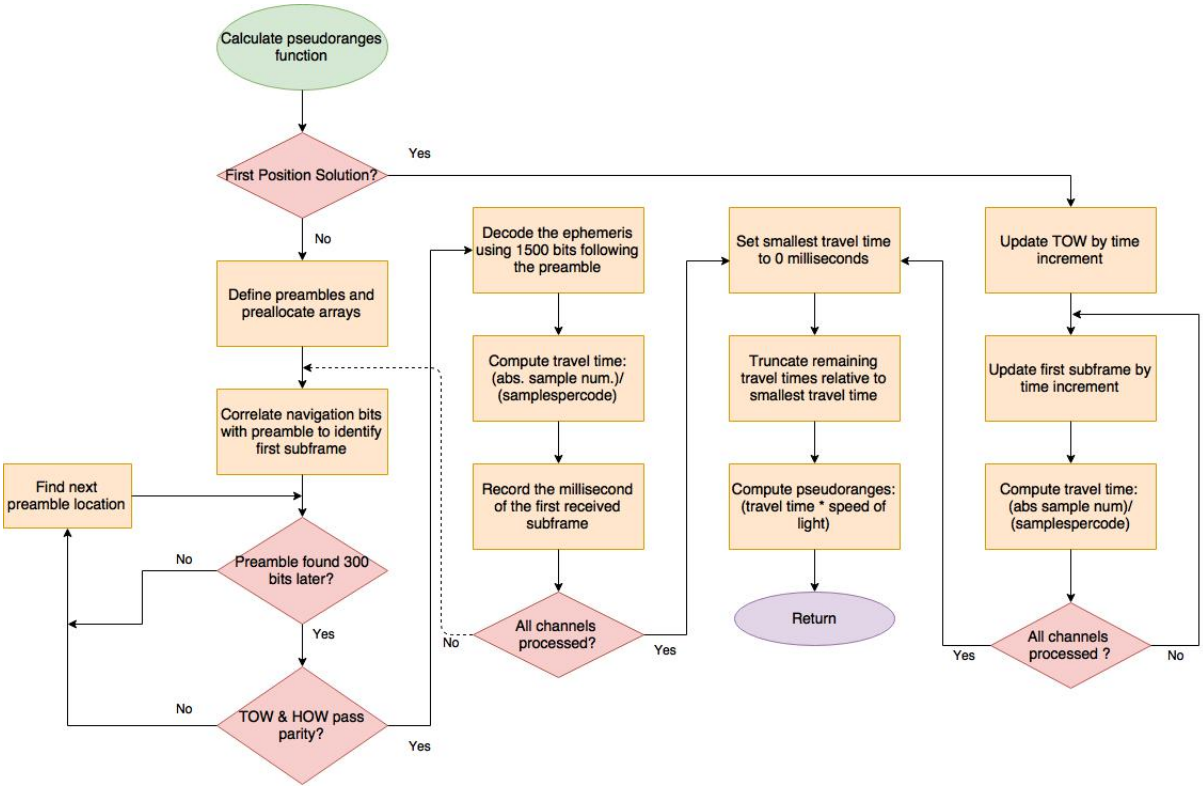

Figure 5.5: PseudoRange Flow Diagram

5.4.1 The Initial Set of Pseudoranges

To find the initial set of pseudoranges between the receiver and the tracked satellites, atleast 12s of data are needed. When the receiver has collected more than 12s of data, the data are correlated with the preamble from the TLM word. The TLM word is located at the start of each subframe.

The length of the preamble is 8 samples, so if the TLM word is present in the data, there will be a correlation value of 8 in 6 seconds and a correlation value corresponding to the preamble twice in the 12 s of data. This is used to check if

it really is a subframe that is found or it were just a bit combination similar to the preamble. The correlation is carried out for all the tracked satellites.

After the preamble is identified, the start of a subframe is found for all the available satellites. It is known that the travel time from the satellites to the Earth is 64-83 ms. This is used to set the initial pseudorange. The satellite closest to the Earth is the satellite with the earliest arriving subframe. In this case the satellite in channel 1 has a travel time of 68 ms. The travel time of the remaining channels is then computed with respect to channel 1. In the example this leads to travel times and pseudoranges for the four channels as listed in Table.

In the present example, the time resolution is 1 ms, which corresponds to a pseudorange of 300,000m. To make the pseudorange more useful, the tracking loop needs to find the start of the C/A code in the specific frame. This means that the time resolution is the sampling time, and in this case the sampling frequency is 38.192 MHz. This sampling frequency leads to a pseudorange accuracy of 8 m.

With the initial pseudoranges the receiver position can be computed. The output of that computation is a receiver position (X, Y, Z) and a receiver clock offset dt. The clock offset can be used to adjust the travel time of the reference satellite. In this case it was channel 1 with the travel time of 65 ms. And with this procedure the receiver can estimate the actual pseudoranges after two computations.

5.4.2 Estimation of Subsequent Pseudoranges

When computing the subsequent pseudoranges, keep track of two issues. The first issue is the difference in ms between the start of the subframe compared to the reference satellite. The second issue is the start of the C/A code, which gives the exact pseudorange for the channel.

When the receiver is computing the subsequent pseudoranges, the receiver moves all the indexes 500 ms. (The receiver moves the indexes 500 ms if the receiver is set up to compute positions 2 times per second) Then the start of the C/A code is found for all the new indexes for all the tracked channels. In this way it is possible to produce pseudoranges every millisecond and the receiver computes positions 1000 times per second.

5.5 Computation of Receiver Position

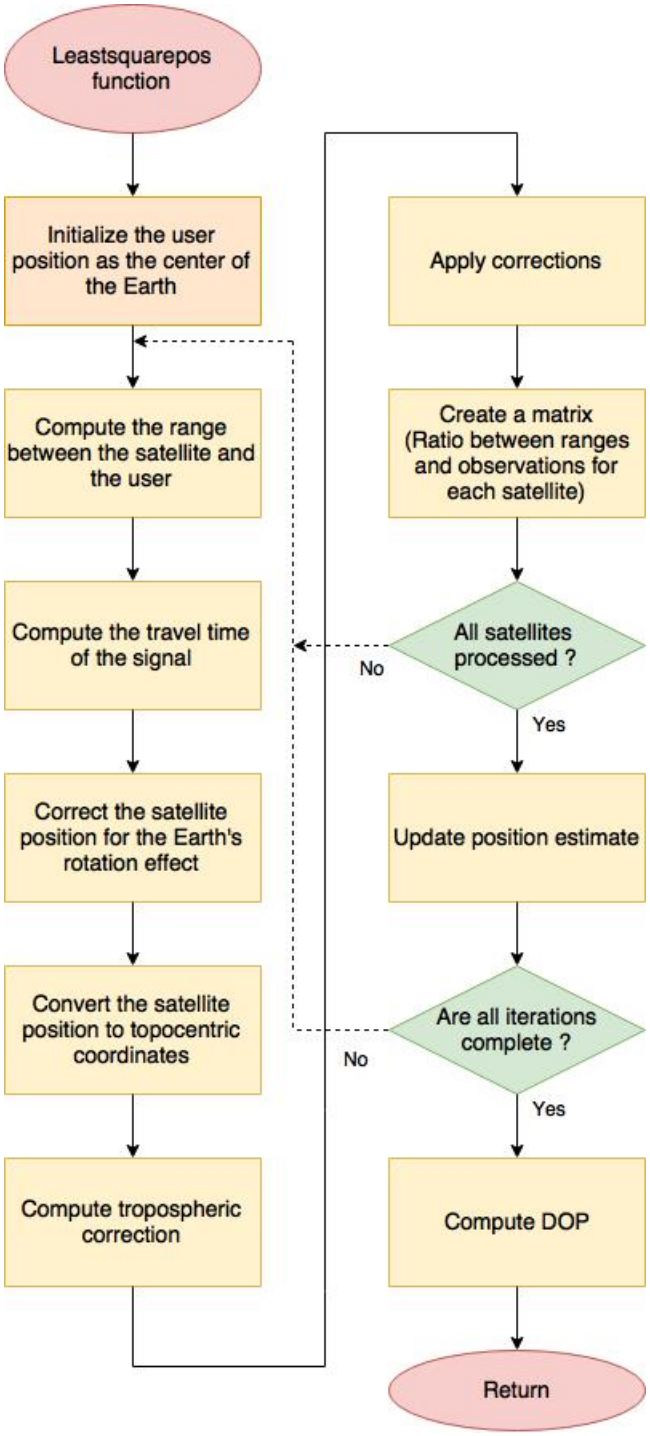

Figure 5.6: Flow Diagram of Least Square Position method for finding Position

5.5.1 Time

The travelling time between satellite k and receiver i is denoted τ_i^k. Let c denote the velocity of light in vacuum and the pseudorange P_i^k/c is defined as

$$t_i - t^k = \tau_i^k = P_i^k/c$$

.

Let any epoch in GPS time (GPST) be called t^{GPS}. The clock at satellite k and the clock at receiver i do not run perfectly aligned with GPST. Thus, we introduce clock offsets defined as

$$t_i = t^{GPS} + dt_i,$$

$$t^k = (t_i - \tau_i^k)^{GPS} + dt^k$$

In the latter equation we substitute $dt^k = a_0 + a_1(t^k - t_{oe}) + ...$ as given from the ephemeris:

$$(t_i - \tau_i^k)^{gps} = t^k - (a_0 + a_1(t^k - t_{oe}) + ...)$$

The left term is used as argument when computing the satellite position. Rearranging equation 9,we get

$$t^k = t_i - P_i^k/c.$$

One way of using this equation is to consider t_i as given. That is the epoch time as defined in terms of the receiver clock. The pseudorange P_i^k is likewise known as an observable. Hence t^k can be computed and after correcting for the satellite clock offset dt^k we obtain the transmit time in GPST. This is the procedure used for hardware receivers.

For software receivers the situation is a little different. The time t_{common} common to all pseudorange observations is defined as the time of transmission at the satellites. Hence the computation of position of satellite k is done at

$$t^k = t_{common} - dt^k.$$

The only "receiver time" used is the relative time of reception from each of the satellites and which makes the individual pseudorange.

A consequence of this time definition is that the computed satellite coordinates immediately refer to the ECEF system, and therefore satellite coordinates are not to be rotated about the Z-axis by an angle equal to the travel time times the Earth's rotation rate.

Linearization of the Observation Equation

The most commonly used algorithm for position computations from pseudoranges is based on the least-squares method. This method is used when there are more observations than unknowns. This section describes how the least-squares method is used to find the receiver position from pseudoranges to four or more satellites.

Let the geometrical range between satellite k and receiver i be denoted ρ_i^k, let c denote the speed of light, let dt_i be the receiver clock offset, let dt^k be the satellite clock offset, let T_i^k be the tropospheric delay, let I_i^k be the ionospheric delay, and let e_i^k be the observational error of the pseudorange. Then the basic observation equation for the pseudorange P_i^k is

$$P_i^k = \rho_i^k + c(dt_i - dt^k) + T_i^k + I_i^k + e_i^k.$$

The geometrical range ρ_i^k between the satellite and the receiver is computed as

$$\rho_i^k = \sqrt{(X^k - X_i)^2 + (Y^k - Y_i)^2 + (Z^k - Z_i)^2} + c(dt_i - dt^k) + T_i^k + I_i^k + e_i^k.$$

The tropospheric delay T_i^k is computed from a priori model that is coded as tropo; the ionosphere delay I_i^k may be estimated from another a priori model, the coefficients of which are part of the broadcast ephemerides. The equation contains four unknowns X_i, Y_i, Z_i and dt_i; the error term e_i^k is minimized by using the least-squares method. To compute the position of the receiver, atleast four pseudoranges are needed.

Equation 17 is nonlinear with respect to the receiver position (X_i, Y_i, Z_i), so the equation has to be linearized before using the least-squares method.

We analyze the nonlinear term in equation 17 as

$$f(X_i, Y_i, Z_i) = \sqrt{(X^k - X_i)^2 + (Y^k - Y_i)^2 + (Z^k - Z_i)^2}.$$

Linearization starts by finding an initial position for the receiver: $(X_{i,0}, Y_{i,0}, Z_{i,0})$. This is often chosen as the center of the Earth (0,0,0).
The increments $\Delta X, \Delta Y, \Delta Z$ are defined as

$$X_{i,1} = X_{i,0} + \Delta X_i,$$

$$Y_{i,1} = Y_{i,0} + \Delta Y_i,$$

$$Z_{i,1} = Z_{i,0} + \Delta Z_i.$$

They update the approximate receiver coordinates. So the Taylor expansion of $f(X_{i,0} + \Delta X_i, Y_{i,0} + \Delta Y_i, z_{i,0} + \Delta Z_i)$ is

$$f(X_{i,1}, Y_{i,1}, Z_{i,1}) = f(X_{i,0}, Y_{i,0}, Z_{i,0}) + \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial X_{i,0}} \Delta X_i + \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Y_{i,0}} \Delta Y_i +$$

$$\frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Z_{i,0}} \Delta Z_i.$$

The above equation includes only first-order terms; hence the function determines an approximate position. The partial derivatives in above equation are

$$\frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial X_{i,0}} = -\frac{X^k - X_{i,0}}{\rho_i^k},$$

$$\frac{\partial f(X_{i,0}, Y_{i,0} Z_{i,0})}{\partial Y_{i,0}} = -\frac{Y^k - Y_{i,0}}{\rho_i^k},$$

$$\frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Z_{i,0}} = -\frac{Z^k - Z_{i,0}}{\rho_i^k}.$$

Let $\rho^{k_{i,0}}$ be the range computed from the approximate receiver position; the first-order linearized observation equation becomes

$$P_i^k = \rho_{i,0}^k - \frac{X^k - X_{i,0}}{\rho_{i,0}^k}\Delta X_i - \frac{Y^k - Y_{i,0}}{\rho_{i,0}^k}\Delta Y_i - \frac{Z^k - Z_{i,0}}{\rho_{i,0}^k}\Delta Z_i + c(dt_i - dt^k) + T_i^k + I_i^k + e_i^k,$$

where we explicitly have

$$\rho_{i,0}^k = \sqrt{(X^k - X_{i,0})^2 + (Y^k - Y_{i,0})^2 + (Z^k - Z_{i,0})^2}.$$

5.5.2 Using the Least-Squares Method

A least-squares problem is given as a system Ax = b with no solution. A has m rows and n columns, with m ¿ n; there are more observations $b_1, ..., b_m$ than free parameters $x1, ..., x_n$. The best choice, we will call it $\hat{x}$, is the one that minimizes the length of the error vector $\hat{e} = b - A\hat{x}$. If we measure this length in the usual way, so that $\|e\|^2 = (b - Ax)^T(b - Ax)$ is the sum of squares of the m separate errors, minimizing this quadratic gives the normal equations

$$A^T A\hat{x} = A^T b or \hat{x} = (A^T A)^{-1} A^T b$$

and the error vector is

$$\hat{e} = b - A\hat{x}.$$

The covariance matrix for the parameters $\hat{x}$ is

$$\Sigma_{\hat{x}} = \hat{\sigma}_0^2 (A^T A)^{-1} with \hat{\sigma}_0^2 = \frac{\hat{e}^T \hat{e}}{m - n}.$$

The linearized observation equation can be rewritten in a vector formualation

$$P_i^k = \rho_{i,0}^k + \begin{bmatrix} -\frac{X^k - X_{i,0}}{\rho_{i,0}^k} & -\frac{Y^k - Y_{i,0}}{\rho_{i,0}^k} & -\frac{Z^k - Z_{i,0}}{\rho_{i,0}^k} & 1 \end{bmatrix} \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \\ cdt_i \end{bmatrix} - cdt^k + T_i^k + I_i^k + e_i^k.$$

We rearrange this to resemble the usual formulation of a least-squares problem Ax = b:

$$\begin{bmatrix} -\frac{X^k - X_{i,0}}{\rho_{i,0}^k} & -\frac{Y^k - Y_{i,0}}{\rho_{i,0}^k} & -\frac{Z^k - Z_{i,0}}{\rho_{i,0}^k} & 1 \end{bmatrix} \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \\ cdt_i \end{bmatrix} = P_i^k - \rho_{i,0}^k + cdt^k - T_i^k - I_i^k - e_i^k.$$

A unique solution cannot be found from a single equation. Let $b_i^k = P_i^k - \rho_{i,0}^k + cdt^k - T_i^k - I_i^k - e_i^k$. Then the final solution comes from

$$Ax = \begin{bmatrix} -\frac{X^1 - X_{i,0}}{\rho_{i,0}^1} & -\frac{Y^1 - Y_{i,0}}{\rho_{i,0}^1} & -\frac{Z^1 - Z_{i,0}}{\rho_{i,0}^1} & 1 \\ -\frac{X^2 - X_{i,0}}{\rho_{i,0}^2} & -\frac{Y^2 - Y_{i,0}}{\rho_{i,0}^2} & -\frac{Z^2 - Z_{i,0}}{\rho_{i,0}^2} & 1 \\ -\frac{X^3 - X_{i,0}}{\rho_{i,0}^3} & -\frac{Y^3 - Y_{i,0}}{\rho_{i,0}^3} & -\frac{Z^3 - Z_{i,0}}{\rho_{i,0}^3} & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ -\frac{X^m - X_{i,0}}{\rho_{i,0}^m} & -\frac{Y^m - Y_{i,0}}{\rho_{i,0}^m} & -\frac{Z^m - Z_{i,0}}{\rho_{i,0}^m} & 1 \end{bmatrix} \begin{bmatrix} \Delta X_{i,1} \\ \Delta Y_{i,1} \\ \Delta Z_{i,1} \\ cdt_{i,1} \end{bmatrix} = b - e.$$

If $m \geq 4$, there is a unique solution: $\Delta X_{i,1}, \Delta Y_{i,1}, \Delta Z_{i,1}$. This has to be added to the approximate receiver position to get the next approximate position:

$$X_{i,1} = X_{i,0} + \Delta X_{i,1},$$

$$Y_{i,1} = Y_{i,0} + \Delta Y_{i,1},$$

$$Z_{i,1} = Z_{i,0} + \Delta Z_{i,1}.$$

The next iteration restarts from 26 to 29 equations with i,0 substituted by i,1. These iterations continue until the solution $\Delta X_{i,1}, \Delta Y_{i,1}, \Delta Z_{i,1}$ is at meter level. Often two to three iterations are sufficient to obtain that goal.

5.5.3 Coordinate Transformations

The immediate result of a satellite positioning is a set of X-, Y-, Z-values, which we often want to convert to latitude φ, longitude λ and height h. We start by introducing the relation between Cartesian (X,Y,Z) coordinates and geographical (φ, λ, h) coordinates:

$$X = (N_\varphi + h)cos\varphi cos\lambda,$$

$$Y = (N_{\varphi+h}cos\varphi sin\lambda),$$

$$Z = ((1-f)^2 N_\varphi + h)sin\varphi.$$

The reference ellipsoid is the surface given by $X^2 + Y^2 + (\frac{a}{b})^2 Z^2 = a^2$. The radius of curvature in the prime vertical(which is the vertical plane normal to the astronomical meridian) is given as

$$N_\varphi = \frac{a}{\sqrt{1 - f(2-f)sin^2(\varphi)}}.$$

CHAPTER 6

Results

Synthesis and Implementation of the design is done on Virtex 7(VC707) FPGA Evaluation board.

Utilization report after Implementation is as follows

Site Type	Used	Fixed	Available	Util%
Slice LUTs	7751	0	303600	2.55
LUT as Logic	7726	0	303600	2.54
LUT as Memory	25	0	130800	0.02
LUT as Distributed RAM	0	0		
LUT as Shift Register	25	0		
Slice Registers	3489	0	607200	0.57
Register as Flip Flop	3489	0	607200	0.57
Register as Latch	0	0	607200	0.00
F7 Muxes	32	0	151800	0.02
F8 Muxes	0	0	75900	0.00

Figure 6.1: Utilization report after Implementation

Timing report of the design is shown below:

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.295ns  (required time - arrival time)
  Source:                 CART2POL/STAGES[9].x_reg[10][6]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@4.550ns period=9.100ns})
  Destination:            DIV/STAGES[0].y_reg[1][34]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@4.550ns period=9.100ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            9.100ns  (clk rise@9.100ns - clk rise@0.000ns)
  Data Path Delay:        8.818ns  (logic 5.062ns (57.407%)  route 3.756ns (42.593%))
  Logic Levels:           50  (CARRY4=43 LUT1=2 LUT2=1 LUT5=3 LUT6=1)
  Clock Path Skew:        -0.028ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    4.775ns = ( 13.875 - 9.100 )
    Source Clock Delay      (SCD):    5.144ns
    Clock Pessimism Removal (CPR):    0.340ns
  Clock Uncertainty:      0.035ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.000ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns
```

Figure 6.2: Timing Report

There are multiple register to register paths and the maximum delay path is between cart2pol register(cordic for finding absolute value) to Div register(cordic for calculating division) as shown in above figure. Above timing report is obtained with a clock of 9.1ns(109.89 MHz), which provides a slack of 0.295ns.
The data path delay is 8.818ns which concludes that the design can operate at a maximum frequency of 113.40MHz.

CHAPTER 7

Conclusion, Summary and Future work

7.1 Conclusion

Design and implementation of GPS L1 signal tracking is done on Virtex 7(VC707) FPGA Evaluation board. Data processing of Navigation data is done through software for getting the receiver position.

7.2 Summary

A GPS receiver has to perform three main steps to decode the navigation data: *Acquisition, Tracking and Navigation Data Processing.* The GPS receiver need data from atleast four GPS satellites to find its location. Sampling rate is set at 4 MHz satisfying Nyquist criterion.

Tracking module is a digital implementation of a Costas Loop and Delay Locked Loop and it involves three real time correlations with Early, Late and Prompt replicas of C/A code.Stripping off C/A code and Doppler and keeping track of phase gives NAV BITS which have very low frequency (50 HZ) and they have to be accumulated in frames. With the collected data using Tringulation method and iterative approach user position is calculated.The Navigation Data Processing is done through software. The rapid FPGA prototyping, design, verification and Processing of Navigation data for finding the position are the main focus of the thesis.

7.3 Future work

Integration of all modules has to be done. This project is implemented for GPS system, after some modifications it can serve both as a GPS and IRNSS receiver. Algorithms related to Tracking of weak signals has to be examined.

References

1. K. Borre, D. M. Akos, N. Bertelsen, P. Rinder and S. H. Jensen, A software defined GPS and Galileo receiver:a single-frequency approach. Springer science and Business Media, 2007.

2. J. S. Walther, A unified algorithm for elementary functions, in Proceedings 38th Spring Joint Computer Conference, Atlantic City, New Jersey, 1971, pp. 379385

3. Navstar GPS Interface Specification (IS-GPS-200)

4. E. Kaplan and C. Hegarty, Understanding GPS: principles and applications. Artech house, 2005