MASTER THESIS

To achieve the academic degree
MASTER

Virtual Prototypes of Different Galileo/GPS Correlation Engines in System-C

Done by
Ahmet Günes
supervised by
Prof. Dr.-Ing. Axel Hunger

Computer Science and Communications Engineering
Faculty of Engineering
University of Duisburg-Essen
Duisburg, 15.10.2006

Thesis Assignment

International Study of Engineering in 'Computer Science & Communication Engineering'

By:	Ahmet Günes
Supervised by:	Prof. Dr.-Ing. Axel Hunger
Topic:	Virtual Prototypes of different Galileo/GPS Correlation Engines in System-C

Nature of the task:

Europe is currently developing the novel satellite navigation system Galileo, whereas GPS is the existing US system. While GPS currently transmits only one civil signal, Galileo will offer five different services with ten different signals in four frequency bands. Galileo deploys advanced signal structures with higher chip rates, improved spreading codes, and novel modulation schemes. Both systems combined provide the best benefit for many emerging applications, such as location based services and E-911. Galileo/GPS satellite navigation receivers determine their position by measuring signal propagation delay differences between signals transmitted by satellites from different positions in earth orbit. The delay differences combined with the knowledge of the satellite positions allows calculating the receiver position. These delay differences are obtained by synchronizing locally generated direct sequence spread spectrum codes with the transmitted codes from the satellites. Synchronization can be achieved by various methods of correlation.

The target of this thesis is to develop virtual prototypes of different correlation engines for the Galileo and GPS signals. They shall be implemented in System-C and the quantization resolutions shall be adjustable. Common to all versions of the correlation engine is the digital downconversion, despreading, and coherent integration. The first version then continues with the noncoherent integration, while the second version applies the novel differential correlation. The third version evaluates the phase of the differential correlation result to adjust the reference frequency of the digital downconversion stage. The fourth version additionally adjusts the duration of the coherent integration interval based on the estimated frequency deviation. The fifth version furthermore rotates the intermediate differential correlation results to compensate the resulting phase variations. The statistical properties at the outputs of the signal processing stages shall then be evaluated for different quantization resolutions.

_____

Supervisor

Second Examiner:

D E C L A R A T I O N

I hereby declare, that I have compiled this document without outside help other than the official direction given to me by my supervisor. All sources used, as well additional means of help are properly referenced. Verbatim use of text abstracts and adopted pictures and diagrams have been acknowledged in every case.

Duisburg, _____ _____

Signature of candidate

Abstract

The introduction of the Galileo navigation system will improve and foster the civil use of the satellite navigation systems. Since Galileo and GPS use a common carrier frequency for signal transmission, using a dual-mode receiver is possible. This would also improve the accuracy of the positioning as the higher the number of satellites in sight, the higher the accuracy. Galileo is announced to be fully functional in 2010, and naturally there are no Galileo receivers in the market yet. The implementation of a virtual prototype of a Galileo/GPS receiver can help to analyse and develop the potential hardware.

For the implementation of such a dual-mode receiver, there are different approaches possible. The conventional algorithm is already used for GPS receivers. The novel approach, differential correlation algorithm, has a sensitivity gain of 1.5dB in average. Both of the correlation techniques are implemented in this work. SystemC is used for the implementation of the algorithms. SystemC offers flexibility for further use and being based on the C++ programming language, it allows fast simulations.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AWGN	Additive White Gaussian Noise
PRN	Pseudo Random Noise Sequence
BPSK	Binary Phase Shift Keying
ASIC	Application specific integrated circuit
C/A code	Code acquisition code
BOC	Binary offset carrier
XOR	exclusive-or
LSB	Least significant bit
MSB	Most significant bit
Mcps	chip rate in terms of millions (M refers to Mega)

1 Introduction

1.1 GPS and Galileo Navigation Systems

GPS is the first navigation system and along with the GLONASS system they are the only global satellite navigation systems. Although designed for military use in the beginning, the importance of the system for public is seen and it serves to both now. GPS, composed of ca 30 satellites orbiting the earth at 20200km. GPS satellites transmit several signal components through 2 frequencies.

Although other local navigation systems are in use or being planned, none of them has a scale as large as the Galileo Navigation System which is developed by European Union. The system is expected to be operational in 2010. The two major advantages of Galileo over GPS are, it is more precise than GPS and it has a service guarantee. It is composed of 30 satellites including the spare satellites. The civilian signal codes of Galileo were published only half a year ago, in April 2006 and the codes may be subject to change.

1.2 Overview of Contents

Chapter 2 provides an introduction of the GPS and Galileo navigation systems and a closer look at the signal structure of these two. It also introduces the development media: SystemC. The SystemC tools which are used later on are explained deeper. Finally, an overview of the receiver algorithms is given. Chapter 3**Error! Reference source not found.** explains the receiver algorithms from a theoretical point of view. The functions of the components in the architectures are introduced in detail. Chapter 4 shows how the components are created using the SystemC-based program. In chapter 5, the programs are used to analyse the architectures and to optimize the components of the architectures. In the conclusion, chapter 6, the thesis is summarized and some possible improvements of the architecture are mentioned.

2 Literature Review

2.1 GPS and Galileo Signal Structures

GPS uses C/A codes. Each satellite transmits a different code, enabling the differentiation and recognition of the satellites. The C/A codes consist of 1023 chips and have a period of 1ms, giving 1.023Mcps chip rate. The codes of GPS are Gold codes. The autocorrelation and cross-correlation of these codes have a few low values and a high correlation peak [1]. This property of the codes makes it easier to determine the chip delay of the codes and eases the synchronization of the codes.

Autocorrelation results of a C/A GPS code				
Frequency of Occurrence	12.5%	12.5%	75%	Once (peak)
Amplitude (value)	-65	63	-1	1023

Table 1 : The autocorrelation results of a GPS Gold code.

Another advantage of the Gold codes is its generation. The Gold codes are derived from a class of codes so called Pseudo Random Noise sequences [2]. These sequences look like a random sequence having half of the chips 0 and the other half 1. Also the occurrence of the sequential patterns of the chips of the code is proportional. A quarter of the chips follow the 00 pattern and another quarter follow 11 pattern. But the code is not really random and the name PRN reflects this fact. The whole sequence can be generated by using two 10-bit shift registers and some XOR operations. This reduces the hardware requirements of the local code generation in the receiver greatly.

On the other hand, the Galileo codes are selected among the purely random sequences. There is no guarantee that the codes will be generated in a similar manner as GPS codes. Even if there is a short cut for the generation, as mentioned in section 1.1 the codes may change any time, making the found generation method obsolete. The Galileo codes have different lengths all being multiples of 1023. The name of the signal components and some properties are shown in Table 2 and are found in [3]. This work will perform multiple simulations with the L1-B signal component.

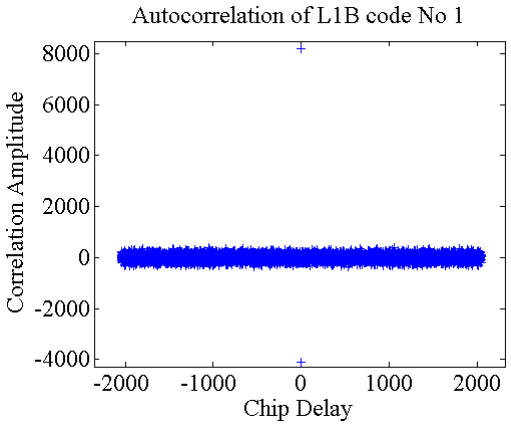

Figure 1 : The autocorrelation function of a Galileo signal component.

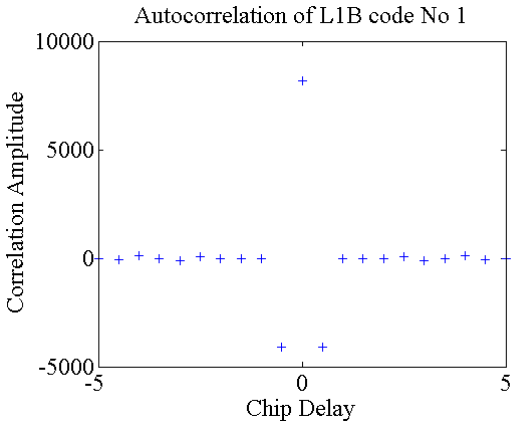

Figure 2 : The autocorrelation function of the same signal component in Figure 1, in the neighborhood of the peak.

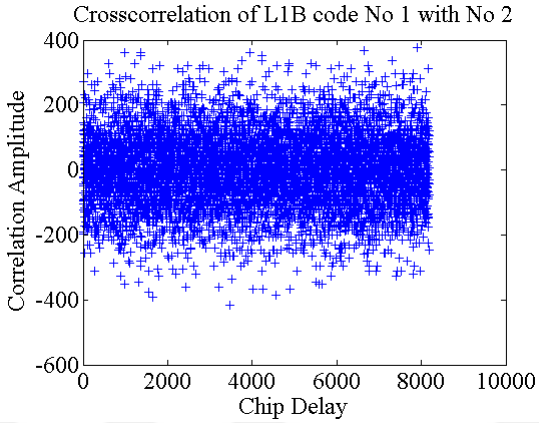

Figure 3 : The cross-correlation function of two Galileo signal components.

Channel	***Primary Code Length (chips)***	***Number of hex. Symbols***	***Number of Filled up zeros***	***Number of defined codes***
E5a-I	10230	2558	2	50
E5a-Q	10230	2558	2	50
E5b-I	10230	2558	2	50
E5b-Q	10230	2558	2	50
L1-B	4092	1023	0	50
L1-C	4092	1023	0	50

Table 2 : Galileo channels and signal components.

The autocorrelation and cross-correlation of the signal components of Galileo has also low values except the correlation peak. A major difference of the Galileo signals from the GPS signals is that they are BOC(1,1) modulated. This results in great negative values in the neighbourhood of the correlation peak as seen in Figure 1 and Figure 2. The negative peaks make the system synchronization easier than normal because

searching for 3 peaks, i.e. a triangle, is a better synchronization recipe than 1 peak. Galileo L1-B signal component has a period of 4ms, with 1.023MHz chip frequency. Both GPS and Galileo transmitted signals are BPSK modulated.

2.2 SystemC

SystemC is a system design class library for the C++ programming language with open source licence. Because of its C++ heritage, it has a wide application spectrum. SystemC operates in 2 phases: Elaboration and execution. The border between these two phases is the *sc_start()* function call under *sc_main()*. The execution of statements above *sc_start()* is called the elaboration. During this phase, the SystemC simulation kernel organizes the components and processes. When thinking in terms of hardware simulation, components are the hardware blocks with links between the block. Processes are the functions of these blocks. For example, defining an output port means creating a component, but writing on this port is a process. If the components work concurrently, the clocks and other tools of concurrency must be created and linked in the elaboration phase. Then, the illusion of concurrency is created by the SystemC simulation kernel. The clocks are the only synchronization tools in this work. In other terms, the elaboration is when the virtual hardware structure is created. At the end of the elaboration phase, the hardware is ready to operate. After the *sc_start()* command, the execution starts. The system clock starts working and via clock cycles the system is stimulated and runs. Along one clock cycle, all tasks which start with this clock cycle are calculated and all tasks are executed, results are updated and finally the system time advances. Execution carries on until there are either no more tasks to do or until time limits which can be set in *sc_start()* (*sc_start*(*end_time*,*time_unit*)) exceed or until the *sc_stop()* function call is evoked. After execution, memory cleanup takes place.

SystemC supports a hierarchical structure. The components are created, linked and further organized. All this connected structure is placed on a bed. This bed is the major structure supporting the components with clocks and other important additional

components. The relation between the classes and the program body is shown in Figure 4.

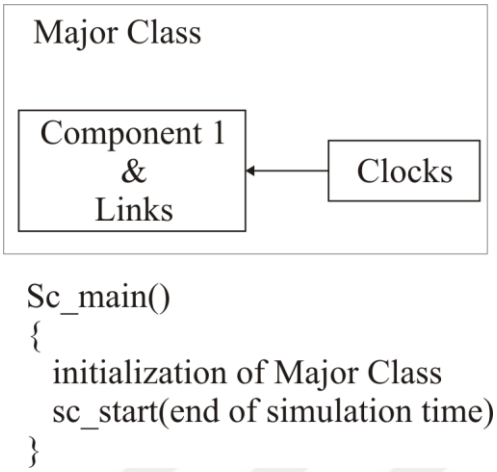

Figure 4 : Data structure and start of simulation in SystemC.

In SystemC terms, the first step to create a component is to create a class for that component. This is done by using the *SC_MODULE* macro of SystemC. The general structure of a class declaration by using *SC_MODULE* is shown in Figure 5. As it can be seen, the general structure of the hardware component is declared. The ports of a SystemC module are not different than the pins of an IC. The clock signal input is also declared as one of these ports. There are three template classes for ports: Input, output and input-output. The clock input port is of Boolean type. The clock cycles in SystemC are nothing different than periodical square waves.

There are several ways to define a process in a component. In this work, *SC_THREAD* is used for all components. *SC_THREAD* allows initializing and processing in one function. The structure of the *SC_THREAD* can be seen in Figure 6.

The variables of the process are initialized in the first step. Then *SC_THREAD* enters an infinite loop, since *SC_THREAD* is called only once through an execution phase.

In this study, there is a common requirement for all components: Once the component starts to run, it must not end until SystemC ends the whole simulation. This is realized by the infinite loop. The clock cycles are watched by *wait()* function. It pauses the execution until any stimuli in the sensitivity list is encountered. The sensitivity list is

declared under the *SC_MODULE* declaration. With every positive edge of the clock, the process runs until it encounters a new *wait()* function. This is provided by the *Clk.pos()* command in the sensitivity list.

```
SC_MODULE(component_name)
{
  // ports
  sc_in in_prt;
    •
    •
    •
  // functions
  void process_name;
    •
    •
    •

  SC_CTOR(component_name)
  {
    SC_THREAD(process_name);
    sensitive << Clk.pos();
  }
};
```

Figure 5 : Structure of a SC_MODULE declaration.

An enable check is done each time when the positive clock edge is encountered. If the block is not enabled, then the process is skipped for that clock cycle. So, it does not prevent the system clock to be updated but it does prevent processing of invalid data. Before data is written on a port, the port has the value 0. It takes some time for the valid data to reach the component and the component must not run before the data reaches it. This is succeeded by enabling the ports as well as clocking the ports.

2.3Possible Receiver Architectures

This work aims to create an ASIC model for GPS/Galileo receiver architectures with SystemC. There are two algorithms for the receiver architecture to be analyzed: First, the noncoherent integration algorithm which will be referred as the conventional

algorithm from now on. Second, the differential correlation algorithm, which was introduced in [4].

```
void component_name::function_name ()
{
  // variable initializations
  int a;
    •
    •
    •
// infinite loop
  for(;;)
  {
    wait();
    component enabled ?
    •
    •
    •
  }
}
```

Figure 6 : An *SC_THREAD* related function declaration.

The receiver objective is to have high detection ratios despite the low signal strength and the transmission medium which is modelled as AWGN channel. The attenuated outdoor received signal and the total signal are plotted in Figure 7.

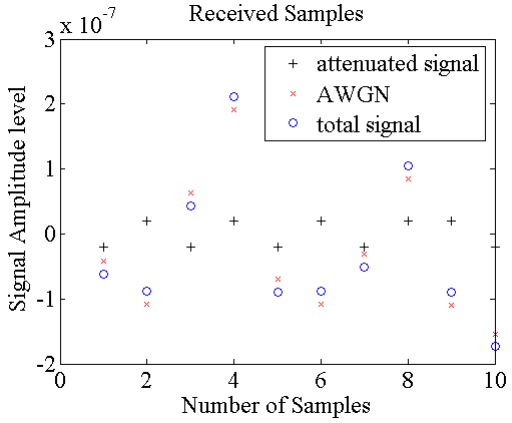

Figure 7 : Some received signal samples and respective noise amplitudes.

The block diagram of the conventional algorithm is shown in Figure 8 for one satellite and channel. The principal blocks are shown here. In latter sections, for the sake of

synchronization and block fitting some new blocks will be added. But this block diagram is based on the mathematical calculations, in other words theory, rather than the practical aspects. A similar plot of this functional block diagram is found in [5].

Another algorithm is introduced in [4]. Figure 9 shows the architecture of the differential correlation algorithm. For the same observation period, differential correlation algorithm gives 1.5 dB better results in average. Obviously, there are not many differences between the structures of the two algorithms. Almost all receivers are based on the same generic receiver structure as in Figure 8 [6].

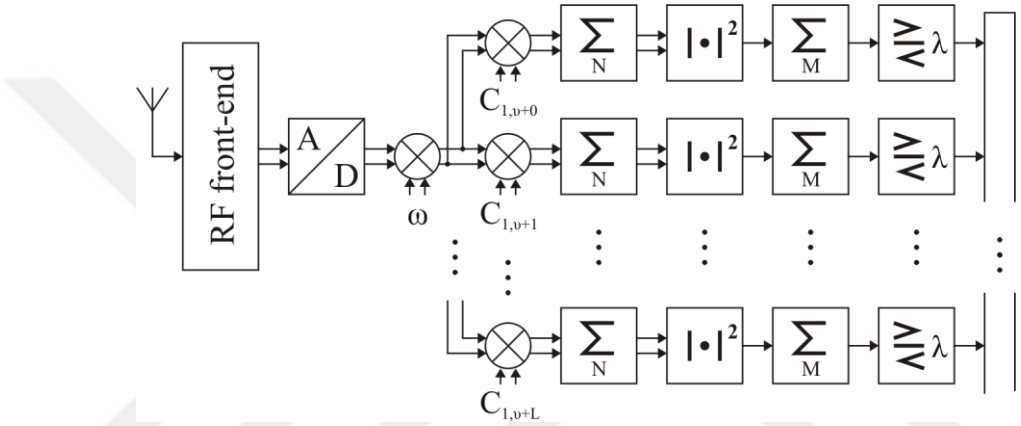

Figure 8 : Structure of the receiver based on conventional algorithm.

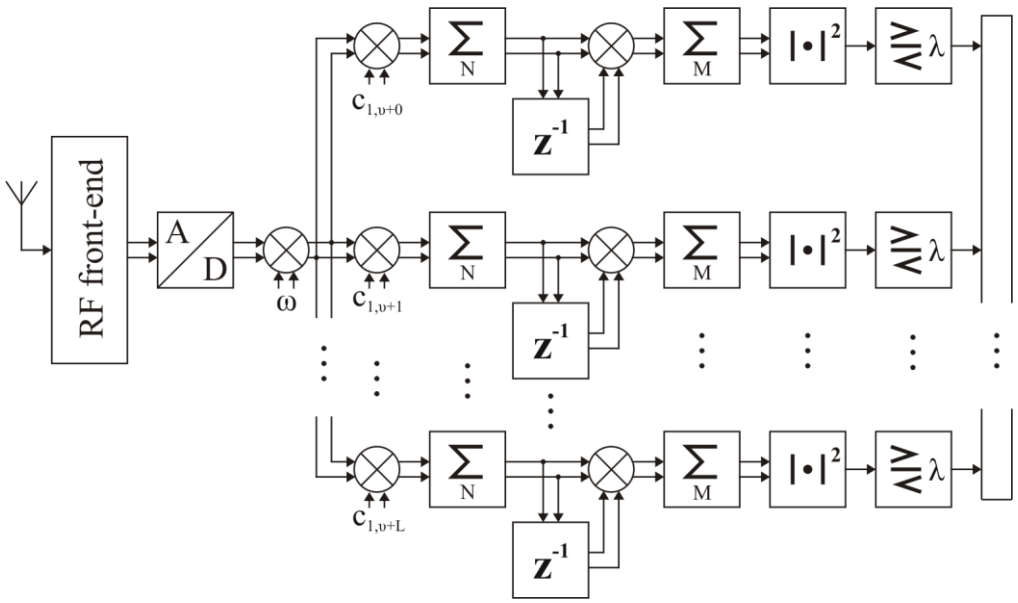

Figure 9 : Structure of the receiver based on differential algorithm.

The problem of synchronization of the chips is shown in Figure 10. The chip sequence of the received signal must be tuned with one of the bins in order to be detected. This synchronization is done by correlating the incoming chip sequence with different chip sequences. The difference between the chip sequences of every bin is a shift of one chip. The number of bins depends on the total length of the chips, the front-end bandwidth and the type modulation. For example, an L1-B signal component of Galileo has 4092 chips which are transmitted in 4ms. This number is doubled due to BOC modulation to 8184 chips. If 2.046MHz front-end filter band-width is applied, then there would be 16368 bins to search. The chip sequences are circular, after the last chip (labelled with L in the figure) the sequence starts from the beginning with the first chip (labelled with 0…L-1 in Figure 10). Finally, when the outputs of all bins are put side-by-side together, the resulting output would be a function like autocorrelation if the incoming chip sequence is the same as the matching chip sequence. If the incoming sequence belongs to another signal component then the output function would be like a cross-correlation function.

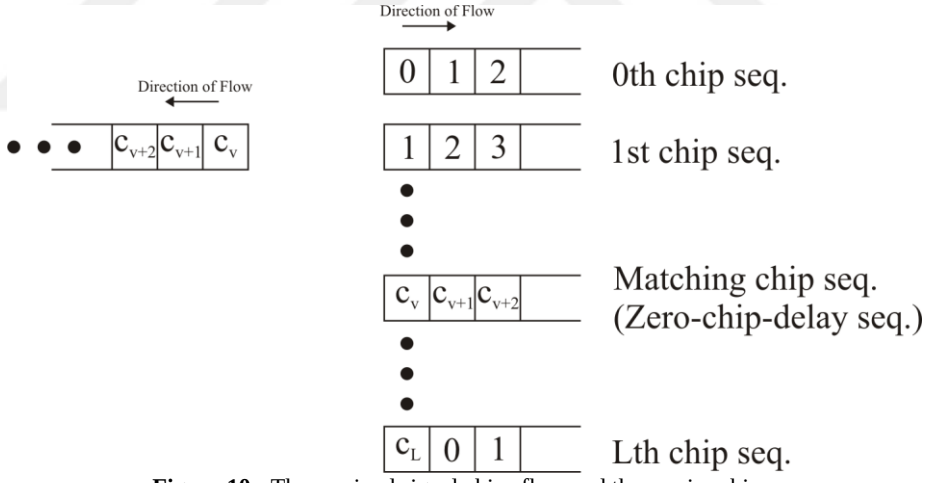

Figure 10 : The received signal chips flow and the receiver bins.

A chip in this work has the value of +1 or -1. The correlation functions are not as clear as the theoretical ones. Because the signals also contain additive thermal noise, to get clear results from the system there is additional work to do.

3 GPS/Galileo Receiver Architecture and Components

The received signal is sampled in the early stages of the receiver and the samples are processed before the first quantization of the samples. This work will focus on the digital signal processing not the analogue. That is why the RF front-end is drawn as a whole and its smaller components are not shown in Figure 8 and Figure 9. But the properties and characteristics of the sampled signal at the input side of the ADC must be explained briefly.

The signal powers of GPS and Galileo signals are below -155dBW [3] [7]. The opensky nominal signal powers of GPS signal components are shown in Table 3.

Received Minimum RF Signal Strength			
SV Blocks	**Channel**	**Signal**	
		P(Y)	**C/A or L2 C**
II/IIA/IIR	L1	-161.5dBW	-158.5dBW
	L2	-164.5dBW or -164.5dBW	
IIR-M/IIF	L1	-161.5dBW	-158.5dBW
	L2	-161.5dBW	-160.0dBW

Table 3 : GPS signal components [7].

Even if the received signal is synchronized with one of the bins, most of the times there are minor shifts which are less than one chip period. To overcome this problem, in-phase and quadrature components of the signal are used for the calculations, i.e. the signal samples are complex valued. The signals at the input of the ADC block are

$$\begin{aligned} s_{\mathrm{I}}(t) &= c_{\nu}\sqrt{2C}\cos(\omega\nu T_s) + n \\ s_{\mathrm{Q}}(t) &= c_{\nu}\sqrt{2C}\sin(\omega\nu T_s) + n \end{aligned} \tag{1}$$

with

$$w = 2\Pi f_{\mathrm{m}} \tag{2}$$

where $s_{\mathrm{I}}(t)$ and $s_{\mathrm{Q}}(t)$ are the in-phase and the quadrature components of the signal respectively, f_{m} is the mid-frequency, v is the index, c_v is the v-th chip that can be +1 or -1 and T_{s} is the sampling period, C is the transmitted signal strength which depends on the simulation conditions (environmental conditions) and the signal components. The signal samples are discrete-time and continuous-valued. n is the thermal Gaussian white noise and it has the same probability density for quadrature and in-phase components

$$\mathrm{P}_n^{(v)} = \frac{1}{\sqrt{2\Pi\sigma^2}} \exp\left(n^2 \big/ 2\sigma^2\right) \tag{3}$$

The statistical characteristics of the samples have great importance for the implementation part. While tracking the signals mathematically, the means and variances of the signals samples on the zero-chip-delay bin are observed. In the beginning, the mean and variance for each component are

$$\begin{aligned} m_x &= \sqrt{2C} \\ \sigma_x &= \sigma \end{aligned} \tag{4}$$

m_x is the mean and σ_x is the standard deviation of each component, σ is the standard deviation of the AWGN part of the sampled signal.

The mean of the signal samples is equal to the mean of attenuated transmitted signal and the variance is equal to the variance of the noise. The implementation of the straight forward structures in Figure 8 and Figure 9 would be a simple but bulky hardware. It does not exploit any potential advantages of the system. Subsequently, the proposed efficient architecture will be presented from a mathematical point of view.

3.1 Conventional Algorithm

3.1.1 ADC Block

The signal samples as in Eq. 1 must be quantized before further processing. There are several criteria, to keep in mind for the quantization. First, the quantization operation introduces quantization noise. Second, amount of quantization bits have the effect of increasing or decreasing the complexity of the system. And finally, the method of quantization must be simple for a fast and simple architecture. An example of number of bits and amount of quantization loss is seen in Figure 11 and Figure 12.

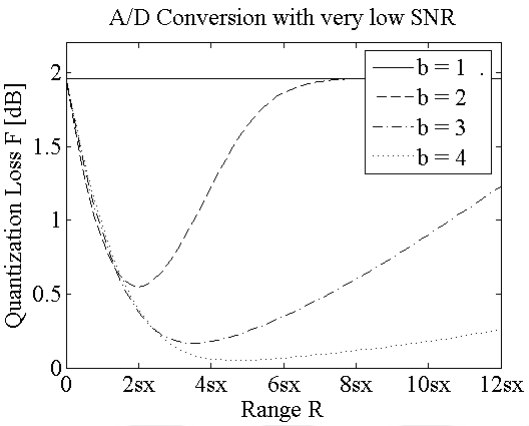

Figure 11 : Change of quantization losses with quantization range for 1 to 4 bits.

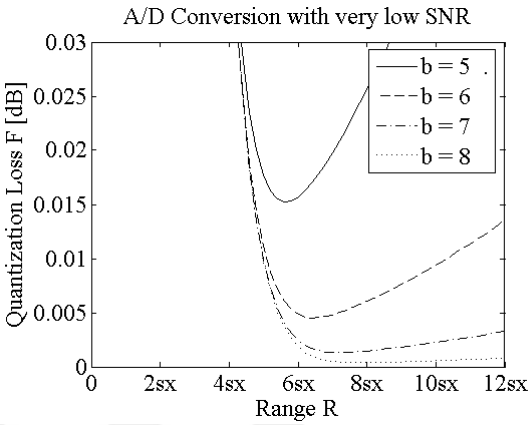

Figure 12 : Change of quantization losses with quantization range for 5 to 8 bits.

A formula for the mid-riser quantization is

$$y_v = \begin{Bmatrix} 2^b - 1, & x(t) \geq \frac{R}{2} \\ 2\left\lfloor x(t)\frac{2^b - 2}{R} \right\rfloor, & -\frac{R}{2} \leq x(t) < \frac{R}{2} \\ -2^b + 1, & x(t) < -\frac{R}{2} \end{Bmatrix} \quad (5)$$

where y_v is the result of quantization, b is the number of bits and R is the quantization range. $\lfloor \cdot \rfloor$ is the floor operation, which rounds the number to the next integer smaller or equal. The range depends on the noisy analogue signal amplitude and it is too formulated in terms of signal variance as

$$R = c \cdot \sigma_x \quad (6)$$

The quantization range in physical terms can be seen in

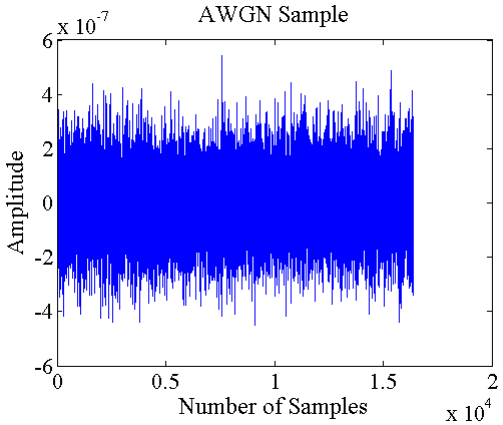

Figure 13 : Noise on the received signals.

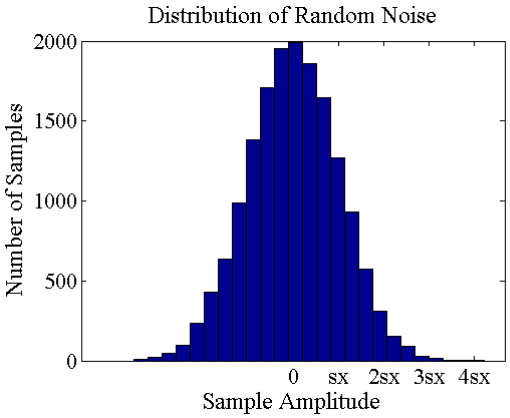

Figure 14 : Histogram of the noise of the received signals.

In Figure 13, an AWGN noise sequence with the standard deviation, $\sigma = 1.2800\text{e-}007$ is seen. In Figure 14, the histogram of the same sequence is drawn. It is seen that as R increases, the number of samples that can be quantized without saturation also increases. Any sample amplitude which has an absolute value higher than R is saturated and is mapped to the maximum (or minimum for negative numbers) integer which depends on the amount of quantization. The negative numbers which are on the left side of Figure 14 result in a negative saturation and right side of the figure would result in a positive saturation.

b = 3							
000	001	010	011	100	101	110	111
-7	-5	-3	-1	1	3	5	7

b =4															
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
-15	-13	-11	-9	-7	-5	-3	-1	1	3	5	7	9	11	13	15

Table 4 : Possible outputs for the quantization of 3 and 4 bits and their corresponding integer values.

The mapping is different than in a PC. For example, the number 111 means -3 in a PC and 7 in base 2. In our mapping, 111 is mapped to 7 in case of 3 bits of quantization. Mapping lists for *b=3* and *b=4* are shown in Table 4.

The mean and the variance of the signal samples change with the quantization to

$$m_y = \sum_{\kappa=0}^{2^b-1} y_\nu^{(\kappa)} P\left\{y_\nu^{(\kappa)}\right\}$$
$$\sigma_y^2 = \sum_{\kappa=0}^{2^b-1} \left(y_\nu^{(\kappa)} - m_y\right)^2 P\left\{y_\nu^{(\kappa)}\right\} \quad (7)$$

where $\mathrm{P}\left\{y_\nu^{(\kappa)}\right\}$ is the probability of the samples to have the value y_ν.

3.1.2 Mixer

The BPSK modulated signal is first down-converted in the RF front-end to the intermediate frequency, f_m. The second down-conversion to baseband is done in the mixer block. The operation is the multiplication of the modulated signal with its conjugate: $\exp(-\mathrm{j}\omega\nu t)\cdot\exp(\mathrm{j}\omega\nu t)$. When in-phase and quadrature components are separated, the down-conversion is written in the following as

$$y_{\mathrm{I},\nu} = x_{\mathrm{I},\nu}\cos(\omega\nu T_\mathrm{s}) + x_{\mathrm{Q},\nu}\sin(\omega\nu T_\mathrm{s})$$
$$y_{\mathrm{Q},\nu} = x_{\mathrm{Q},\nu}\cos(\omega\nu T_\mathrm{s}) - x_{\mathrm{I},\nu}\sin(\omega\nu T_\mathrm{s}) \quad (8)$$

where $x_{\mathrm{I},\nu}$ is the in-phase component of the input of the mixer block and $x_{\mathrm{Q},\nu}$ is the quadrature component.

The resolution inside the blocks is higher than the resolution used for data communication between the blocks. This means sine and cosine functions give a higher resolution than the input signal samples. The multiplication of these two terms increases resolution even to a higher value. The result of these multiplication operations, $y_{\mathrm{I},\nu}$ and $y_{\mathrm{Q},\nu}$ must be quantized once more.

The same function as in ADC block is used in this block as well. The relation of output mean m_y and output variance σ_y^2 to input mean m_x and input variance σ_x^2 are

$$m_y = m_x$$
$$\sigma_y^2 = \sigma_x^2 F \quad (9)$$

where *F* is the quantization loss and depends on the amount of quantization.

3.1.3 PRN Code Generator

The PRN code generator delivers the local chips in every bin. The sequence in each bin is already explained in Figure 10.

3.1.4 Despreading

The demodulated signal samples of the spread spectrum signal are despreaded in this block. This operation for each is like finding the autocorrelation of incoming chip sequence, except that the incoming samples are noisy. It is a simple multiplication of data bits with the local chips which are received from the local PRN generator.

The mathematical expression for this operation is

$$y_\nu = c_{\nu+T \bmod L} \cdot x_\nu \tag{10}$$

where *T* is the chip delay. It is any integers from 0 to *L*-1.

Because the $c_{\nu+T \bmod L}$ can have only +1 and -1 as values, this operation is equivalent to the XOR operation applied on the sign bit of x_ν. On the zero chip delay bin, this function gives a positive mean. But because of the noise component of the signal, further processing is required. The despreading operation does not change the mean and variance of the components, but it is the last operation applied on the individual signal samples. The accumulation uses the output of the despreading block.

3.1.5 1st Accumulator

This block symbolized by $\sum_N$ in the middle in the block diagram. The operation is

$$y_{I,\mu} = \sum_{\nu=(\mu-1)N+1}^{\mu N} x_{I,\nu}$$
$$y_{Q,\mu} = \sum_{\nu=(\mu-1)N+1}^{\mu N} x_{Q,\nu} \quad (11)$$

where *N* is the number of accumulated samples.

In , the effect of the *N* on the signal mean and variance is seen. *N* increases the mean of the sampled signals to a level which is comparable to the noise in the signal. Thus, the value of *N* should be adjusted according to the SNR. The smaller the SNR, the higher *N* must be. But the increase of *N*, also increases the observation time for the results.

The addition operation causes the content of the accumulator memory to increase and expand. The LSB denotes the lowest digit of the result of addition and MSB is the highest bit of the result of the addition as shown in Figure 15. The proportion of the of the LSB to the MSB decreases with the increase of MSB. At some point, this proportion becomes so small that some LSBs can be neglected. This negligence is not done directly at the addition but at the output writing. This operation is shown in Figure 15. The MSB is used at any case and it is aimed not to be neglected. If the

MSB of an accumulation is too big and the number of bits is not enough then the result must be saturated making all bits 1.

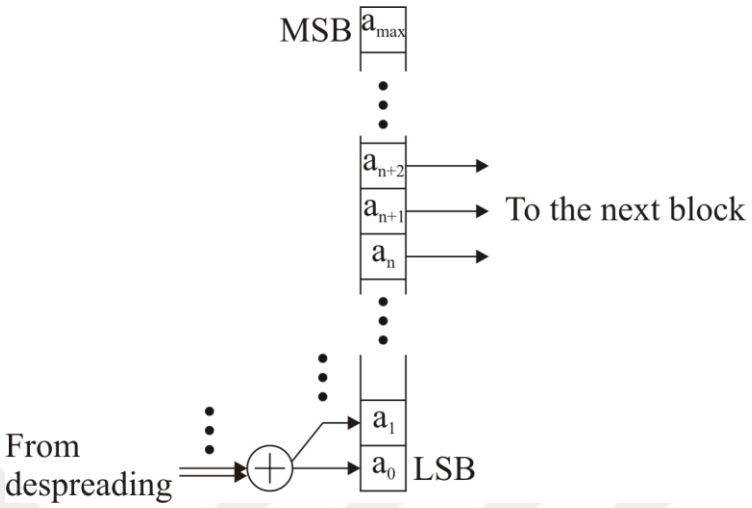

Figure 15 : Addition operation in the 1st accumulator block and neglecting the LSB.

The mean and variance of the accumulation operation of the signal for the zero-chip-delay bin are

$$\begin{aligned} m_y &= N \cdot m_x \\ \sigma_y^2 &= N \cdot \sigma_x^2 \end{aligned} \tag{12}$$

At this block, the output SNR increases considerably. In line-of-sight signals, the second accumulator may not be even needed. It all depends on the initial SNR of the sampled signal.

3.1.6 Squaring

The two components must be squared in order to construct the envelope. The squaring operation realizes this. Both components are combined with the operation

$$y_\nu = x_{\mathrm{I},\nu}^2 + x_{\mathrm{Q},\nu}^2 \tag{13}$$

The squaring operation doubles the required number of bits. It also squares the mean and the variance of the samples. The mean and variance at the end are

$$\begin{aligned} m_y &= m_x^2 \\ \sigma_y^2 &= \sigma_x^4 \end{aligned} \tag{14}$$

The squaring block can be subject to the same bit dropping policy as in the 1st accumulator block. At the output of this block, the in-phase and quadrature components have been merged.

3.1.7 2nd Accumulator

In case of low SNR, to increase the observation period a second accumulator can be used. The accumulator operation is

$$y_\mu = \sum_{\nu=(\mu-1)M+1}^{\mu M} x_{I,\nu} \tag{15}$$

where M is the number of outputs of squaring block. This operation changes the mean and the variance

$$\begin{aligned} m_y &= M \cdot m_x \\ \sigma_y^2 &= M \cdot \sigma_x^2 \end{aligned} \tag{12}$$

The bit dropping operation can be applied in this step too.

3.1.8 Threshold Comparison

All the bins of the architecture meet at the end. The results of the bins are compared with a threshold value and the bin with the result higher than the threshold is accepted as the synchronized bin, i.e. zero-time-delay bin.

3.2Differential Correlation Algorithm

The architecture of the differential algorithm has two additional blocks: the differential correlation block and a buffer block to store and delay the sequence.

3.2.1 Buffer

The buffer block is shown with z^{-1}. All samples from 1st accumulator are directed in both the buffer block and the differential multiplication block. The only operation done in this block is the conjugate which is simply sign change of the quadrature component.

3.2.2 Differential Product

The samples from the buffer block and the 1st accumulator meet again at the differential product block. Through this product, the independence of the noise in the accumulated sample packets is exploited. The operation in this step is

$$\begin{aligned} y_{\mathrm{I},\mu} &= x_{\mathrm{I},\mu}x_{\mathrm{I},\mu-1} + x_{\mathrm{Q},\mu}x_{\mathrm{Q},\mu-1} \\ y_{\mathrm{Q},\mu} &= x_{\mathrm{Q},\mu}x_{\mathrm{I},\mu-1} - x_{\mathrm{I},\mu}x_{\mathrm{Q},\mu-1} \end{aligned} \tag{16}$$

$y_{\mathrm{I},\mu}$ and $y_{\mathrm{Q},\mu}$ are the μ-th output of the product, $x_{\mathrm{I},\nu}$ and $x_{\mathrm{Q},\nu}$ The negative sign in the quadrature component multiplication is to show the nature of the conjugate operation.

The mean and the variance change is

$$\begin{aligned} m_y &= m_x^2 \\ \sigma_y^2 &= \left(2\sigma_x^4 + 2\sigma_x^2 m_x^2\right)F \end{aligned} \tag{17}$$

As the change in the mean of the signal is the same in squaring block, LSB dropping can also be done for this block. The results from this block are directed to the squaring block and the flow of data carries on as in the conventional method.

3.3 Efficient Architecture for Conventional Algorithm

The structure of the parallel architecture is a bulky architecture requiring many hardware components. Every operation is done in a separate block and even though the architecture looks parallel the processes are not processed in parallel in the blocks. But the components used in all correlation paths are more or less similar and in most cases identical. A new structure is designed in order to exploit the advantage of these similarities. The modifications in the blocks to convert the structure to the new architecture are explained subsequently.

In the new structure, one component undertakes the workload of more than one bin. The new structure is also branched but each branch processes the work of more than one bin. This makes the new structure more compact and the components of the new architecture must work faster in order to succeed to process more work in the same time limits. This requires a higher process speed, decreasing the clock period, thus increasing the clock frequency. The number of components must be multiples of the number of bins. For the Galileo signal component, the number of branches must be multiples of 4092. The highest feasible frequency is selected as 400MHz. For the

Galileo signal component, there are 16368 bins. 11 is a multiple and if it and if 11 branches are used then the clock frequency must be increased 1488 times (16368÷11). Then the clock increases to 1.522GHz (1.023MHz×1488). Obviously, the new clock frequency is over the upper limit of 400MHz. A new component, mixer-despreading buffer, is introduced in order to reduce the clock frequency further. To decrease the frequency under the limit, 4 samples must be buffered at each clock cycle.

The signal properties do not change along the blocks and the previous mathematical interpretations of the conventional architecture apply for the new blocks as well. The mathematical expressions are therefore not repeated. The parallel architecture gives a better picture of what is happening inside where the efficient algorithm creates shortcuts and modifies the architecture for better use of resources.

3.3.1 ADC and Mixer Components

Only one ADC, operating on intermediate frequency, is required for the entire receiver. Then there is one digital down-conversion mixer required for each frequency bin to be searched in parallel.

3.3.2 Mixer-Despreading Buffer

The clock frequency of the blocks increases when the workloads of the different blocks are gathered in one block. The aim of this block is to reduce the clock rate to a value, that is in the range, i.e. less than 400MHz, by storing the sampled signal for a while. During that storage time, the previous samples can be processed. In Figure 16, four samples are gathered. The clock of the registers on the left side of the line is the same as in the mixer block. The structure here is a shift-register. The right side of the

line is different though. The buffer outputs are updated with a positive clock edge. The clock frequency on the right side is one fourth of the left side.

Hence, the mixer-despreading buffer reduces the sample rate by a factor of four when it distributes the input stream to four interleaved output streams.

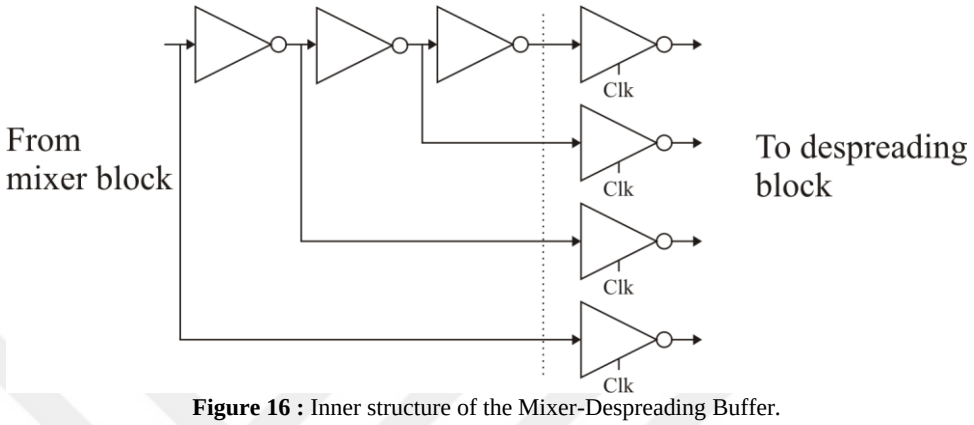

Figure 16 : Inner structure of the Mixer-Despreading Buffer.

The block has only one input and it is sufficient to have only one buffer block for each frequency search channel.

3.3.3 PRN Code Generator

In the conventional architecture the PRN code generator can be composed of only one block. A possible structure can be as in Figure 17. This block has direct ways to each code search bin, thus it requires all memory to be composed of read registers. Observe that for each possible chip there is an output. *L* in this diagram includes the number of samples per chip.

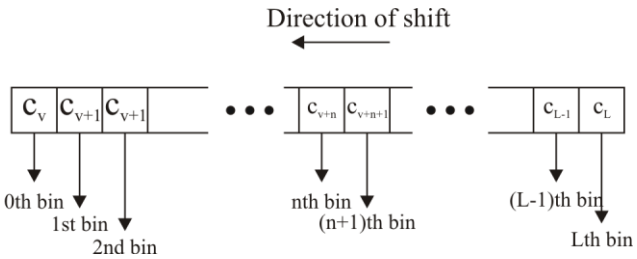

Figure 17 : Inner structure of the PRN code generator in the conventional architecture.

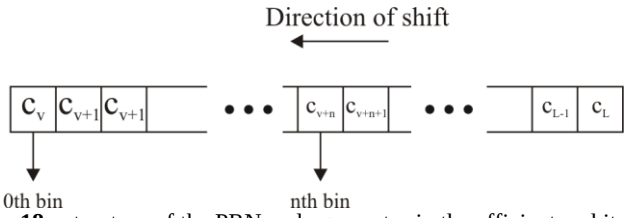

Figure 18 : structure of the PRN code generator in the efficient architecture.

In the second image, Figure 18, the PRN generator serves only *K* different despreading blocks. That is to say, there are *K* different bins. Also, the *L* in Figure 17 is ¼ of the *L* in Figure 18. In the second diagram, the number of memory registers is equal to the number of chips. The number *K* should be chosen such that *K* is a multiple of it, i.e. $n \bmod K = 0$ where *K* is the number of outputs from the block. The reason of low number of local chips will be revealed under the despreading section, 3.3.4.

Code	1				-1				1				1			
GPS	1		1		-1		-1		1		1		1		1	
Galileo	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1

Table 5 : Original chips and its GPS and Galileo equivalent for 2 samples per chip.

In Table 5, a possible chip sequence and its corresponding chips sequences in the zero-chip-delay bins both in a GPS and a Galileo receiver are shown.

To transmit all the chips to the despreading block, the (*K*-1)-th chip must be shifted to 0^{th} register. The clock period of the PRN generator is denoted by *T*c. This relation of *K* cycles per 1 sample would give us the ratio $T_c = K \cdot T_s$. Here the mixer-despreading buffer shows its effect. The clock period is increased to $4T_s$ at the output of this

component. The factor of 4 is explained in 3.3. Finally, the clock period can be increased to $T_c = 4 \cdot K \cdot T_s$.

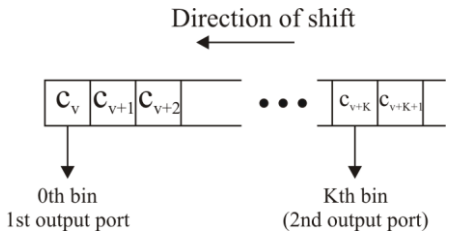

Figure 19 : The state of the bits in PRN code generator at *t*.

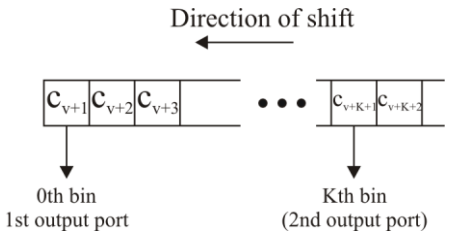

Figure 20 : The state of the bits in PRN code generator at *t+Tc*.

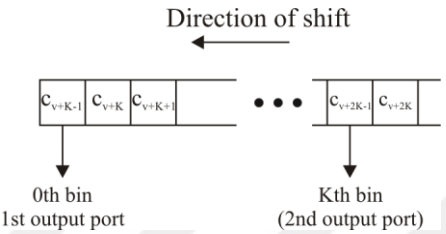

Figure 21 : The state of the bits in PRN code generator at $t+2T_c$.

3.3.4 Despreading Component

The local chips from the PRN code generator meet here. In 3.1.4, it is already explained that the despreading is equivalent to the XOR operation. This operation can take place, in K different XOR gates. The input is from the mixer-despreading buffer. This structure is valid for a Galileo receiver with 4 samples per chip. For GPS, the first two channels would not be required. The inversed channel is due to the BOC modulation. The BOC modulation of Galileo E1-B1C signal component generates 2 samples from one chip, the first with inverse and the second with the same sign as the original chip. This relation can be seen in Table 5. It can be seen that the first and the second chips give the same results as well as the third and the fourth chips, so that the structure can be further simplified by joining the operations. They can be united decreasing the required hardware to half of its size, thus leading to the final diagram in Figure 22.

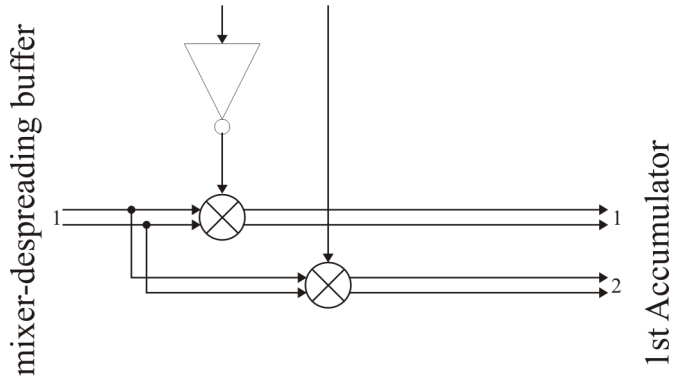

Figure 22 : Improved inner structure of Despreading Component for a Galileo signal component.

One sample is not multiplied with only one reference chip but with full range of chips. So, before the new samples arrive the present samples must be multiplied with the required reference chips. The clock rate must increase. The amount of increase depends on the number of cycles the PRN generator needs in order to deliver the full epoch of the local code.

This block shall work with the same clock period as the despreading block because of the obvious synchronization requirement between the two blocks.

3.3.5 1st Accumulator Component

The inputs from the despreading block are directed to the different addresses. The basic operations for a Galileo signal are shown in Table 6. The 4 samples are multiplied and the results are directed to the corresponding memory addresses.

	chips				
Samples by Sampling times		c_v	c_{v+1}	c_{v+2}	c_{v+3}
	T	n	n+1	n+2	n+3
	$T+T_s$	n+1	n+2	n+3	n+4
	$T+2T_s$	n+2	n+3	n+4	n+5
	$T+3T_s$	n+3	n+4	n+5	n+6

Table 6 : The flow and addressing of the data with time inside the 1st accumulator component.

For an incoming packet of 4 samples and 4 chips of the reference code there are 16 outputs. But these outputs are of 7 different memory registers. The owner of the results also shifts. Such a movement in ownership cannot be solved by simply shifting the memory registers because all the 4 multiplications are executed simultaneously. Using a shift register increases the clock rate because the system would work as if the operations are done at different times.

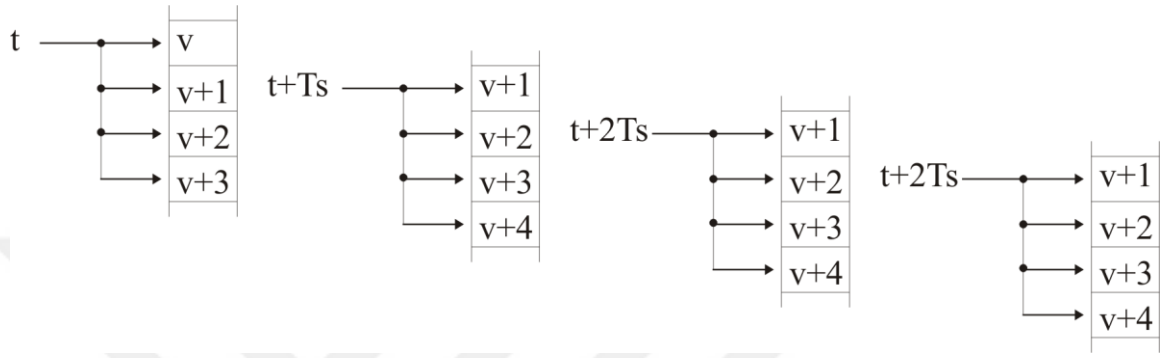

Figure 23 : Flow of the data with time and the memory addresses in the 1st accumulator.

The solution for concurrent operation is directing the outputs to the right registers at each time instant, i.e. with the same positive clock edge. A concurrent system would be as in Figure 24. It is important to note that for each adder unit there is an additional input that is not shown. A feedback from the memory registers enters the adder too.

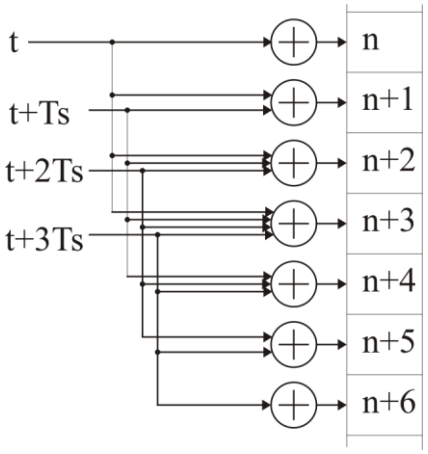

Figure 24 : The inner structure of the 1st accumulator.

The memory block of the accumulator is also a shift register. With every positive clock edge, it shifts by number of samples per chips for GPS and

$(2 \cdot \text{number of samples per chips})$ for Galileo receiver. When the number of samples that are processed and accumulated reaches *N*, then the outputs are forwarded to the next block and with the next positive clock edge, the feedback from the memory registers is disabled, so that a new adding process starts from zero. The adder structure is shown in Figure 25.

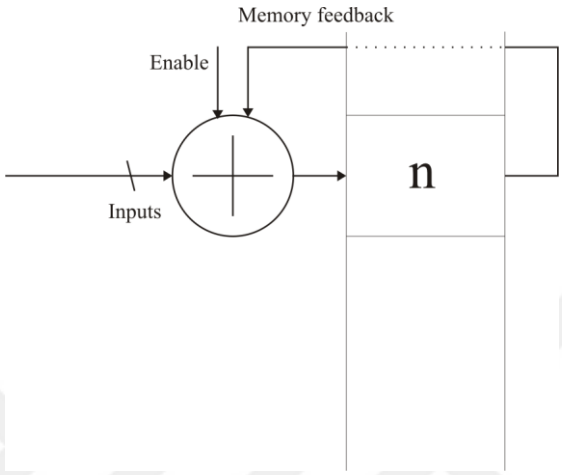

Figure 25 : The structure of an adder in the 1st accumulator.

3.3.6 Squaring

The squaring block is composed of only one memory block. When the last samples are received at the input-stream block, the squaring block starts to accept the results from the 1st accumulator. During the last time T_s, the squaring block rotates together with the 1st accumulator block and saves their results. When the squaring of all memory cells is finished, the squaring block hands over the samples to the 2nd accumulator block.

3.3.7 2nd Accumulator

The 2nd accumulator receives the samples from the squaring block. There are no special operations here, because the position of 2nd accumulator is adjusted.

3.4 Efficient Architecture for Differential Correlation Algorithm

3.4.1 Buffer

The buffer block is composed of a memory block to which the contents of the 1st accumulator are stored and this block needs not to be changed.

3.4.2 Differential Multiplication

There is no need to change the differential multiplication block. The positioning or shifting are not applied on these blocks as they do not store any data and forward the results directly to the next blocks.

4 Implementation with SystemC

The SystemC implementation of the system starts with building the smaller components and then linking them. The program should be flexible to some extend in order to experiment with different configurations. This is achieved by creating a configuration file where all variables can be adjusted and from where the program can read. Figure 27 shows the structure of the SystemC classes. Signals between the blocks are labelled with *_sig* and the ports of the components are labelled with *_prt*.

There is an important rule which is valid for all of the blocks. The latter blocks cannot conduct any operations before the data reaches them. If the first sample is considered, it takes four sample periods until it reaches the despreading block. During this time delay all of the components of the prototype will be operational. In other words, the despreading block will either wait or process invalid random data or process initialization values that are read from the channels. To overcome this problem, the enable signalling is used. Every channel enables the next block when valid data is on the channel, ready to be read. This switching may also be used to control the next block. Every block in the architecture has an enable port except the first block which is input stream of the SystemC structure or ADC of the architecture.

The tests are done for the outdoor and indoor applications. The difference is the attenuation of the signal rather than the noise power which depends on the ambient noise temperature. The indoor signals are simulated to have 20dB less signal power than the outdoor signals. The usual signal observation period is not enough when the signal strength decreases. A usual recipe against this is, increasing the observation period. The decrease in the signal strength denotes a decrease in the accuracy of the system as well. Some examples for GPS can be seen in Table 7 from [8].

Application	***Coverage***	***Accuracy(m)***	***Time-to-first-fix(s)***
Navigation	Urban, rural	1-25	1-5
People finder	Indoor, urban	1-100	5-15
Roadside assistance	Urban, rural	75-125	5-30
Location sensitive billing	Indoor, urban, rural	5-100	1-5

Table 7 : Some applications, accuracy and time-to-first-fix of Navigation systems [8].

4.1 Input stream

The signal at the beginning of the ADC block is a sampled BPSK modulated signal. To have a stand alone program, the signal samples must be generated on the fly. So the ADC block is embedded in a new block together with the signal sample generator.

The noise component in Eq. 1 is generated by converting the white noise which is generated by the random number generator of C++ to white Gaussian Noise with the help of a Box-Muller method. It is strongly recommended that simulations with different configurations must be executed with the same noise sequence. That is realized by setting the same random seed of the C++ compiler in the beginning of every simulation.

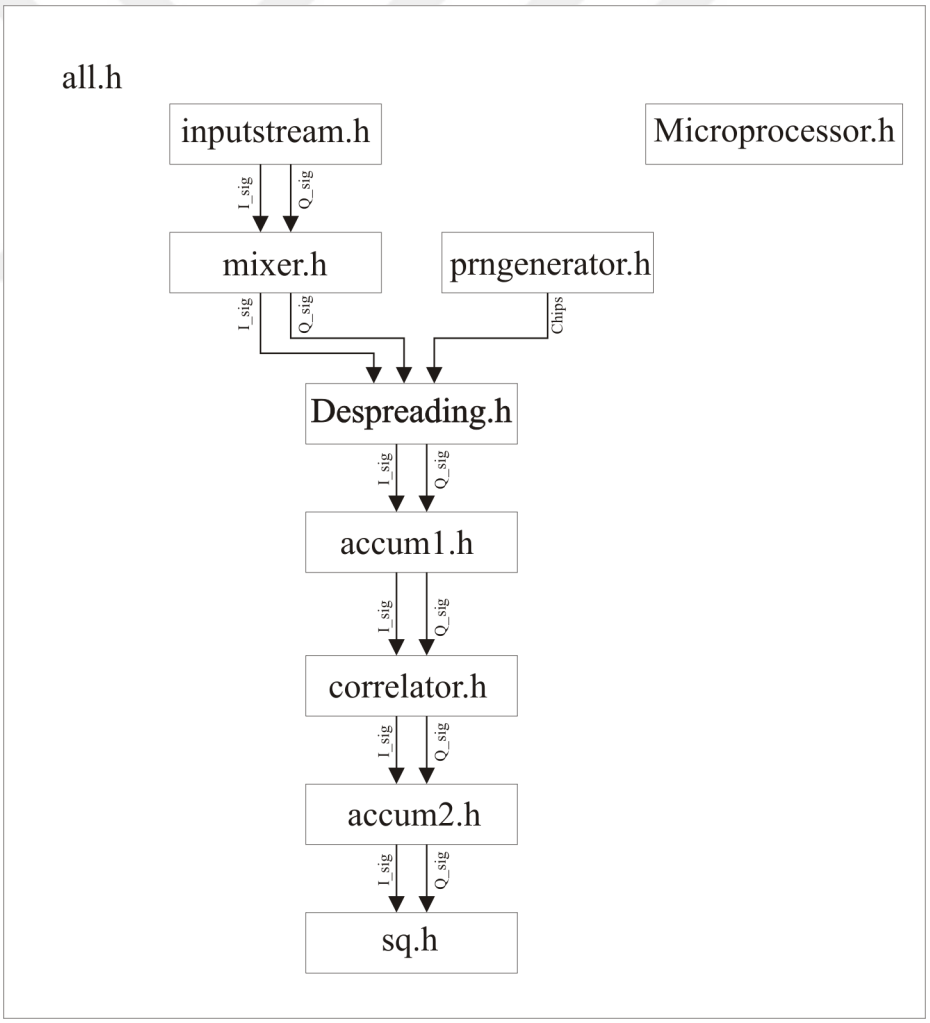

Figure 26 : Block diagram of the component declarations with header files and the channels between them in the SystemC implementation of the conventional algorithm.

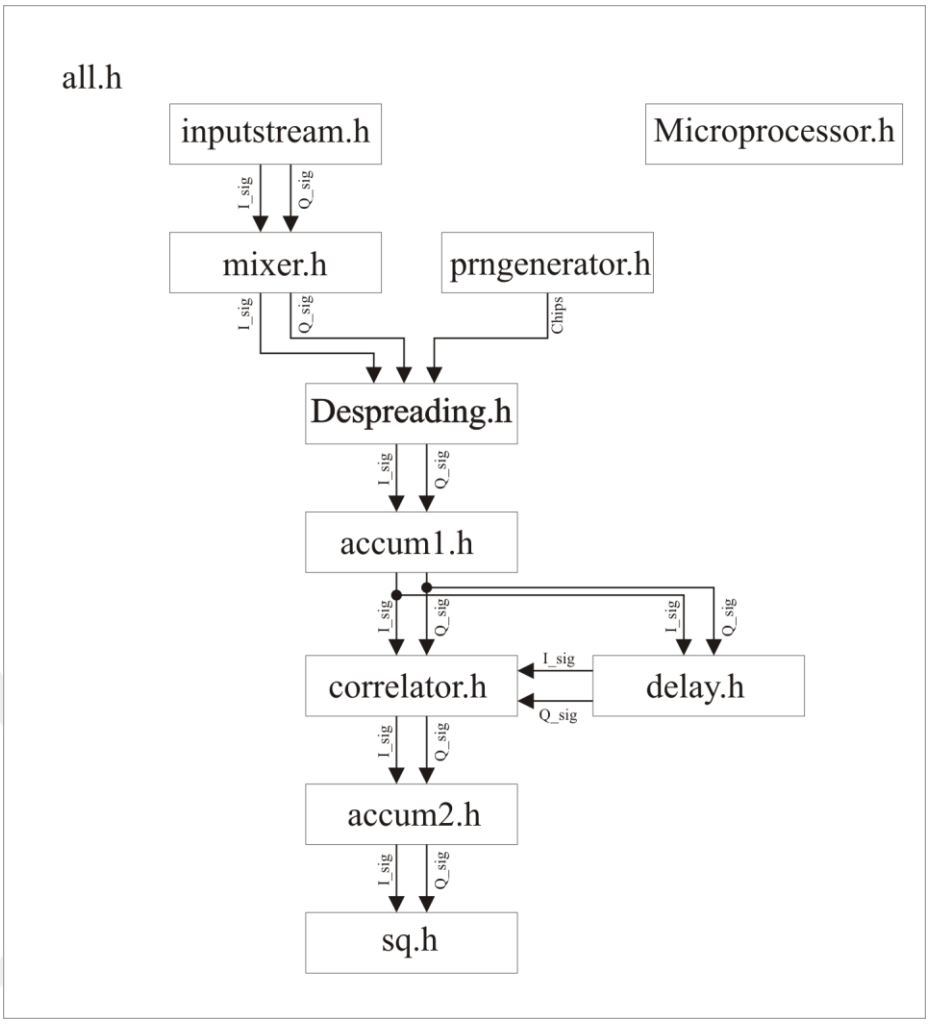

Figure 27 : Block diagram of the component declarations with header files and the channels between them in the SystemC implementation of the differential correlation algorithm.

The sine and cosine functions in the attenuated signal component are generated by using the *sc_time* class of the SystemC. After generating the sampled signals, finally the signal reaches the ADC. In the ADC, the signal samples are converted from discrete-time, continuous-valued signals to discrete-time discrete-value signals, i.e. digital signals.

The continuous-valued signals are declared to be double type. It is precise enough to be considered as continuous-value signal. In Figure 12, it can be seen that the quantization loss drops to less than 0.005dB already at 8-bit quantization. Then, the samples are quantized the results are written on the in-phase and quadrature output ports. Also the enable signal is changed to true. The enable signals are used to prevent invalid data processing. The channels do not carry data in the beginning. The data spreads with every clock cycle to the next block. The enable channels prevent the

blocks working earlier than data arrival. The data which is written on a channel, is available for reading with the next positive clock edge.

4.2 Mixer

The mixer block converts the signal to the baseband. The signal samples after multiplication are converted from integers to double reflecting the higher number of bits, which is because of the sine and cosine function multiplications. Then, the samples are quantized with the amount of bits specified in the configuration file.

4.3 PRN Code Generator

Under the process defined with an *SC_THREAD* macro, the memory block which contains the local chips is implanted by a one dimensional array only.

4.4 Despreading

The despreading block reads the inputs from the PRN code generator and mixer blocks and after the multiplication, it writes the results immediately to the output port. All the despreading blocks are assembled into one greater component with many input/output ports.

4.5 1st Accumulator

The accumulation period or total number of samples to accumulate is determined by the *N_turns* variable in the configuration file. After all samples are added, and some bits are dropped and clipped, they are written to the output ports and the content is set to zero. This specification for the bit adjustments is also given in the configuration file. If the output port is written for the first time, the enabling channel for the next component is set.

4.6 Squaring Block and Differential Multiplication Block

The squaring block also consists of an array. Due to the squaring operation, the number of bits tends to double. The bits which are sent to the next block are set in the configuration file and with the first output the next block is enabled. The differential multiplication block has a similar structure as the squaring block due to the similar operations.

4.7 2nd Accumulator

This block is always included. However, for this block to be functional, the *M_turns* variable which is found in the configuration file must be higher than 1. Otherwise, it will only store and forward, instead of accumulate and forward. After the accumulated inputs to this port reaches *M_turns* the next block is enabled. The instructions about which bits to send are found in the configuration.

4.8 Buffer

Buffer block has the duplicate memory array of 1st accumulator block. It does not have any variables. A clock which synchronizes the output of the 1st accumulator is connected and a process is created in order to buffer the outputs of the 1st accumulator.

4.9 Linking and Combining the Components

After all components are created, they are combined via signal channels. The ports which execute a read or write operation on the channel are combined to the same channel. For instance, the in-phase signal components of the two cascading components ADC and the mixer block are combined through the channel, *is_outI_sig*. The output port of the ADC is connected to the channel. This port is a write-only port. This side is the transmitter side. The receiver side at the end of the channel is found in mixer block. The mixer reads the data through the input-only port. Given the write only and the read only ports at the two components which are connected to the same channel, one can say that the flow of data is one directional.

These connections are done in a higher level class. Its structure is different than the component structures.

4.10 Simulation Setup and Running the Program

Under *sc_main*, first the components are declared. A virtual prototype is created there. Then the initializations are done. For example, resetting all enable signal values to *false* is done in the beginning. Now that the program is fully created, the *sc_start* command starts the simulation. When the program encounters a *sc_stop* or it comes to the end of the time interval set by the *sc_start*, it terminates.

```
SC_MODULE(component_name)
{
  // component list
  InputStream is1;
     •
     •
     •

  // clock list
  sc_clock transmitter_clk;
     •
     •
     •

  // ports
  sc_in<int> in_prt;
     •
     •
     •

  // functions
  void process_name;
     •
     •
     •

  SC_CTOR(component_name)
    :   // clock and component declerations
    transmitter_clk(”transmitter_clk”,244379,SC_PS),
     •
     •
     •

    is1(”is1”),
     •
     •
     •

  {
    // linking
    is1.outI_prt(is_outI_sig);
     •
     •
     •

  }
};
```

Figure 28 : The the SC_MODULE declaration of a generic component.

5 Results

After the components are defined for both noncoherent and differential correlation algorithms, they are configured. There are some criteria for the configuration. The most important is the clock frequency. The upper limit for the clock rates is decided to be 400MHz. The numbers in Figure 29 are based on Galileo L1-B signal component.

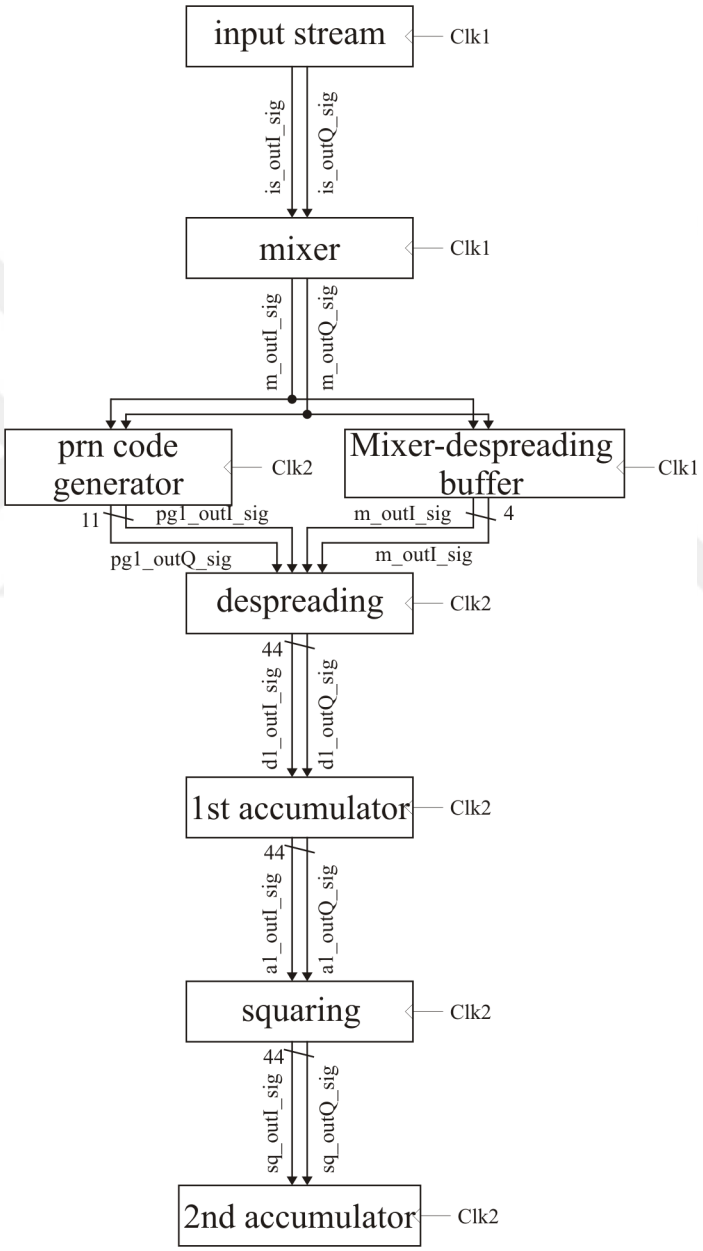

Figure 29 : The flow diagram of the efficient architecture for the differential correlation algorithm.

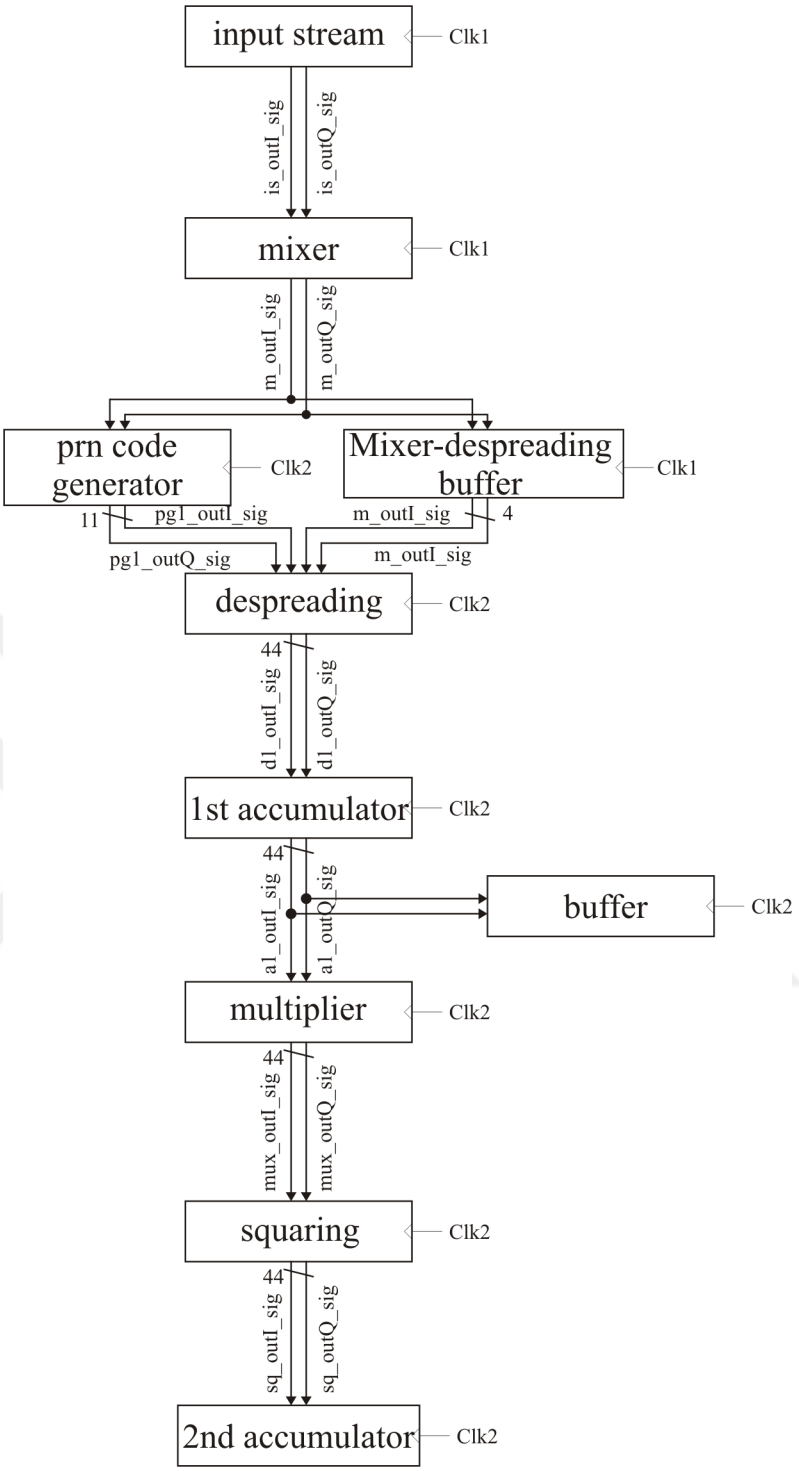

Figure 30 : The flow diagram of the efficient architecture for the conventional algorithm.

The calculations of the variables in Figure 29 are shown in 3.3. The mid-frequency used for the BPSK modulation is not critical but it must be at least double the chip frequency. With regard to this criterion, the mid-frequency is selected as 4MHz.

The efficiency of the architecture depends not only on the number of components but also the resolution of the data. The optimum quantization is searched for each component. *The optimization criterion is, the least number of bits which gives 0.1 dB less SNR than the floating-point result, is the optimum number of bits.* Literally, floating-point means infinite number of bits which is not realistic to calculate with a PC. In Figure 11 and Figure 12, the number of bits and the quantization losses are shown for 1 to 8 bits. Already with 8-bits, the loss is less than 0.005 dB. With 12-bits, the loss is below 0.001 dB. In other words, 12-bits quantization is close enough to approximate the floating-point calculation for this application. This criterion is followed in most cases but the first two. The earlier a component is, the more critical it becomes. The resolutions of the previous blocks affect the next blocks. Increasing the number of bits of ADC and mixer blocks will increase the complexity of the whole system and this decision requires an increase in the following components as well. Thus, the applied threshold is higher in the early stages than the following components. Instead of 0.1dB degradation, 0.25dB is chosen as limit for these two components.

The SNR of the system is defined as the SNR at the output. In section 3.1, the deterministic part of the signal is denoted as the desired signal and it contributes to the mean of the signal. On the other hand, the fluctuations are caused by the AWGN. These statistical values are calculated by saving 1500 output samples. Then SNR of the sample sequence is

$$SNR = \frac{m_x^2}{\sigma_x^2} \tag{19}$$

where m_x^2 is the mean and σ_x^2 is the variance of the output samples.

The quantization of the components for the noncoherent correlation architecture is tested first. But because the architecture is for both conventional and differential algorithms, the results of both architectures will be observed. It is important to mention that the architecture does not change anything in the amount of quantization

of the components. Because the different structures realize the same algorithms and mathematically they do the same operations.

Another important point is the observation period of the signals. The tests are applied for indoor and outdoor scenarios. For outdoor scenarios the signal strength is taken as the possible nominal line-of-sight power and the signal components are sampled for one period of the signal component (4ms for Galileo E1-B1C). The length of this period depends on the navigation system. For the Galileo test signal component it is 4ms where as for GPS it is only 1ms. The difference in the configurations between outdoor and indoor scenarios is that the outdoor applications receive more signal power than the indoor scenarios, thus outdoor applications have usually a higher SNR. The difference is assumed to be 20dBW in the signal strength due to [8]. This difference is subtracted from the signal strength in the configuration file for the indoor test configurations. The observation period increases with the decreasing SNR. The observation period is taken as 200ms, i.e. 50 times more for Galileo signal component and 200 times more for GPS.

The quantization tests are applied for 4 different test scenarios per algorithm. These are GPS and Galileo signal components and their indoor and outdoor versions. Each test has its own test results and at the end the structures differ accordingly.

The bit width of the channels can be defined through SystemC data types. This is done with *sc_int* data type. But the processing speed decreases when this data type is used. A typical quantization test takes a few hours and any modifications that does not change the results and speed up the simulations is welcome. So the type used for data exchange is built-in type int. Although the inner structure of the differential correlation algorithm components does not differ from the conventional algorithm version, because the final SNR of the system with the differential algorithm is higher than with the conventional algorithm, there is the possibility of a higher amount of LSB dropping. The nonlinear contribution of the linear multiplication block is also a factor which makes the tests obviously necessary for the blocks after differential correlation block.

5.1 ADC in the Conventional Algorithm

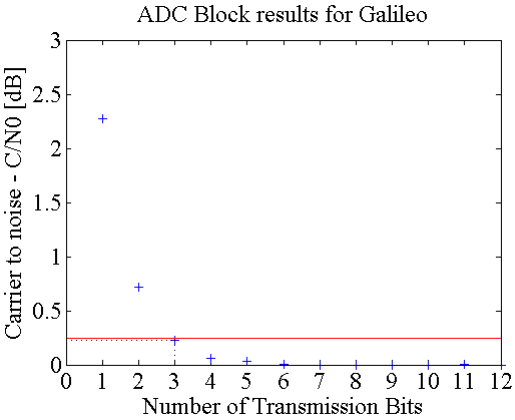

Figure 31 : Number of Bits at the output of ADC block of Conventional algorithm architecture for the Galileo signal.

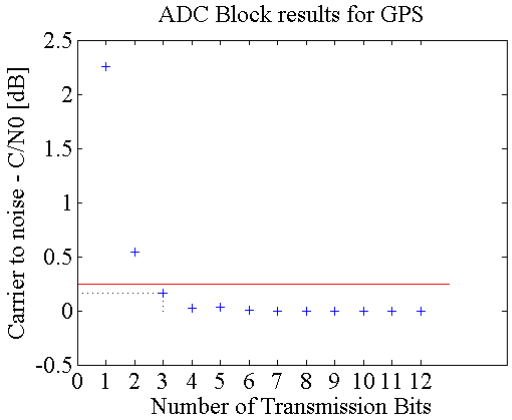

Figure 32 : Number of Bits at the output of ADC block of Conventional algorithm architecture for the GPS signal.

Figure 31 shows the response of the system to the change in the number of bits at the ADC output for the Galileo signal component. The response is measured by the SNR at the output of the whole system. The criterion for determining the number of bits dictates the SNR boundary which is highlighted by the red line. The graphs are not the direct SNR plots but the degradation plots of the system. As expected, with the increasing number of bits, the quantization loss decreases and also the degradation decreases. In each graph, the chosen result is shown with the dotted lines. In Figure 31, 3 bits are selected as result. Figure 32 shows the SNR degradation versus number of bits at the output of the ADC block for the GPS signal component. Both tests are done for stronger signals, i.e. outdoor scenarios. The strategy to cope with the weaker signals is, to increase the observation period, not the resolution of the system. This decision made the previous simulations for the indoor versions unnecessary. Thus, there are only 2 graphs instead of 4.

According to the criterion that is explained above, 3 bits is chosen for the number of bits for both the GPS and Galileo receiver tests. The receiver is expected to be functional for all GPS and Galileo signals. In Figure 32, the same algorithm is tested with a GPS signal component. This test also gave the same results and the number of bits should have been 3 for this block too.

5.2 Mixer in the Conventional Algorithm

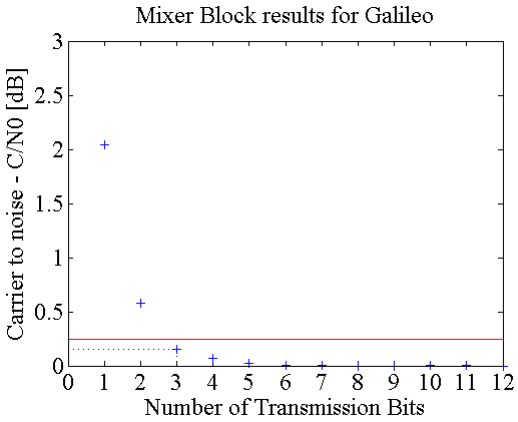

Figure 33 : Number of Bits at the output of mixer block of Conventional algorithm architecture for the Galileo signal.

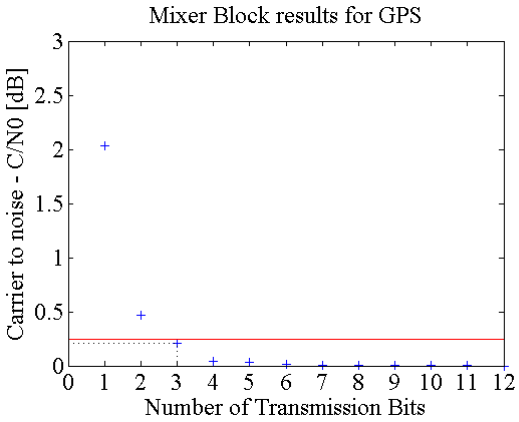

Figure 34 : Number of Bits at the output of mixer block of Conventional algorithm architecture the GPS signal.

In the mixer component, the same way as in the input stream ADC is used. By keeping the quantization of the input stream component at 3 bits for the conventional correlation algorithm and switching the quantization amount at the mixer component from 1 to 12 bits, the tests are executed once more. The results for GPS and Galileo systems imply again a resolution of 3 bits for all test configurations.

5.3 1st Accumulator in the Conventional Algorithm

While the ADC and mixer components define the number of bits, the rest of the components simply receive the samples and do the operations. The amplitudes of the signals in these components depend on observation time and the type of operations.

The process of the 1st accumulator does not generate floating numbers. It only accumulates the incoming samples by adding them. Here another variable appears. The amplitude of the signal must be determined. In this step, also the MSB is to be set so that no unused hardware will be built. In Figure 35, some samples at the output of the 1st accumulator are plotted by blue + signs for outdoor scenario of Galileo signal component tests. This shows the signal amplitude level at after the operation in the 1st

accumulator. The red line above the samples in Figure 35 is the maximum possible number which can be reached by using MSB of 14, excluding the sign bit. This means a total of 15 bits is required for the 1st accumulator for Galileo signal component. The same test configuration for is tested for LSB dropping and the results are shown in Figure 37. The red line is the 0.1dB limit. Dropping 7 bits would satisfy the criterion.

In Figure 36, the GPS signal component test results are seen. The MSB is again 15 bits. The LSBs to drop for GPS signal component is chosen to be 6 bits for outdoor scenario. But because the receiver must be a dual receiver, the dropped LSBs are set to 7 for both.

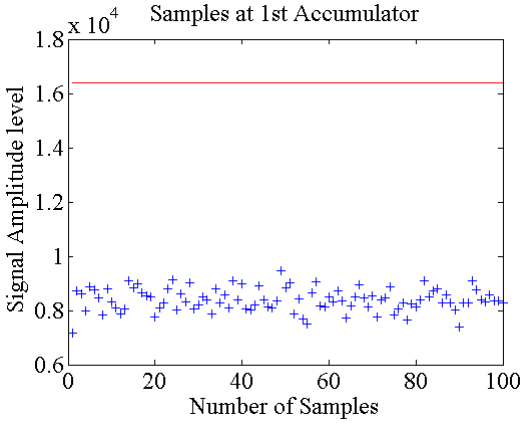

Figure 35 : Output level of the accumulated signal samples at the output of the 1st accumulator block of the Conventional algorithm for the high SNR Galileo signal.

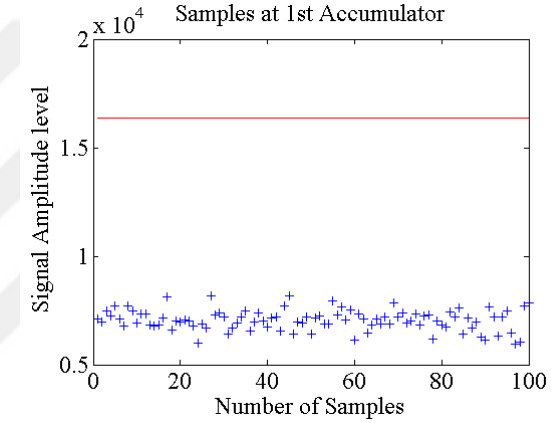

Figure 36 : Output level of the accumulated signal samples at the output of the 1st accumulator block of the Conventional algorithm for the high SNR GPS signal.

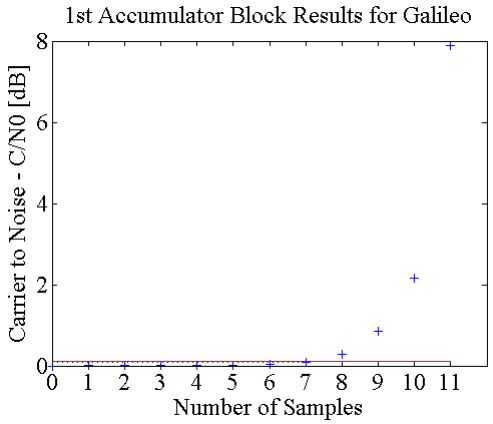

Figure 37 : Number of dropped LSBs at the output of the 1st accumulator block of Conventional algorithm architecture for the high SNR Galileo signal.

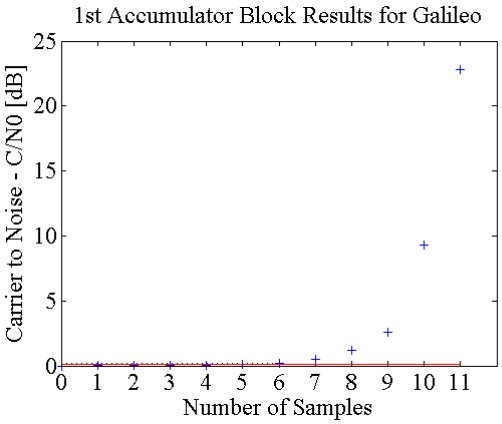

Figure 38 : Number of dropped LSBs at the output of the 1st accumulator block of Conventional algorithm architecture for the high SNR GPS signal.

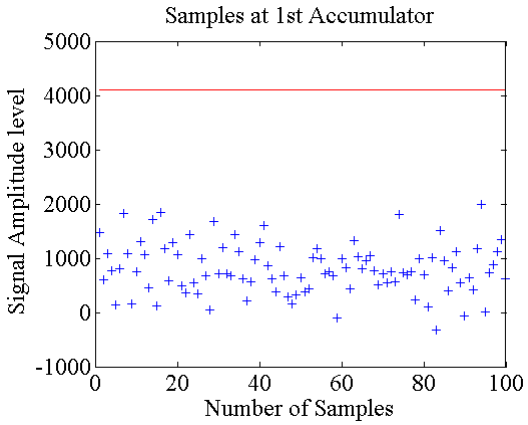

Figure 39 : Output level of the accumulated signal samples at the output of the 1st accumulator block for the low SNR Galileo signal.

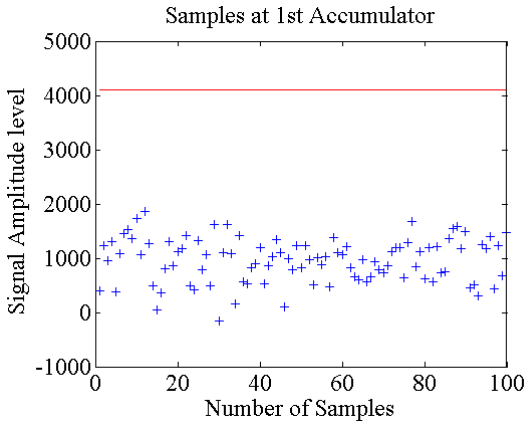

Figure 40 : Output level of the accumulated signal samples at the output of the 1st accumulator block for the low SNR GPS signal.

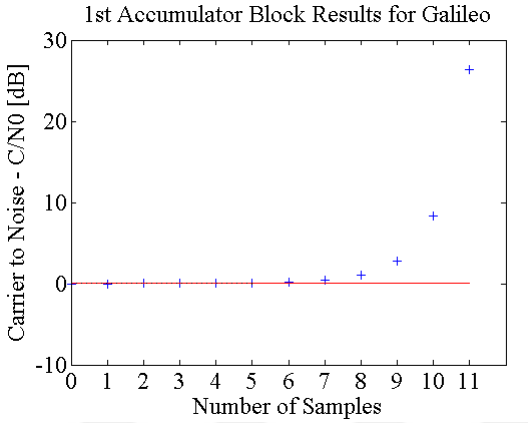

Figure 41 : Number of dropped LSBs at the output of the 1st accumulator block of Conventional algorithm architecture for the low SNR Galileo signal.

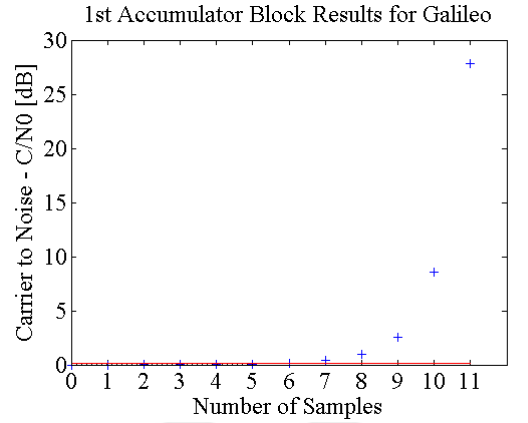

Figure 42 : Number of dropped LSBs at the output of the 1st accumulator block of Conventional algorithm architecture for the low SNR GPS signal.

The results above are for the outdoor scenario. These results are for the best signal case. The signal strength drops when the indoor scenarios are considered. To simulate the system behavior under the indoor conditions, the parameters of AWGN, which is a temperature and signal frequency band based noise, are kept constant and the signal strength is decreased by 20dBW. To compensate the effect of reduced SNR due to increased attenuation, the observation period is increased by 50 times, an observation period of 200ms (50*4ms). In practice, it is obvious that the reaction against a change in SNR of received signal cannot be changing the hardware. The only applicable change is to change the observation time. This means higher number of samples shall be accumulated in order to increase the SNR of the system. The accumulation process starts after mixer block, thus there will be no adjustments for the ADC and mixer.

At the end of the increased observation time, the mean of the signal is still decreased as seen in Figure 39 for Galileo. The upper limit is not higher than 13 bit limit of high SNR case including the sign bit. The reduction of the mean of the signal does not conflict with the increased number of samples due to the fact that the number of samples accumulated in the 1st accumulator does not change but the signal strength is lower. However, because the mean decreases, the LSBs represent more significant values and not as many bits as under outdoor condition are allowed to be dropped. For the Galileo signal component, the number of bits to drop decreases to 5 from 7.

In case of GPS, the signal components change similar to Galileo. In Figure 40, it is seen that the Galileo signal amplitude level is similar to the GPS level and it can be represented by using 13 bits. The bit dropping is also the similar to Galileo as seen in Figure 42. 5 bits can be ignored for the transmission of the data to the next component.

5.4 2nd Accumulator in the Conventional Algorithm

It has been already mentioned that the additional samples which are sampled due to the increased observation time are not accumulated in the 1st accumulator but in the 2nd. This component does not have to be used for stronger signals because only one period is enough for high detection ratios. If the signal strength is high, then the accumulated signal samples from the 1st accumulator are transmitted as they are without any operations applied.

In case of the indoor scenario, both the Galileo and the GPS signal components are limited with 12 bits as seen in Figure 43 and in Figure 44 respectively. Dropping 4 bits for both the GPS and the Galileo receiver satisfies the 0.1dB degradation criterion.

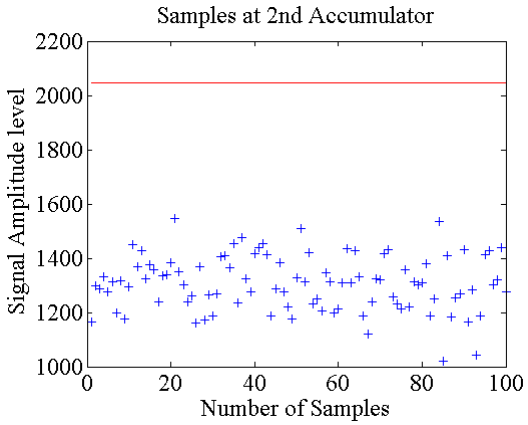

Figure 43 : Output level of the accumulated signal samples at the output of the 2nd accumulator block for the low SNR Galileo signal.

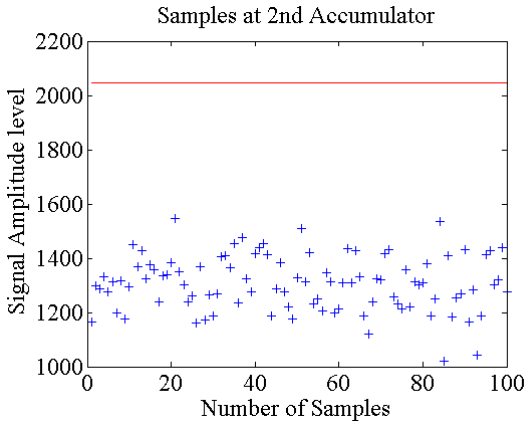

Figure 44 : Output level of the accumulated signal samples at the output of the 2nd accumulator block for the low SNR GPS signal.

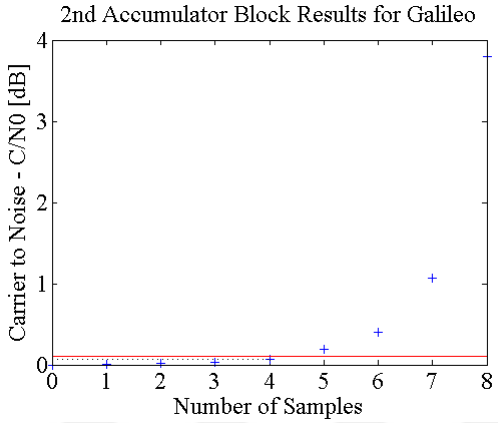

Figure 45 : Number of dropped LSBs at the output of the 2nd accumulator block of Conventional algorithm architecture for the low SNR Galileo signal.

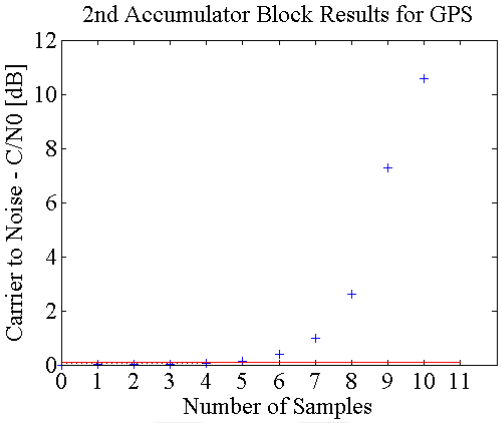

Figure 46 : Number of dropped LSBs at the output of the 2nd accumulator block of Conventional algorithm architecture for the low SNR GPS signal.

5.5 Squaring in the Conventional Algorithm

The squaring operation increases the MSB level dramatically. The MSB results, which are seen in Figure 47, are the results for Galileo tests for the outdoor scenario. It implies a 18-bit result. The number of bits for the same scenario is also 18 as seen in Figure 48. The signal amplitudes of the weaker signal strength versions of these tests are shown in Figure 51 and Figure 52 for Galileo and GPS respectively. From these figures, both in Galileo and GPS receivers, 15 bits are sufficient. The bit dropping for the Galileo receiver is shown in Figure 49. 7 bits must be dropped. In Figure 50, it is less flexible and 3 bits is allowed to be dropped. This difference may be due to the lower signal strength of the GPS signal component. Under indoor conditions, 15 bits are enough for the process. For Galileo 8 bits can be dropped where for GPS 7 bits satisfy the criterion. 8 bits is chosen.

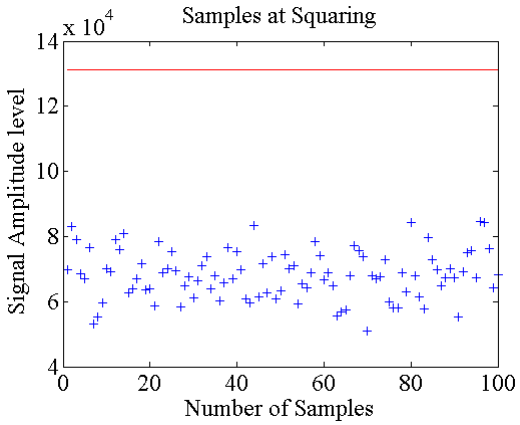

Figure 47 : Output level of the signal samples at the output of the squaring block for the high SNR Galileo signal.

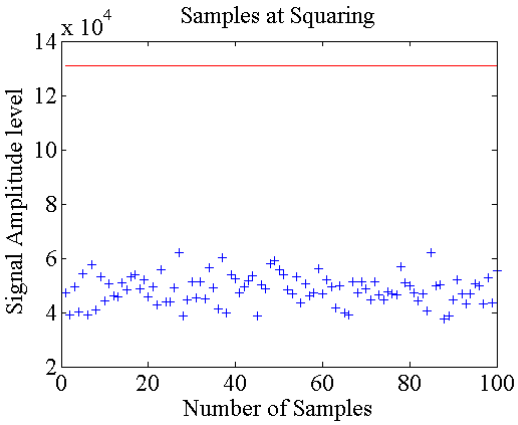

Figure 48 : Output level of the signal samples at the output of the squaring block for the high SNR GPS signal.

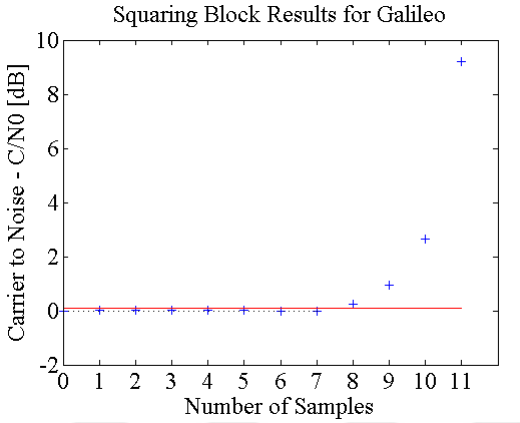

Figure 49 : Number of dropped LSBs at the output of the squaring block of Conventional algorithm architecture for the high SNR Galileo signal.

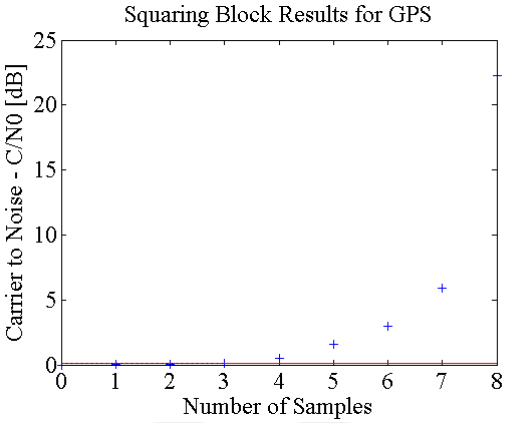

Figure 50 : Number of dropped LSBs at the output of the squaring block of Conventional algorithm architecture for the high SNR GPS signal.

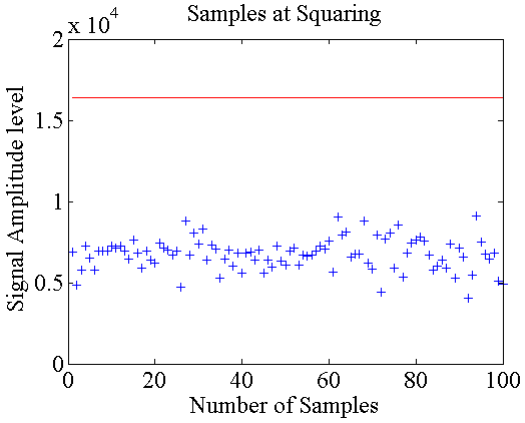

Figure 51 : Output level of the signal samples at the output of the squaring block for the low SNR Galileo signal.

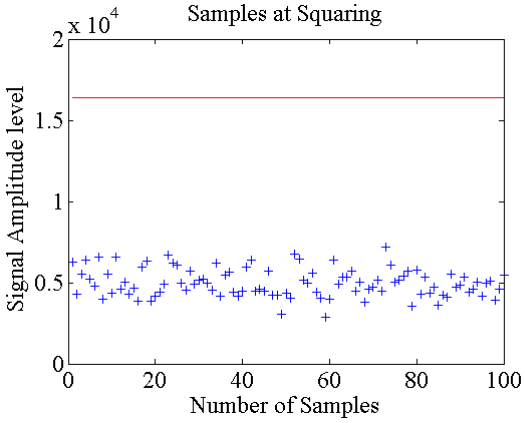

Figure 52 : Output level of the signal samples at the output of the squaring block for the low SNR GPS signal.

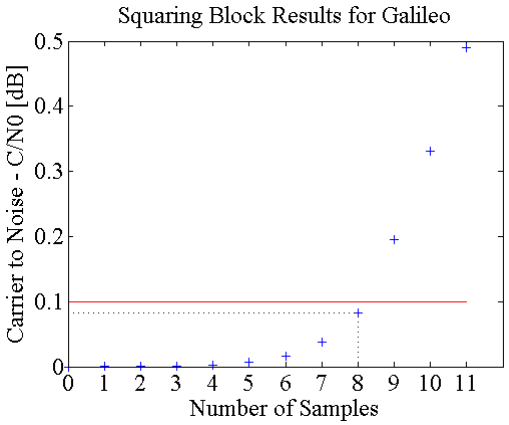

Figure 53 : Number of dropped LSBs at the output of the squaring block of Conventional algorithm architecture for the low SNR Galileo signal.

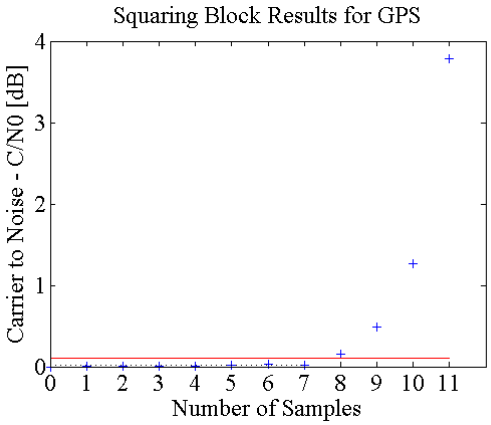

Figure 54 : Number of dropped LSBs at the output of the squaring block of Conventional algorithm architecture for the low SNR GPS signal.

5.6 ADC in the Differential Algorithm

In Figure 55 and Figure 56, the signal components for Galileo and GPS are tested with the differential correlation algorithm. The amount of quantization is taken as 3 bits in the differential correlation as well.

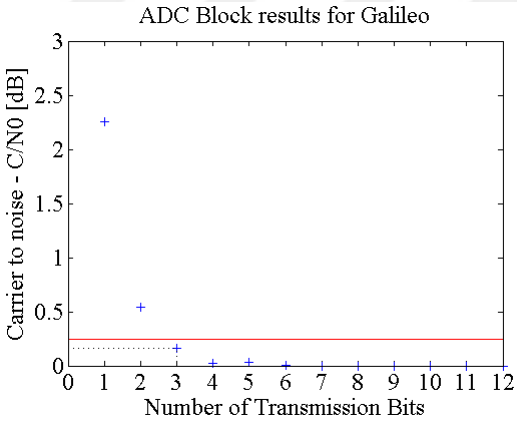

Figure 55 : Number of Bits at the output of ADC block of Differential algorithm architecture for the Galileo signal.

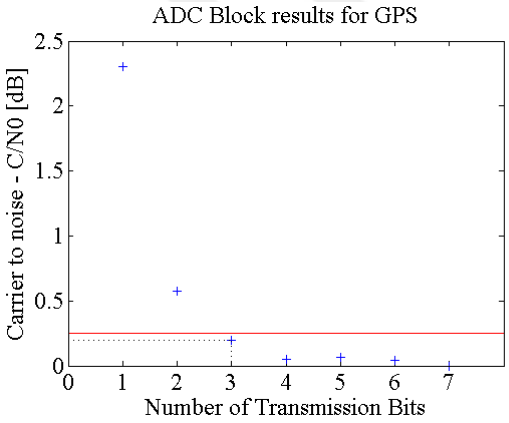

Figure 56 : Number of Bits at the output of ADC block of Differential algorithm architecture for the GPS signal.

5.7 Mixer in the Differential Algorithm

In Figure 57 and Figure 58, the results for Galileo and GPS signal components are shown respectively. The resolution is chosen to be 3 bits for mixer component.

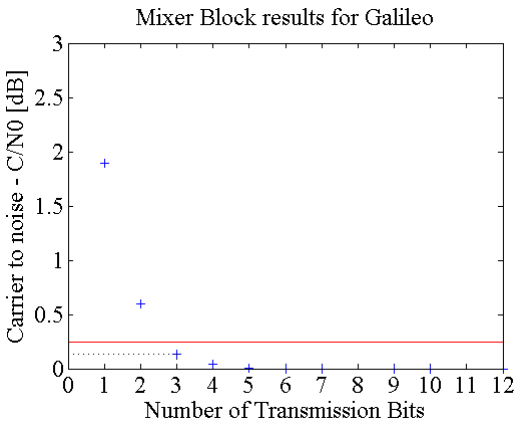

Figure 57 : Number of Bits at the output of mixer block of Differential algorithm architecture for the Galileo signal.

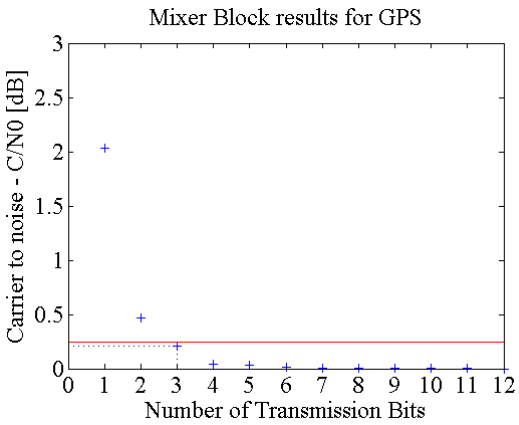

Figure 58 : Number of Bits at the output of mixer block of Differential algorithm architecture for the GPS signal.

5.8 1st Accumulator in the Differential Algorithm

The MSB in all test configurations is the same as for the conventional algorithm because there are no different operations in the system until the differential multiplication. Thus, the signal amplitudes are the same but the SNR of the system differs.

The LSB dropping schemes of the Galileo receiver are shown in Figure 59 for outdoor and in for Figure 61 for indoor environments. The determined results are 7 bits for outdoor and 5 bits for the indoor model. The results for GPS are presented in Figure 62 and Figure 60 to be 6 bits for the outdoor and 5 bits for the indoor conditions. Finally, 7 bits are chosen for dropping in the outdoor model and 5 bits for the indoor.

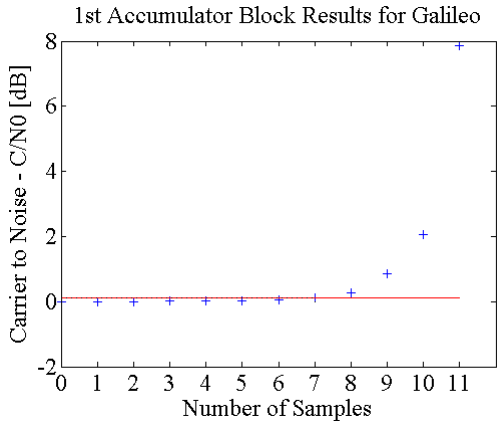

Figure 59 : Number of dropped LSBs at the output of the 1st accumulator block of Differential algorithm architecture for the high SNR Galileo signal.

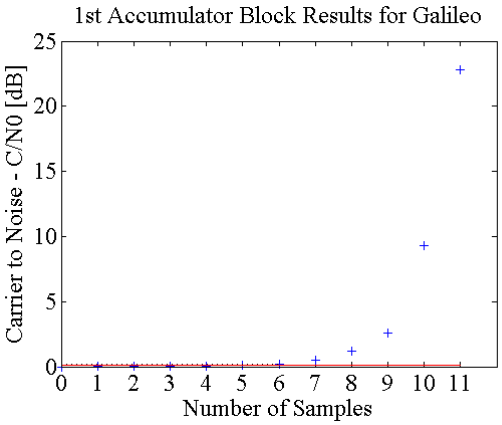

Figure 60 : Number of dropped LSBs at the output of the 1st accumulator block of Differential algorithm architecture for the high SNR GPS signal.

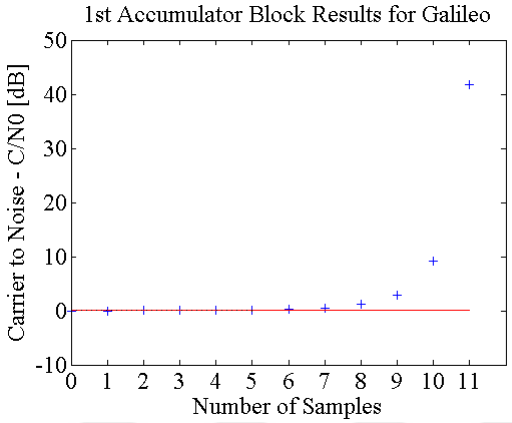

Figure 61 : Number of dropped LSBs at the output of the 1st accumulator block of Differential algorithm architecture for the low SNR Galileo signal.

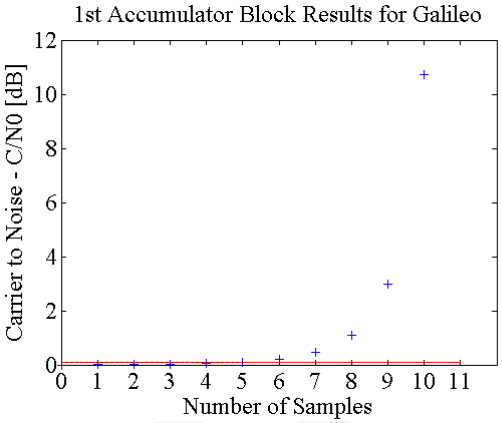

Figure 62 : Number of dropped LSBs at the output of the 1st accumulator block of Differential algorithm architecture for the low SNR GPS signal.

5.9 Differential Multiplication in the Differential Algorithm

In Figure 63 and Figure 64, the results of the outdoor conditions are shown and the MSB is 14 bits for both Galileo and GPS. These amounts reduce to 13 bits for indoor conditions as seen in Figure 67 and Figure 68 for Galileo and GPS, respectively. The LSB dropping has some differences as well. In Figure 65 and Figure 66, it is seen that 4 LSBs can be dropped. The indoor conditions do not change as seen in Figure 69 and Figure 70.

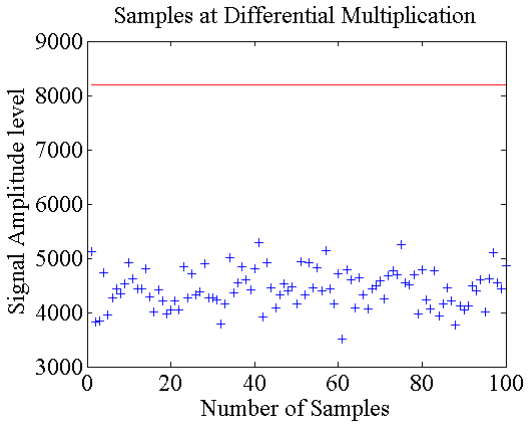

Figure 63 : Output level of the accumulated signal samples at the output of the differential multiplication block of the Differential algorithm for the high SNR Galileo signal.

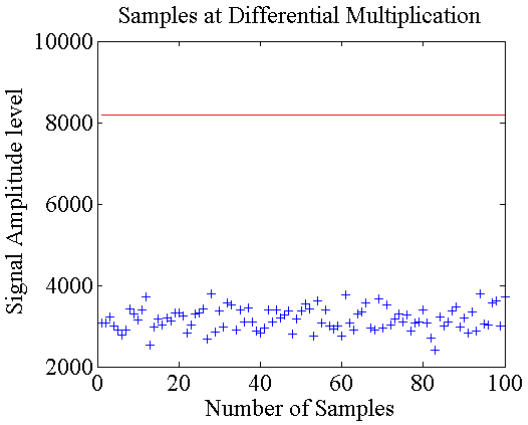

Figure 64 : Output level of the accumulated signal samples at the output of the differential multiplication block of the Differential algorithm for the high SNR GPS signal.

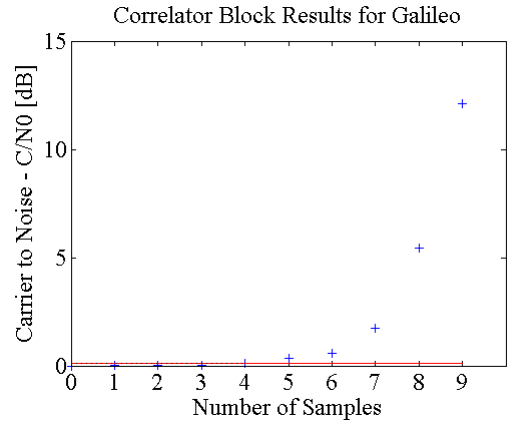

Figure 65 : Number of dropped LSBs at the output of the differential multiplication block of Differential algorithm architecture for the high SNR Galileo signal.

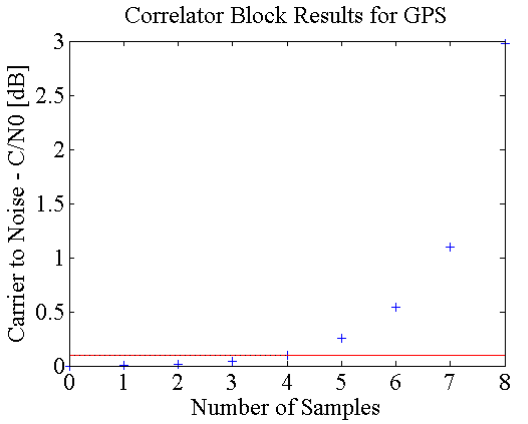

Figure 66 : Number of dropped LSBs at the output of the differential multiplication block of Differential algorithm architecture for the high SNR GPS signal.

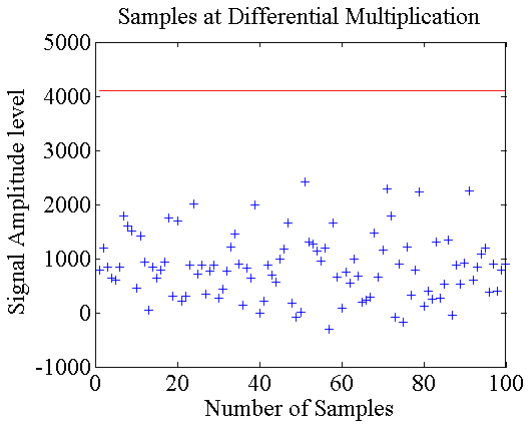

Figure 67 : Output level of the accumulated signal samples at the output of the differential multiplication block of the Differential algorithm for the low SNR Galileo signal.

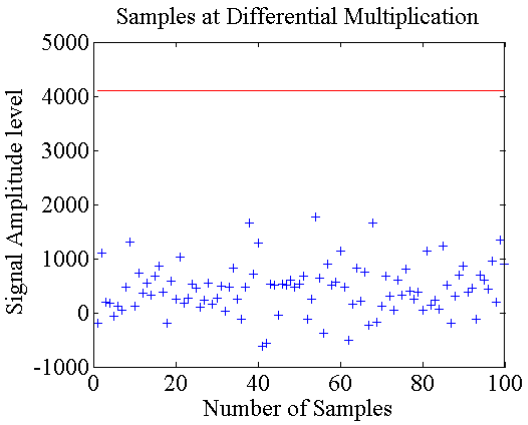

Figure 68 : Output level of the accumulated signal samples at the output of the differential multiplication block of the Differential algorithm for the low SNR GPS signal.

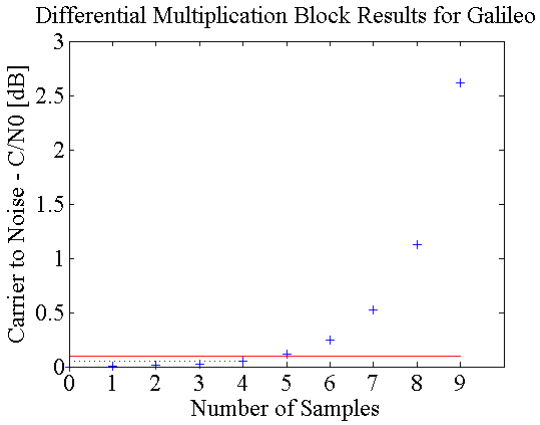

Figure 69 : Number of dropped LSBs at the output of the differential multiplication block of Differential algorithm architecture for the low SNR Galileo signal.

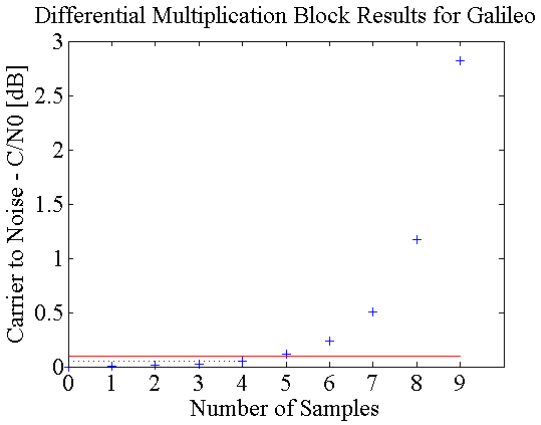

Figure 70 : Number of dropped LSBs at the output of the differential multiplication block of Differential algorithm architecture for the low SNR GPS signal.

5.10 Squaring in Differential Algorithm

For outdoor environments, the MSB of the squaring components of the GPS and Galileo receivers must be 8 with respect the Figure 71 and Figure 72. But the LSBs to drop are slightly different, being 6 for the Galileo receiver and 5 for the GPS receiver, from Figure 73 and Figure 74 respectively. The indoor tests imply that for the Galileo receiver again 8 bits are enough but GPS requires 10 bits. But both signals allow 4 bits to be omitted.

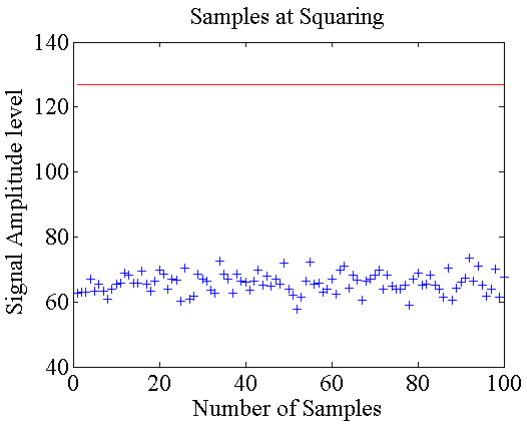

Figure 71 : Output level of the accumulated signal samples at the output of the squaring block of the Differential algorithm for the high SNR Galileo signal.

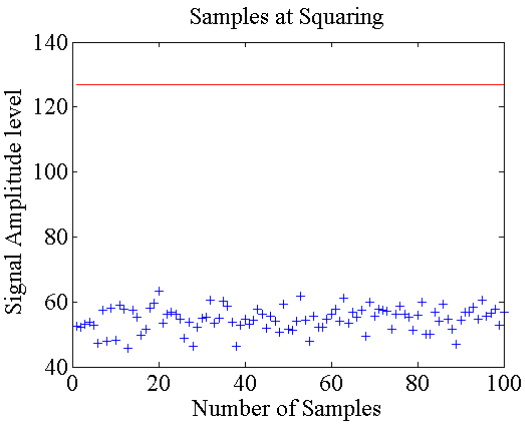

Figure 72 : Output level of the accumulated signal samples at the output of the squaring block of the Differential algorithm for the high SNR GPS signal.

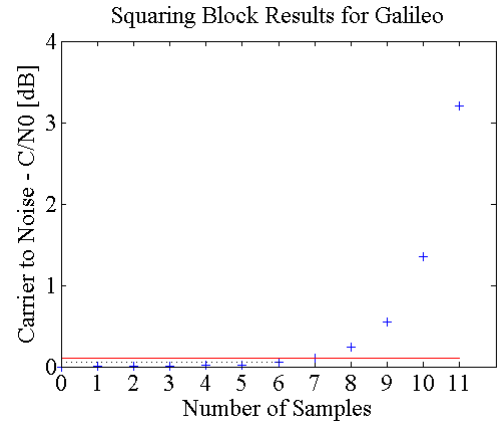

Figure 73 : Number of dropped LSBs at the output of the squaring block of Differential algorithm architecture for the high SNR Galileo signal.

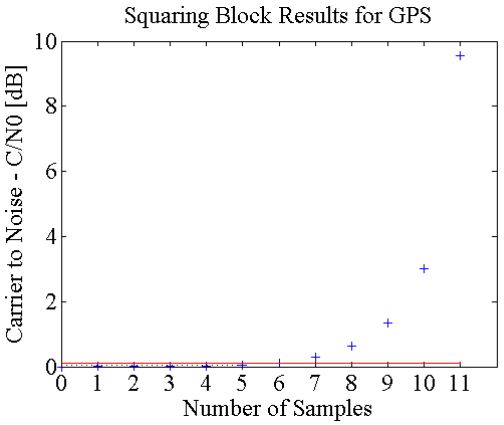

Figure 74 : Number of dropped LSBs at the output of the squaring block of Differential algorithm architecture for the high SNR GPS signal.

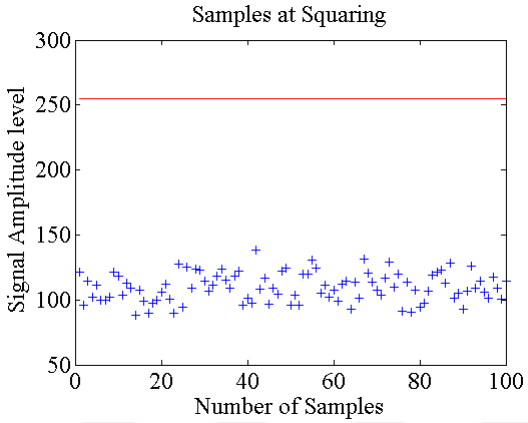

Figure 75 : Output level of the accumulated signal samples at the output of the squaring block of the Differential algorithm for the low SNR Galileo signal

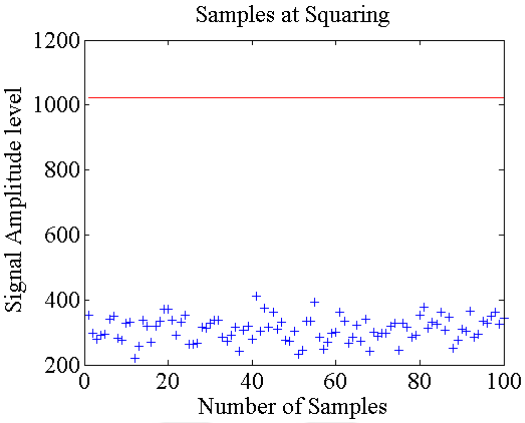

Figure 76 : Output level of the accumulated signal samples at the output of the squaring block of the Differential algorithm for the low SNR GPS signal

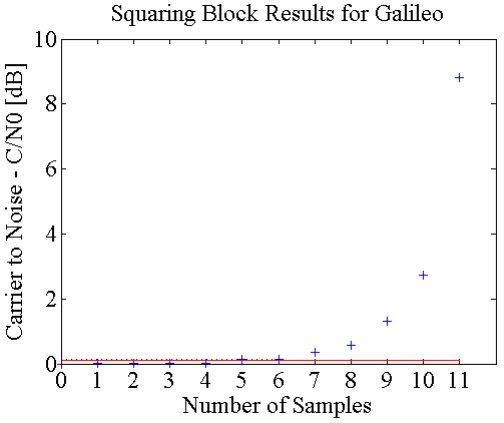

Figure 77 : Number of dropped LSBs at the output of the squaring block of Differential algorithm architecture for the low SNR Galileo signal.

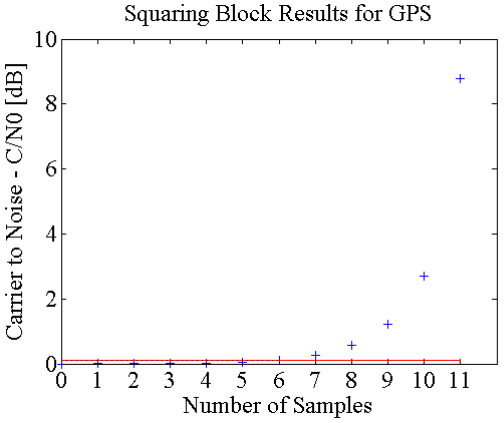

Figure 78 : Number of dropped LSBs at the output of the squaring block of Differential algorithm architecture for the low SNR GPS signal.

5.11 2nd Accumulator in the Differential Algorithm

The second accumulator is again used only for the indoor conditions. In Figure 79 and Figure 80, it is seen that the highest bit is the 13th bit. Figure 81 and Figure 82 show the number of negligible bits at the output. The results are 5 bits for both Galileo and GPS receivers.

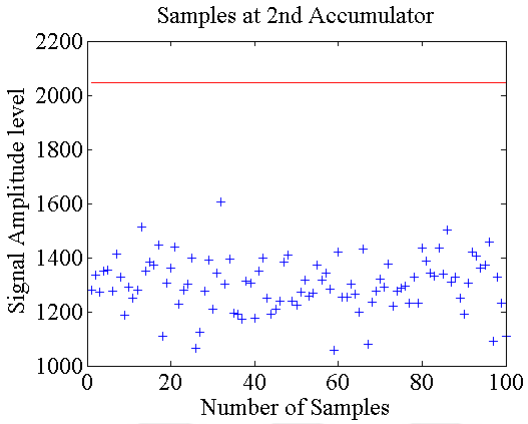

Figure 79 : Output level of the accumulated signal samples at the output of the 2nd accumulator block of the Differential algorithm for the low SNR Galileo signal.

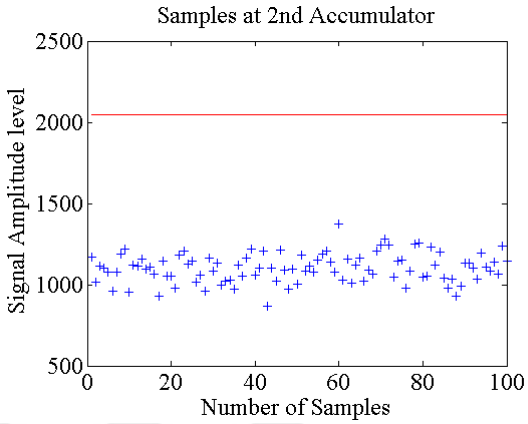

Figure 80 : Output level of the accumulated signal samples at the output of the 2nd accumulator block of the Differential algorithm for the low SNR GPS signal.

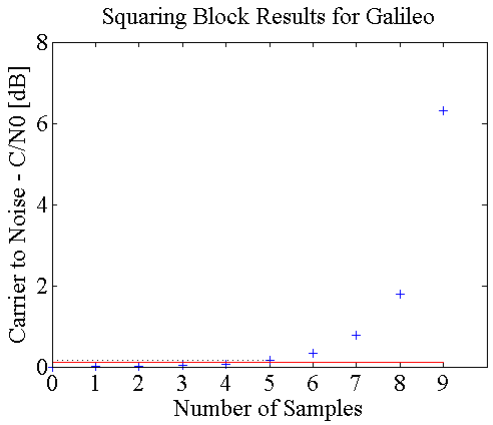

Figure 81 : Number of dropped LSBs at the output of the 2nd accumulator block of Differential algorithm architecture for the low SNR Galileo signal.

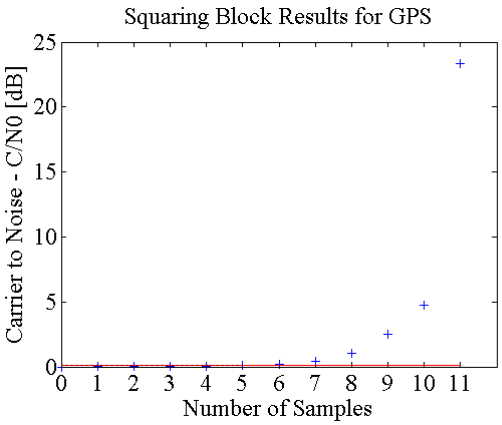

Figure 82 : Number of dropped LSBs at the output of the 2nd accumulator block of Differential algorithm architecture for the low SNR GPS signal.

5.12 Efficiency of the Architecture

The parallel architecture is the straight forward implementation of the mathematical operations for each bin. The hardware required for this method is not flexible, if the number of samples per chip is doubled, most of the hardware must be doubled due to increased number of bits per period of signal components. Thus, the hardware expansion is proportional to the expansion of the bins. Naturally, this proportion applies only to the components of on the bins. These components start with the despreading component. In other words, the ADC and mixer blocks are needed only once in any case.

On the other hand, the efficient algorithm suggests a more compact hardware approach. The number of pre-despreading blocks remains the same, only 1 ADC and 1 mixer blocks. But the rest of the architecture is both pipelined and due to the higher clock frequencies, the workload of several parallel components of the same type can be undertaken by only one component with the same type. For the implemented signal components, this hardware reallocation method makes it enough to use 1/11 of the hardware that would be used for parallel structure.

6 Conclusion

This thesis is intended to build a virtual GPS/Galileo receiver prototype. The prototype is analysed with behavioural tests and the optimum structure for components are searched. The prototype can be configured for the desired test conditions.

The architecture can be improved by adapting it to the Doppler shifts in the frequency. This frequency offset detection policy requires some additional blocks but then can easily be integrated into the existing structure after creating the new components. These new components are introduced in [6] and the simulation results are published in [7].

Another possible improvement is to switch off the invalid-time-delay bins by the earlier detection of the zero-time-delay bin. This strategy aims to decrease the power consumption of the prototype by stopping the process of irrelevant data. This can be done by using a dynamical threshold instead of constant. The dynamical threshold can be found out by comparing the zero-time-delay bin with the other bins.

Bibliography:

[1] Spilker, J.J., Jr., “GPS Signal Structure and Performance Characteristics”, Navigation, Journal of the Institute of Navigation, Vol. 25, No.2, 1978.

[2] Spilker, J.J., Jr., “Global Positioning System: Theory and Applications”, Washington, DC, USA: American Institute of Aeronautics and Astronautics, 1996, vol.1, ch.3: GPS Signal Structure and Theoretical Performance, pp. 57-119.

[3] European Space Agency / Galileo Joint Undertaking, “Galileo Navigation Primary Codes”, April 2006

[4] A. Schmid and A. Neubauer, “Performance Evaluation of Differential Correlation for Single Shot Measurement Positioning”, Proc. ION GNSS Int. Technical Meeting of the Satellite Division, pp. 1998-2009, Sept. 2004.

[5] Philip Ward, NAVWARD GPS Consulting, “Understanding GPS: Principles and Applications”, 1996, ch.5: Satellite Signal Acquisition and Tracking, pp. 119-208.

[6] A. J. Van Dierendonck, “Global Positioning System: Theory and Applications”, Washington, DC, USA: American Institute of Aeronautics and Astronautics, 1996, vol.1, ch.8: GPS Receivers, pp. 329-407.

[7] Arinc Engineering Services, El Segundo, USA, “Interface Specification: Navstar GPS Space Segment/Navigation User Interfaces”, March 2006.

[8] N. Agarwal, J. Basch, “Algorithms for GPS Operation Indoors and Downtown”, published online, 13 November 2002.