IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Implementation and Design of FIR Filter using Verilog

[1]Riddhi Gupta, [2]Vaishali Agrawal, [3]Vanshika Jain

[123]Student

Department of Electronics & Communication Engineering,

ABES Engineering College, Ghaziabad, Uttar Pradesh

Abstract: Digital filters are systems that work with quantized, sampled input signals (digital signals) to accomplish filtering goals such as passing or rejecting specific bands of frequencies from the input signal bandwidth or allowing certain frequency ranges to pass at either a high or low frequency. An essential component of digital signal processing is the use of digital filters. Finite impulse response and infinite impulse response are the two types of digital filters that are employed. Among the design methods for the FIR filter are the windowing method and the distributed arithmetic algorithm.

This approach employs various window functions like rectangular, triangular, hamming, hanning, and Kaiser Window to truncate the infinite impulse response (e.g., Sink signal) and obtain a finite impulse response.

Distributed Arithmetic reduces the need for multiply-and-accumulate (MAC) blocks. It employs a pipeline structure and a divided Look-Up-Table (LUT) method to enhance system speed and decrease memory requirements. The Distributed Arithmetic structure stores MAC values in LUTs, enabling the retrieval of values based on input data. This approach conserves hardware resources by replacing MAC units with LUTs, offering a more efficient implementation of FIR filters with significant resource savings.

Index Terms - Windowing method, Distributed Arithmetic Algorithm, LUTs, Pipeline structure.

I. INTRODUCTION

1.1 Filters

A filter is an electronic circuit that modifies the waveform, amplitude-frequency, or phase-frequency attributes of a signal. Its purpose is to eliminate noise, extract signal information, or separate combined signals. Filters are crucial in engineering, particularly in electrical, electronics, and communication fields. They are typically frequency-selective, allowing only specific frequency ranges to pass, such as low or high frequencies. Filters are broadly classified as **analog** or **digital**, depending on their implementation.

Analog filter- Any filter that operates on continuous-time signals to remove noise, allow certain frequency ranges to pass through or block them, or achieve other filtering objectives is called an analog filter.

Digital Filter- A digital filter is a kind of filter that uses a digital input signal to perform filtering operations and produces a digital output signal.

1.2 Types of Digital Filters

i. Finite Impulse Response Filter (FIR)

A Finite Impulse Response (FIR) filter is a digital filter whose output is the weighted sum of the current and past input samples. It is defined by the equation:

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

The impulse response h(k) of an FIR filter, as indicated by its difference equation, is finite from 0 to N, where N is the number of taps or coefficients. The output y(n) depends on present and past input signal values. The transfer function of the FIR filter is obtained by taking the z-transform of the impulse response, as mentioned below:

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}$$

The following diagram illustrates a block diagram of a digital FIR filter, showing components such as input, output, delays, coefficients, and summation.

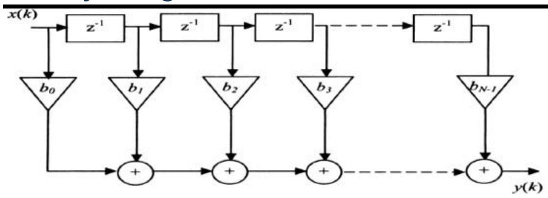

Fig. 1.2 (a) FIR Filter block diagram

ii. Infinite Impulse Response (IIR)

An Infinite Impulse Response (IIR) filter is a type of digital filter characterized by its recursive nature, where the output is a weighted sum of present and past input samples as well as past output samples. The general equation for an IIR filter is:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

The problem with the direct form equation for an IIR filter is that it involves an infinite impulse response, which is impractical for computation. Instead, the equation is expressed in a recursive form, known as the difference equation of the IIR filter, given below:

$$y(n) = \sum_{k=0}^{N} bkx(n-k) \; - \; \sum_{k=1}^{M} aky(n-k)$$

Where a k and b k are the coefficients of the IIR filter.

The difference equation for the IIR filter indicates that the output signal y(n) depends on the present and past values of the input signal, as well as past output values. This feedback mechanism is a characteristic feature of IIR filters.

II. EASE OF USE

It takes thorough and easily accessible information to navigate the complexities of digital signal processing and filter design. The goal of this review paper on Finite Impulse Response (FIR) filter implementations is to provide a comprehensive and easily navigable resource for novice researchers as readers expect a more in-depth examination of each methodology.
Purposeful Segmentation: The beginning clearly and purposefully introduces each methodology, carefully segmenting the content. This division makes it easier for readers to cognitively arrange the data and comprehend how one approach logically leads to another.
Practical Implementation Focus: To put the conversation in context, real-world examples are used to demonstrate the methodologies' practical uses. The introduction makes sure that readers can quickly understand the applicability and possible influence of the approaches in their respective fields by providing examples of how each technique is used in practice.
Decision Guidance: The introduction's clear purpose statement, which is to direct engineers and researchers, is one of its main features. This section alerts readers to the review's practical value right away and lets them know that they can anticipate insightful information that will help them make decisions about the application of FIR filters.
Accessible Language: The introduction's wording is clear and simple, free of superfluous technical jargon that could make it difficult to understand. This method guarantees that readers can interact with the material and understand the core ideas being offered, irrespective of their degree of experience.

III. Overview of designing and implementation methodologies of FIR Filter

Finite Impulse Response (FIR) filter implementation is a fundamental component of digital signal processing, with applications spanning from biomedical signal analysis to communications. This study reviews in detail the designing process and hardware implementation approaches used to create FIR filters, providing insight into their schematics and uses. This study seeks to provide a comprehensive understanding of FIR filter design by examining several methodologies

3.1 DESIGNING OF FIR FILTER-

The FIR filter's design specification is divided into the following five steps: -

(i) **Specification**: Define the filter requirements such as the desired frequency response, passband, stopband, and transition width. Determine parameters like filter order and cutoff frequency.

(ii) **Design Method Selection**: Choose a design method based on the specifications and application requirements. Common methods include windowing (e.g., Hamming, Kaiser), frequency sampling, or optimization techniques.

(iii) **Coefficients Calculation**: Calculate the filter coefficients based on the chosen design method. For windowing methods, apply the chosen window function to the desired impulse response to obtain the filter coefficients.

(iv) **Filter Structure Selection**: Decide on the filter structure, such as Direct Form, Cascade Form, or Transposed Form, based on factors like implementation complexity, resource utilization, and numerical stability.

(v) **Implementation**: Implement the designed filter structure in hardware or software. This involves coding the filter equations, simulating the design to verify its performance, and finally, deploying the filter in the target system.

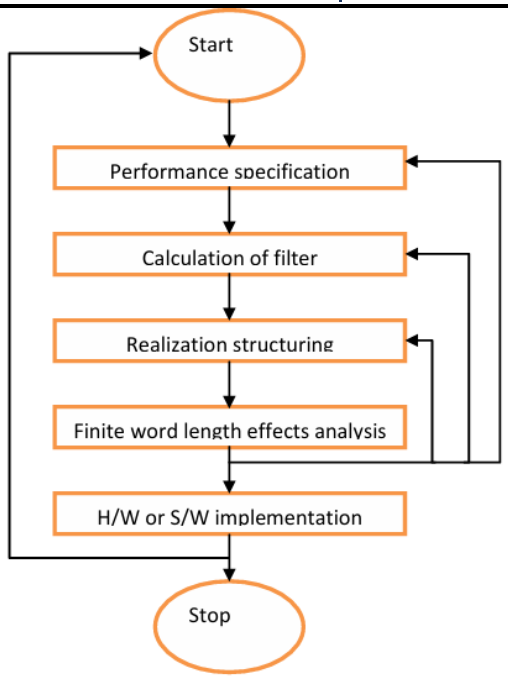

Fig 3.1 (a) FIR Filter Designing Flowchart [15]

3.1.1 Windowing Technique

To obtain the finite impulse response of the filter, the windowing method of FIR filter design involves using window functions such as rectangular, hamming, hanning, and Kaiser Window and multiplying it with the desired infinite impulse response. This truncation causes ripples and overshoots in both the passband and the stopband of the FIR filter frequency response. Gibbs phenomenon is the term for this ringing effect that occurs close to the FIR filter's band edge; the amplitude of this phenomenon varies depending on which window function is utilized in the design. Each window function has a unique side lobe attenuation and transition width, as well as unique features in the time and frequency domains. When designing an FIR filter, the optimal window function has a decent filter response, fewer side lobes, and relatively fewer ripples in the pass- and stop-bands.[1]

Window functions-

Window functions are employed to limit the infinite impulse response, such as in the design of FIR filters. Below, various window functions are elaborated upon.

(i) Rectangular window function

A rectangular window function is a simple type of signal-processing technique used in DSP and spectrum analysis. It is characterized by a constant value within a specified interval and zero outside of it, resulting in a rectangular shape when plotted.
While straightforward to implement, its main drawback lies in its abrupt transition between zero and non-zero values, leading to spectral leakage and poor frequency resolution in Fourier analysis.
Despite this limitation, it finds application in basic signal-processing tasks where simplicity outweighs the need for precise frequency analysis.

$$W_R(n) = \begin{cases} 1, & 0 < n < L-1 \\ 0, & otherwise \end{cases}$$

ii. Hanning Window Function

The Hanning window function, also known as the Hann window, is a signal-processing technique used to reduce spectral leakage in Fourier analysis. Similar to the Hamming window, it features smooth tapering towards the window's edges, effectively minimizing abrupt transitions and spectral leakage.
The Hanning window is characterized by a central lobe with reduced sidelobes compared to the rectangular window, resulting in improved frequency resolution.
Widely employed in various fields such as audio and vibration analysis, the Hanning window offers a balance between simplicity and enhanced frequency analysis accuracy, making it a popular choice in signal processing applications.

$$W_{HN}(n) = \begin{cases} 0, & otherwise \\ 0.5 - 0.5\cos\left(\frac{2\pi n}{L-1}\right), & 0 < n < L-1 \end{cases}$$

iii. Hamming window function

The Hamming window function is a type of signal-processing technique utilized to reduce spectral leakage in Fourier analysis. Unlike the rectangular window, it employs a smooth tapering technique, gradually decreasing the amplitude towards the window's edges. This gradual tapering helps minimize abrupt transitions, resulting in reduced spectral leakage and improved frequency resolution. The Hamming window is characterized by a central lobe with smaller sidelobes compared to the rectangular window.
Widely used in various applications such as audio and image processing, the Hamming window strikes a balance between simplicity and enhanced frequency analysis accuracy.

$$W_H(n) = \begin{cases} 0, & otherwise \\ 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right), & 0 < n < L-1 \end{cases}$$

iv. Blackman Window Function

The Blackman window is a windowing method used in FIR filter design, known for its excellent stopband attenuation and narrow main lobe width. Its advantages include superior frequency response characteristics, making it suitable for applications requiring high precision.
However, it can introduce higher computational complexity compared to simpler window functions. Additionally, designing filters with the Blackman window may require careful parameter selection to achieve optimal performance, potentially leading to increased design complexity.

$$W_B(n) = \begin{cases} 0, & otherwise \\ 0.42 - 0.5\cos\left(\frac{2\pi n}{L-1}\right) + 0.08cos\left(\frac{4\pi n}{L-1}\right), & 0 < n < L-1 \end{cases}$$

v. Kaiser Window Technique

The Kaiser window is a versatile windowing method in FIR filter design, offering a flexible trade-off between transition width and stopband attenuation. Its advantages lie in its adaptability to meet specific passband and stopband requirements. Adjusting parameters, enables precise control over frequency response characteristics. However, its implementation may require more computational resources compared to simpler window functions. Furthermore, determining optimal parameters for the Kaiser window can be challenging, potentially adding complexity to the design process.

$$d_n = \begin{cases} \sqrt{\dfrac{\sum_{i=0}^{n} w[i]}{\sum_{i=0}^{N} w[i]}} & \text{if } 0 \leq n < N \\ \sqrt{\dfrac{\sum_{i=0}^{2N-1-n} w[i]}{\sum_{i=0}^{N} w[i]}} & \text{if } N \leq n \leq 2N-1 \\ 0 & \text{otherwise.} \end{cases}$$

Simulation Results

The response of FIR Low Pass Filter designing using various windowing techniques are shown below/

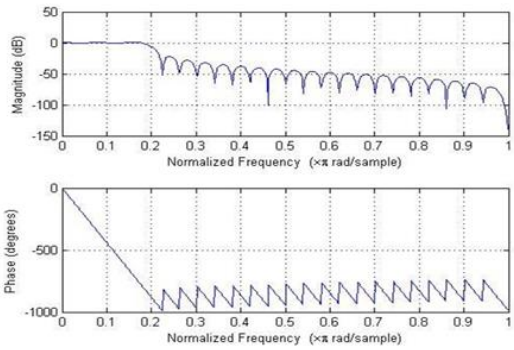

Fig. 3.1.1 (a) Response of the Low pass FIR filter using the Rectangular Windowing Method [5]

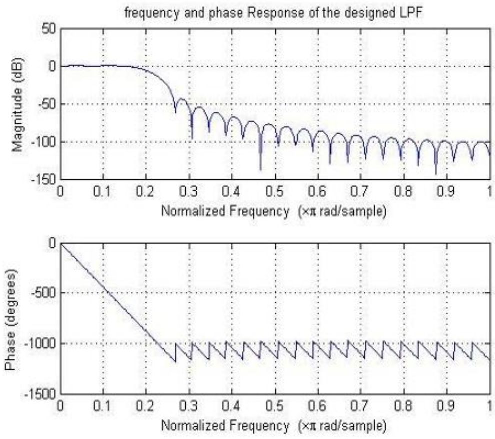

Fig 3.1.1 (b) Response of the Low pass FIR filter using the Hanning Windowing Method [5]

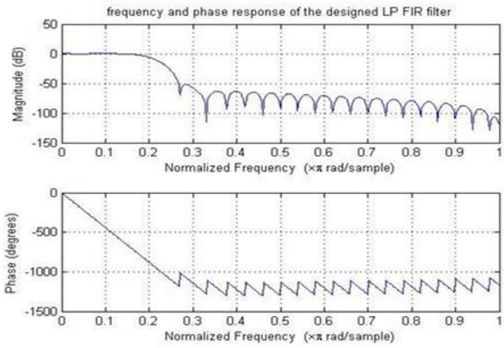

Fig 3.1.1 (c) Response of the Low pass FIR filter using the Hamming Windowing Method [5]

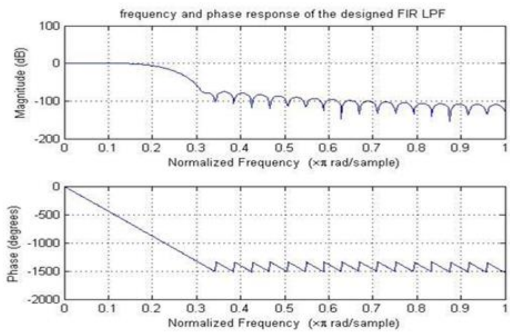

Fig 3.1.1 (d) Response of the Low pass FIR filter using the Blackman Windowing Method [5]

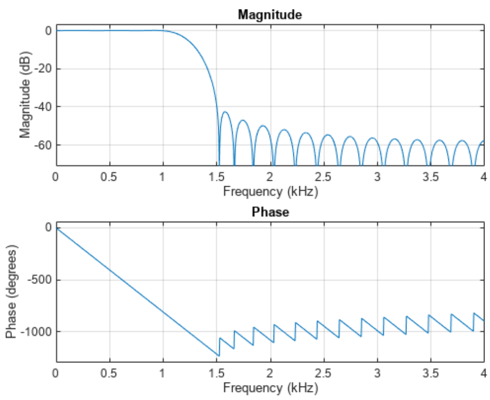

Fig 3.1.1 (e) Response of the Low pass FIR filter using the Kaiser Windowing Method [5]

3.2 HARDWARE-IMPLEMENTATION OF FIR FILTER

3.2.1 Distributed Arithmetic Algorithm

3.2.1.1 Theory

Distributed arithmetic is a method for efficiently implementing digital signal processing algorithms, particularly for FIR filters. It replaces traditional multiply-accumulate operations with shift operations and look-up tables (LUTs). In this technique, each coefficient of the FIR filter is converted into a binary representation and stored in a LUT.

To calculate the filter output for a given input sample, the binary form of the input sample is used as an index to access the corresponding coefficient from the LUT. These coefficients are then shifted and summed to obtain the filter output.

This approach offers benefits such as reduced computational complexity, lower power consumption, and faster processing. However, it requires additional memory for storing the LUTs and may not be suitable for filters with a large number of taps due to the size of the LUTs.

3.2.1.2 Implementation

Among the most popular approaches to applying FIR filters is Distributed Arithmetic. In the case of FIR filters, where the coefficients are known beforehand, the DA calculates the inner product equation.

The following describes a FIR filter of length K:

$$y[n] = \sum_{k=0}^{K-1} h[k]x[n-k] \quad \text{……….(1)}$$

Here, h[k] represents the filter coefficient, and x[k] denotes the input data. To facilitate analysis, we introduce a modified input data notation, where x′[k]=x[n−k] , thereby adjusting Equation (1) accordingly.

$$y = \sum_{k=0}^{K-1} h[k].x^{'}[k] \quad \text{………(2)}$$

Next, we represent the input data with the binary numbers of B-bit two's complement:

$$x^{'}[k] = -2^{B} \cdot x_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^{b} \quad \text{……….(3)}$$

Where $x_b[k]$ represents the b'th bit of x[k] with x[k]b∈{0,1}x[k]b∈{0,1}.

$$y = \sum_{k=0}^{K-1} h[k] \cdot (-2^{B} \cdot x_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^{b})$$

$$= -2^{B} \cdot \sum_{k=0}^{K-1} h[k] \cdot x_B[k] + \sum_{b=0}^{B-1} 2^{b} \cdot \sum_{k=0}^{K-1} h[k] \cdot x_b[k]$$

$$= -2^{B} \cdot f(h[k], x_B[k]) + \sum_{b=0}^{B-1} 2^{b} \cdot f(h[k], x_b[k]) \quad \text{…..(4)}$$

Now we have,

$$f(h[k], x_b[k]) = \sum_{k=0}^{K-1} h[k] \cdot x_b[k] \quad \text{…….(5)}$$

Equation (4) implies that storing filter coefficients in a Look-Up Table (LUT), addressed by xb=[x0b,x1b,...,xK−1b], simplifies FIR filter operations. This method reduces MAC blocks to simple access and summation with the LUT.
Digital filters employing this arithmetic utilize registers, memory resources, and a scaling accumulator for implementation
The initial LUT-based Direct Architecture (DA) implementation of a 4-tap FIR filter is depicted in Figure 3.3. This architecture comprises three components: the shift register unit, the DA-LUT unit, and the adder/shifter unit.

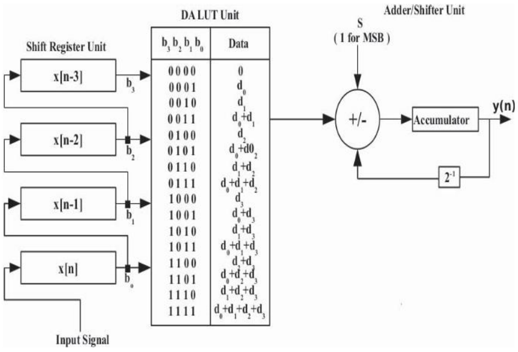

Fig 3.2.1.2(a) LUT-based Distributed Arithmetic Algorithm Structure [16]

3.2.1.3 Look Up Table Less Distributed Arithmetic Architecture

The LUT-less architecture for distributed arithmetic is a method that removes the reliance on look-up tables (LUTs) when implementing digital signal processing algorithms like FIR filters. Instead of using LUTs to store and retrieve coefficient values, this approach calculates the coefficient values directly through simple logic operations and shift registers.
In this technique, the binary representation of filter coefficients is directly employed in arithmetic operations, eliminating the need for a separate memory lookup step. The algorithm computes the filter output by executing shift and add operations on the input data and coefficient values, without requiring pre-stored values in a LUT.

Why LUT Less Structure?

Both the memory size and the filter order grow as they go up. Thus, the look-up table (LUT) size grows as a result. Hence, for improved performance, we substitute combinational logic for lookup tables. Given that multiplexers and complete adders can replace each LUT unit, the suggested DA-LUT unit significantly lowers memory consumption.

It is evident that the upper half of b_3 LUT (locations where 3 = 0) and h [3] add up to the same value as the lower half of LUT (locations b_3 where 3 = 1). Because of this, the LUT size can be cut in half by adding a complete adder and an extra 2x1 multiplexer, as shown in Fig 3.2.1.3(a)

The final LUT-less DA designs are achieved by the same LUT reduction process, as Fig 3.2.1.3(b) illustrates. Conversely, using a combination logic circuit will have an impact on the filter's performance. However, in situations when the filter taps are prime, we can achieve a trade-off between filter performance and low resource consumption by utilizing 4-input LUT units with extra multiplexers and complete adders.

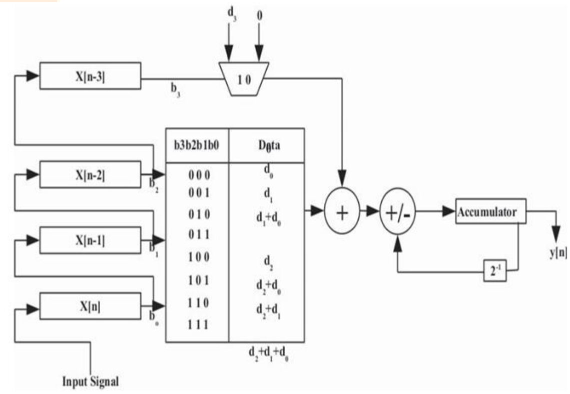

Fig 3.2.1.3(a) DA Structure of a 4-tap filter
(2^3 word LUT Implementation of DA) [16]

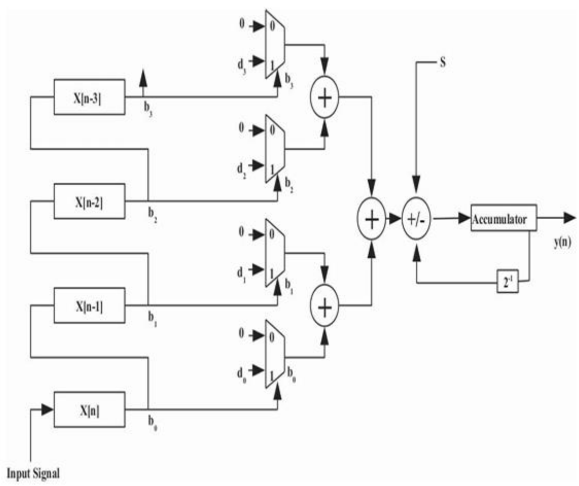

Fig 3.2.1.3 (b) LUT-Less DA- Structure [16]

IV. Simulation Result

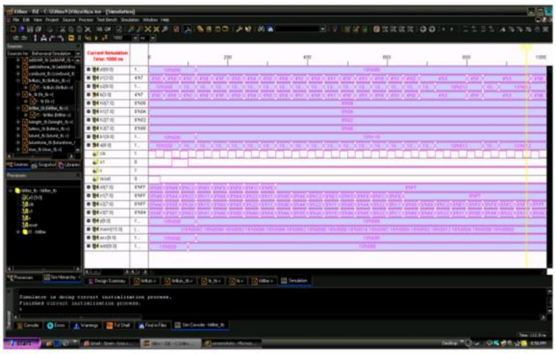

Fig 3.2.1.3 (c) LUT- Based DA implementation of 4-tap FIR filter [4]

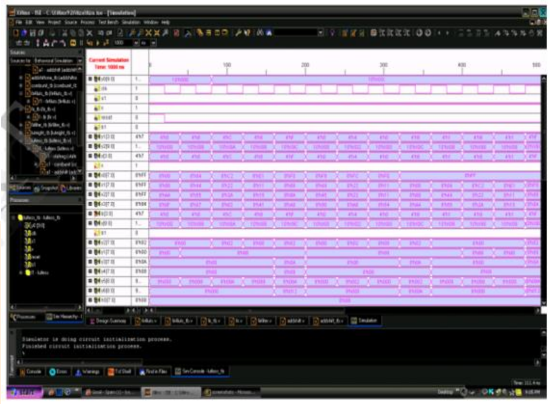

Fig 3.2.1.3 (d) LUT-Less DA- implementation of 4-tap FIR Filter [4]

Analysis of Windowing Simulation Results

The Rectangular window has the narrowest main lobe width but exhibits significant side lobe width, leading to poor stopband attenuation. The Hamming window offers improved side lobe attenuation at the expense of widening the main lobe slightly. The Hanning window further improves side lobe attenuation compared to Hamming while maintaining a similar main lobe width. The Blackman window provides even better side lobe attenuation but may slightly widen the main lobe compared to Hanning. The Kaiser window allows for adjustable trade-offs between main lobe width and side lobe attenuation, making it highly versatile for filter design, especially in scenarios where precise control over these parameters is essential. Overall, the choice of windowing technique depends on the specific requirements of the application, balancing trade-offs between main lobe width and side lobe attenuation.

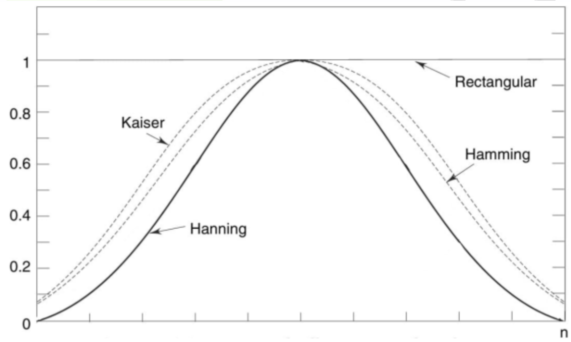

Fig. 4.1(a) Comparative Analysis of different types of windowing methods.

Table 4.1 Frequency-domain characteristics of window functions

Window	Main Lobe Width	Peak of First Sidelobe (dB)	Minimum Stop Band Attenuation (dB)
Rectangular	4π/M	-13 dB	+21 dB
Hanning	8π/M	-32 dB	+44 dB
Hamming	8π/M	-43 dB	+53 dB
Blackman	12π/M	-58 dB	+74 dB
Kaiser	5.8π/M	-34.5 dB	+50 dB

V. CONCLUSION

The comparison of windowing techniques for FIR filter design highlights the superiority of the Kaiser window due to its ability to achieve superior main lobe width and side lobe suppression compared to other methods like rectangular, Hamming, Hanning, and Blackman. Additionally, in hardware implementation, utilizing a LUT-less Distributed Arithmetic (DA) structure proves advantageous in reducing memory elements. By employing Kaiser windowing in conjunction with DA, FIR filter implementations can achieve efficient resource utilization while maintaining high performance. Thus, for applications demanding stringent performance criteria and resource efficiency, the combination of Kaiser windowing and DA structure emerges as a compelling choice for FIR filter design and implementation.

VI. ACKNOWLEDGEMENT

We sincerely thank Dr, Raman Kapoor Associate Professor at ABES Engineering College, for his wise counsel, steadfast assistance, and perceptive comments during this paper's research and writing. Our understanding of FIR filter technologies has been greatly shaped by Dr. Raman Kapoor's encouragement and expertise, which has also improved the overall caliber of this work. In addition, we would like to thank Vaishali Agrawal, and Riddhi Gupta, our coauthors. Their dedication, teamwork, and spirit of cooperation have enhanced the variety of viewpoints represented in this paper.

We would also like to express our sincere gratitude to the academic community, whose groundbreaking studies provided the impetus for our investigation into multiplexer innovations. We are grateful for the contributions made by academics and researchers whose work has greatly influenced how we understand digital circuit design.

Lastly, we would like to express our gratitude to our families and friends for their understanding and support during the challenging stages of this research project. This work is evidence of the mentorship and collaborative spirit that characterizes the academic experience at ABES Engineering College, and we are appreciative of the chance to advance the field under Dr. Raman Kapoor's guidance.

REFERENCES

[1] Sanjay Sharma, "Digital Signal Processing", Katson Books.

[2] S Salivahanan, A Vallavaraj, C Gnanapriya: "Digital Signal Processing", pp 381-401, Tata McGraw-Hill, second edition, 2010.

[3] John G. Proakis, Dimitris G. Manolakis, "Digital signal Processing" PEAESON, 4th edition.

[4] Keerthi, M., et al. "FPGA implementation of distributed arithmetic for fIR filter." *International Journal of Engineering Research and Technology* 1.9 (2012).

[5] Ado, Abbati Muhammad, Jitendra Vaswani, and Muhammad Ibrahim Zubair. "Design, simulation and response analysis of FIR filter designed by windowing technique." *Int. Research J. of Engineering and Technology* 9.9 (2021): 1037-1044.

[6] Gopika, P. "FIR window method: A comparative analysis." IOSR Journal of Electronics and Communication Engineering 10.4 (2015): 15-20.

[7] L. Rosler, "Design and analysis of an FPGA based low tap bandstop FIR filter," Master thesis, Youngstown State University, 2021

[8] Subramaniam, B., Siddik, Z. H., & Nagoor, N. H. (2020). Optimization of nanostructured lipid carriers: Understanding the types, designs, and parameters in the process of formulations. Journal of Nanoparticle Research, 22, 1-29

[9] Subramaniam, B., Siddik, Z. H., & Nagoor, N. H. (2020). Optimization of nanostructured lipid carriers: Understanding the types, designs, and parameters in the process of formulations. Journal of Nanoparticle Research, 22, 1-29.

[10].Patali, P., and Kassim, S.T."High throughput FIR filter architectures using retiming and modified CSLA based adders." IET Circuits, Devices & Systems, Vol. 13, No. 7, pp. 1007–1017, 2019.

[11] Hardware Implementation of Parallel FIR Filter Using Modified Distributed Arithmetic by Gourishankar Das , Krishanu Maity , Suman Sau 2018 2nd IEEE International Conference.

[12] Yong Du. "The implementation of Digital filter based on Matlab and FPGA (Altrra/Verilog)". Beijing: Electronic Industry Press, 2015.03.

[13] .Cui W. "Design and Analysis of High-Performance FPGA System Timing. Microcomputer Applications".2014(2).

[14] Yong Du. "The implementation of Digital filter based on Matlab and FPGA (Altrra/Verilog)". Beijing: Electronic Industry Press, 2015.03.

[15]Narwaria, R. P., and Manish Kumar. "Comparison of FIR filter using different window functions." *International Research Journal of Engineering and Technology (IRJET)* 5.10 (2018): 2395-0056.

[16] NagaJyothi, Grande, and Sriadibhatla SriDevi. "Distributed arithmetic architectures for fir filters-a comparative review." *2017 International conference on wireless communications, signal processing and networking (WiSPNET)*. IEEE, 2017.

[17] Su Peng, “Design and analysis of FIR filters based on Matlab”, Degree Project of Bachelor thesis, Linnaeus University, school of computer science, physics and Mathematics.

[18] John G. Proakis, Dimitris G. Manolakis, “Digital signal Processing” PEAESON, 4th edition.

[19] McClellan , J.H. Parks, T.W. Rabiner, L.R. "A computer program for designing optimum FIR linear phase digital filters":. IEEE Trans.Audio Electroacoust. Vol. 21, No.6, pp:506-526, 1973.

[20] Sonika Gupta, Aman Panghal Performance Analysis of FIR Filter Design by Using Rectangular,Hanning and Hamming Windows Methods,IJARCSSE Vol-2,Issue-6,Jun2012,pp273-278

[21] Patrick Longa, Ali Miri, "Area-Efficient Fir Filter Design on FPGAs using Distributed Arithmetic" IEEE International Symposium on Signal Processing and Information Technology, pp:248-252,2006

[22] Shivani Sharma & Ravindra Pratap Narwaria, “Noise Removal Techniques for ECG signal – A Survey Approach” published in Current Trends in Signal Processing (STM Journals), ISNN: 2277-6176, 2013, pp 19-23.

[23] Shivani Sharma & Ravindra Pratap Narwaria, p-ISSN: 2395-0072 “Performance Evaluation of various window techniques for Noise cancellation from ECG signal” published in International Journal of Computer Application, Volume 93 and No. 19, ISNN: 0975-8887, May 2014, pp 1-5.

[24] Mohan Das Choudhary and Ravindra Pratap Narwaria, “ ECG signal processing with FIR digital filter by window techniques” published in International Journal of Research in Engineering & Applied Sciences, volume 2, issue 8 (August 2012), pp 38-46.

[25] Manjinder Kaur, Sangeet Pal Kaur, “FIR Low Pass Filter Designing Using Different Window Functions and their Comparison using MATLAB”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 5, Issue 2, February 2016.

[26] Manira Khatun, Implement a new Window function and Design FIR filters by using this new window, IJECS,Volume-3,Issue-3, March 2014, pp4087-4090

[27] Heejong Yoo, David V.Anderson, "Hardware-Efficient Distributed Arithmetic Architecture For High-order Digital Filters", IEEE International Conference on Acoustics, Speech and Signal Processing, Vol.5,pp: 125-128,March,2005