DIGITAL IMPLEMENTATION OF GMSK MODULATOR AND DEMODULATOR BASED FPGA

*Adith T M *Anandu Vinod *M Arathi Krishnan *Meghavika **Padmakumar K ***Dr. Sindhu R

*UG Scholar - ECE, NSSCE Palakkad **Engineer/Scientist G, VLDD, FCG, AVN, VSSC
***Professor – Dept. of ECE, NSSCE Palakkad

N S S COLLEGE OF ENGINEERING
Palakkad, Kerala, India

Abstract

The project focuses on the development of a robust Gaussian Minimum Shift Keying (GMSK) modulation and demodulation system implemented on Field-Programmable Gate Arrays (FPGAs). The primary objective is to enhance communication reliability and efficiency for space and rocket applications where stringent constraints on size, weight, and power consumption exist.

The GMSK modulation technique is chosen for its spectral efficiency and resilience to channel impairments. Leveraging FPGA technology allows for real-time processing, crucial for space missions where low-latency communication is imperative. The modulator and demodulator are designed to adhere to the specific requirements of space environments, ensuring radiation tolerance and fault resilience. Key features of the proposed system include adaptive filtering, error correction coding, and advanced synchronization mechanisms to mitigate the challenges posed by deep space communication. Additionally, the FPGA implementation offers flexibility for adapting to varying communication protocols and frequencies encountered in space missions.

The successful realization of this FPGA-based GMSK modulator and demodulator is anticipated to contribute significantly to the advancement of communication systems in space and rocket applications, improving data transmission reliability and enabling efficient command and control functions.

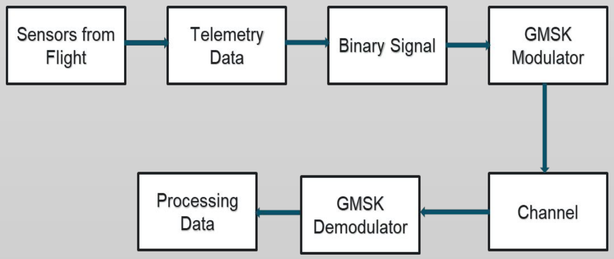
Introduction

- In the rapidly evolving landscape of digital communication systems, Gaussian Minimum Shift Keying (GMSK) modulation has emerged as a cornerstone technique due to its inherent advantages in spectral efficiency and resilience against noise and interference. Meanwhile, Field Programmable Gate Arrays (FPGAs) have become pivotal tools for implementing complex digital systems, offering flexibility and efficiency. This project endeavors to converge these domains by digitally implementing a GMSK modulator and demodulator using FPGA technology.
- The primary objective of this project is to design, simulate, and implement a GMSK modulator and demodulator on an FPGA platform. This entails translating the theoretical principles of GMSK modulation and demodulation into digital logic circuits that can be synthesized and executed efficiently on an FPGA chip.
- The project will involve the development of algorithms for GMSK modulation and demodulation, which will subsequently be translated into hardware description language (HDL) code such as Verilog or VHDL. Through FPGA synthesis, the HDL code will be transformed into a configuration bitstream compatible with the target FPGA device, ensuring optimal resource utilization and timing performance.
- The anticipated outcomes of this project include the realization of a functional GMSK modulator and demodulator meeting standard specifications, along with an in-depth analysis of their performance metrics such as spectral efficiency and bit error rate (BER).
- By documenting the design methodology, implementation specifics, and performance evaluation results, the project aims to contribute to the body of knowledge in digital communications and FPGA-based signal processing.

Methodology

- Determine the parameters for GMSK modulation such as modulation index, symbol rate, and carrier frequency. Define the desired performance metrics, including bit error rate (BER), spectral efficiency, and throughput.
- Develop algorithms for GMSK modulation and demodulation using Python. Use Python libraries such as NumPy for numerical computation and Matplotlib for visualization.
- Extract the filter coefficients for Gaussian pulse shaping from the Python simulation. Translate the Python algorithms and filter coefficients into hardware description language (HDL) code, such as Verilog.
- Design modules for GMSK modulation, demodulation, Gaussian pulse shaping, filtering, carrier recovery, and symbol demapping. Implement the Verilog code targeting the FPGA platform.Utilize FPGA synthesis tools to synthesize the Verilog code and generate a configuration bitstream.
- Optimize the design for resource utilization, timing constraints, and power efficiency. Perform simulation of the synthesized design using FPGA simulation tools or hardware-in-the-loop (HIL) simulation.
- Verify the functionality and performance of the GMSK modulator and demodulator.
- Compare simulation results with Python simulation to ensure consistency. Implement the synthesized design on the target FPGA device.
- Validate the performance of the GMSK modulator and demodulator in a real-world environment.

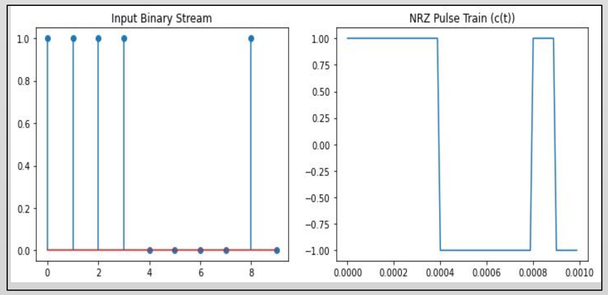
Block Diagram



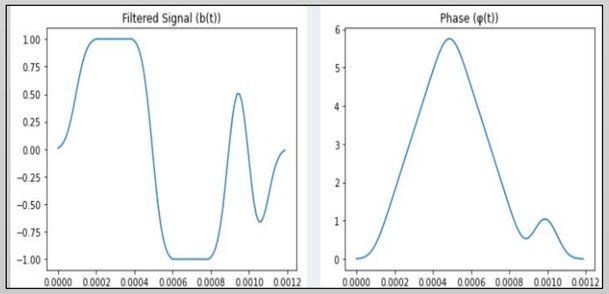
Software Implementation

- Use software tools like Python with libraries such as NumPy and SciPy to design the Gaussian filter with the desired parameters like bandwidth and roll-off factor.
- Encode the input digital data stream into symbols suitable for modulation. Then, perform phase accumulation and frequency modulation based on the encoded symbols to generate the modulated signal.
- Increase the sample rate of the modulated signal by upsampling it and interpolate to refine the waveform and meet spectral characteristics.
- Apply the designed Gaussian filter to the upsampled signal to shape it according to the GMSK modulation scheme and output the filtered signal for further processing or transmission.
- Generate the filter coefficients for the Gaussian filter using Python, and export them to Vivado which is compatible with Verilog.
- Encode the input digital data stream into symbols within Verilog, perform phase accumulation, and frequency modulation based on the encoded symbols to generate the modulated signal.
- Increase the sample rate of the modulated signal within Verilog by upsampling and interpolate to refine the waveform and meet spectral characteristics and output the modulated signal for further processing or transmission within Verilog.

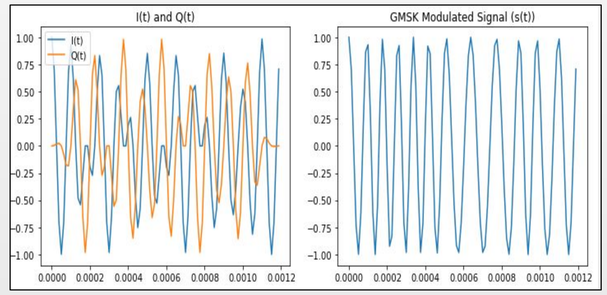
Results



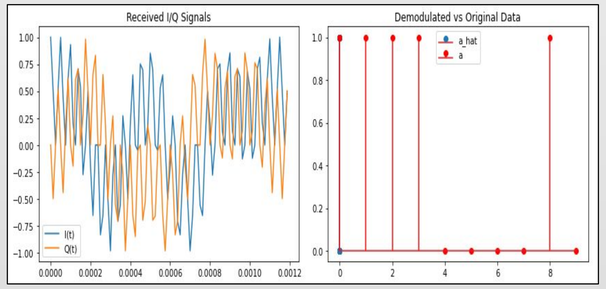
Binary data is converted into NRZ signal.



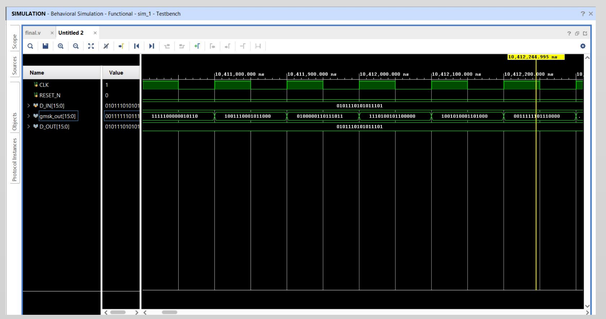
The filtered NRZ impulse train is generated.



The in-phase and quadrature phase components of the signal are added together and GMSK modulated signal is produced.



The GMSK demodulated signal is the same as the original input signal.



Verilog result for GMSK Modulator and Demodulator comparing both input and output.

Conclusions

- The system was first executed in Python to check its feasibility and later on converted to Verilog for hardware implementation.
- The input signal was modulated first and retrieved after demodulation.