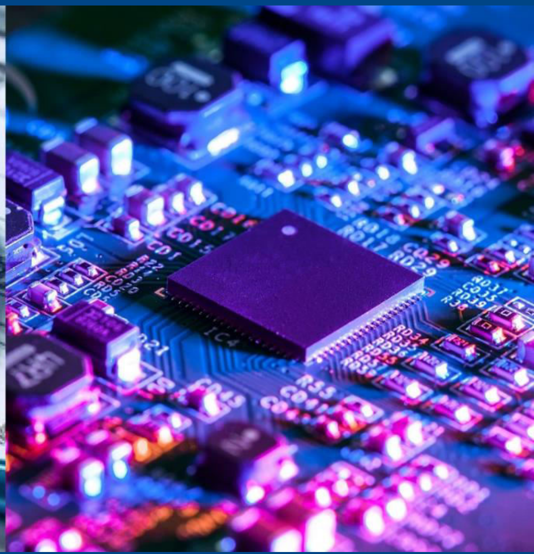
ISSN: 2582-7219

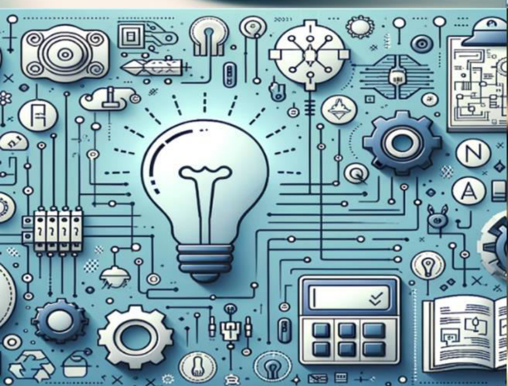




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 9, Issue 3, March 2026

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Design & Implementation of FIR Filter using Verilog

B.Nagasharmi[1], G.Jeevan Sai[2], P.Chandra Adithya[3], Shaik Imran[4], K.Sowmya Sri [5]

U.G. Students, Department of ECE, SVIET Engineering College, Nandamuru, Pedana, Andhra Pradesh, India[1-4]

Assistant Professor, Department of ECE, SVIET Engineering College, Nandamuru, Pedana, Andhra Pradesh, India[5]

ABSTRACT: Digital Signal Processing (DSP) is essential in modern systems such as communication, audio/video processing, and biomedical applications. Digital filters are used to remove noise and enhance signal quality. Among them, Finite Impulse Response (FIR) filters are widely preferred due to their stability, linear phase response, and ease of hardware implementation. This project focuses on the design and implementation of an FIR filter using Verilog Hardware Description Language (HDL).The FIR filter is developed based on its mathematical model, where the output is a weighted sum of present and past input samples. The design includes delay elements, multipliers, and adders, modelled using RTL techniques in Verilog. The system is modular and suitable for FPGA or ASIC implementation. Simulation is performed to verify functionality and performance, ensuring accurate signal processing. This project highlights practical DSP implementation using hardware design. It also covers VLSI concepts such as RTL coding and verification. The design can be further improved with optimization techniques, making it suitable for real-time applications.

KEYWORDS: Digital Signal Processing (DSP), FIR Filter, Verilog HDL, RTL Design, FPGA, ASIC, Digital Filters, Signal Processing, VLSI Design, Simulation

I. INTRODUCTION

Digital Signal Processing (DSP) is a branch of engineering and mathematics that focuses on manipulating signalssuch as sound, images, or sensor data—after they have been converted into a digital format. While natural signals are analog (continuous waves), computers and processors can only work with digital (discrete numbers). DSP provides the tools to analyze, filter, and compress these numbers to improve the quality or efficiency of the information.

Simply it is the mathematical manipulation of an information signal to modify or improve it in some way. It is characterized by the representation of discrete time, discrete frequency, or other discrete domain signals by a sequence of numbers or symbols and the processing of these signals. In the real world, most signals are analog, meaning they are continuous and vary constantly—like the sound of a human voice or the temperature throughout a day. For a computer or a specialized DSP chip to work with these signals, they must first be converted into a digital format through a process called analog-to-digital conversion, which involves sampling the signal at specific intervals and quantizing those values into binary code.

Paper is organized as follows. Section II describes FIR filter using Verilog modelling style. The simulation diagram represents the final waveform of the FIR filter using Verilog. After Simulation of code, the filter is showed how the present value of the input signal is going too multiplied with the past value of the impulse response. Section III. Section IV presents experimental results showing simulation result of FIR filter. Finally, Section V presents conclusion.

II. RELATED WORK

Digital signal processing (DSP) emerged in the late 1970s and early 1980s as specialized hardware designed to overcome the speed limitations of general-purpose microprocessors [1]. The introduction of the Texas Instruments TMS32010 was a turning point, as it utilized a Harvard architecture and a dedicated hardware multiplier to perform complex mathematical operations in a single clock cycle[7]. This breakthrough moved DSP from theoretical mathematics into practical, real-time applications like telecommunications and industrial control systems, setting the stage for the digital revolution in consumer electronics. [8]

In recent decades, the evolution of DSP has shifted from standalone chips toward integration into complex systems-on-chip (SoCs) and parallel processing architectures[4]. The 1990s introduced floating-point processors and Very Long Instruction Word (VLIW) designs, which significantly boosted the efficiency of 2G and 3G mobile networks[2]. Today, traditional DSP functions are often integrated alongside CPUs, GPUs, and AI accelerators to handle massive data tasks like 5G beam forming and real-time video encoding, prioritizing high energy efficiency and extreme computational density [3].

A filter is a mathematical operation or system designed to suppress unwanted components or features from a signal, such as noise or specific interference, while allowing desired frequencies to pass through[2]. Unlike physical analog filters made of resistors and capacitors, a digital filter operates on a sequence of numbers using algorithms implemented in software or specialized hardware[3]. The primary function of a filter is to modify the signal's frequency spectrum, which is achieved by performing a series of multiplications and additions on the current and previous input samples, as well as previous output samples[10]. This process is generally categorized into two main types: Finite Impulse Response (FIR) filters, which are stable and have a linear phase, and Infinite Impulse Response (IIR) filters, which are more computationally efficient and mimic traditional analog filter behaviors[11].

The impulse response of a system is the output it produces when presented with a very brief input signal called a unit impulse or "Dirac delta" function[5]. This theoretical input is often representing an infinitely sharp spike of energy. Because an impulse contains all possible all of frequencies in equal measure, the way a system reacts to it reveals its unique "DNA"—it shows exactly how the system modifies, delays, or decays any signal passing through it. Mathematically, the impulse response is the fundamental characterization of a Linear Time- Invariant (LTI) system, providing a complete description of its filtering properties and stability in the time domain[6].

The significance of the impulse response lies in a mathematical operation called convolution, which allows engineers to predict how a system will react to any arbitrary input signal[7]. By knowing the impulse response, you can calculate the output for any complex sound or data stream simply by "combining" that input with the system's unique signature. In practical terms, this is how digital filters are designed; for example, a Finite Impulse Response (FIR) filter's coefficients are literally the samples of its desired impulse response[6]. Whether you are simulating the acoustics of a famous concert hall or designing a communication link for a satellite, the impulse response serves as the essential bridge between a theoretical system design and its real-world performance[10].

A filter is a mathematical operation or system designed to suppress unwanted components or features from a signal, such as noise or specific interference, while allowing desired frequencies to pass through. Unlike physical analog filters made of resistors and capacitors, a digital filter operates on a sequence of numbers using algorithms implemented in software or specialized hardware. The primary function of a filter is to modify the signal's frequency spectrum, which is achieved by performing a series of multiplications and additions on the current and previous input samples, as well as previous output samples. This process is generally categorized into two main types: Finite Impulse Response (FIR) filters, which are stable and have a linear phase, and Infinite Impulse Response (IIR) filters, which are more computationally efficient and mimic traditional analog filter behaviors.

The practical application of filtering is foundational to modern technology, enabling everything from the clarity of voice calls to the precision of medical instrumentation. By defining a "passband" for desired frequencies and a "stopband" for those to be rejected, engineers can isolate a specific radio station from a crowded broadcast spectrum or smooth out jagged data from a digital sensor. In a DSP system, the characteristics of a filter—such as its cutoff frequency, ripple, and roll-off—are determined by a set of coefficients. Because these coefficients can be changed instantly in code, digital filters offer a level of flexibility and adaptive power that is impossible to achieve with fixed hardware, allowing a single device to dynamically adjust its filtering strategy based on changing environmental conditions.

The importance of Finite Impulse Response (FIR) filters in digital signal processing stems primarily from their inherent stability and their ability to provide a linear phase response. Because FIR filters do not use feedback—meaning the output is calculated only from current and previous input samples—they are guaranteed to be stable and will never oscillate uncontrollably, regardless of the coefficients used. The linear phase property is particularly critical in applications like digital communications and high-fidelity audio, as it ensures that all frequency components of a signal are delayed by the exact same amount of time. This prevents "phase distortion," which would otherwise smear the signal in the time domain, making FIR filters indispensable for maintaining the shape of pulses in data transmission or the clarity of transients in music.

Furthermore, FIR filters offer a level of design flexibility and simplicity that makes them the preferred choice for many modern embedded systems. They can be easily designed to meet specific frequency requirements using straightforward windowing methods or frequency sampling techniques. While they often require more computational power (more "taps") than their IIR counterparts to achieve a sharp cutoff, the rise of powerful, dedicated DSP hardware has made this less of a constraint. Their non-recursive structure also makes them highly resistant to the effects of finite word-length and quantization errors, meaning they perform more predictably when implemented on fixed-point processors. This combination of reliability, phase integrity, and robust performance makes FIR filters a cornerstone of modern pulse shaping, equalization, and noise reduction systems.

In many real-time signal processing applications, there is a critical need for efficient hardware implementations of digital filters that can operate at high speeds with deterministic latency. Software-based implementations on general-purpose processors often suffer from sequential execution bottlenecks, limiting their utility in high-frequency domains. This project addresses the need for a hardware-accelerated solution by implementing a parallelized Direct Form FIR architecture in Verilog, offering a pathway to high-throughput signal processing.Therefore there is need for verilog implementation of FIR filter so that it can be implemented through FPGA boards for hardware systems for faster results and future scope, Hence this project submits the solution for this hardware implentation issue.

III. METHODOLOGY

The project begins with understanding the basic concepts of Digital Signal Processing and FIR filter design. The filter specifications such as filter order and coefficients are determined based on requirements. The FIR filter mathematical model is analysed, where output is calculated as a weighted sum of input samples.

The design architecture is developed using delay elements, multipliers, and adders. The entire system is implemented using Verilog HDL with RTL modelling. The design is written in a modular and scalable manner for better flexibility. Simulation is carried out using standard tools to verify functional correctness. Test cases are applied to validate input-output response. Finally, the design is analysed for performance and suitability for FPGA or ASIC implementation.The motivation for this project is rooted in the critical need for high-performance, real-time signal processing that transcends the limitations of traditional software-based systems. While software implementations on general-purpose processors are often hindered by sequential execution bottlenecks and inconsistent latency, a hardware-accelerated approach using Verilog allows for massive parallelism and deterministic performance.

This is particularly essential in modern high-bandwidth domains such as 5G communications, radar, and medical imaging, where signals must be processed with extreme speed and reliability. By implementing a 4-tap Direct Form FIR filter, this project bridges the gap between abstract DSP theory—such as Z-transforms and discrete convolution—and practical VLSI design involving registers and Furthermore, the project highlights the inherent advantages of Finite Impulse Response filters, specifically their absolute stability and linear phase characteristics, which are vital for preserving the integrity of data in communication and audio applications. Using a hardware-centric approach on an FPGA provides a "sweet spot" in the design space, offering significantly higher throughput than microcontrollers while maintaining a level of reconfigurable flexibility that fixed ASICs lack. Ultimately, developing a dedicated hardware block for functions like the Moving Average filter offloads the main processor and enhances overall system efficiency, providing a foundational architecture that is scalable for next- generation signal processing demands.

Once a signal is digitized, it is processed by systems characterized by specific mathematical properties.

Linear Time-Invariant (LTI) Systems: Most DSP systems, including FIR filters, are LTI, meaning their behavior does not change over time and they follow the principle of superposition.

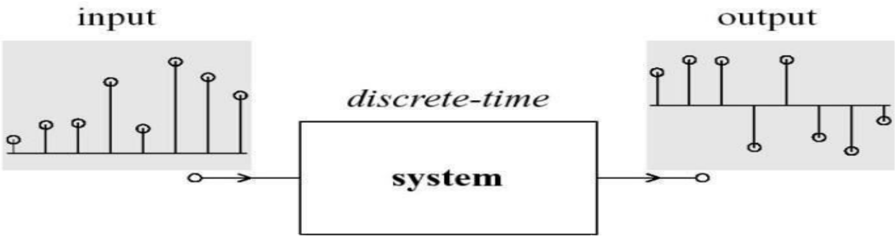
Convolution: This is the fundamental mathematical operation behind filtering, where an input signal is combined with a system's impulse response to produce an outputFiltering is the core application of DSP, used to remove unwanted components or noise from a signal.

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI: 10.15680/IJMRSET.2026.0903229

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Finite Impulse Response (FIR): These filters have a finite duration response and are known for absolute stability and linear phase characteristics defined by an impulse response that settles to zero within a finite number of sample intervals. Unlike filters that use feedback, the FIR architecture is strictly non-recursive, meaning the current output depends only on a finite set of current and previous input values. This structural characteristic ensures that the filter is inherently stable and cannot oscillate uncontrollably, regardless of the coefficients or input dat used. One of its most significant advantages in signal processing is the ability to achieve linear phase response

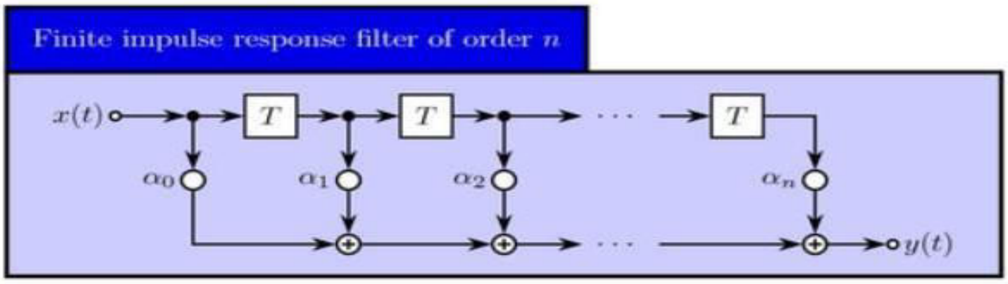


Fig a: Block diagram representation of FIR filter

IV. EXPERIMENTAL RESULTS

Figures (b) shows the simulation results of the 4-tap Direct Form FIR filter were obtained using waveform analysis. The clock signal exhibits stable periodic behavior, ensuring proper synchronous operation of the design. The input signal xinx_{in}xin is applied sequentially, and the delayed versions of the input are observed across the register stages x_reg[0]x_reg[0]x_reg[0] to x_reg[3]x_reg[3]x_reg[3], confirming correct shift register operation.

The filter coefficients h[0]h[0]h[0] to h[3]h[3]h[3] remain constant throughout the simulation, indicating proper coefficient assignment. As the input propagates through the delay elements, the output yout y_{out}yout is generated as the accumulated weighted sum of present and past inputs.The output waveform shows a gradual increase corresponding to the convolution operation, validating the correct functionality of the FIR filter. The results confirm that the design produces accurate filtered outputs with proper timing and synchronization, demonstrating for the successful hardware implementation of the FIR filter.

Fig b:- Simulation results of FIR Filter Using Verilog

V. CONCLUSION

We have implemented the Direct Form FIR filter successfully demonstrates the implementation of DSP concepts using Verilog HDL. The design achieves high-speed, single-cycle throughput with improved efficiency over processor-based methods. Simulation results confirm accurate and reliable performance. The design is modular and scalable, making it suitable for real-time applications and further enhancements in digital filter design.

REFERENCES

[1] Oppenheim, A. V., & Schafer, R. W. (2010). Discrete-Time Signal Processing. Pearson.
[2] Proakis, J. G., & Manolakis, D. G. (2006). Digital Signal Processing. Prentice Hall.
[3] IEEE Standard for SystemVerilog (IEEE Std 1800-2012).
[4] Design and realization of FIR digital filters based on MATLAB | IEEE Conference Publication IEEE Xplore https://share.google/iBy2900zxLsrkMIEa
[5] Aliparast, Peiman, Ziaadin D. Koozehkanani, and FarhadNazari. "An Ultra High Speed Digital 4-2 Compressor in 65-nm CMOS."International Journal of Computer Theory & Engineering 5.4 (2013).
[6] N. Weste and David Harris, "CMOS VLSI Design- A Circuits & System Perspective", Pearson Education, 2008.
[7] ChandraMohan U, "Low Power Area Efficient Digital Counters", Proceedings of the 7th VLSI Design and Test Workshops, VDAT, August 2003.
[8] Narendra C P & Ravi K M Kumar, "Efficient Comparator based Sum of Absolute Differences Architecture for Digital Image Processing Applications", Foundation of Computer Science, New York, USA, International Journal of Computer Applications,
[9] 96(4):17-24, June 2014.
[10] W.-C. Yeh, "Arithmetic Module Design and its Application to FFT," Ph.D. dissertation, Dept. Electron. Eng., National Chiao-Tung University, ,Chiao-Tung, 2001.
[11] R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-of-products," in Proc. Asilomar Conf. Signals, Syst. Comput., Pacific Grove, Washington, DC, 2003, pp. 867–872.
[12] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. Oxford: Oxford Univ. Press, 2000