

Design and Implementation of High Performance FIR and IIR Digital Filters for ECG Signal Processing

Dr. Mukthi S L¹, Sandeep O²

Associate professor, Dept of ECE, Bangalore Institute of Technology, Bangalore, Karnataka, India¹

PG Student, Department of ECE, Bangalore Institute of Technology, Bangalore, Karnataka, India²

Abstract: The design and implementation of digital filtering methods to improve electrocardiogram (ECG) signals. MATLAB is used for the development and analysis of low-pass filters with finite impulse response (FIR) and infinite impulse response (IIR) of different orders. The main goal is to eliminate noisy elements like high-frequency interference and baseline drift while keeping the therapeutically meaningful aspects of ECG signals. In terms of signal preservation and noise reduction, the FIR filters with Kaiser windows and the IIR filters based on elliptic responses perform better. The comparative study provides valuable insights for biomedical applications by highlighting the trade-off between filter, complexity, and signal clarity. The results demonstrate that digital filtering techniques are useful instruments for processing ECG signals accurately, which are use in diagnostic and healthcare monitoring systems.

Keywords: Digital Filters, FIR, IIR, ECG Signal Processing, MATLAB, Vivado.

I. INTRODUCTION

In signal processing, a digital filter is a mathematical method that modifies particular aspects of a discrete-time signal. Digital filters use parts like ADC for signal sampling, CPU, memory, and DAC for output, in contrast to analog filters that process real-time analog data. In order to build the digital filter and perform mathematical computations on the collected data, the CPU carries out software instructions. A field-programmable gate array (FPGA) can speed up filtering activities in high-performance applications by taking the role of a general-purpose CPU. Like individual logic gates, FPGAs may perform basic or complicated tasks thanks to their programmable logic blocks and interconnects. A essential non-invasive diagnostic technique for tracking the electrical activity of the human heart is the electrocardiogram (ECG). Regarding conduction anomalies, heart rhythm, and potential indicators of cardiovascular disorders, it offers important information. ECG signal interpretation accuracy is critical for prompt diagnosis and treatment, particularly in applications like wearable medical technology, remote telemedicine platforms, and intensive care monitoring. Electromyographic (EMG) noise from muscular activity, power line interference at 50/60 Hz, and baseline wander from breathing or body movement are the most frequent causes of noise in ECG readings. The ECG waveform may be distorted by these undesired elements, which may also obscure clinically important characteristics including the T-wave, P-wave. Advanced filtering techniques are therefore needed to improve ECG quality by reducing noise while maintaining the crucial diagnostic data. For biomedical, digital signal processing (DSP) methods—in particular, digital filtering—have become essential for increasing the precision of physiological signals.

Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) are the two primary classifications for digital filters. Because of its linear phase properties and guarantee stability, FIR filters are commonly utilized to preserve the shape of the ECG waveform. On the other side, IIR filters can provide finer frequency responses with smaller filter orders, enabling computational economy, but they may create phase distortion if not properly constructed. In order to improve the ECG signal, FIR and IIR low-pass filters of different orders are constructed and simulated in this study using MATLAB. Various design techniques are investigated and contrasted, including window functions for FIR and conventional IIR approaches.

MATLAB is used to create and assess digital low-pass FIR and IIR filters for improving ECG signals. The effectiveness of various filter orders and design techniques to reduce noise while preserving the integrity of diagnostic characteristics is examined. Since these are frequent issues in clinical and wearable ECG monitoring systems, special attention is paid to baseline wander elimination and high-frequency interference reduction. The

best filter designs for biological applications are determined by comparing the outcomes.

II. CONCEPT THEORY OF MAIN TOPICS IN THE RESEARCH

Digital signal processing chips or application-specific integrated circuits (ASICs) are commonly used to create digital filter algorithms. For moderate-volume applications, however, the use of Field-Programmable Gate Arrays (FPGAs) to construct digital filters can provide benefits including more flexibility, reduced costs, and higher sampling rates. This chapter describes an FPGA-based method for implementing digital filters.

A. Digital Filter

A mathematical procedure used in hardware and/or software to process digital input signals in order to accomplish a filtering goal is called a digital filter. Both the input and output signals are discrete in this discrete-time signal system. DSP, or digital signal processing, is widely used in many different fields. Higher precision, lower filter order requirements, and the ability to customize a variety of filter types and configurations to meet individual demands are just a few of the many benefits that digital filtering offers over analog filtering. Digital filters rely on set filter coefficients, which ensure stability over time, in contrast to analog filters, They depend on changing component values for their cutoff frequencies and get more complex. Digital filters operate in the time domain with reduced overshoot, a faster roll-off, and a smaller transition width. As such, real-time applications such as biomedical engineering, image processing, data collection, and recording are particularly suitable for them. Furthermore, digital filters of the Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) types provide flatter responses without the ripples typically seen in analog filters.

I. Finite Impulse Response (FIR)

In finite impulse response The filter's response decays to zero once the impulse has circulated through the system. It reacts impulsively for a brief period. This equation shows the convolution sum between the input and output signals of the filter. In Figure 1, the FIR filter is implemented in a straight forward transversal fashion.

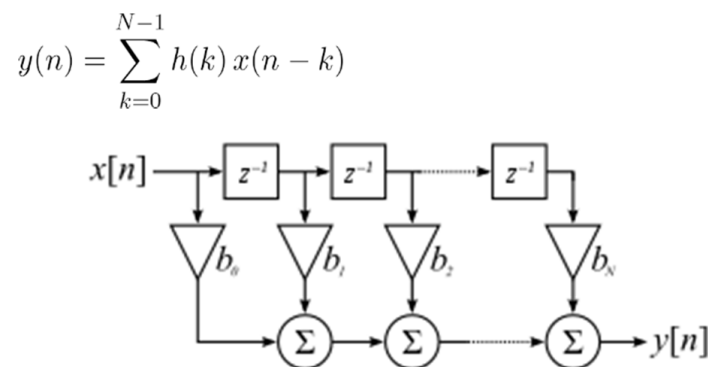


Fig. 1, Architecture of FIR filters

Additionally, lattice and cascade structures can be implemented. The number of delay lines incorporated into the design of a filter determines its order. The filter kernel is defined as the coefficients $b[0], b[1], \dots, b[N]$. Moving Average and Windowed Sine are two of the various types of FIR filters that are available. There are three ways to calculate the filter coefficients in FIR filters: window, optimal, and frequency sampling. The Window approach was used since it requires little calculation and is straightforward. When the number of periods in the signal being measured is not an integer, FIR filters experience spectral leakage, which results in larger spectral lines and discontinuities at the endpoints. This may provide a signal that differs from the initial continuous-time signal in certain ways. The most common types of Window re:

1. Rectangular Window.

The value of this window is 1 for a specific time period and 0 otherwise; it is used when none of the other windows are provided. A rectangular window multiplied by a signal allows us to extract the part of the signal that exists while the window's values are not zero. The equation for a rectangular window where M represents the number of filter order and n is the number of samples.

$$w_{rec}(n) = 1, -M \leq n \leq M$$

2. Hanning and Hamming windows

Narrow side lobes and large peaks characterize the sinusoidal Hanning and Hamming windows. In contrast to the Hamming window, which has a slight signal break because it doesn't quite reach zero, the Hanning window has equal zero ends. No other window is as good at canceling out the nearest side lobe as the Hamming window. we can measure noise levels with these windows. Generally speaking, the Hanning window works well due to its excellent frequency resolution and low spectrum leakage. The equations are shown below

$$w_{han}(n) = 0.5 + 0.5 \cos\left(\frac{n\pi}{M}\right), -M \leq n \leq M$$

$$w_{ham}(n) = 0.54 + 0.46 \cos\left(\frac{n\pi}{M}\right), -M \leq n \leq M$$

3. Kaiser Window

It achieves minimal side lobe height, minimal side lobe distance, and accurate amplitude. Here is the equation. In the frequency domain window, the real-valued parameter determines the trade-off between side-lobe level and main-lobe width, where I_0 is a first-order modified Bessel function.

$$w(n) = \frac{I_0\left(\beta\sqrt{1 - \left(\frac{2n}{M} - 1\right)^2}\right)}{I_0(\beta)}$$

II. Infinite Impulse Response (IIR)

The exponential decay from feedback causes the infinite response of IIR filters to remain non-zero. A convolution sum connects their input/output signals, and they can be digitally implemented with feed-forward and feedback coefficients.

IIR can be implemented using a variety of structures, including Lattice, Direct-Form-I, Direct-Form-II, and Lattice Ladder. It includes numerator and denominator coefficients. The Direct-Form-I realization is shown in Figure 2. Digital IIR filters are derived from traditional analog designs and have different properties, such as varying transition sharpness between the passband and stopband. Four different types of IIR filters can be used in this work:

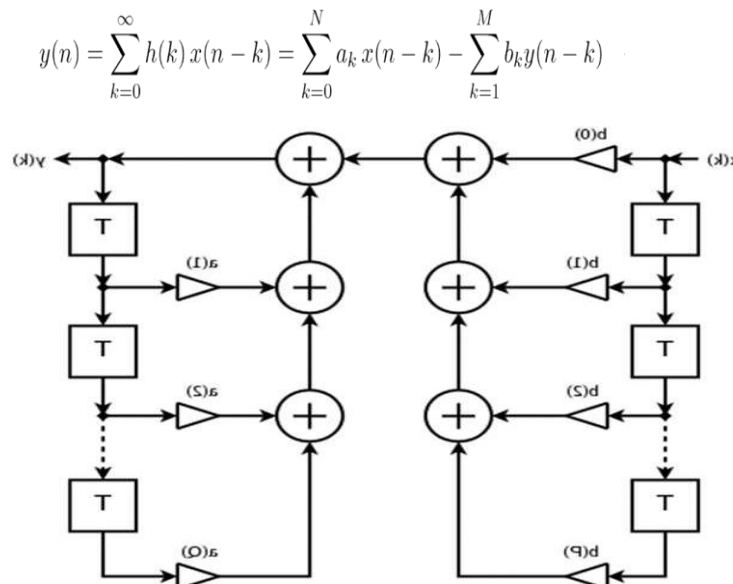


Fig. 2, Architecture of IIR filters

1. Butterworth filter with homogeneous response across all frequencies and a steady drop from the cut-off frequency.
2. The Chebyshev filter features a harder roll-off than Butterworth, a passband with equiripple magnitude response, and a stopband with a magnitude response that decreases monotonically.
3. Chebyshev II filter Similar to Chebyshev, but with a drop in the stopband rather than the passband.
4. Elliptic filter with the sharpest transition between the passband and the stopband and equiripples in both.

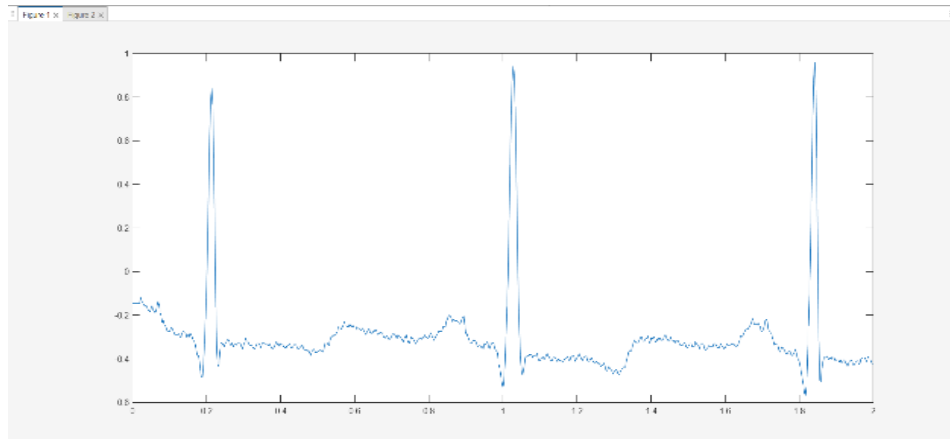


Fig. 3, ECG signal without noise signals

Plotting of the initial ECG signal took place across about two seconds. The most prominent aspects of the ECG are the acute vertical spikes, or QRS complexes, which are indicative of ventricular depolarization. The baseline fluctuations and minor elements like the P-wave and T-wave are easier to see in this recording since it clearly keeps the essential morphological components, unlike the noisy version of the signal. Compared to the noisy ECG, the amplitude is steady and constant with less interference, showing a clearer depiction appropriate for clinical interpretation.

III. PERFORMANCE ANALYSIS AND RESULTS

Circuits that filter signals tainted by random noise and additional sine waves are shown using modeling and practical application. The study is to compare the effectiveness of various digital filter types, windowing strategies, and filter orders in order to assess how well they suppress noise and preserve signals. To develop and simulate the filters, MATLAB and the Xilinx ISE develop Suite were utilized. The designs were then loaded onto the hardware by JTAG programming and the Xilinx System Generator. MATLAB Hardware Co-Simulation was then utilized to display and examine the outcomes, guaranteeing a strong correlation between the simulated and real-world outputs. This dual strategy demonstrates how accurate and dependable digital filters are in real-time situations. In addition to validating theoretical concepts, the methodology shows that they are feasible for use in communication and biomedical applications where signal integrity is crucial. such as higher-order FIR and IIR designs, which provide exceptional noise removal capabilities while maintaining crucial waveform properties.

A. Hardware Simulation in MATLAB and Vivado

An ECG signal with noise is shown by a waveform in Fig.5. The heart's ventricular depolarization phase is shown by the QRS complexes, which are represented by the abrupt upward deflections. Random high-frequency noise and baseline fluctuations, which can be viewed as irregular variations around the primary waveform, contaminate the signal, which appears relatively flat between these peaks. Digital filtering (FIR/IIR) is used to suppress undesired components while maintaining the diagnostic morphology of the ECG signal since the noise makes it harder to see smaller features like the P-wave and T-wave.

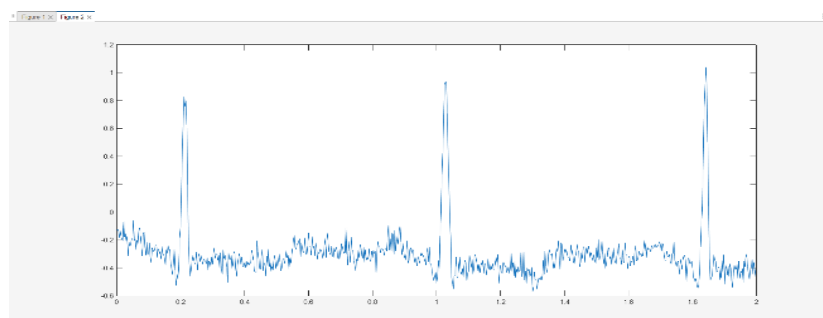


Fig.5, ECG signal with noise

A FIR filter design's structural implementation following synthesis in a hardware description environment. In the Fig.5, the convolution action necessary for FIR filtering is performed by an extended network of interconnected logic

blocks, multipliers, adders, and registers. A filter tap are associated with step, where the input signal sample is multiplied by the corresponding coefficient and then added to the output of subsequent taps. In the graphic, the dense interconnections show how data channels are routed to accomplish simultaneous computations, while the regular cascade structure shows the sequential ordering of these filter taps.

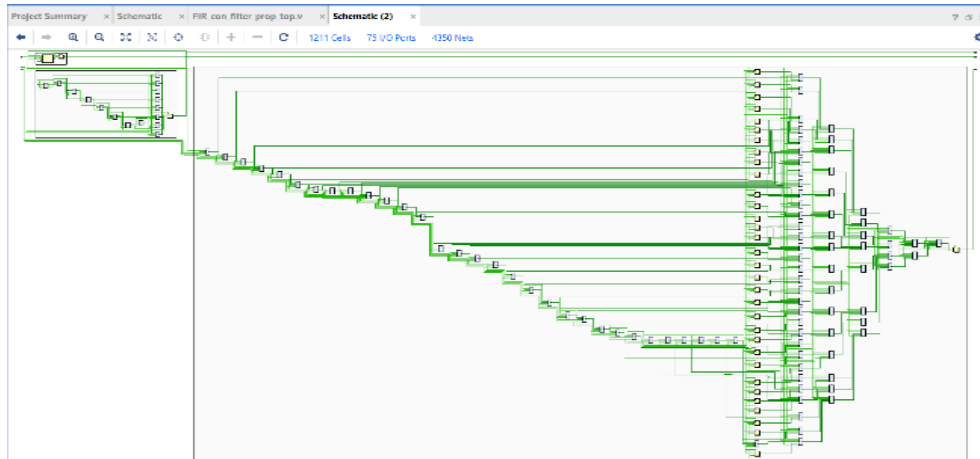


Fig. 6, Hardware schematic of the proposed design in Vivado

The Control signals are also incorporated into the architecture for synchronization, guaranteeing proper data flow across the pipeline. All things considered, this Fig.6, shows how a high-level FIR filter design written in Verilog or VHDL is translated into low-level logic circuits, which qualifies it for real-time signal processing applications like the reduction of noise in ECG signals. High-speed signal processing is made possible by the filter's pipelined data processing, which is illustrated by the cascading and parallel connections. For real-time applications where precise filtering and minimal latency are crucial, like ECG signal augmentation, this type of design is especially well-suited.

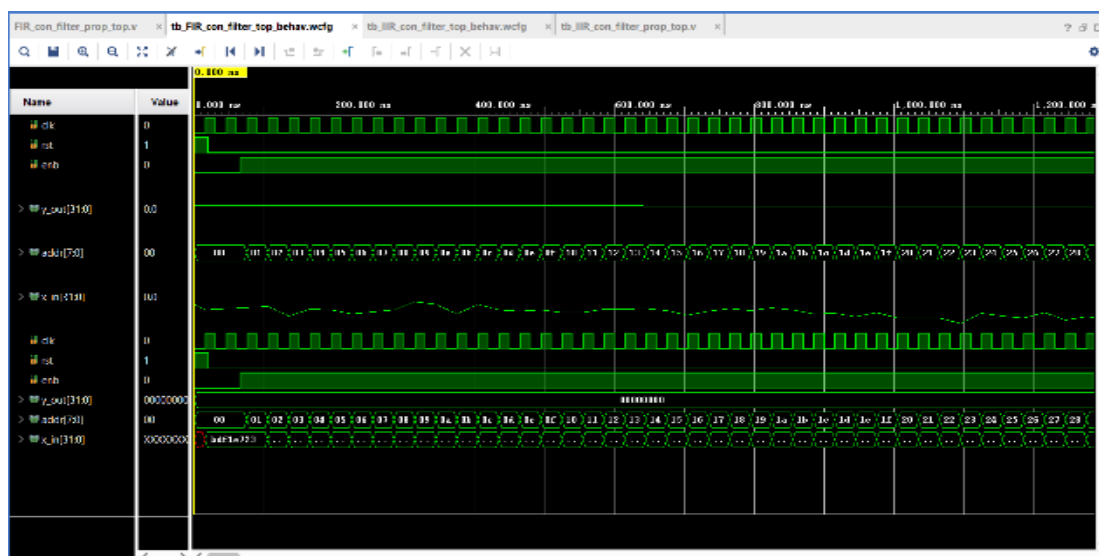


Fig.7, Simulation waveform of FIR filter

The Fig.7, shows the FIR filter design's simulation results following behavioral verification with Verilog HDL. To ensure correct sample fetching from memory, the counter generates matching address values (addr[7:0]) that are sequentially read with the input signal (x_in[31:0]). The filtered response is displayed in the output signal (y_out[31:0]), where the relevant signal is retained and noise components are reduced. The system's synchronization and operation are managed by the clock (clk), reset (rst), and enable (enb) signals; the reset guarantees correct initialization at the beginning of the simulation. The waveform confirms that the FIR filter architecture functions as expected by showing proper address creation, data flow, and output calculation.

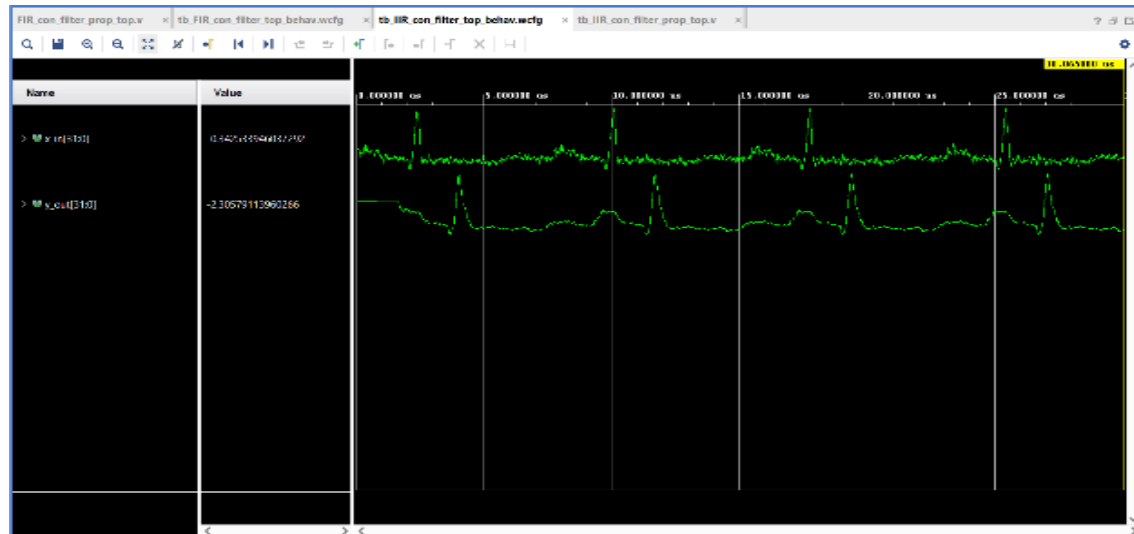


Fig.8, Simulation waveform of IIR filter

The Fig.8, shows the IIR filter design's simulation results, which were acquired by behavioral verification and Verilog HDL implementation. The top trace represents the input ECG signal (x_in[31:0]), which includes undesired noise in addition the desired information. The bottom trace shows the filtered output signal (y_out[31:0]), where the primary QRS complexes and overall waveform morphology are retained but noise components are much reduced. The output shows fewer baseline variations and high-frequency interference, but the periodic peaks in the input are still discernible. The filter's real-time efficiency is demonstrated by the timing scale, which is measured in microseconds. This qualifies the concept for use in biomedical settings where precise and instantaneous ECG analysis is necessary.

Table. 1, Shows the utilization of Resource

Resource	Estimation	Available
LUT	28739	230400
LUTRAM	2997	101760
Flip Flop	28729	460800
BRAM	0.50	312
I/O	75	360

The resource usage of the suggested filter design on FPGA is shown in this table.1. The architecture ensures efficient logic and memory consumption by requiring only a small percentage of the available resources, 28,739 LUTs and 2,997 LUTRAMs. Additionally, it uses 28,729 Flip-Flops, which is comfortably within the 460,800 capacity, and just 0.50 BRAM blocks out of 312—both of which demonstrate extremely low memory use. In addition, 75 of the 360 available I/O pins are utilized, indicating moderate demand for interface. The approach is scalable, resource-efficient, and appropriate for real-time biomedical signal. applications like ECG filtering, according to these results.

IV. CONCLUSION

The design and implementation of high-performance FIR and IIR digital filters for noise reduction and ECG signal improvement are demonstrated in this work. The efficiency of IIR filters based on elliptic designs and FIR filters using functions in eliminating high-frequency interference and baseline drift while maintaining diagnostic features was assessed. Hardware validation using Vivado and Verilog synthesis guaranteed real-time feasibility, while MATLAB simulations validated precise filter performance. The findings demonstrate that, at lower orders, IIR filters yield higher computational efficiency, but FIR filters preserve waveform integrity with linear phase characteristics. Analysis of resource usage revealed that the design is hardware-efficient and scalable, which qualifies it for real-time biomedical applications. With clearer ECG waveforms, this method facilitates wearable technology, telemedicine platforms, and accurate diagnosis.

REFERENCES

- [1]. Yousif Samer Mudhafar and Saif Hasan Abdalnabi “*High Performance FIR and IIR Filters Based on FPGA for 16 Hz Signal Processing*” 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA),Iraq, 2023.
- [2]. Muhammad N. Khan, Syed K. Hasnain and Mohsin Jamil, “*Digital Signal Processing: A Breadth-First Approach*” River Publishers, Netherlands, 2016.
- [3]. Uwe Meyer-Baese, “*Digital Signal Processing with Field Programmable Gate Arrays*”, 2001.
- [4]. B. Rashidi, B. Rashidi and M. Pourormazd, “*Design and implementation of low power digital FIR filter based on low power multipliers and adders on xilinx FPGA,*” 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, India, 2011, pp. 18-22.
- [5]. Thakur, Rakhi & Khare, Kavita. (2013). “*High Speed FPGA Implementation of FIR Filter for DSP Applications.*” International Journal of Modeling and Optimization. 92-94.
- [6]. S. M. Rabiul Islam, R. Sarker, S. Saha and A. F. M. Nokib Uddin, “*Design of a programmable digital IIR filter based on FPGA,*” 2012 International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, Bangladesh, 2012, pp. 716-721.
- [7]. G. Jinding, H. Yubao and S. Long, “*Design and FPGA Implementation of Linear FIR Low-pass Filter Based on Kaiser Window Function,*” 2011 Fourth International Conference on Intelligent Computation Technology and Automation, Shenzhen, China, 2011, pp. 496-498.
- [8]. Dixit, H. V., & Gupta, V. (2015, January). “*IIR filters using Xilinx System Generator for FPGA implementation.*” In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 640-644). IEEE.
- [9]. Dietrich Schlichthärle, “*Digital Filters Basics and Design*”, 2nd edition, 2011.
- [10]. R. A. Barapate, and J. S. Katre, “*Digital Signal Processing*”, Second Edition, Tech-Max Publication Pune, India, 2007.
- [11]. Juan Jose Rodriguez Andina, Eduardo de la Torre Arnanz, Maria Dolores Valdes , “*FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics*”, 1st edition, Taylor & Francis Group, USA, 2017.