ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/311948864

GMSK Modulator/Demodulator Design and Implementation on FPGA for Cube Satellites

Thesis · June 2016

DOI: 10.13140/RG.2.2.24158.64321

CITATIONS
0

READS
4,246

1 author:

Oguz Kislal
Bogazici University

13 PUBLICATIONS **8** CITATIONS

SEE PROFILE

All content following this page was uploaded by Oguz Kislal on 28 December 2016.

The user has requested enhancement of the downloaded file.

ISTANBUL TECHNICAL UNIVERSITY

ELECTRICAL – ELECTRONICS ENGINEERING FACULTY

Software Defined GMSK Modulator/Demodulator Design for Cube Satellites

BSc Thesis by

Ahmet Oğuz KIŞLAL

040110075

Department	**:**	**Electronics and Communication Engineering**
Programme	**:**	**Electronics and Communication Engineering**

Supervisor: Dr. H. Bülent YAĞCI

MAY 2016

PREFACE

I wish to express my sincere gratitude to my supervisor Dr. H. Bülent YAĞCI who gave me the opportunity to carry out this work. I also want to thank Onur ÇAKAR, Arda DEMİRAY, H. Bahadır TUĞREL, Alican ÇAĞLAR and all other members and students at the ITU RF Electronics Laboratory. A special thanks to Dr. Osman CEYLAN for his guidance about theoretical and practical approaches to solve the problems. I would also like to thank my family for encouraging and supporting me along my life. Finally, I want to thank R. Can TOK for making my whole university life interesting.

May 2016 A. Oğuz Kışlal

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

SUMMARY

Nowadays, traditional radio communication systems, swiftly, are replaced with software defined radios (SDR). The idea behind SDRs is implementing functions in radios by software instead of analogue components. Traditional radios were using separate hardware for many application like modulation, demodulation and filter. All of these components are vital for a communication system and can be implemented with software basis. Along with this thesis, GMSK modulation/demodulation has implemented on FPGA with VHDL to use in a SDR project which is planned to be placed in a cube satellite. The motivation for this project is, providing a flexible system that even baseband carrier frequency can be changed by reprogramming FPGA. Since FPGA’s have the advantages of parallel and high speed processing and field programmable, the core of the SDR system has built around the FPGA.

The first test for GMSK modulator and demodulator are done with a simulation program. After that, a transmitter board has designed and the system is tested on this board. Finally, the GMSK modulator/demodulator is merged with the old system and tested with a satellite computer model. As a satellite computer an MSP430 family device is used. The merged system has the options serial peripheral interface (SPI), Universal asynchronous receiver/transmitter (UART) and Inter-Integrated Circuit (I2C) to communicate with other devices and satellite computer. All of these standards are also explained.

ÖZET

Günümüzde, geleneksel radyo iletişim sistemleri yerini yazılım tanımlı radyolara bırakmaktadırlar. Yazılım tanımlı radyonun arkasındaki temel mantık, normalde donanım tabanlı olarak gerçeklenen modülasyon, demodülasyon, süzgeç gibi bir haberleşme sisteminin önemli parçalarını yazılım ile gerçeklemektir. Bu tez kapsamında, küp uydularda kullanılmak üzere tasarlanan bir yazılım tanımlı radyo projesi için FPGA kullanılarak VHDL yardımı ile GMSK modülatör ve demodülatör tasarımı yapılmıştır. Projenin temel amacı, temel bantta ki taşıyıcının bile değişebildiği FPGA yı sahada programlayar değiştirilebilecek, esnek bir sistem ortaya koymaktır. FPGA'ların paralel ve hızlı işlem yapabilme ve de sahada programlanabilme özellikleri, YTR projesi için ideal olduğuna karar verilmiş ve sistem FPGA etrafında kurulmuştur.

GMSK modülasyon ve demodülasyonuyla ilgili ilk testler bir simülasyon programı yardımıyla yapılmıştır. Daha sonra, bir verici kartı tasarlanmış ve sistem bu kart üzerinde denenmiştir. Son olarak GMSK modülatör/demodülatör eski sistem ile birleştirilmiş ve bir uydu bilgisayarı modeli ile denenmiştir. Uydu bilgisayarı olarak MSP430 ailesinden bir işlemci kullanılmıştır. Birleştirilmiş sistem başka entegrelerle ve uydu bilgisayarı ile haberleşebilmek adına SPI, UART ve I2C opsiyonlarına sahiptir. Bahsi geçen bu standartlar tez de anlatılmıştır.

1. INTRODUCTION

Communication systems consist of different blocks and features. The most important ones are: Source coding, channel coding, modulation and demodulation. Source coding is for making message signal shorter which means that lesser bit should be transmitted. Channel coding is, adding some bits according to a rule to the message signal so that even if some problem occur in the channel, receiver can guess the message signal and solve the message correctly. The modulation is the transmitting the message signal which has low frequency with the help of a high frequency carrier

The FSK modulation has the advantages of can be easily demodulated with amateur radios, low frequency band need and can be used with 9.6k baud rate efficiently [1,2]. The GMSK modulation on the other hand, due to the constant envelope feature, has very good spectral efficiency, resilient against the amplitude base noise, acceptable receiver complexity and can be used with non-linear amplifiers which means that cheap and high gain C type amplifiers can be used. [3,4].

A CubeSat is a type of mini satellite for space research that have sizes of aproximatelly 10×10×11 cm. CubeSats have a mass of no more than 1.5 kilograms for per unit, and most commonly put in orbit by deployers on the International Space Station or launched as secondary component on a rocket. Conceptually, small satellites are becoming increasingly popular because of their relatively low cost and the short design period required. Small satellites are used for various missions such as education, monitoring of space debris, earth observation, environmental monitoring, space weather measurements, gravimetry, and luminous observations [5]. Figure 1.1 shows ITUpSAT1, a CubeSat that was launched in 2009. In ITU Electrics-Electronics faculty RF electronics laboratory, the SDR for cube satellite project is continuing. Planned launch of the cube satellite, which have been designing in Istanbul Technical University, is at the last quarter of 2016.

At the satellite communication systems, the preferred modulation type is important. Many cube satellite has mostly 2 different modulation option for communication. Frequency shift keying (FSK) and Gaussian minimum shift keying (GMSK).

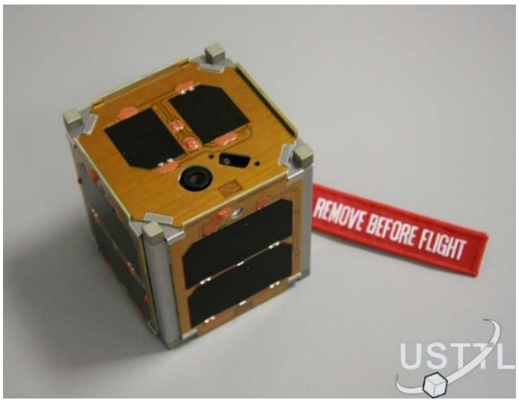

Figure 1.1: ITUpSAT1 1 unit sized Cube Satellite

Our motivation for this project is to provide a flexible system for cube satellites. There are some serious challenges for satellite communication like Doppler Effect, phase and frequency shift and most of these problem can be solved with software defined architecture. GMSK is one of the most preferred modulation type among amateur radio community due to the fact that, it has really good performance in terms of bandwidth and it can be implemented with acceptable complexity.

Section 2 explains the details of software defined radio. Section 3 gives a background about continuous phase modulations (CPM) and explains the details of GMSK. Section 4 and section 5 shows the simulation and implementation results of GMSK. Finally, section 6 conclude the thesis and give some ideas for future works.

2. SOFTWARE DEFINED RADIO

The software defined radios are getting popular every day. Due to digital electronics technology evolution, the SDRs become more cheap and available. In the past, such systems are only used at military purposes however, these days, SDRs preferred in many applications. Software defined radio is defined as a system where all or a part of physical layers of a radio implemented by means of software [6]. The original SDR concept is suggested by Joseph Mitola. Figure 2.1 shows the ideal SDR system which defined by Mitola [7].

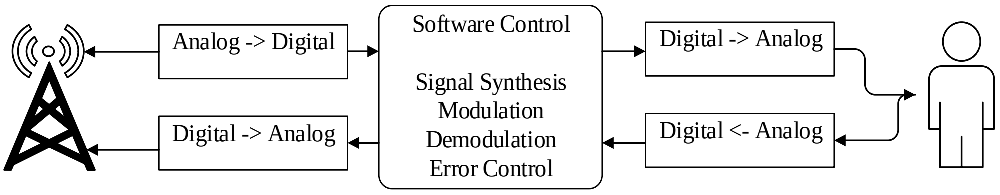

Figure 2.1: Ideal SDR system suggested by Mitola

Main purpose of the SDR’s are processing the radio signal. Modulation/demodulation, signal synthesising, channel and source coding processes are can be done with a DSP, FPGA or a similar processor which leads system to have less hardware component. Besides, system is reprogrammable so under an unexpected situation, the system might be reprogrammed. This feature makes these systems life longer. In contrast traditional systems are fixed, which means not flexible, can be effected by any means of nature much more easily and more expensive than SDRs makes them a worse alternative. There are numerous advantages of SDR. Some of them are: Simple design and production, posibility of reprogramming, usability of any signal processing tecnique, flexiblility, less separate element need and can be used for more than one application at the same time.

Figure 2.2. shows that a basic SDR systems hardware and software layers. System’s hardware stage consist of, receiver antenna which takes the RF signal, a receiver circuit to turn RF signal to IF, an analog digital converter (ADC) to convert IF signal to digital, a digital analog converter (DAC) to convert processed digital data to analog signal and the transmittter circuit which converts the signal at IF frequency to RF frequency with desired power. Line coding, modulation, demodulation, filter, mixer and such signal processings are implemented with an FPGA or a DSP.

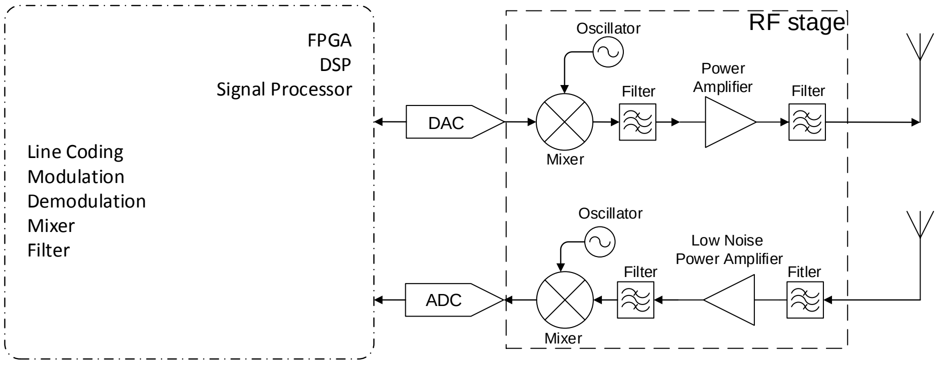

Figure 2.2: SDR structure

Choosing the appropriate signal processor is cruical for an SDR. The system's needs should be defined well and considered when chosing it. FPGAs, DSPs and ASICs are most common choices for SDRs. Figure 2.3 shows the differences between FPGAs, DSPs, and ASICs [8,9].

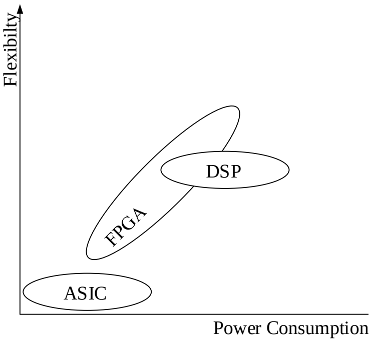

Figure 2.3: Comparision of DSP, FPGA and ASIC

Modern communication systems have high baud rate features. That's why, power consumption, flexibility and parellel processing feature is important when choosing the signal processesor. Parallel processing at DSP based system only possible with multiple DSP chip and this takes them under a bad position [10]. FPGA's parallel processing ability, low power consumption and flexible nature makes them the perfect choice for this project.

3. CONTINIOUS PHASE MODULATION

Continuous phase modulation (CPM) is a form of digital modulation which used commonly in wireless communication. Its spectral power efficiency is higher in comparison to the digital phase modulations. For example, in QPSK, the carrier might change instantly from a cosine shape signal to a sine shape signal. This leads discontinuity at phase which causes some problems in terms of out-of-band power. Continuous phase frequency shift keying (CPFSK), minimum shift keying (MSK) and Gaussian minimum shift keying (GMSK) are common examples for continuous phase modulations.

3.1 Continuous Phase Frequency Shift Keying

In traditional FSK, $\log_2 M$ frequencies are used to transmit the symbols where M is the bit number to define a symbol and defined as.

$$S_i(t) = \sqrt{\frac{2E}{T}}\cos(2\pi f_i t) \tag{3.1}$$

Where E is the energy of the signal, T symbol duration, f_i is the carrier frequency depend on the data and t is the time variable. However, this does not guarentee the phase contunity. Hence, CPFSK is defined as

$$s(t) = \sqrt{\frac{2E}{T}}\cos(2\pi f_c t \ + \left(\frac{h}{2T}\right)\int_0^t m(\tau)\, d\tau) \tag{3.2}$$

Where m(τ) is the message signal and h is the modulation index. The integral is in the cosine is given the continuous nature of CPFSK because, a finite integral doesn’t contains any discontinuity, even when the message signal has discontinuity which is mostly the case.

3.2 Minimum Shift Keying

MSK is a variation of CPFSK which h=0.5 and M=2. With these specifications, and after some trigonometric transform the modulated signal is obtained as

$$s(t) = a_I(t)\cos\left(\frac{\pi t}{2T}\right) cos2\pi f_c t + a_Q(t) sin\left(\frac{\pi t}{2T}\right) sin2\pi f_c t \tag{3.3}$$

The Figure 3.1 shows the every component of this signal.

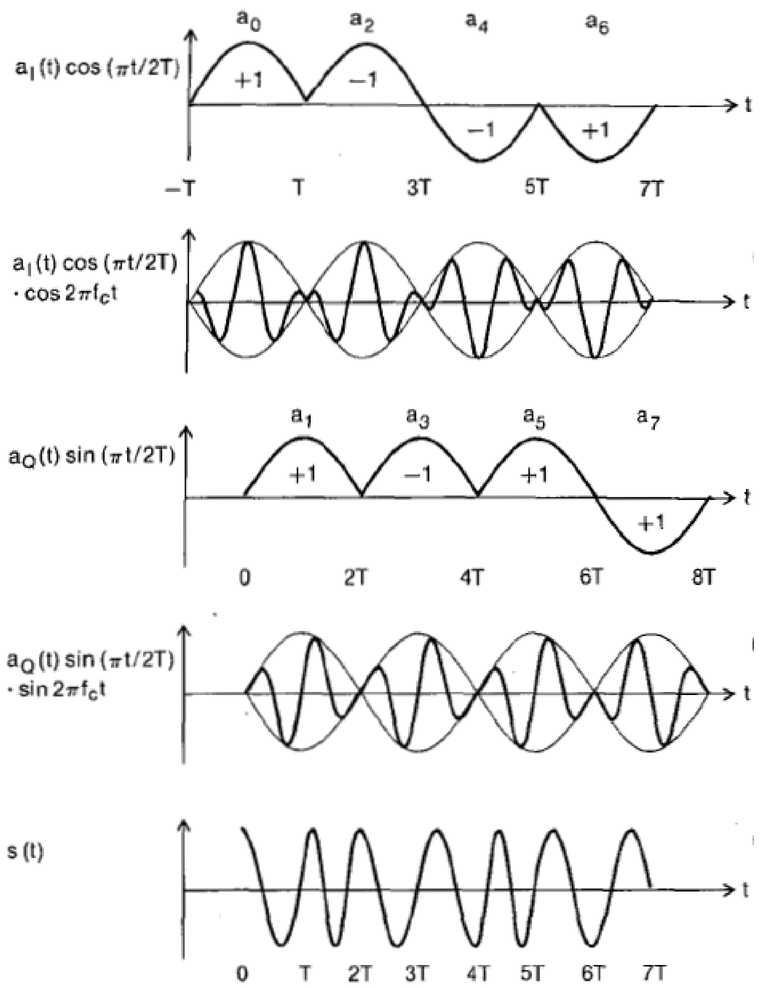

Figure 3.1: Signal components of MSK [11]

3.3 Gaussian Minimum Shift Keying

In many communication system, it is desired to suppress the out of band radiation power in the adjacent channel as much as possible [12]. To have a narrow bandwidth, some operations might be needed on the modulated signal. The signals harmonics and impact of these harmonics to the signal determine the signal's smoothness. If a signal's phase change continuously, that signal can be stated as a smooth signal.

GMSK is a modulation type which is can be achieved by a MSK modulator with a pre-modulation Gaussian low pass filter. It became a standard for satellite communication after (CCSDS) achieve the communication with GMSK.

GMSK are used in many application due to its low out of band power characteristic and the constant envelope characteristic. These features makes it suitable for both coherent and non-coherent detection. The constant envelope makes it more immune to the noise and the inexpensive class-C amplifiers may be used with this modulation.

The energy from a one symbol slot may be spread out over neigbour symbol slots. This effect is called intersymbol interferance (ISI) and it may be caused by the channel, when the RMS delay spread becomes an appreciable fraction of the bit period, or by the filtering the data pulses even before the modulation process, to supress out of band power. This effect is important because, ISI is introduced at GMSK modulation due to the fact that a gaussian filter is needed before the modulation process starts.

The Gaussian shaped filter is defined with a variable called bandwidth-time product, BT, which B is stand for bandwidth and T is stand for for a unit bit interval. This variable has great importance because, when it gets smaller, ISI is becoming a bigger problem but the out of band power and spectral efficiency is getting better. When the BT product is getting bigger, ISI is getting lower but spectral efficiency gets worse. A better understanding can be seen in Figure 3.2.

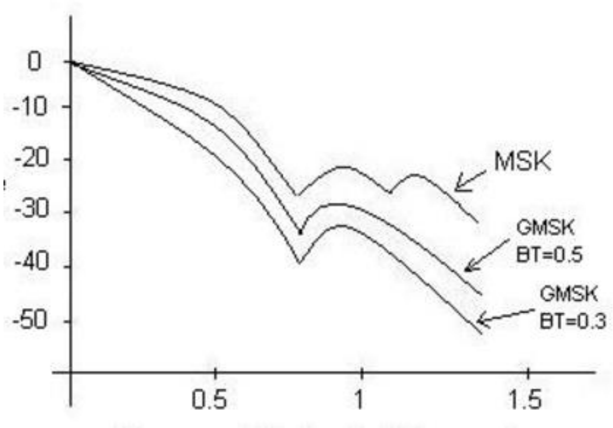

Figure 3.2: Power spectral density of MSK/GMSK for different BTs

3.3.1 The Modulation of GMSK

3.3.1.1 GMSK Modulation with FM Modulator

The GMSK modulation was founded by Murota [13]. He propose that, using a Gaussian filter on a 2 PAM signal, and then using an FM modulator with a modulation index h = 0.5, as shown in Figure 3.3, produce a GMSK modulated signal [14].

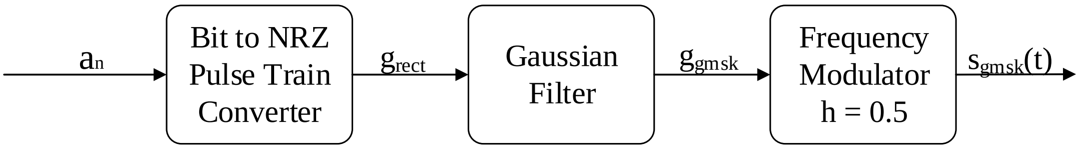

Figure 3.3: GMSK Modulator with FM modulator aproximation

Where a_n is the bit input either ‘1’ or ‘0’, g_{rect} is the non return zero pulse train which is shown in Figure 3.4 and defined as [15].

$$g_{rect}(t) = \begin{Bmatrix} U_m, -\frac{T_b}{2} \leq t \leq \frac{T_b}{2} \\ 0 \quad , \ otherwise \end{Bmatrix} \tag{3.4}$$

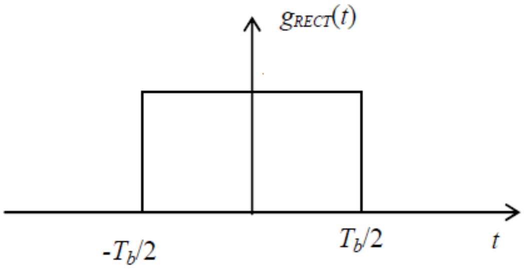

Figure 3.4: Rectangular pulse

Where U_m can be 1 or -1 depending on the input bit and T_b is the symbol interval. After g_{rect} produced it is filtered with a Gaussian filter. Impulse response of the Gaussian filter is shown in Figure 3.5 and given by [15].

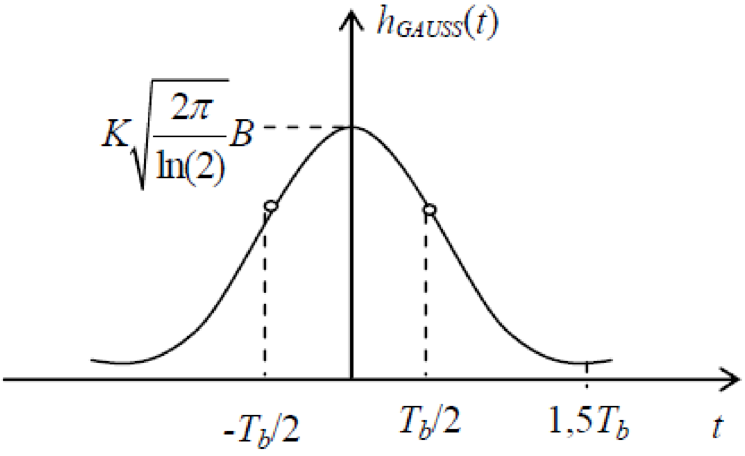

Figure 3.5: Impulse Responce of Gaussian Filter

$$h_{gauss}(t) = K\sqrt{\frac{2\pi}{ln2}}\text{B}e^{-\frac{2(B\pi)^2}{\ln 2}t^2} \tag{3.5}$$

Where B is defined as bandwidth of the gaussian filter. The g_{gmsk} shown in (3.6) can be defined as

$$g_{gmsk}(t) = g_{rect} * h_{gauss} \tag{3.6}$$

Where * sign is for convolution operation. When the convolution is applied, the result is shown in Figure 3.6 and the signal is [15]

$$g_{gmsk}(t) = -\frac{K}{2\sqrt{ln2}}\left(Erf\left(2B\pi t\frac{t-\frac{T_b}{2}}{\sqrt{ln4}}\right) - Erf(2B\pi t\frac{t+\frac{T_b}{2}}{\sqrt{ln4}})\right) \tag{3.7}$$

Where Erf is the error function and defined as [16]

$$Erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \tag{3.8}$$

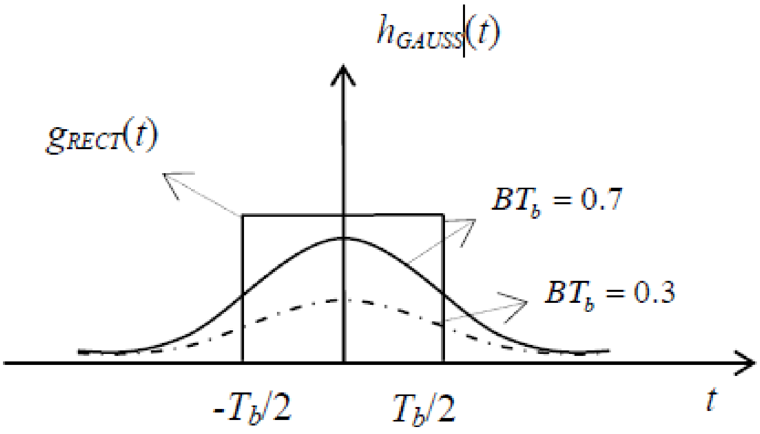

Figure 3.6: The pulse shape after the Gaussian filter for different BT products

After that, g_{gmsk} applied to a FM modulator and the result is given as

$$s_{gmsk}(t) = 2\pi\frac{m}{T_b}\int_0^t \sum_{i=0}^{\infty} b_i g_{gmsk}(\tau - iT_b)d\tau \tag{3.8}$$

Evet though this method is usefull to understand the concept and simulation purposes, a more practical approximation is needed since setting modulation index h exactly to 0.5 is very challenging hence a better approximation is needed.

3.3.1.2 GMSK Modulation with I/Q Modulator

In digital communications, modulations are often expressed in terms of I and Q. On a polar diagram which is a way to show I and Q parts of a signal, the I axis lies on the zero degree phase reference and the Q axis is the orthogonal axis to the I axis. The signal vector projection onto the I axis is its "I" component which means "In-phase" and the projection onto the Q axis is its "Q" component which stands for the "Quadrature" component of the signal. Figure 3.7 shows a visual explanation of I/Q representation of signal.

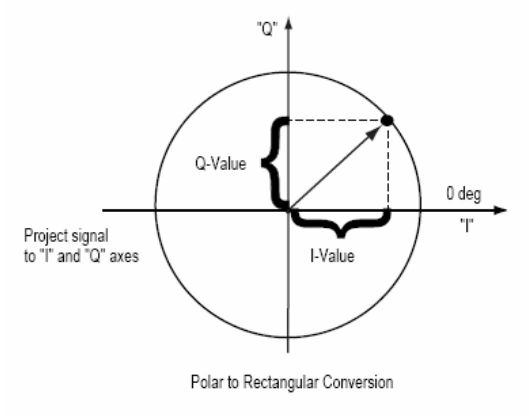

Figure 3.7: I/Q Representation of a signal

The Laurent decomposition can be applied to the GMSK signal with the approxmation of finite impulse responce of Gaussian filter [17]. As these decomposition's results shows that GMSK modulated signal is actually superposition of two OQPSK signals[14]. Hence, by using I-Q modulating technics, GMSK modulator can be implemented as well. The block diagram of the modulator which is obtained by this approxmation is given in Figure 3.8.

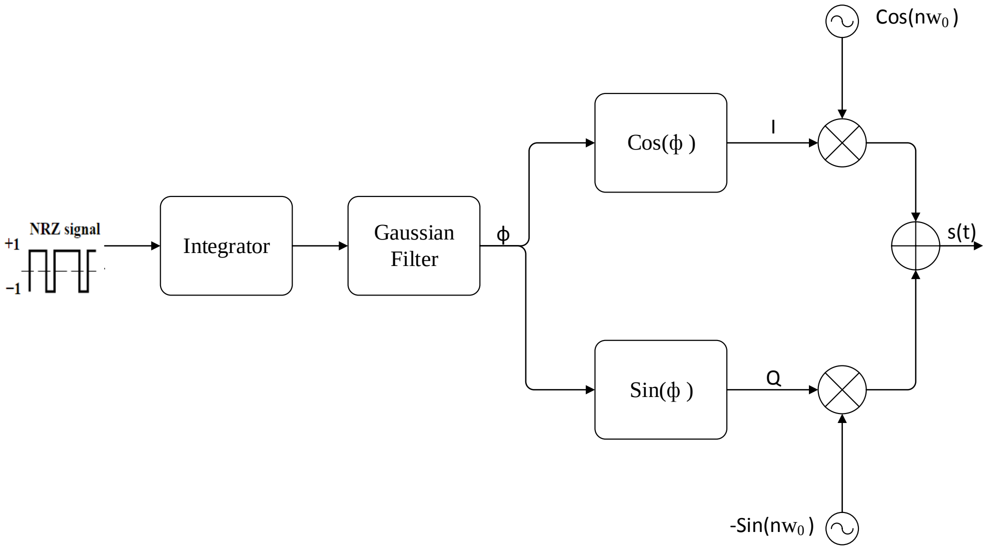

Figure 3.8: OQPSK based I-Q GMSK modulator

According to this approximation the output signal can be defined as

$$s(t) = I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t) \tag{3.9}$$

One of the crucial thing that should be considered about this approximation is, BT product of the Gaussian filter should be between 0.25 and 1, for this modulator [14]. Many applications of GMSK prefer mainly 3 different BT product: 0.3, 0.5 and 0.7. Since, BT = 0.3 gives the best performance in terms of power spectral density and it is also used at GSM standart, BT product is chosen as 0.3 at this project.

3.3.2 The Demodulation Theory of GMSK

In order to solve the GMSK modulated signal, the phase change of the signal should be found for every symbol interval T_b. The phase difference can be defined as [16]

$$\Delta\phi_b(t) = \phi(t) - \phi(t - T_b) = \frac{\pi}{2T_b}\int_{t-T_b}^{t}\big(g_{gmsk}(\tau)\big)d\tau \tag{3.10}$$

The phase change is limited with T_b, which means that in one period, maximum phase change has occured. The phase change can be positive or negative according to the bit transmitted.

To find ϕ(t), I and Q components of the modulated signal should be obtained. hence, modulated signal should be multiplied with cosine and sine signals which are defined as

$$I(t) = s_i(t)\cos(2\pi f_c t) \tag{3.11}$$

$$Q(t) = s_i(t)\sin(2\pi f_c t) \tag{3.12}$$

Where $s_i(t)$ stands for the GMSK modulated signal and f_c stands for the carrier frequency of GMSK modulated signal.

After that, dividing them and calculating the arctan value of the result gives the g_{gmsk} aproximately, which defined in (3.6). Finally, a derivative operation and a decider to decide either the signal represent ‘1’ or ‘0’ should be performed. The block diagram of demodulator given in Figure 3.9.

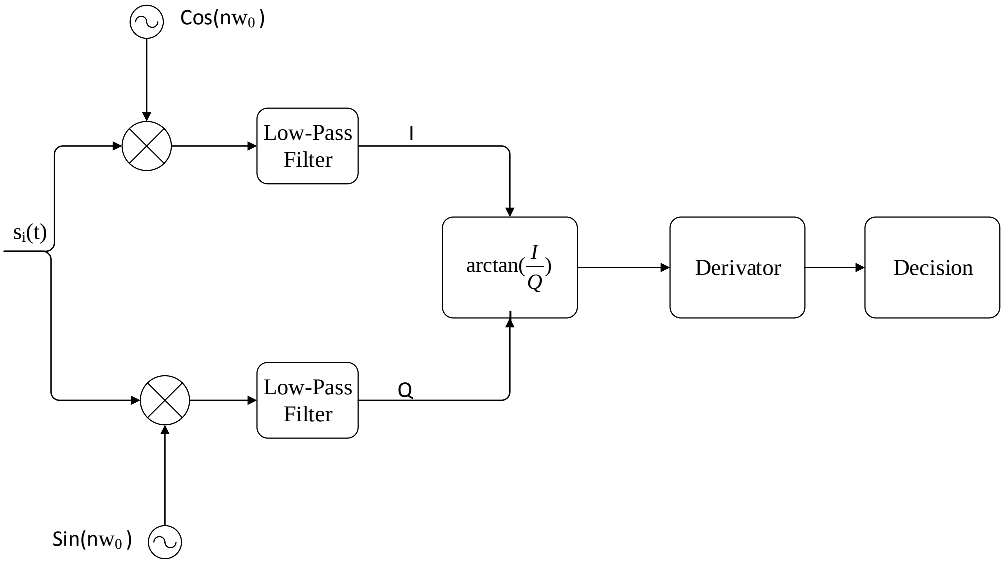

Figure 3.9: I-Q based GMSK demodulator

4. GMSK SIMULATION RESULTS

All simulations run in matrix laboratory (MATLAB) software. I-Q based GMSK modulator and demodulator are used in simulations.

At first binary data is produced and NRZ mapping is done as shown in Figure 4.1

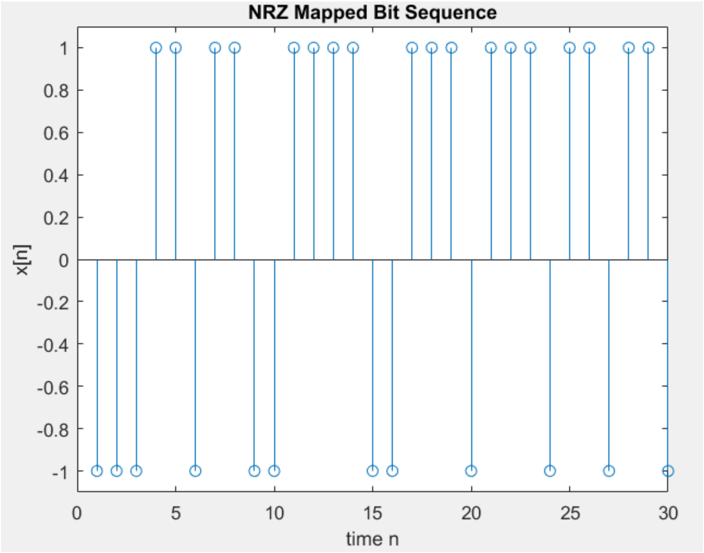

Figure 4.1: Non-return zero mapped input sequence.

And then turned to be a pulse train to make sure every symbol is filtered as shown in Figure 4.2.

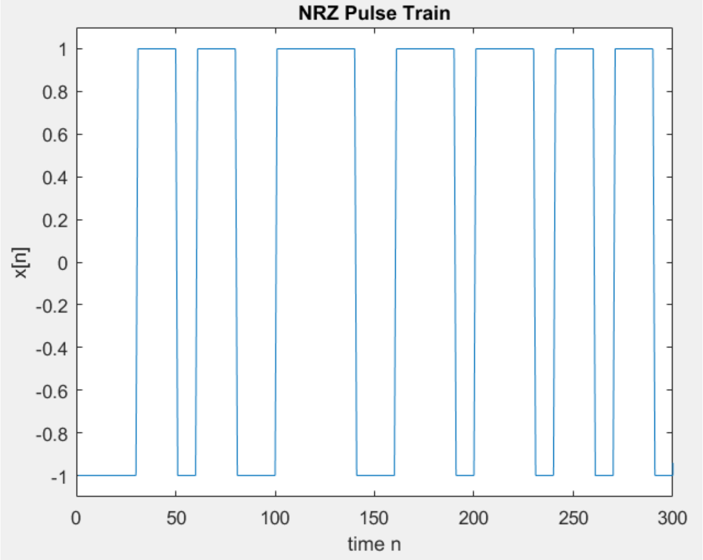

Figure 4.2: NRZ Pulse Train

Now the data is going to be filtered with a low pass Gaussian filter. The impulse responce of the designed Gaussian filter is given in Figure 4.3.

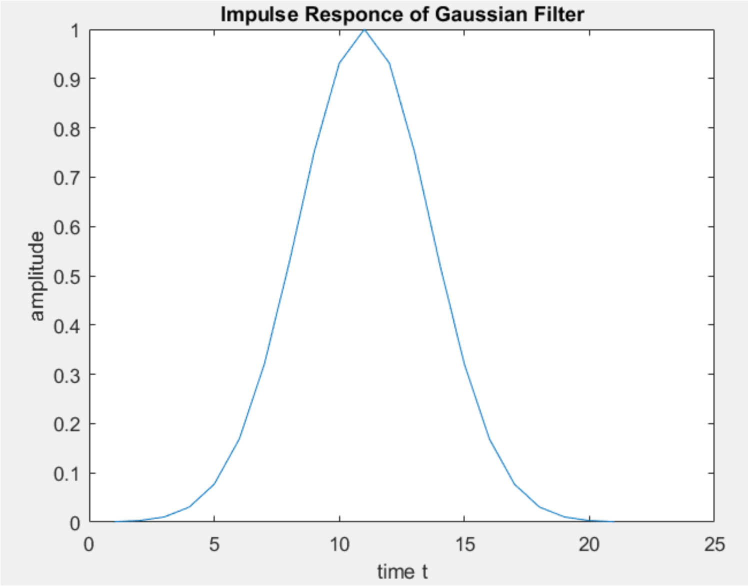

Figure 4.3: Impulse Responce of Gaussian Filter

The filtered NRZ impulse train is shown in Figure 4.4.

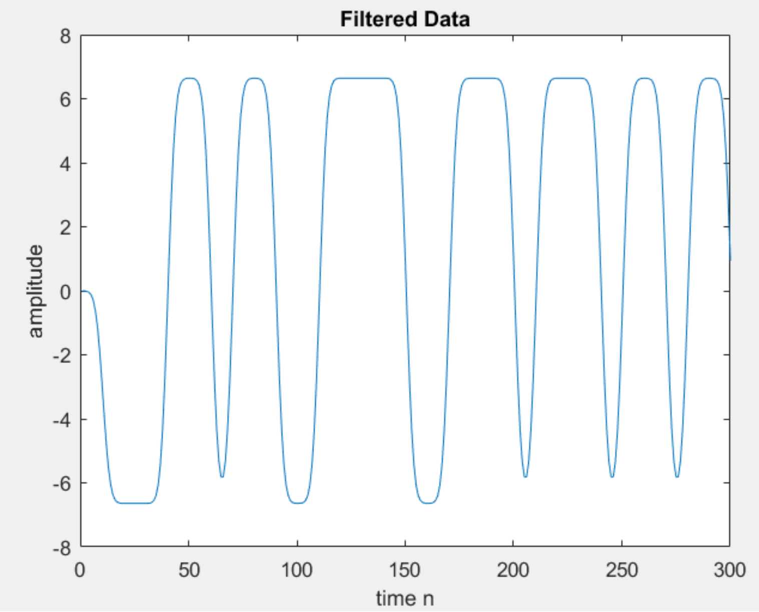

Figure 4.4: Fitlered NRZ impulse train

After that, an integral operation is performed. Integral operation is actually a first degree infinite impulse responce (IIR) filter. The result is shown in Figure 4.5.

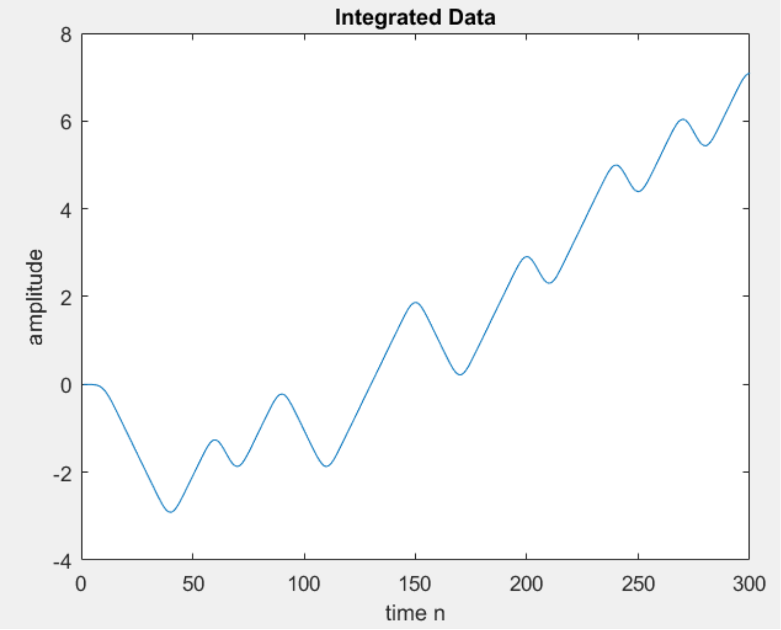

Figure 4.5: Integrator Output

Then, the in-phase and quadrature part of the signal is calculated by multiplying the integrated data by cos($2\pi f_c t$) and -sin($2\pi f_c t$) respectively. The multiplication results are shown in Figure 4.6 and Figure 4.7.

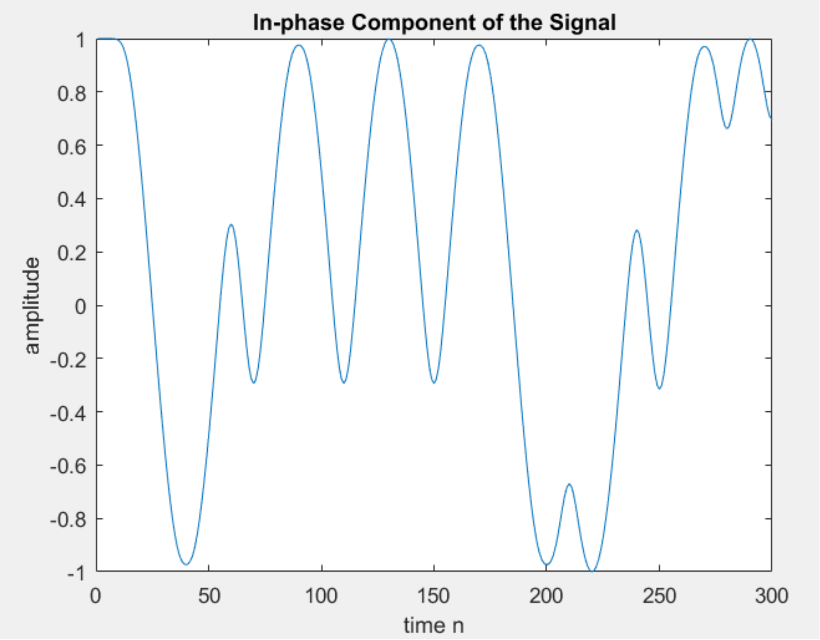

Figure 4.6: In-phase component of the signal

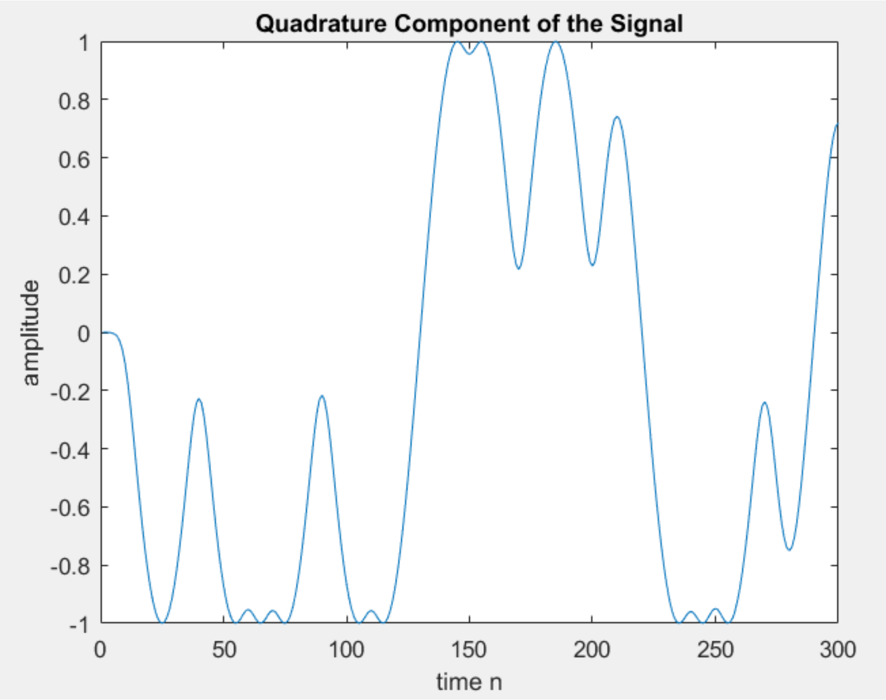

Figure 4.7: Quadrature component of the signal

Finally these, component of the signal is added to each other and GMSK modulated signal is produced. The addition result can be seen in Figure 4.8.

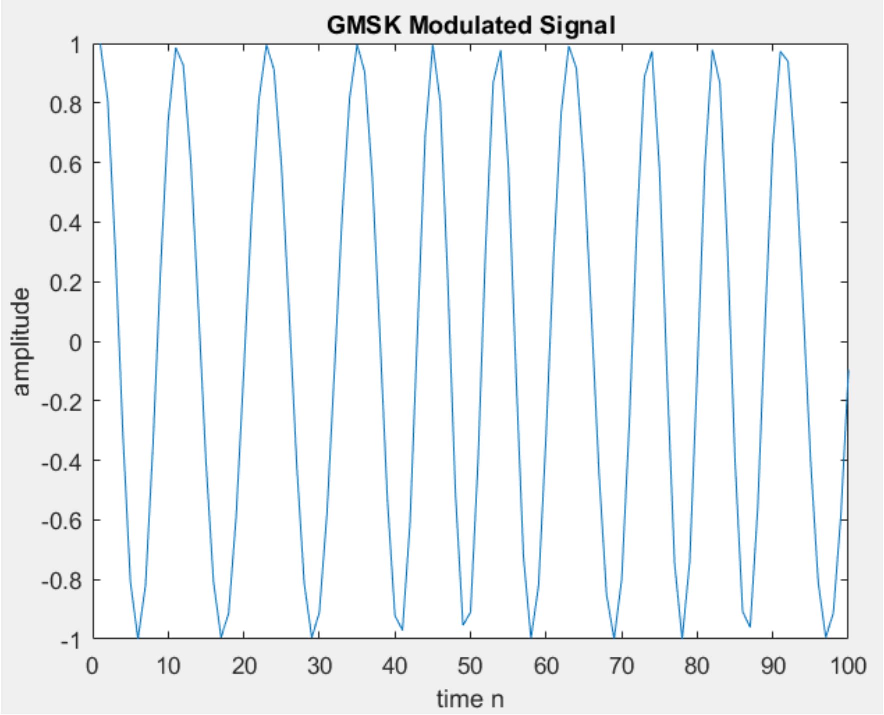

Figure 4.8: GMSK Modulated Signal

The GMSK demodulator takes GMSK modulated signal as input and at first decompose the input signal to its quadrature and in-phase components by multiplying the input by cos(2ωf_ct) and sin(2ωf_ct). The multiplication results are shown in Figure 4.9 and Figure 4.10.

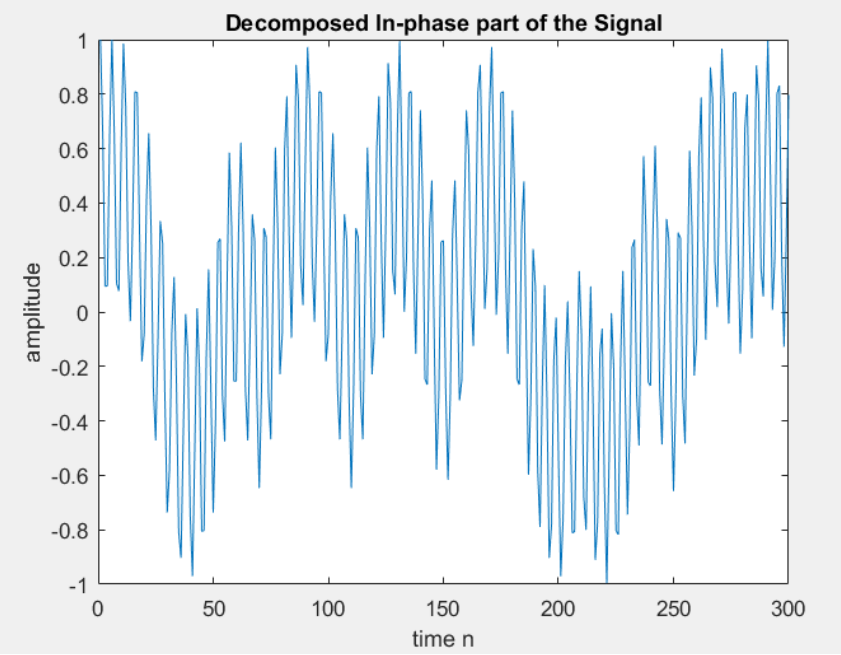

Figure 4.9: Decomposed I component of the signal

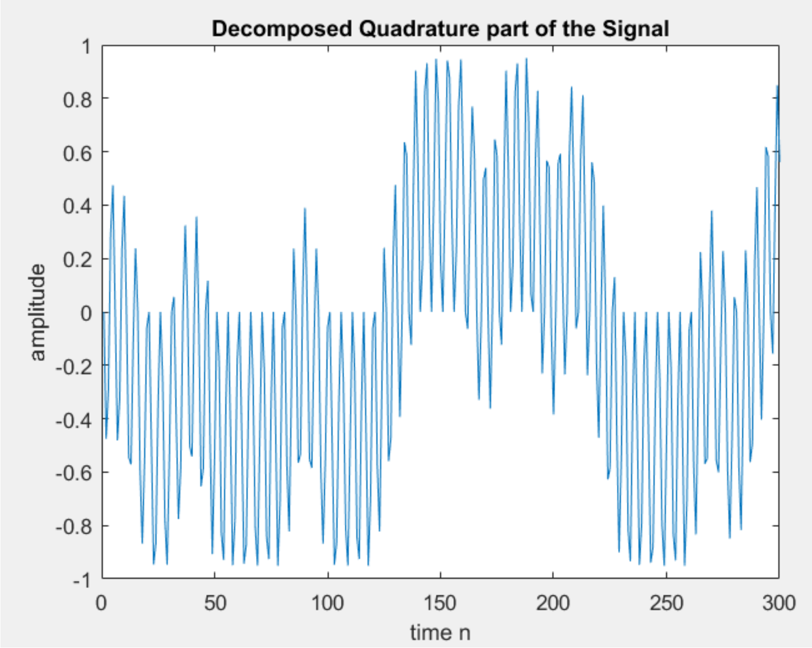

Figure 4.10: Decomposed I component of the signal

A low pass filter is used to suppress high frequency component on the signal and the filtered results are shown in Figure 4.11 and Figure 4.12.

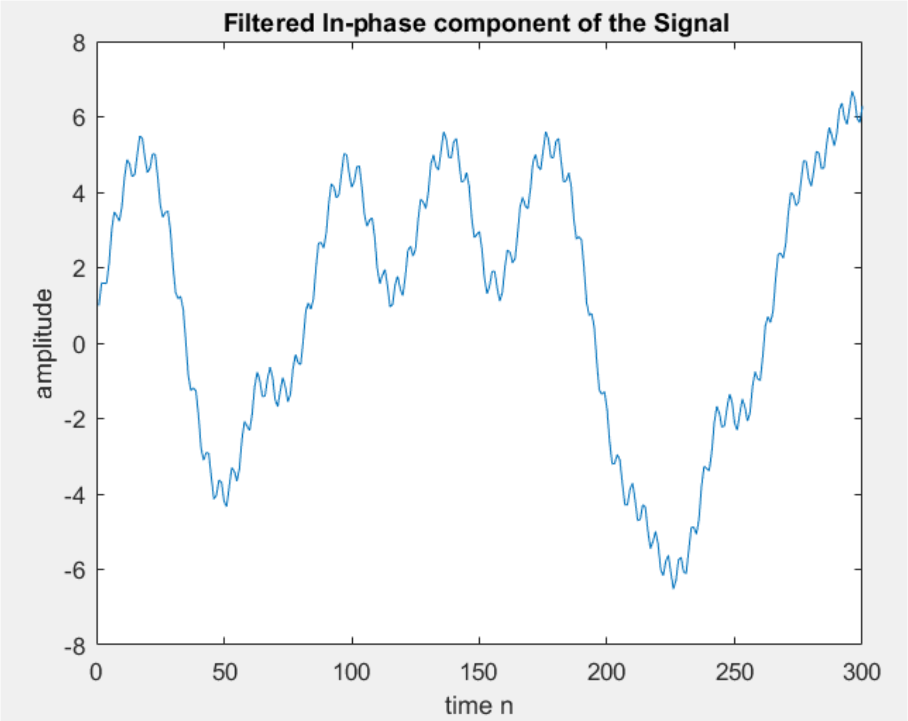

Figure 4.11: Filtered I component of the signal

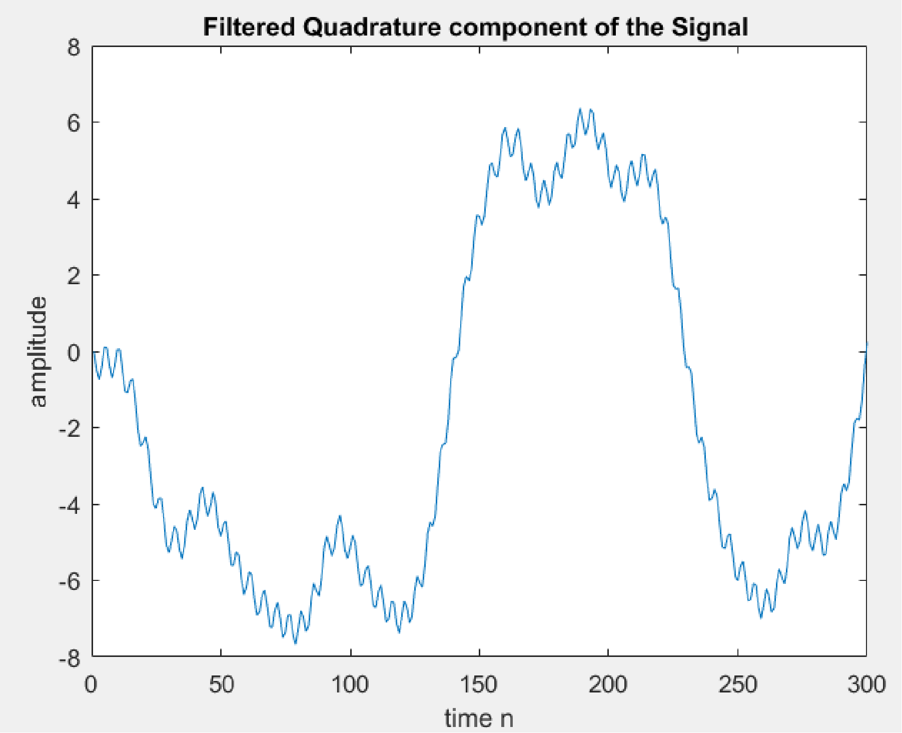

Figure 4.12: Filtered Q component of the signal

After that, the arctangent and derivation operations are done. The result is shown in Figure 4.13.

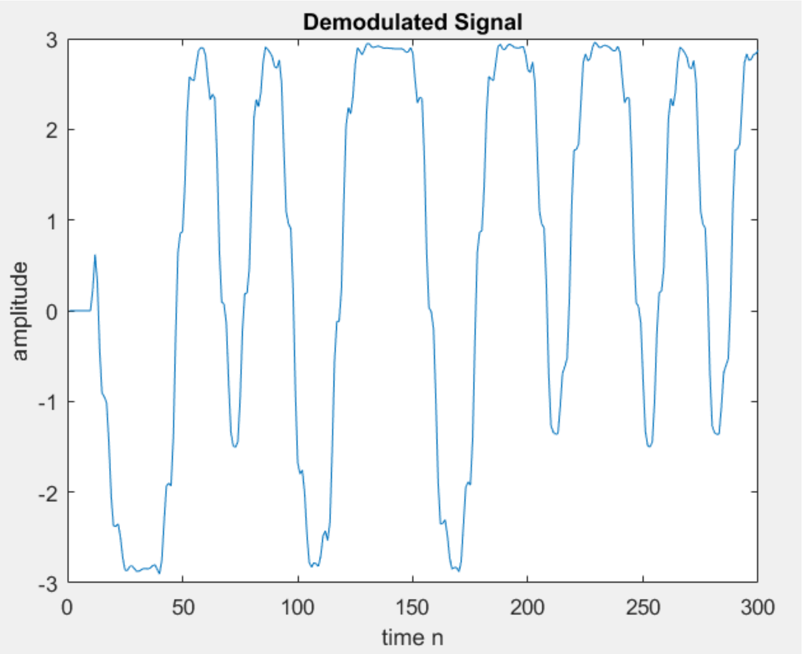

Figure 4.13: Demodulated GMSK Signal

Finally the decison should be made to convert NRZ impulse train to bit sequence. The decision result is given in Figure 4.14.

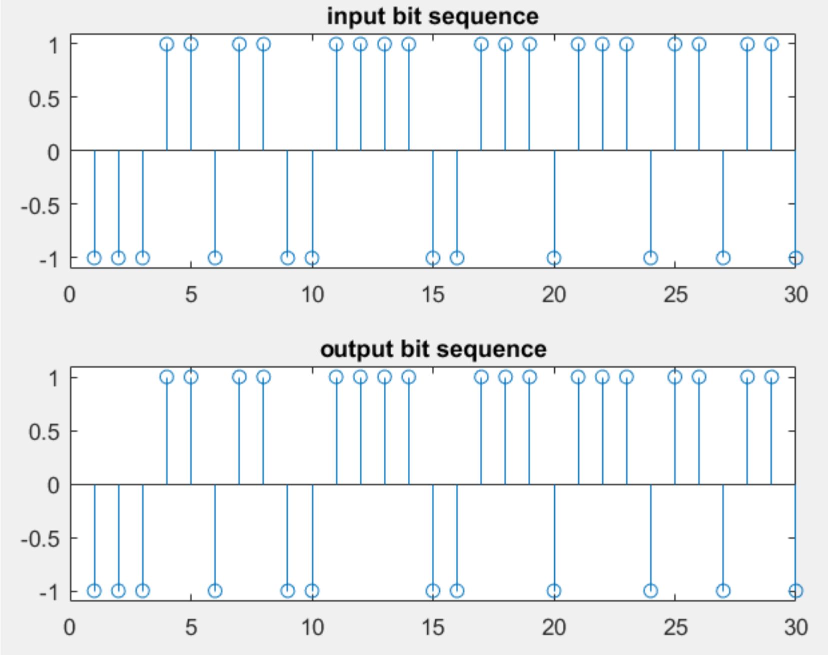

Figure 4.14: Input and output bit sequence

5. IMPLEMENTATION OF GMSK MODULATOR/DEMODULATOR

For implementation, each of the block is implemented shown in Figure 3.9 for modulator and demodulator in Figure 3.10. Altera De0 Nano board, which has Cyclone IV model FPGA on it, is used and it is shown in Figure 5.1.

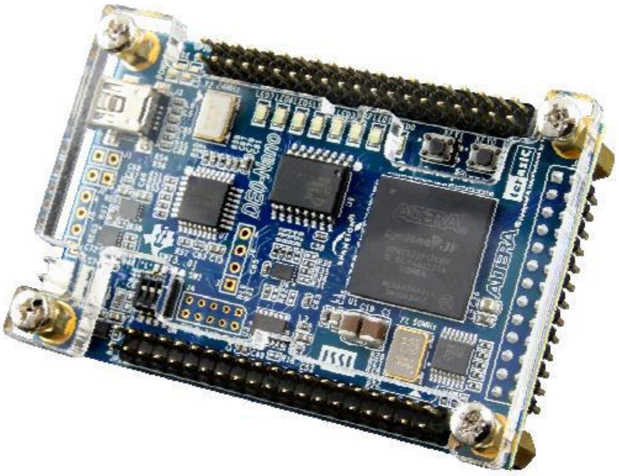

Figure 5.1: Top view of the DE0 Nano Board

To test and for final implementation Cyclone III model FPGA, which has more logic component in it, is used. RF stage of this system is not argued, since it is implemented with a total different architecture and does not related with FPGA. To program FPGA, Quatrus II version 13.1 is preferred and all functions are implemented with very high speed hardware design language (VHDL). As a digital analog converter (DAC), 12 bit DAC902E, and as an analog digital converter (ADC), 8 bit ADS830 is used. Both of them are products of Texas Instrument and both of them are fully parallel hence having IF carrier around 1 to 10 MHz is possible. As IF carrier frequency, 2 MHz is preferred but it can be changed anytime, thanks to the nature of numerically controlled oscillator. The sampling frequency of the system decided as 32 MHz. Even though for 2 MHz IF carrier, 32 MHz sampling frequency seems like so high and not needed, it has some significant advantages. First of all, designing a numerically controlled oscillator with high sampling frequency is easier. Secondly, if the user need a higher IF carrier, he does not have to change the sampling frequency which means that redesigning the system is a lot more easier when compared to deciding the sampling frequency and redesigning every part for the system for the new sampling frequency.

The system uses a Gaussian filter with BT product 0.3. The speed of the input is 250Kbaud and System's bandwidth is 75 KHz. All system defined with 2's compliment signed bit vectors.

5.1 Modulator Implementation

I-Q decomposition method is used to implement the GMSK modulator. All of the results, except the modulator output is taken from Signal Tap II which is a program inside the Quatrus II, shows every signal defined and implemented in the FPGA. The input sequence is also generated in FPGA for testing purpose and it is shown in Figure 5.2.

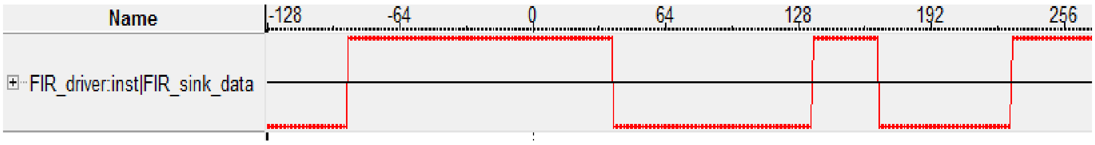

Figure 5.2: NRZ transformed input bit sequence

5.1.1 Gaussian Filter

As stated before, BT product of Gaussian filter is decided as 0.3. The impulse and frequency response of the filter is shown in Figure 5.3 and Figure 5.4 respectively.

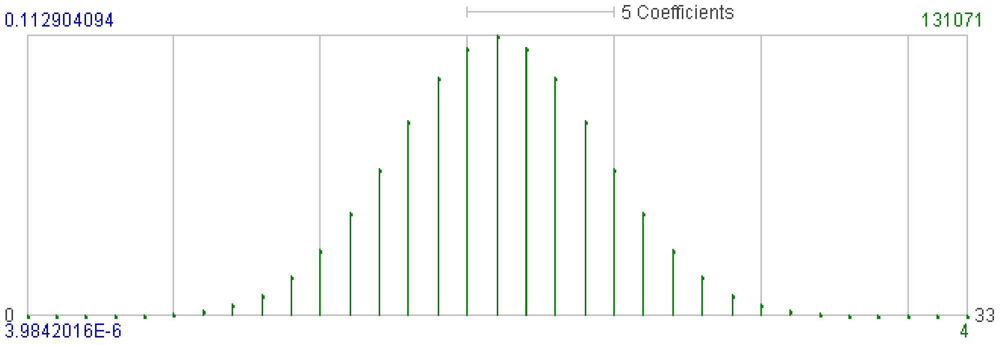

Figure 5.3: Impulse responce of the designed filter

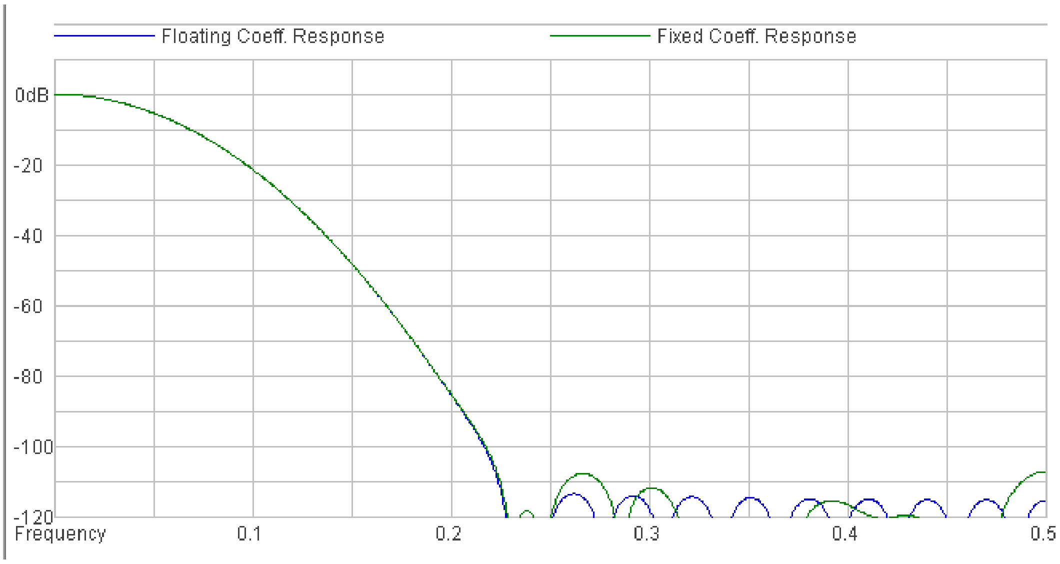

Figure 5.4: Frequency responce of the designed filter

The filter is implemented like a FIR fitler and type I is prefrred when implementing. The type I FIR filter general block diagram is given in Figure 5.5. The input and the output of the Gaussian filter is given in Figure 5.6.

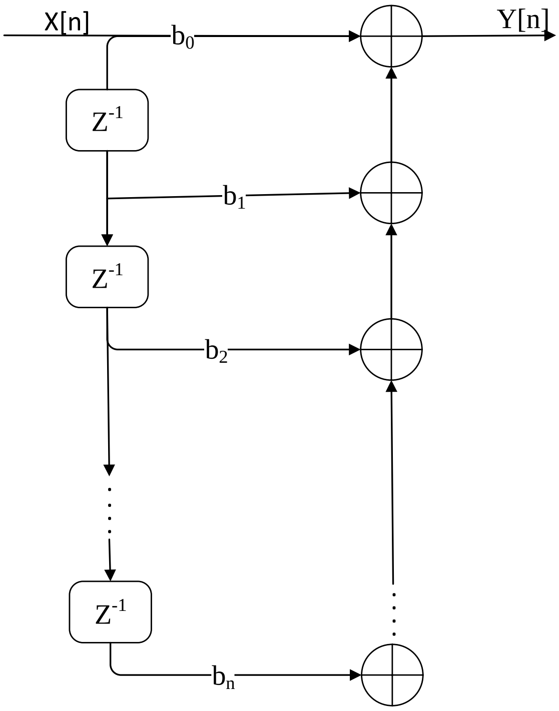

Figure 5.5: Type I FIR filter structure

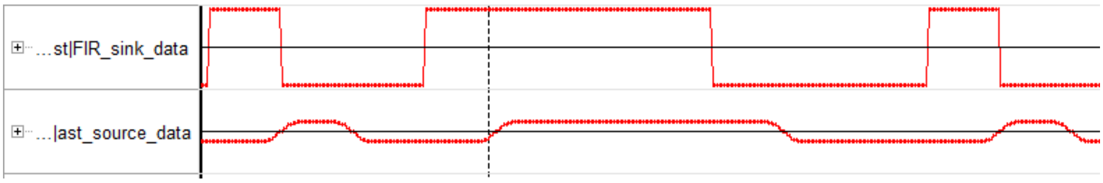

Figure 5.6: Input and output of the Gaussian filter

5.1.2 Integrator

Designing an integrator circuit is easy both as analogue and as digital as well. The integrator is an IIR filter actually and, mathematical representation of it given in (5.1) and block diagram of it given in Figure 5.7 and the input and output of the integrator is given in Figure 5.8.

$$y(n) = y(n-1) + x(n) \tag{5.1}$$

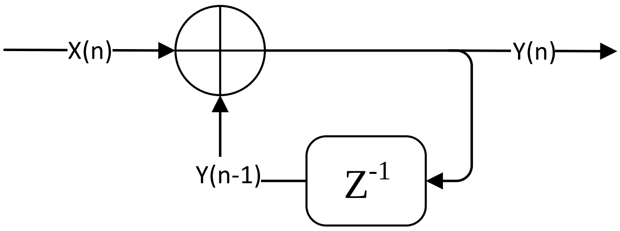

Figure 5.7: Block diagram of Integrator

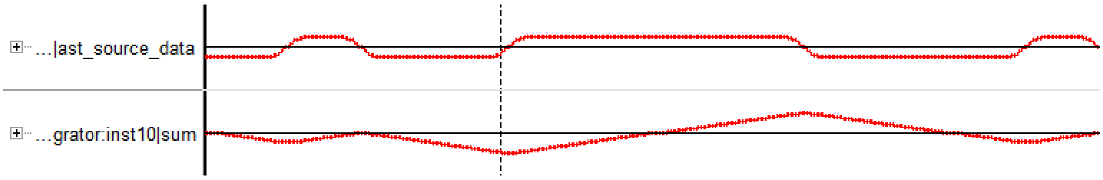

Figure 5.8: Input and output of the integator

5.1.3 Cosine and Sine Functions

There are 2 practical methods to implement cosine and sine functions: preparing look up tables and using coordinate rotation digital computer algorithm (CORDIC). Since CORDIC algorithm is easy to define and more efficient in terms of resource usage, CORDIC algorithm is preferred.

CORDIC algorithm is used to calculate hyperbolic and trigonometric functions in real time with high resolution without any multiplication needed. It is especially useful for microprocessors and FPGAs due to fact that, they do not have any hardware multiplier in them [18]. It is founded by Jack Bolder and developed by John Walther to calculate hyperbolic, trigonometric, square root and vector magnitude calculating functions [19]. The algorithm offers an iterative solution and contains only addition, subtraction and shifting thus, no multiplier is needed. There are 2 main modes of CORDIC algorithm: rotation mode and vectoring mode.

The idea behind this algorithm is, rotating the input vector for many times get close to the result. CORDIC algorithm consists of three inputs x_{in}, y_{in} and z_{in}. x and y for a 2-D vector and z is for the angle input which the vector will be rotated. The expected outputs of the system is given in (5.2) and (5.3) [19].

$$x_R(n) = x_{in}\cos(\theta) - y_{in}\sin(\theta) \tag{5.2}$$

$$y_R(n) = x_{in}\sin(\theta) + y_{in}\cos(\theta) \tag{5.3}$$

The matrix form is given in (5.4)

$$\begin{bmatrix} x_R \\ y_R \end{bmatrix} = ROT(\theta)\begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{5.4}$$

Where ROT(θ) is defined as

$$ROT(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{5.5}$$

To simplify it decompose the angle as shown in (5.6).

$$\theta = \sum_{j=0}^{\infty} \mathrm{\alpha_j} \qquad (5.6)$$

After that, ROT(θ) may be redefined like

$$ROT(\theta) = \prod_{j=0}^{\infty} ROT(\mathrm{\alpha_j}) \qquad (5.7)$$

So that, x_R and y_R also can be redefined like

$$x_R(j+1) = x_R(j)\cos(\mathrm{\alpha_j}) - y_R(j+1)\sin(\mathrm{\alpha_j}) \qquad (5.9)$$

$$y_R(j+1) = x_R(j)\sin(\mathrm{\alpha_j}) + y_R(j+1)\cos(\mathrm{\alpha_j}) \qquad (5.10)$$

Both x_R and y_R can be scaled and rewritten like

$$x_R(j+1) = \cos(\mathrm{\alpha_j})\left(x_R(j) - y_R(j)\tan(\mathrm{\alpha_j})\right) \qquad (5.11)$$

$$y_R(j+1) = \cos(\mathrm{\alpha_j})\left(y_R(j) - y_R(j)\tan(\mathrm{\alpha_j})\right) \qquad (5.12)$$

To make calculation easier $\mathrm{\alpha_j}$ can be chosen like

$$\mathrm{\alpha_j} = \tan^{-1}\left(\mu_j(2^{-j})\right) = \mu_j \tan^{-1}(2^{-j}) \qquad (5.13)$$

Where μ_j is -1 or 1 according to the rotation direction. Finally x_R and y_R can be defined as

$$x(j+1) = x(j) - \mu_j 2^{-j} y(j) \qquad (5.14)$$

$$y(j+1) = y(j) + \mu_j 2^{-j} x(j) \qquad (5.15)$$

To decide the rotation direction z(j) should be calculated as shown in (5.16)

$$\mathrm{z(j+1)} = z(j) - \mu_j \tan^{-1}(2^{-j}) \qquad (5.16)$$

And the rotation direction μ_j can be decided according to z(j) sign as shown in (5.17)

$$\mu_j = \begin{cases} 1 \; if \; z(j) \geq 0 \\ -1 \; if \; z(j) < 0 \end{cases} \qquad (5.17)$$

A total scaling factor K should be added to the formulas and it is defined as

$$\mathrm{K} = \prod_{j=0}^{\infty}\left(1 + 2^{-2j}\right) \cong 1.6468 \qquad (5.18)$$

The final forms of x, y and z vectors are defined as

$$x_{final} = K(x_{in}\cos(\theta) - y_{in}\sin(\theta)) \qquad (5.19)$$

$$y_{final} = K(x_{in}\sin(\theta) + y_{in}\cos(\theta)) \qquad (5.20)$$

$$z_{final} = 0 \tag{5.21}$$

Finally, to compute cosine and sine functions, giving initial condition of x and y, 1/K and 0 respectively would be enough.

As shown in (5.14), (5.15) and (5.16) no multiplier is needed for this algorithm, only adders, substractors and bit shifters are enough. This algorithm is called CORDIC rotation mode. For vectoring mode only a slight modification needed. First, the iteration, ideally, should be continued until y = 0 instead z = 0. Secondly, the rotation direction should be decided according to the y vectors sign instead of z. The final results defined as

$$x_{final} = K\sqrt{(x_{in}^2 + y_{in}^2)} \tag{5.22}$$

$$y_{final} = 0 \tag{5.23}$$

$$z_{final} = z_{in} + \tan^{-1}\left(\frac{y_{in}}{x_{in}}\right) \tag{5.24}$$

Obviously, in physical world, no calculation made for infinite thus, an appropriate iteration number should be selected. For calculating the sine and cosine functions, 12 iteration is preferred.

The input and outputs of this block is given in Figure 5.9.

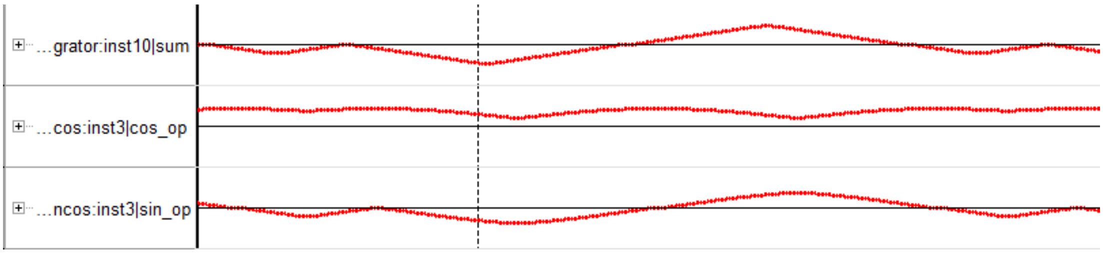

Figure 5.9: Input and outputs of the CORDIC rotation mode block

5.1.4 Numerically Controlled Oscillator

The NCO is used to produce cosine and sine waves. Since sampling frequency is high no filter is used after the NCO. The block diagram of the NCO is given in Figure 5.10.

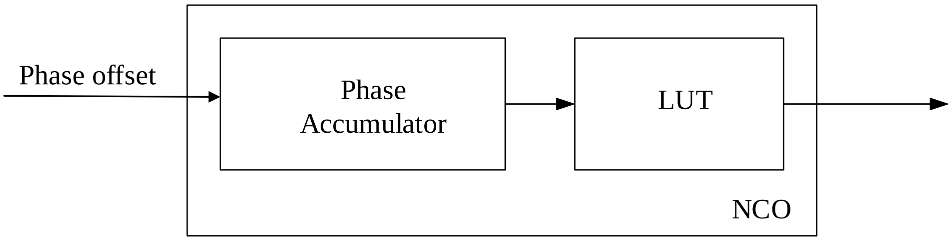

Figure 5.10: Block diagram of the NCO

The phase accumulator is actually an integrator as mentioned in 5.1.2. The look up table (LUT) is contains sine or cosine values to produce waves at desired frequency.

5.1.5 Multiplication

Multipliers are for shift the signal to carrier frequency. The multiplication results are shown in Figure 5.11.

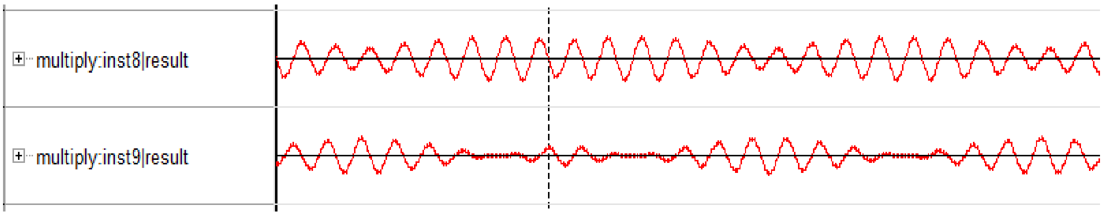

Figure 5.11: Multiplications result

5.1.6 Addition

The final part of the modulator is a simple addition operation to combine I and Q component of the signal. The result of the addition is given in Figure 5.12.

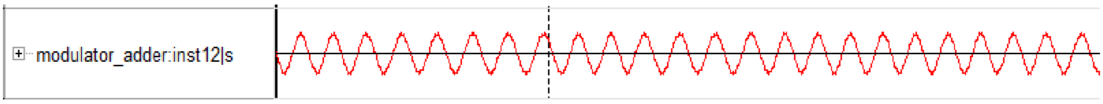

Figure 5.12: The GMSK modulated signal

5.2 Demodulator Implementation

5.2.1 Multiplication

The demodulator starts with multiplicatıon with the carrier signal. Carrier signal is synthesized with an NCO as explained in 5.1.4. The multiplication results is given in Figure 5.13

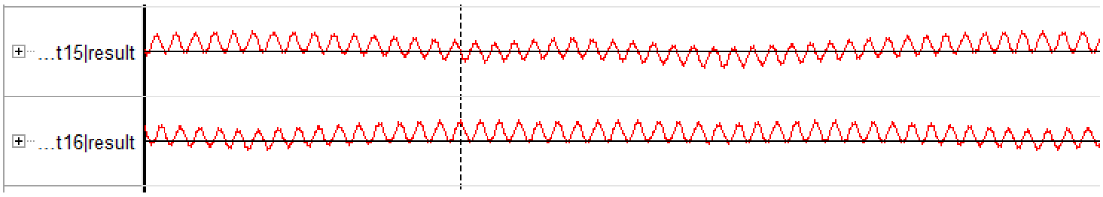

Figure 5.13: Demodulator multiplication results

As expected, the high frequency components are corrupting the signals.

5.2.2 Low Pass Filter

A loop filter, which is a first degree IIR filter, is used as a low pass filter. Its transfer function is,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-0.9375z^{-1}} \quad (5.25)$$

The block diagram of the loop filter is given in Figure 5.14.

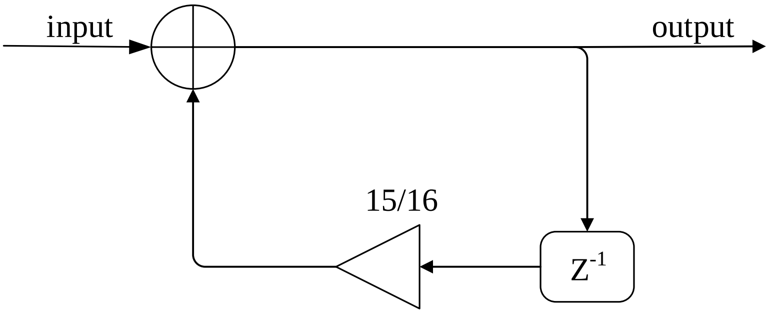

Figure 5.14: Block diagram of the loop filter

Since multiplying is a problem for FPGAs, $\frac{15}{16}$ is implemented as 1-2^{-4} thus only substraction and bit shifting is enough to implement this filter. The outputs of this filter is given in Figure 5.15.

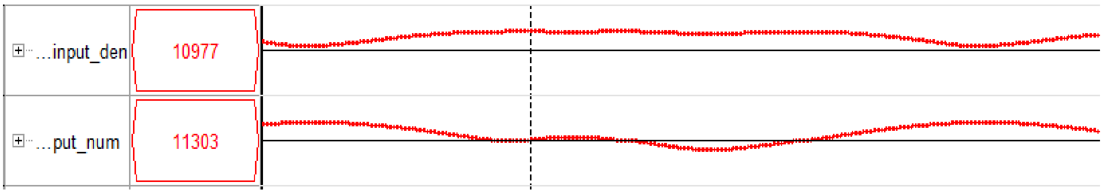

Figure 5.15: Output of the low pass filter

5.2.3 Division and Arctangent function

To implement such an operation, CORDIC algorithm in vectoring mode is the best solution for an FPGA. The details of it is explained in 5.1.3. If $z_{in} = 0$ then, as shown in (5.24), z_{final} would be

$$z_{final} = \tan^{-1}\left(\frac{y_{in}}{x_{in}}\right) \tag{5.26}$$

Which is the function exactly needed for demodulator. The output of this block is given in Figure 5.16.

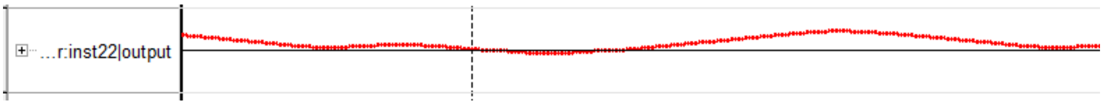

Figure 5.16: Output of the CORDIC block

5.2.4 Derivator

Derivation is the inverse process of integration and the last part of the demodulator which is expected since the first part of the modulator is an integrator. Derivator's equation in n space is defined as

$$y(n) = x(n) - x(n-1) \tag{5.27}$$

Its block diagram is given Figure 5.17.

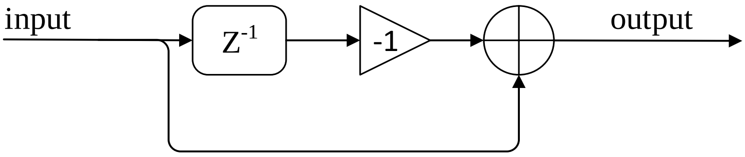

Figure 5.17: Block diagram of derivator

Since the output of the derivator is noisy, a pulse shaper, raised cosine filter is used after derivator. The output of the system is given in Figure 5.18.

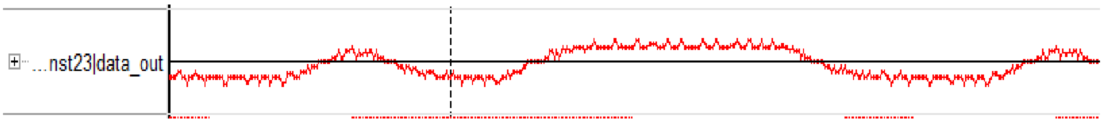

Figure 5.18: Output of the system

Both input bit sequence and the decided bit sequence is shown in Figure 5.19.

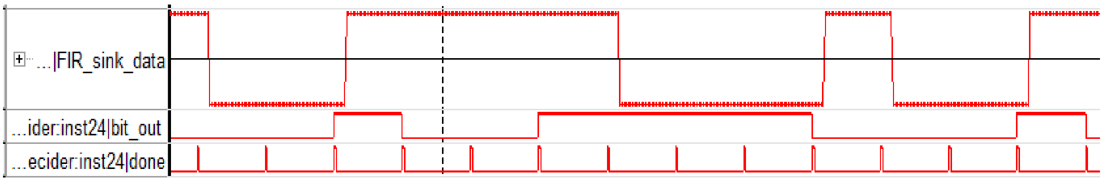

Figure 5.19: The input and the resolved bit sequence

Since the system has serial structure, phase difference occured betwen input and the output bit sequence, which is not a significant problem.

6. SERIAL COMMUNICATION SYSTEM DESIGN

There are different systems on the system like, satellite computer, camera, power system and all of these systems should be either controlled or communicated with our system and it can do that without any serious challenge. The system needs a protocol and a method to communicate with devices or control ICs.

6.1 Protocol

When creating the protocol that proposed system use, the helium 3 modem’s protocol is taken as an example. The command data structure is given in Table 6.1.

Table 6.1: Command data structure

Header	Payload (Data)	Final
6 Byte	0-255 (N) Byte	1 Byte

Where header is given in Table 6.2.

Table 6.2: Header structure

Sync (0x52 & 0x46)	Command	Data Length
2 Byte	2 Byte	2 Byte

The commands and the explanations are given in Table 6.3.

Table 6.3: Command list

Command Code	Command Name	Sended/Received data length
0x1001	No-Op	0
0x2001	No-Op Acknowledge	0
0x1002	Reset	0
0x2002	Reset Acknowledge	0
0x1003	Transmit	N byte
0x2003	Transmit Acknowledge	0
0x1006	Set Transfer Config	1
0x2006	Set Transfer Config Acknowledge	0
0x1010	Beacon Data	Nb byte
0x2020	Beacon Data ACK	0
0xFFFF	Not Acknowledge	0

Where N is stands for the data length that transmitted with the proposed SDR system and Nb is stands for the fixed beacon data length which is decided as 13 byte.

6.2 Methods

System has 3 options as a method to communicate with other internal devices: SPI, UART and I2C. Even though, SPI is an option, it is not used in many cases since SPI has only simplicity advantage and nothing more. UART and I2C on the other hand, have many benefits. A better comparison is shown at Table 6.4 [20].

Table 6.4: Comparison of various communication methods

UART	USB	SPI	I^2C
• Well Known • Cost Effective • Simple	• Fast • Plug&Play • Simple • Low Cost	• Fast • Universally accepted • Low Cost	• Simple • Well Known • Plug&Play • Cost Effective
• Limited functionality • Point to point	• Powerful Master Required • Specific drivers required	• No plug&play • No fixed standard	• Limited Speed

Since SPI is simple and nothing complex to explain, only UART and I2C is argued.

6.2.1 Universal Asynchronous receiver/transmitter

UART is a serial communication protocol that provide an interface for computer to communicate with serial connected devices. The data is transmitted based on the change of DC value. There are different standards for UART like RS-232, RS-422 and RS-485 [21]. In this project, RS-232 standard is preferred. According to the RS-232 standard, -12V is stand for ‘1’ and 12V is stand for ‘0’. Since FPGA’s have output between 0 to 3.3V a conversion is needed. The MAX232 integrated circuit is used for this purpose. There is no clock for this standard thus it is asynchronous. To synchronize devices, baud rate is defined and fixed even before the communication starts. Baud rate is actually the data transmission speed and stands for the bit number that can be send in a second. For example 9600 baud rate means that only 9600 bits can be transmitted through the channel in one second. Most common use of UART is to communicate with the personal computer hence ASCII standard is preferred. In ASCII standard each symbol has 8 bits and 2 bits to start and stop the UART communication thus total 10 bits should be sent. When the channel is not used, it stand at high voltage. When a ‘0’ sent the channel is fall down and this means for the device communication is initiated. After 8 bit is sent the stop bit sent and the communication ends. After the communication ends, the data channel stands at high as expected because it is not used anymore. A new communication might be started with the same procedure. The UART package structure is given in Table 6.5.

Table 6.5: UART package structure

Start Bit	Data bit (MSB)	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit (LSB)

If "10110010" message is desired to send via UART, how the channel would be seem is given in Figure 6.1.

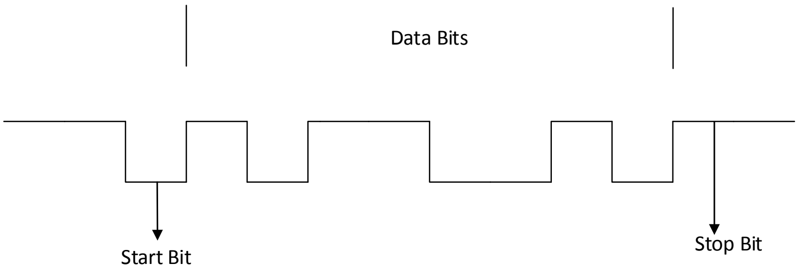

Figure 6.1: An example for UART transmission

6.2.2 Inter-Integrated Circuit

I^{2}C is a method to communicate with IC's by using minimum number of pins [20]. Originally, its top speed was 100kbps. However, to keep up the increasing performance requirements of the new IC's, the top speed increased 400kbps at first and 3.4Mbits after that. There are only 2 lines, which should be pulled up with pull up resistors, needed to communicate with the all slaves since every slave has a unique address. Moreover, the address should be sent to start a communication with the related IC. The system consist of simple master slave structure and master can read or write to slave and interestingly slave can stretch the clock to make sure output is ready. Hence, the main difference between the master and slave is not direct control on the clock but instead starting the communication and source for the clock. Even though, it is not used in proposed system, I^{2}C is actually a multi master method that busses have the ability to collision detection and arbitration to prevent data corruption if more than one master initiate the data transfer [20]. I^{2}C communication procedure have 5 steps. These are:

1. Master should check the bus line if there is any activity.

2. Master should send the address of the IC that want to communicate. When sending the adrees, master put a ‘1’ or ‘0’ in front of the address data to tell either it is going to read from slave or write to the slave. Master also should provide clock on SCL line, data should be valid when SCL line goes low to high.
3. Ask and wait for the acknowledge that shows, IC understands that the master calls for it.
4. Now master can send or read data from slave. After every 8-bit slave should send an acknowledge bit to tell that everthing is going as expected.
5. Finally, after the data is sent or read, stop bit should be sended.

After these steps, the communication is ended successfully and can be restarted if needed.

7. CONCLUSION

In this thesis, SDR basis GMSK modulator and demodulator structure is proposed. At first the simulations are run in MATLAB and then implemented on FPGA. As aimed at first, the proposed system is flexible, minimum resource utilizing, low power consuming and updatable.

All modules are designed by VHDL and nearly all of them is designed as generic in terms of bit length which means that anyone can change a block and do not struggle with the bit length problem. As software to program FPGA, Quatrus II, which is a product of Altera, is used. The modulator design mainly consist of 6 parts. Bit to NRZ pulse signal converter, Gaussian filter, integrator, CORDIC in circular mode to implement sine and cosine functions, quadrature modulator and an adder as a combiner. The demodulator structure consists of 6 parts. Quadrature demodulator, low pass filter, CORDIC in vectoring mode to implement arctangent function, derivator, pulse shaper and finally a decider to decide if the data is ‘1’ or ‘0’.

To improve the systems flexibility, SPI, UART and I^{2}C communication protocols are added to the system as an option to communicate with other internal devices and ICs.

Each of the protocol also written as an independent block and can be removed immediately if the user do not need such options or do not use it.

For future works, an ADC with higher resolution may be preferred because 8 bits for narrow bandwidth signals is not enough. Secondly, even though never mentioned in this project, timing problem may occur in this system. There are many methods but Mueller&Mueller timing algorithm would probably give the best result. Finally, according to the link specifications the threshold at the demodulators decider may be reconfigured to get even a better performance.

REFERENCES

[1] **Androlewicz, J. F.,** 2008. Software-Defined and Cognitive Radio Technology For Military Space Applications.

[2] **Nagarathna, N., Biradar, P. G. and Rao, S. K.,** 2013. BER Analysis of FSK Transceiver for Cognitive Radio Applications.

[3] **Mantis, S.D., Hippenstiel, R. D., Jenn, D. C., and Ha, T. T.,** 2001. Localization of Wireless Communication Emitters Using Time Difference of Arrival (TDOA) Mehods in Noisy Channels, *MSc Thesis,* Naval Postgraduate School, Department of Electrical and Computer Engineering, Monterey, California.

[4] **Kostedt, F., and Kemerling, J. C.,** Practical GMSK Data Transmission, Mixed Signal Ics Application Note, Winston-Salem, USA.

[5] **Ceylan, O., Caglar, A., Cakar, H.O., Kislal A.O., Kula K., and Yagci H.B.,** 2016, Small Satellites Rock A Software-Defined Radio Modem and Ground Station Design for Cube Satellite Communication, *IEEE Microwave Magazine*, **17**, 26-33,

[6] **Hatai, I. and Chakrabarti, I.** 2009. FPGA Implemantation of a Digital FM Modem, *International Conference on Information and Multimedia Technology*, 1-4.

[7] **Bryson, B.,** 2003. A Short History of Radio wih an Inside Focus on Mobile Radio Winter, *Random House*, Toronto, Canada.

[8] **Gelsinger, P.** (2001). Microprocessors for the New Millennium: Challenges, Opportunities and New Frontiers, *IEEE Solid-State Circuits Conference*, 22-25.

[9] **Eyre, J. and Bier, J.** (2000). The Evolution of DSP Processors, *IEEE Signal Processing Magazine*, **17(2)**, 43-51.

[10] **D'Souza, M., Chan, M. and Postula, A.** (2005). Efficient FM Demodulation by Single Tone Detection for FPGA Implementation, *8th International Symposium on DSP and Communications Systems*, **1**, 1-6.

[11] **Kavas, A. K.,** 2006. Design and Realization of GMSK Modem, MSc thesis, Istanbul Technical University, Graduate School of Science Engineering and Technology, Istanbul.

[12] **David, R. S.,** 2004. Digital Transmission Systems, Springer US, New York.

[13] **Murota, K., and Hirade, K.,** 1981, GMSK Modulation for Digital Mobile Radio Telephony, *IEEE Transactions on Communications*, **29,** 1044-1050.

[14] **Visintin, M., Mondin, M., Pent, M., Dovis, F., Falletti, M., and Sellone, F.,** 2002. End-To-End Study of GMSK Modulation, Project Final Report, Torino, Italy

[15] **Švedek, T., Herceg, M., and Matić, T.**, 2009. A simple signal shaper for GMSK/GFSK and MSK modulator based on sigma-delta look-up table, *Radioengineering*, **18**, 231.

[16] **Proakis, J., and Salehi, M.,** 2007 Digital Communications. McGraw-Hill.

[17] **Laurent, P.A.,** 1986. Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP), *Communications, IEEE Transactions on*, **34**, 150-160.

[18] **Andraka, R.,** 1998. A survey of CORDIC algorithms for FPGA based computers, *Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays*, 191-200.

[19] **Sharma, S., Kulakrni, S. and Lakshiminarsimhan, P.,** 2009. Implementation and Application of CORDIC algorithm in Satellite Communication. *15th National Conference on Communication*, Guwahati, India, January.

[20] **Irazabal, J. M., Blozis, S.,** 2003. I^{2}C Manual, San Jose, USA.

[21] **Shibu, K., V.,** 2009, Introduction to Embedded Systems, pp. 51 Tata McGraw Hill Private Limited.

BIOGRAPHY

Ahmet Oğuz Kışlal was born in Sivas in 1993. He graduated from Antalya Anatolian High School in 2011. In 2011, he started the Electronics and Communication Engineering B.Sc. program in Istanbul Technical University. He has started to study at “UHF band SDR for Cube Satellites” project at 2013 as an asistant student. He has one award at International Microwave Symposium (IMS) 2015 in the competion of SDR design. He also has 2 publication which one of them was published at Microwave Magazine at March 2016 and the other published at 2015 TELFOR conferance.

View publication stats