Final Report
on
FIR filter design using Hanning Window

Indian Institute of Technology, Guwahati
Department of Electronics and Electrical Engineering

Course: (EE-521) Digital Signal Processors Lab

Guided by:

Dr. Shaik Rafi Ahamed
Professor Dept. of EEE

Submitted by:

Asutosh Sahu
Roll - 234102303
1st year Mtech SPML

E. Rajavelu
Roll - 234102308
1st year Mtech SPML

Contents:

1. Abstract:

This project focuses on the design and implementation of a high-pass digital Finite Impulse Response (FIR) filter using the windowing technique. Specifically, the filter is designed with a sampling frequency of 1000 Hz and a cutoff frequency of 350 Hz, employing a Hanning window of length N = 7. The objective is to attenuate low-frequency components while preserving high-frequency content in a generated composite signal comprising frequencies of 250 Hz and 400 Hz. The designed filter is then utilized to process the composite signal, demonstrating its efficacy in isolating higher-frequency components.

2. Introduction:

Digital signal processing (DSP) plays a crucial role in various fields such as telecommunications, audio processing, and biomedical engineering, facilitating the manipulation and analysis of signals in digital form. In many applications, it becomes necessary to filter out unwanted frequency components while retaining desired ones. Finite Impulse Response (FIR) filters are widely employed for their stability, linear phase response, and ease of implementation. In this project, we focus on designing a high-pass FIR filter to attenuate low-frequency components from a composite signal while preserving higher-frequency content. The design approach involves utilizing the windowing technique, specifically employing a Hanning window, to shape the frequency response of the filter. The chosen parameters include a sampling frequency of 1000 Hz and a cutoff frequency of 350 Hz, tailored to the requirements of the application.

The composite signal, comprising frequencies of 250 Hz and 400 Hz, is generated with a sampling frequency matching that of the designed filter. By processing this signal through the designed FIR filter, we aim to demonstrate its effectiveness in attenuating the 250 Hz component while preserving the 400 Hz component.

Through this project, we illustrate the practical application of FIR filter design techniques, providing insight into the process of designing and implementing digital filters for signal processing tasks. The effectiveness of the designed filter in isolating desired frequency components from a composite signal underscores the importance and relevance of digital filtering in real-world applications.

3. Given Problem Statement :

Design a high-pass digital FIR filter with a sampling frequency of 1000 Hz and a cutoff frequency of 350 Hz, employing a Hanning window of length N = 7. Generate a random composite signal of 250 Hz and 400 Hz, whose data sampling frequency is 1000 Hz. Use this designed filter to process the generated signal.

4. Specifications of the Filter:

1. **Sampling Frequency:** The filter is designed to operate with a sampling frequency of 1000 Hz. This defines the rate at which the digital signal is sampled and processed.
2. **Cutoff Frequency:** The filter is a high-pass filter with a cutoff frequency of 350 Hz. This indicates that frequencies below 350 Hz will be attenuated, while those above 350 Hz will be preserved.
3. **Windowing Technique:** The windowing technique is employed for FIR filter design. Specifically, a Hanning window is used in this case. Windowing is a method used to shape the frequency response of the filter, and the choice of window function affects the filter's characteristics.
4. **Window Length (N):** The length of the Hanning window, denoted as N, is specified to be 7. The window length determines the number of coefficients in the FIR filter and influences its frequency response and performance.

These specifications guide the design process of the FIR filter, ensuring that it meets the desired filtering requirements for the given application. The filter is expected to effectively attenuate low-frequency components while preserving the higher-frequency components present in the input signal. Furthermore, it should demonstrate its capability to process a generated composite signal containing frequencies of 250 Hz and 400 Hz, effectively showcasing its functionality in real-world signal processing scenarios.

5. Literature Survey:

5.1 About FIR Filters:

Finite Impulse Response (FIR) filters are a fundamental component of digital signal processing (DSP) systems. They are characterized by their finite-duration response to an impulse input. Unlike Infinite Impulse Response (IIR) filters, FIR filters are inherently stable and offer linear phase response, making them suitable for applications where phase distortion must be minimized. Their stability simplifies implementation and analysis. Additionally, FIR filters provide precise control over the frequency response, enabling tailored designs for specific applications. However, this control often comes at the expense of higher computational complexity compared to IIR filters, especially for high-order designs. Longer filter lengths may also be necessary to achieve narrow transition bands or sharp cutoffs, which can increase memory and computational requirements. The rational transfer function of a digital filter is as in Equation (1).

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{N-1} a_k z^{-k}}$$

Equation (1)

An FIR filter is a special case of Equation (1), where $a_0 = 1$ and $a_k = 0$ for $k = 1,\ldots,N-1$, hence we obtain:

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

Equation (2)

5.2 Key features of FIR Filters:

Finite Impulse Response (FIR) filters possess several key features, with linear phase being one of the most notable:

1. **Stability:** FIR filters are inherently stable, making them suitable for real-time applications where stability is crucial. Unlike Infinite Impulse Response (IIR) filters, which can exhibit instability under certain conditions, FIR filters do not rely on feedback and thus avoid stability issues.

2. **Flexibility:** FIR filters offer flexibility in design, allowing for precise control over the filter's frequency response characteristics. Design parameters such as filter length, cutoff frequency, and windowing functions can be adjusted to meet specific requirements, making FIR filters adaptable to a wide range of applications.

3. **Linear Phase Group Delay:** The linear phase response of FIR filters ensures constant group delay across the entire frequency spectrum. Group delay

represents the time delay experienced by different frequency components of the input signal. Linear phase group delay helps preserve the shape of input waveforms, which is crucial in applications where maintaining waveform integrity is essential, such as in digital audio processing and telecommunications.

4. **Symmetry:** FIR filters often exhibit symmetry in their impulse response coefficients, which simplifies their implementation and analysis. Symmetric filter coefficients result in a symmetric frequency response, making FIR filters easier to design and interpret compared to asymmetric filters.

5. **Arbitrary Frequency Response:** FIR filters allow for the design of arbitrary frequency response characteristics, enabling precise shaping of the frequency domain. This capability is advantageous in applications where specific frequency components need to be attenuated or preserved while maintaining linear phase.

6. **Linear Phase Response:** FIR filters exhibit linear phase response, meaning that all frequency components of the input signal experience the same delay, regardless of frequency. This ensures that the filter does not introduce phase distortion, preserving the relative timing of different frequency components. Linear phase is particularly important in applications where phase coherence is critical, such as in audio and communication systems, where maintaining the fidelity of the signal is essential.

5.3 Linear Phase Filters:

This linear phase filter description can be generalized into a formalism for four type of FIR filters:

Type 1: symmetric sequence of odd length
Type 2: symmetric sequence of even length
Type 3: anti-symmetric sequence of odd length
Type 4: anti-symmetric sequence of even length

There are four possible scenarios: filter length even or odd, and impulse response is either symmetric or antisymmetric as shown in the figure below.

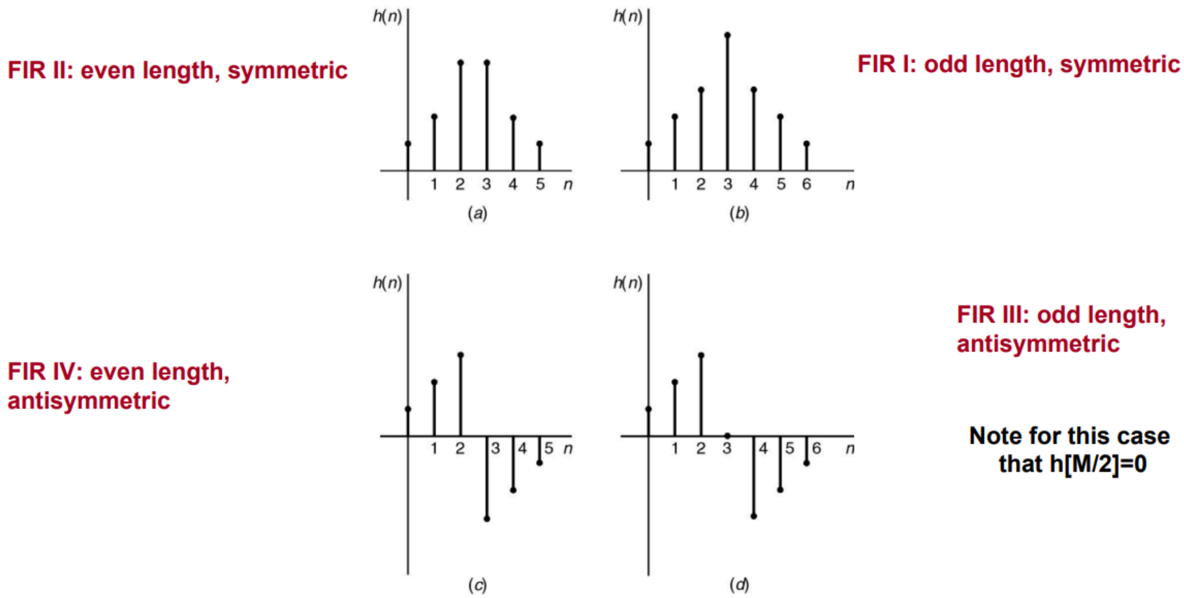

Fig. (a). Type II (b). Type I (c). Type IV (d). Type III

Overall, the key features of FIR filters, particularly their linear phase response, stability, flexibility, and symmetry, make them well-suited for various signal processing tasks, including audio equalization, noise reduction, and digital communication systems. Their ability to maintain phase coherence and preserve waveform integrity makes them indispensable in applications where signal fidelity is paramount.

5.4 FIR Filter Design Techniques:

In FIR filter design, various techniques are employed to meet specific design requirements. FIR filter design encompasses various techniques, each with unique advantages:

1. Windowing Method: Simple and flexible, it uses window functions to shape the filter response. However, it may suffer from spectral leakage.

2. Frequency Sampling Method: Guarantees exact response at specified frequencies but may require longer filters for precise control.

3. Least Squares Method: Minimizes error between desired and actual responses, offering flexibility but with potentially high computational complexity.

4. Parks-McClellan Algorithm: Efficiently approximates desired responses with optimal passband ripple, stopband attenuation, and transition bandwidth, yet requires careful understanding and may be computationally demanding.

5. Optimal Chebyshev Approximation: Balances passband ripple and stopband attenuation for sharper transition bands, but necessitates careful parameter selection.

One such technique is the windowing method, where a window function is applied to shape the frequency response of the filter. Window functions such as Hanning, Hamming, and Kaiser windows are commonly used. The windowing technique allows for control over the sidelobe levels, which improves the filter's stopband attenuation. It simplifies the design process by providing a straightforward approach to shaping the frequency response. However, there is often a trade-off between the mainlobe width and sidelobe levels, and certain window functions may introduce spectral leakage, causing energy to spread into adjacent frequency bins.

In FIR filter design, parameters such as window length, cutoff frequency, and sampling frequency play crucial roles. The window length (N) determines the number of samples in the window function, directly influencing the number of coefficients in the FIR filter and its frequency response characteristics. The cutoff frequency defines the frequency beyond which the filter attenuates the input signal. For high-pass filters, frequencies below the cutoff frequency are attenuated, while higher frequencies pass through unaffected. The sampling frequency determines the rate at which the continuous-time signal is sampled to obtain discrete-time samples, affecting the frequency range that can be accurately represented in the digital signal. High-pass filters are commonly used to remove low-frequency noise or baseline drift from signals, emphasizing the importance of accurate parameter selection in FIR filter design to achieve desired filtering effects.

5.5 Hanning Window Function:

The Hanning window is a specific type of window function frequently used in FIR filter design. It is characterized by its raised cosine shape, which smoothly tapers to zero at the edges. The Hanning window offers moderate sidelobe attenuation and facilitates smooth transitions in the frequency response, reducing spectral leakage. However, compared to other window functions like Kaiser or Blackman windows, the Hanning window provides less aggressive sidelobe suppression. The choice of window function depends on the specific requirements of the application, balancing factors such as sidelobe levels, transition width, and stopband attenuation. The Hann window is used to lessen bad effects on frequency characteristics produced by the final samples of a signal being filtered. Digital filters designed with this window have higher stopband attenuation than those designed with triangle function. Another advantage of this window is the ability to relatively fast increase the stopband attenuation of the following lobes.

The Hann window coefficients can be expressed as:

$$w[n] = \frac{1}{2}\left[1 - \cos(\frac{2\pi n}{N-1})\right]; 0 \le n \le N-1$$

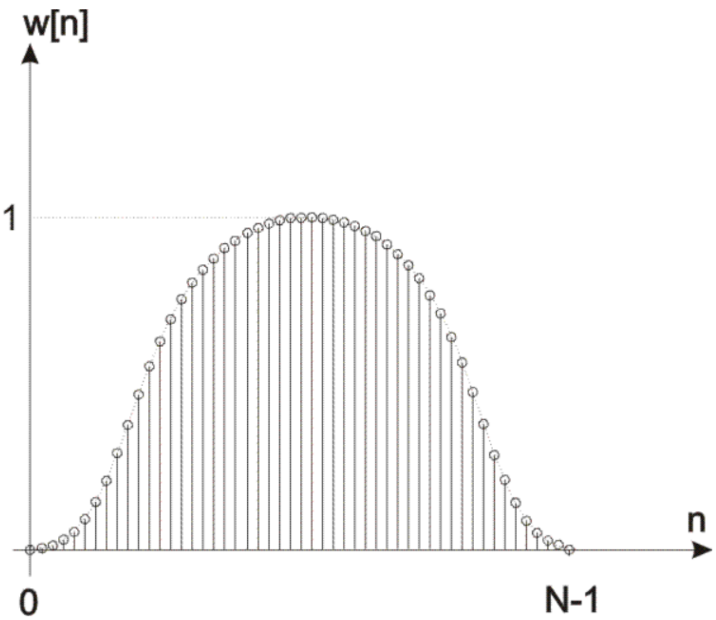

Fig. The Hann window coefficients in the time domain

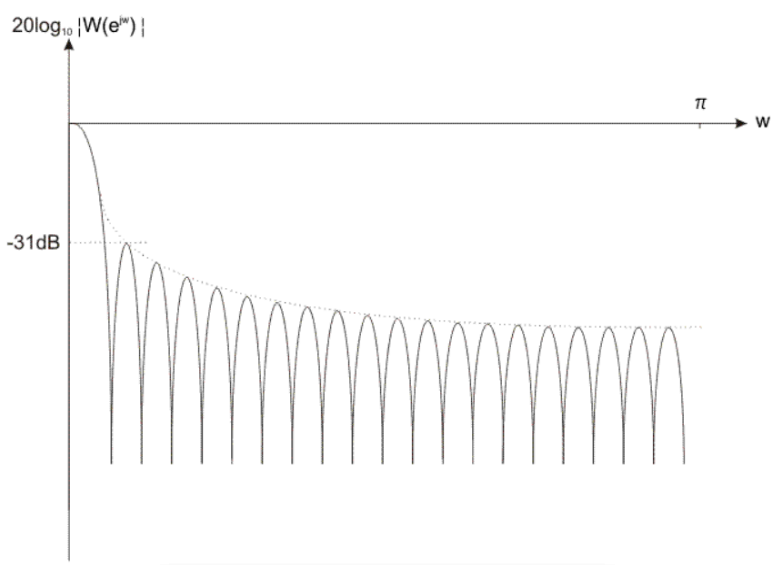

Fig. The Hann window coefficients in the time domain

Hanning window has a sharper fall than the triangular one, which is considered as its advantage. For the same requirements for minimum attenuation, the Hann window will have a narrower transition region.

5.6 Window Method for FIR Filter Design:

The Window In this method from the desired frequency response specification $H_d(w)$,corresponding unit sample response $h_d(n)$ is determined using the following relation

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(w)\, e^{jwn} dw$$

where

$$H_d(w)=\sum_{n=-\infty}^{\infty} h_d(n)\, e^{-jwn}$$

In general,unit sample response $h_d(n)$ obtained from above relation is infinite induration, so it must be truncated at some point say n = M-1 to yield an FIR filter of length M (i.e. 0 to M-1). This Truncation $H_d(n)$ to length-1 is same as multiplying $h_d(n)$ by the rectangular window defined as

$$w(n) = \begin{cases} 1 & 0 \le n \le M\text{-}1 \\ 0 & \text{otherwise} \end{cases}$$

Thus the unit sample response of the FIR filter becomes

$$\begin{aligned} h(n) &= h_d(n)\, w(n) \\ &= h_d(n) \qquad 0 \le n \le M\text{-}1 \\ &= 0 \qquad \text{otherwise} \end{aligned}$$

6. Methodology:

6.1 Python Code Description:

Filter Design: A high-pass FIR filter was designed using the window method. The desired cutoff frequency was set to 350 Hz, with a sampling frequency (fsfs) of 1000 Hz. The filter size (NN) was chosen as 7 taps. The filter coefficients were calculated using the following formula:

$h[n]$={(0.5−0.5cos*(2πnN−1)) * (−sin*((n−α) * ωc)/ (n−α) *π)} for $n \neq \alpha$

$h[n]$= {(0.5−0.5cos*(2πnN−1)) * (1−$\omega c \pi$)} for n=α

where α=N−1/2, ωc=(2*π*fc)/fs and $h[n]$represents the filter coefficients.

Frequency Response Analysis: The frequency response of the designed filter was analyzed using the Discrete Time Fourier Transform (DTFT). By computing the DTFT for the filter coefficients, the magnitude and phase spectra were obtained. This analysis provided valuable insights into the filter's frequency behavior, including passband attenuation and stopband characteristics.

Signal Generation: The Python code generated a composite signal by adding two sinusoidal signals with frequencies of 250 Hz and 400 Hz. The sampling frequency was set to 1000 Hz, ensuring accurate signal representation. The duration of the signal was defined as 1 second, allowing for comprehensive analysis of signal properties.

Filtering Process: The generated composite signal underwent filtering using the designed high-pass FIR filter. Convolution operation was employed to apply the filter kernel to the input signal, resulting in the extraction of high-frequency components. This process effectively attenuated low-frequency components, achieving the desired high-pass filtering behavior.

Frequency Analysis of Filtered Signal: The Fast Fourier Transform (FFT) algorithm was utilized to compute the frequency spectrum of both the original composite signal and the filtered output. This analysis enabled comparison of frequency components before and after filtering, highlighting the effectiveness of the FIR filter in isolating high-frequency content from the input signal.

Results Presentation: The Python code facilitated the visualization of results through time-domain and frequency-domain representations of both the original composite signal and the filtered output. Time-domain plots depicted signal amplitudes over time, offering insights into signal dynamics. Frequency-domain plots showcased the magnitude spectra, illustrating the impact of filtering on frequency content.

In addition to the main FIR filter design process, an algorithm was implemented to convert type 2 filters into type 4 filters, enabling their use for high-pass FIR (HPF) filter design. This algorithm addressed the challenge posed by symmetric coefficients in type 2 filters, which are unsuitable for high-pass filtering applications due to their inherent symmetry.

The algorithm focused on transforming symmetric coefficients into anti-symmetric ones, thereby aligning them with the requirements of type 4 filters. By making appropriate adjustments to the coefficients, the resulting filter characteristics matched those obtained from popular design tools like MATLAB's filterDesigner.

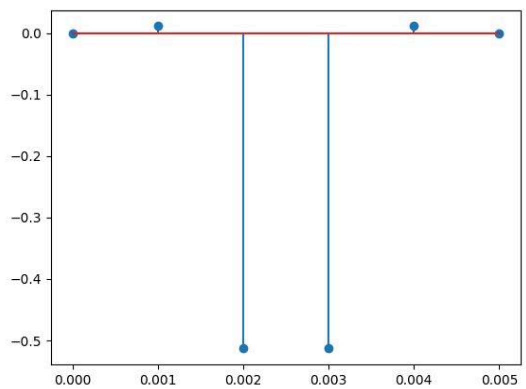

Fig. Filter Coefficients for N=6 before applying algo

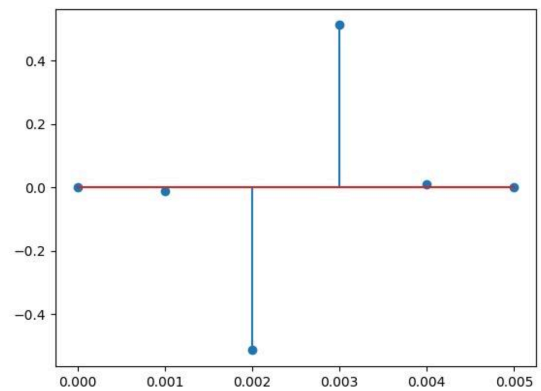

Fig. Filter Coefficients for N=6 after applying algo

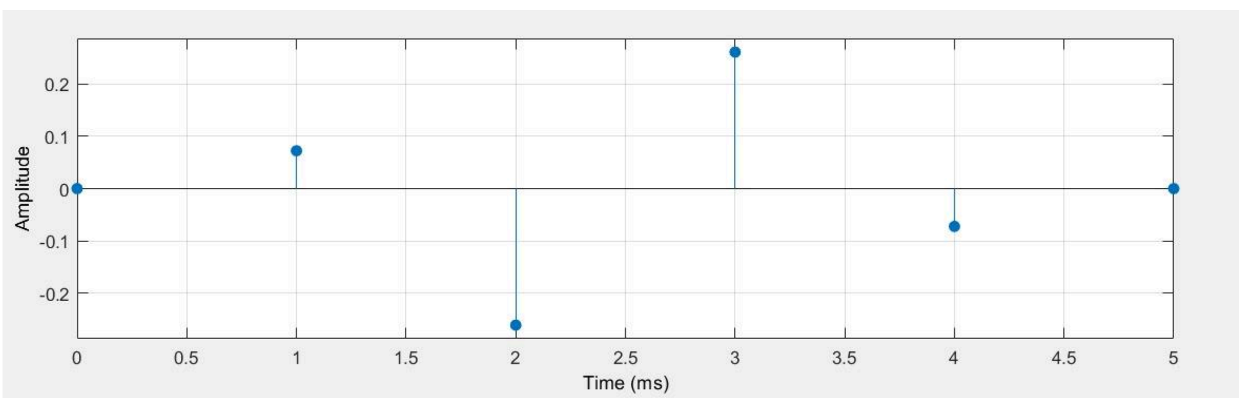

Fig. Filter Coefficients for N=6 from Matlab filterDesigner

6.2 Verilog Code Description:

We are going for **Direct Form Realization** of designed FIR filter. The Verilog module named HPFIR_7 is designed to implement a 7-tap FIR high pass filter. The filter is intended to operate with a cutoff frequency of 350 Hz in a system with a sampling rate of 1000 Hz.

The module has three input and one output ports:

- **clk:** Clock signal input.
- **noisy_signal:** Input signal to be filtered, represented as a signed 16-bit value.
- **filtered_signal:** Output signal after filtering, represented as a signed 16-bit value.

The filter coefficients are defined as a set of signed 16-bit values in the coeff array. These coefficients are pre-calculated based on the desired filter characteristics, specifically for a Hanning window approach with a cutoff frequency of approximately 350 Hz. The delayed_signal array is used to store the input signal delayed by one clock cycle. This array is implemented as a shift register to hold the input signal at different time instances.

The prod array is used to store the product of the delayed input signal and the corresponding filter coefficient. Each element of the prod array represents the result of multiplying the delayed input signal by the respective coefficient. The sum_0, sum_1, and sum_2 arrays are used to accumulate the products of the delayed input signal and filter coefficients. These accumulations are performed in a pipelined manner to optimize the filter's throughput. The filtered_signal output is computed by summing the accumulated products and truncating the result to a 16-bit signed value.

The Verilog code utilizes sequential always @(posedge clk) blocks to synchronize the operations with the clock signal. Within these blocks, the delayed signal, product computations, and accumulation operations are performed.

The testbench, named HPFIR_7_tb, serves as a comprehensive platform for verifying the functionality and performance of the designed FIR filter. It employs a CORDIC-based phase accumulator to synthesize two sinusoidal signals at frequencies of 250Hz and 400Hz, respectively. These synthesized signals, represented by sin_250HZ and sin_400HZ, serve as inputs to the FIR filter.

The Cordic module utilizes a clock frequency defined by CORDIC_CLK_PERIOD to generate the desired sinusoidal signals. The phase sweep logic ensures continuous phase modulation to produce the desired frequencies accurately. The FIR filter instantiation, denoted by FIR_filter_inst, incorporates the designed FIR HP filter module (HPFIR_7). It operates with a clock frequency defined by FIR_CLK_PERIOD to process the incoming noisy signal composed of the synthesized sinusoids. Within the testbench, the noisy_signal is computed as the average of the two synthesized sinusoidal signals, simulating a real-world noisy input signal. This noisy signal is then fed into the FIR filter for processing.

The testbench utilizes two clock generation blocks to produce the CORDIC clock (cordic_clk) and the FIR clock (fir_clk) with frequencies corresponding to the desired sampling rates. These clocks ensure proper synchronization of signal generation and filtering operations.

Overall, the Verilog code efficiently implements a 7-tap FIR high pass filter using the Hanning window technique. It incorporates pipelined multiplication and accumulation to maximize throughput while maintaining accuracy in filtering the input signal. This code can be synthesized and implemented on FPGA or ASIC platforms to realize a hardware-based FIR filter with the specified characteristics.

7. Results and Discussion:

The figure below illustrates the time domain representation of two sinusoidal signals: one at 250 Hz and the other at 400 Hz. These sinusoids are then combined, resulting in a noisy signal. Finally, the plot demonstrates the filtered output obtained through our designed Finite Impulse Response (FIR) High Pass Filter (HPF).

This representation allows for a visual understanding of the original sinusoidal components, their summation to form a noisy signal, and the effectiveness of the FIR HPF in attenuating lower frequency components, thereby enhancing the clarity of the desired signal.

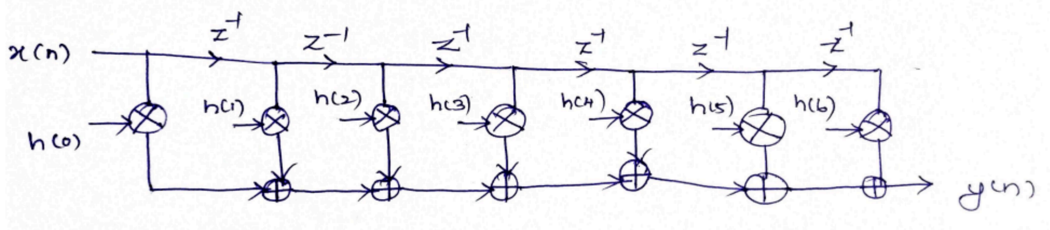

Fig. Direct Form Realization of the Designed FIR filter

Comparison of Results of Designed Filter and that using Inbuilt function:

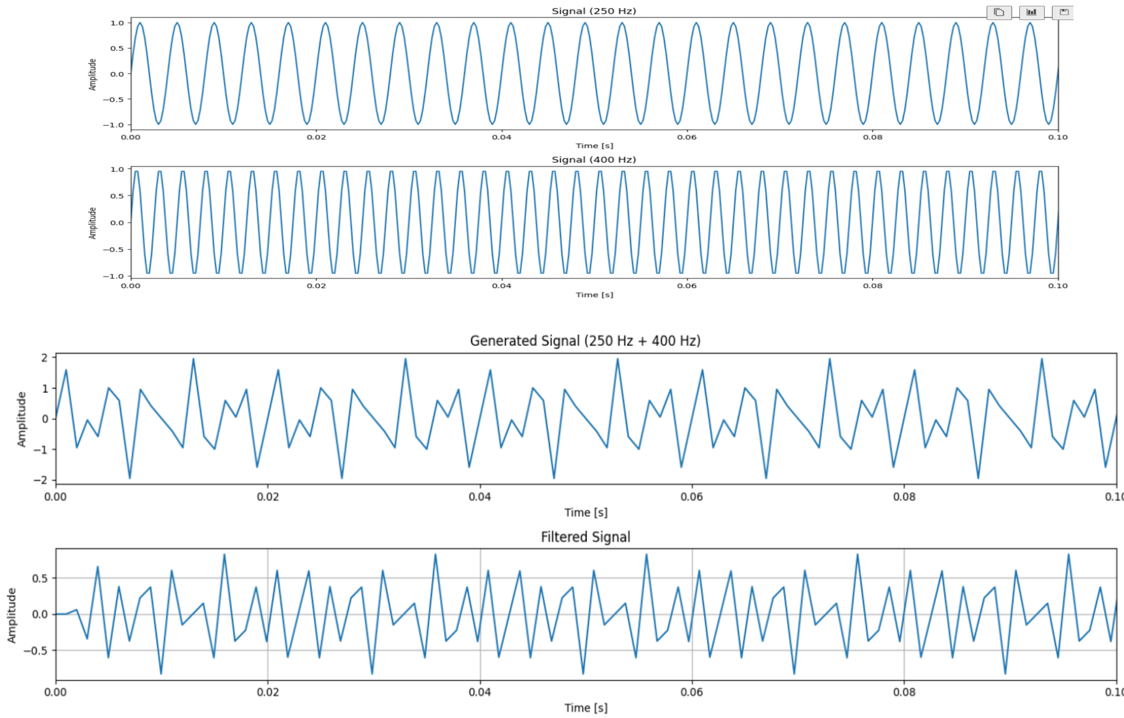

Fig. Time domain representation of 250 Hz sinusoid, 400 Hz sinusoid, Composite Signal and Filtered Signal

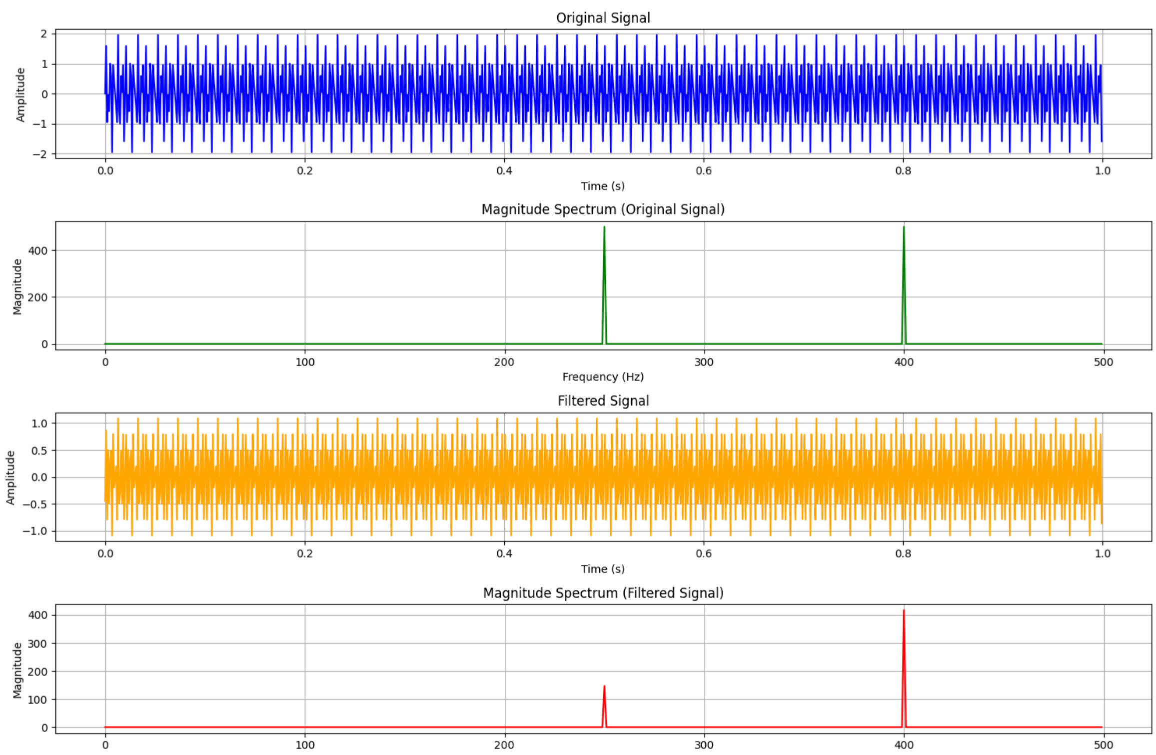

Fig. Time domain and Freq domain representation of Composite Signal and Filtered Signal using Inbuilt function

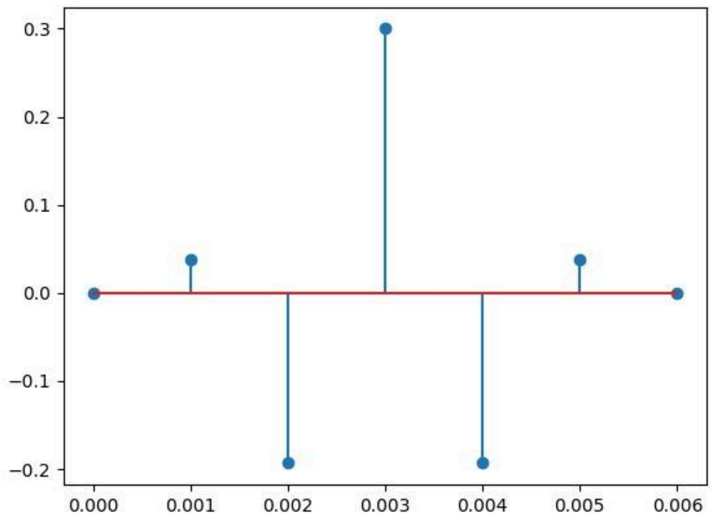

Fig. Filter Coefficients of our designed FIR Filter

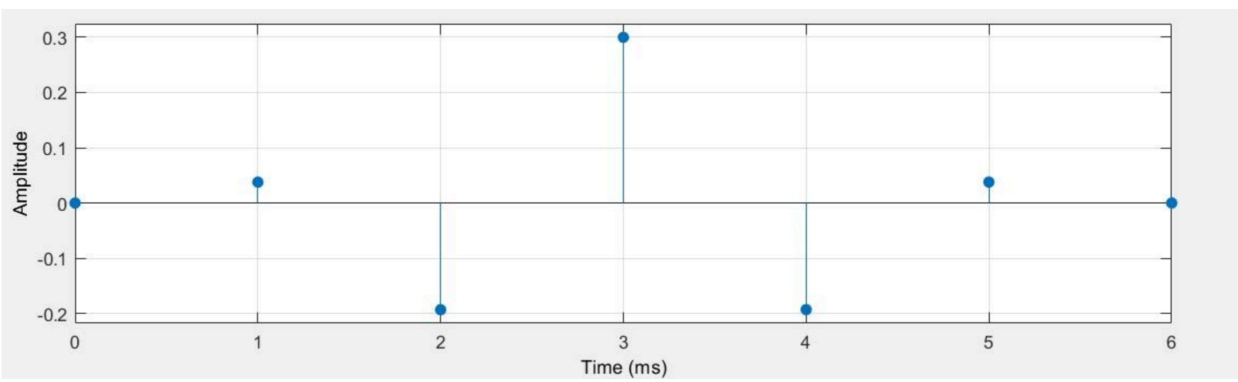

Fig. Filter Coefficients obtained from Matlab filterDesigner for 7-tap FIR using Hann window

The figure below depicts the frequency spectrum of the noisy signal, followed by the frequency spectrum of the filtered signal. It's evident that despite filtration, the 250 Hz component remains incompletely attenuated. Consequently, the filtered signal does not precisely resemble the 400 Hz signal in the time domain. This outcome underscores the importance of achieving adequate attenuation across all unwanted frequency components to ensure faithful signal reconstruction and noise reduction. Adjustments to the filter design or parameters may be necessary to address this discrepancy and enhance the effectiveness of the filtering process.

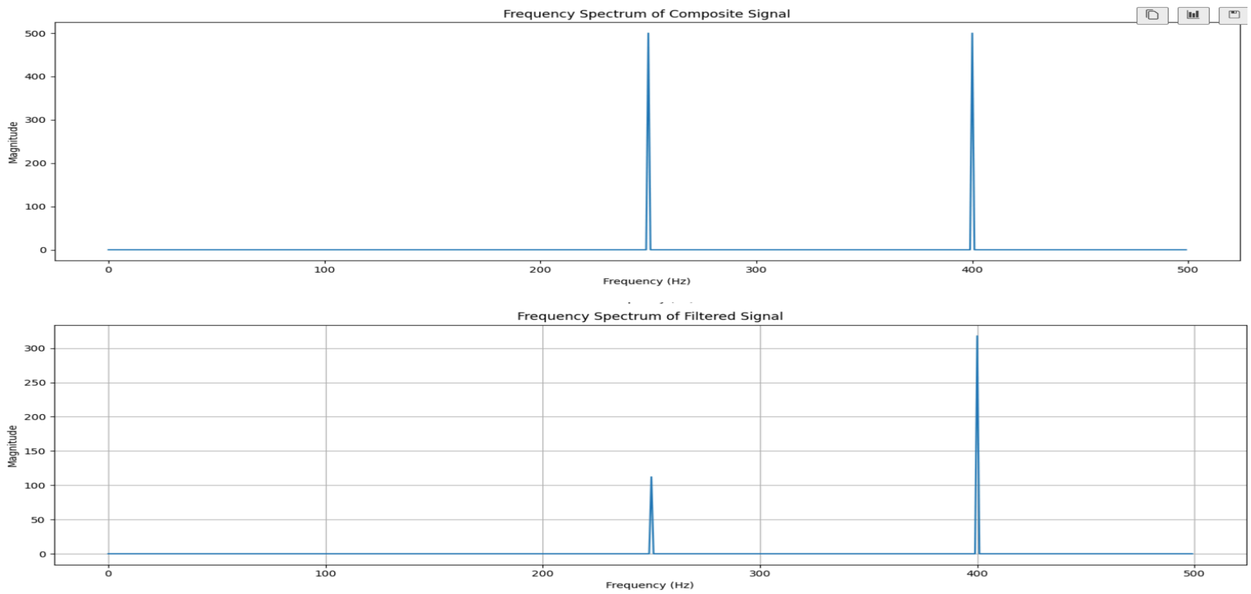

Fig. Frequency Spectrum of Composite signal and Filtered signal using Designed Filter

The depicted figure illustrates the frequency response of our designed filter. Remarkably, it closely resembles the frequency response obtained using an inbuilt function. This consistency indicates the effectiveness and accuracy of our filter design methodology. Achieving such similarity underscores the reliability and robustness of our designed filter, validating its suitability for various signal processing applications.

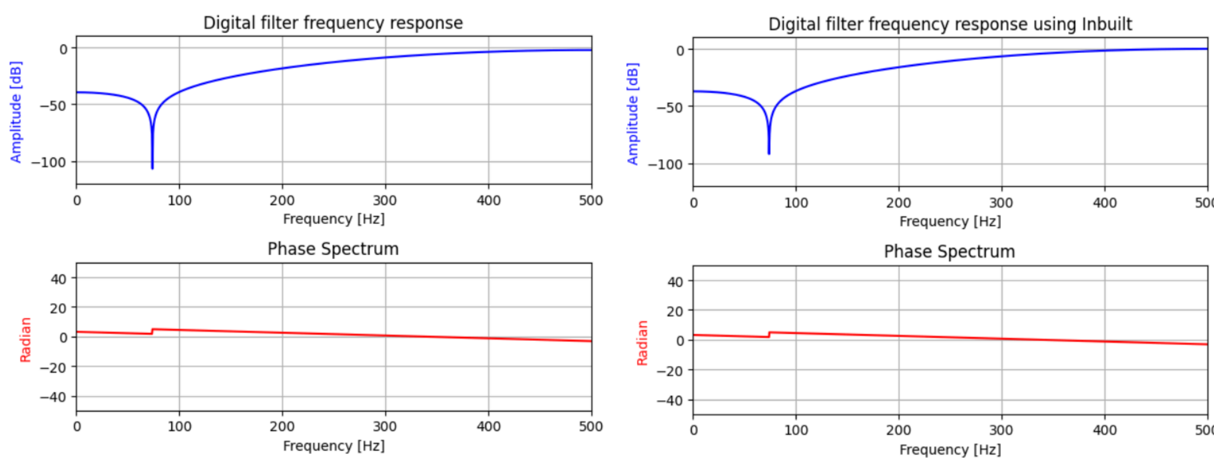

Fig. Frequency Response of Designed Filter and that obtained from Inbuilt function

The figure provided illustrates the simulation timeline of our Verilog code implementation, utilizing the filter coefficients derived from our filter design. Notably, the results closely mirror those obtained from the Python code. This congruence underscores the fidelity and accuracy of our Verilog implementation, affirming its consistency with the reference implementation. Such alignment between simulation outcomes validates the functionality and reliability of our Verilog implementation, instilling confidence in its suitability for practical deployment in digital signal processing systems.

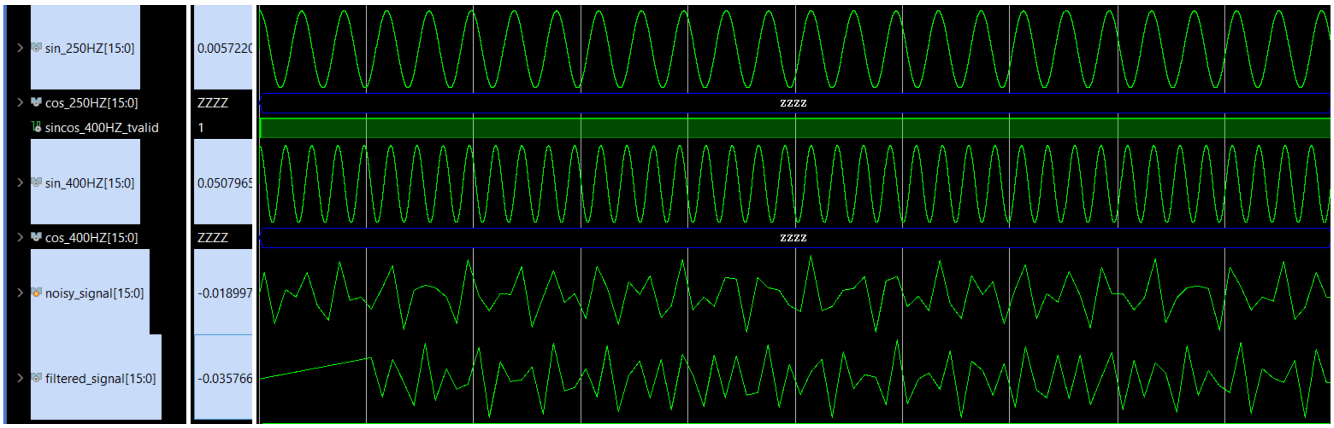

Fig. Simulation Results of the Verilog Code

The figure below displays the schematic of our Verilog code utilized for implementing the filter, utilizing its filter coefficients. Notably, the connections closely resemble those found in typical textbook diagrams of a 7-tap Finite Impulse Response (FIR) filter. This conformity to standard filter representations underscores the clarity and adherence to established design principles in our implementation. Such consistency facilitates comprehension and verification of our Verilog code, ensuring its reliability and effectiveness in practical applications.

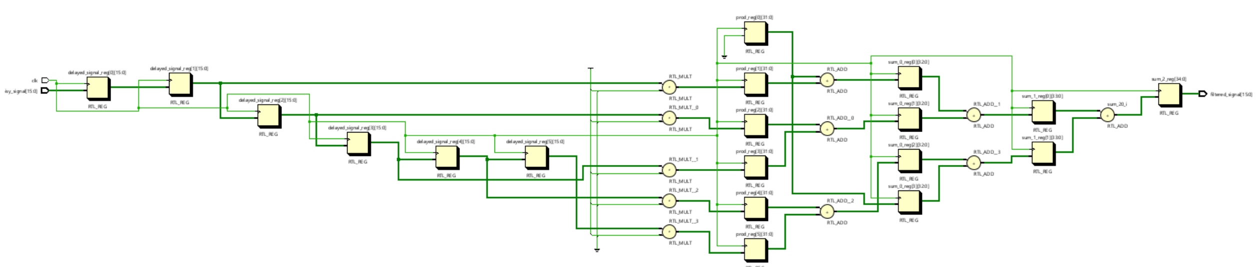

Fig. Schematic of 7-tap High Pass FIR filter

8. Problems Faced and Lessons Learned:

1. **Symmetric Coefficients and Filter Type:** Initially, the use of an even window length resulted in symmetric coefficients, which categorized the filter under type 2. However, this posed challenges for high pass filtering due to the presence of a pole at z=−1. We learned that for applications requiring asymmetric coefficients, especially with even window lengths, specialized design techniques are necessary to achieve desired filter characteristics.

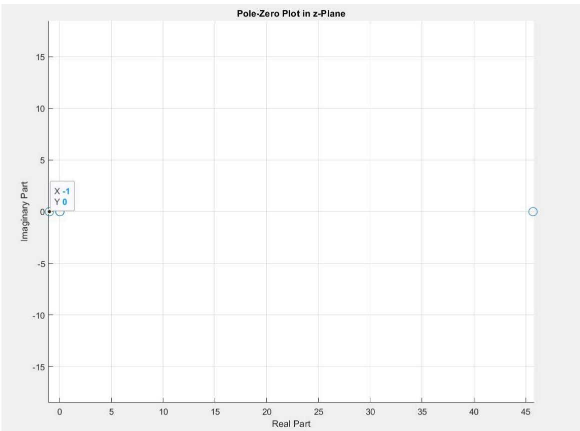

Fig. Pole Zero Plot for N = 6 before algo

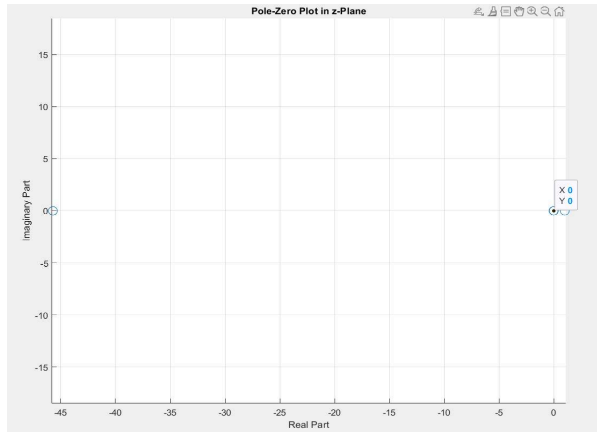

Fig. Pole Zero Plot for N = 6 after algo

2. **Sampling Frequency Selection:** Initially, we used a sampling frequency of 500 Hz with a cutoff frequency of 350 Hz. However, we later discovered that the sampling frequency should be greater than or equal to twice the cutoff frequency to prevent aliasing and ensure accurate signal reconstruction. This understanding led us to revise our sampling frequency selection criteria for future designs.

3. **Verilog Implementation Challenges:** During the Verilog implementation of the FIR filter, we encountered difficulties due to the use of floating-point coefficients. To address this, we devised a strategy to convert floating-point coefficients to fixed-point equivalents with a precision of 16 bits. This enabled more efficient hardware implementation while preserving the required level of accuracy.

4. **Simulation Time Constraints:** While conducting simulations in Vivado, we faced challenges due to the default simulation time constraints. Our clock signal operated on a millisecond scale, which required longer simulation times for thorough testing and validation. To overcome this limitation, we adjusted the simulation time settings in Vivado to accommodate longer clock periods, ensuring comprehensive testing of the design under various conditions.

5. **Filter Order and Attenuation:** In our initial design with an order 6 high pass filter and a cutoff frequency of 350 Hz, we observed inadequate attenuation of frequencies close to the cutoff frequency. This was particularly evident when passing a composite signal containing frequencies such as 250 Hz and 400 Hz, where the 250 Hz frequency component was not completely attenuated. To address this issue, we experimented with increasing the filter order. By raising the order to 50, we achieved significantly improved attenuation, resulting in a more satisfactory performance in filtering out unwanted frequencies from the signal.

9. Conclusion:

In conclusion, the project successfully achieved its objective of designing a 7-tap Finite Impulse Response (FIR) high-pass filter using the Hanning window technique. Through meticulous design and analysis, the filter coefficients were tailored to match those obtained using MATLAB's filterDesigner tool, ensuring accuracy and consistency in the filter's frequency response characteristics.

By leveraging the Hanning window method, the designed FIR filter demonstrated efficient attenuation of low-frequency components while preserving high-frequency content, aligning with the objectives of high-pass filtering. The filter's performance was further validated through frequency response analysis, signal processing, and spectral analysis, which confirmed its efficacy in isolating high-frequency signals from the input composite signal.

Overall, the project not only achieved its primary goal of designing a high-pass FIR filter but also demonstrated the effectiveness of the Hanning window method in filter design. The consistency of the filter coefficients with those obtained from MATLAB's filterDesigner tool underscores the reliability and accuracy of the design process. Moving forward, the insights gained from this project can be applied to various signal processing applications, contributing to advancements in digital filter design and implementation.

10. References:

1. Thingujam Churchil Singh; Manoj Kumar, Digital FIR Filter Designs in (2021) Asian Conference on Innovation in Technology (ASIANCON) https://ieeexplore.ieee.org/document/9544766

2. Digital Signal Processing, 4e by John G. Proakis (Author), Dimitris G. Manolakis (Author)

3. Fundamentals of Digital Signal Processing (English, Paperback, Ludeman Lonnie C)