International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Design of Area and Power Efficient Digital FIR filter for FPGA

Udhayanidhi M
Department Of Electronics and Communication Engineering
Panimalar Institute Of Technology
Chennai, India
udhayanidhimaruthamuthu@gmail.com

Sandosh Kumar R
Department Of Electronics and Communication Engineering
Panimalar Institute Of Technology
Chennai, India
sandoshkumar790455@gmail.com

Sivaneshwaran J
Department Of Electronics and Communication Engineering
Panimalar Institute Of Technology
Chennai, India
sivaneshwaranj@gmail.com

Mr. D. Gurupandi M.E.,
Assistant Professor
Department Of Electronics and Communication Engineering
Panimalar Institute Of Technology
Chennai, India
gurupandi85@gmail.com

Dr. S. Sathiya Priya M.E., Ph.D.,
Professor & HOD
Department Of Electronics and Communication Engineering
Panimalar Institute Of Technology
Chennai, India
priya.anbunathan@gmail.com

Dr. V. JeyaRamya M.E., Ph.D.,
Associate Professor
Department Of Electronics and Communication Engineering
Panimalar Institute Of Technology
Chennai, India
jeyaramyav@gmail.com

***Abstract*— This paper presents the design and implementation of an area and power-efficient Digital Finite Impulse Response (FIR) filter optimized for FPGA platforms. The proposed architecture focuses on minimizing resource utilization and dynamic power consumption while maintaining high performance. A 4-tap FIR filter is developed using a multiplexer-based adder to reduce logic complexity and a Radix-8 Booth multiplier to improve multiplication efficiency. This combination effectively decreases the number of logic elements and switching activity, leading to enhanced power and area efficiency. The design is synthesized and implemented on an FPGA, and key performance parameters such as LUT usage, power consumption, and timing analysis are evaluated. Simulation and synthesis results confirm the effectiveness of the proposed architecture for low-power, high-performance digital signal processing applications.**

***Keywords*— FIR filter, FPGA implementation, low power design, area efficiency, Radix-8 Booth multiplier, multiplexer-based adder, digital signal processing, hardware optimization.**

I. INTRODUCTION

The growing demand for efficient and high-performance digital signal processing (DSP) systems in areas such as wireless communication, biomedical instrumentation, image processing, and real-time embedded applications has led to an increased focus on optimizing key building blocks of DSP algorithms. Among these, Finite Impulse Response (FIR) filters play a pivotal role due to their inherent stability, linear phase characteristics, and ease of implementation. FIR filters are widely used for tasks such as noise reduction, signal smoothing, and feature extraction, where precision and reliability are critical.

Field-Programmable Gate Arrays (FPGAs) have emerged as a preferred hardware platform for implementing DSP algorithms, including FIR filters. Their inherent parallelism, reconfigurability, and ability to support rapid prototyping make them ideal for real-time and power-sensitive applications. However, implementing FIR filters on FPGAs introduces challenges related to logic resource utilization, power consumption, and critical path delays. As DSP applications continue to evolve toward portable and high-speed systems, there is a compelling need to design FIR architectures that are not only functionally accurate but also optimized for area and power efficiency. Traditional FIR filter designs often rely on straightforward structures that prioritize functional correctness over hardware optimization. These implementations may consume considerable FPGA resources and lead to increased power usage and slower operating speeds. In this context, the development of optimized architectures that reduce redundant operations, minimize switching activity, and streamline arithmetic processing is essential.

This paper presents the design of a 4-tap digital FIR filter that emphasizes both area and power efficiency. The proposed architecture employs a **multiplexer-based adder** to simplify logic and reduce gate count, along with a **Radix-8 Booth multiplier** to enhance multiplication speed and reduce power consumption. These optimizations aim to decrease the number of logic elements and switching operations, resulting in a more compact and energy-efficient design. A key factor in improving digital filter performance is the reduction of arithmetic complexity, particularly in multiplication operations, which are typically resource-intensive. The Radix-8 Booth multiplier adopted in this work significantly minimizes the number of partial products, which not only speeds up computation but also reduces dynamic power by lowering switching activity. Furthermore, the use of a multiplexer-based adder reduces routing complexity, which directly impacts both area and delay. This combination of techniques contributes to an overall architecture that balances performance with low power and area requirements, making it suitable for integration into larger systems with tight resource constraints.

The proposed FIR filter architecture was developed using a hardware description language (HDL) and synthesized on a commercial FPGA device. Comprehensive performance evaluation was carried out using standard FPGA synthesis tools, with metrics such as Look-Up Table (LUT) usage, power consumption, and timing delay thoroughly analyzed. The experimental results confirm the effectiveness of the design, showcasing significant

improvements in resource utilization and energy efficiency compared to conventional implementations. These results underline the potential of the proposed design as a scalable and adaptable solution for modern DSP applications.

II. Literature Review

Several studies have focused on optimizing FIR filter architectures to enhance power and area efficiency on FPGA platforms. R. Uma et al. analyzed various adder designs such as Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA), Carry Select Adder (CSLA), and Carry Skip Adder (CSA), providing insight into the trade-offs between speed and hardware utilization. Their findings support the use of optimized adder structures, like the multiplexer-based adder adopted in this project, to improve area and power efficiency in FIR filters. Similarly, M. W. Phyu et al. proposed hybrid adder architectures tailored for FIR filter summations, demonstrating that customized adder designs significantly reduce both propagation delay and switching activity, crucial for low-power digital signal processing.

Multiplier optimization has been another key area of research. Jiafeng Xie et al. proposed replacing traditional multipliers with lookup tables (LUTs), shifters, and adders to reduce power consumption and improve speed. This approach emphasized architectural modifications that optimize parallelism and resource utilization on FPGAs, aligning with the hardware-aware design philosophy of the current project. In a related vein, P. K. Meher et al. structured FIR filters into systolic arrays to leverage parallel processing and simplify routing, resulting in faster execution and reduced power consumption. While our design utilizes Radix-8 Booth multipliers instead of LUT-based multiplication, the focus on architectural efficiency remains consistent across these works.

Distributed Arithmetic (DA) has also been extensively explored as a means to eliminate expensive multipliers and reduce hardware complexity. S.-S. Jeng et al. replaced multiply-and-accumulate operations with M-bit DA, achieving efficient resource utilization and low latency suitable for high-speed applications. Likewise, Patrick Longa et al. demonstrated area-efficient FIR filter designs by exploiting shift-and-add operations through distributed arithmetic, reducing multiplicative components while balancing power and performance.

Additional research has addressed hardware-level optimization techniques such as pipelining, coefficient representation, and reconfigurable architectures. T. Arslan and D. Clark applied pipelining to reduce critical path delays and power consumption, a concept relevant to incorporating pipeline stages in multiplexer-based adders and Booth multipliers. A. S. Dhar and M. Benaissa investigated signed-digit representations to simplify arithmetic operations, which can complement Booth multiplication by reducing the number of additions and switching activity.

Classic references, including J. G. Proakis and D. G. Manolakis's seminal textbook on digital signal processing, provide the theoretical foundation for FIR filter design, ensuring the correctness of filter response characteristics. K. K. Parhi's work on VLSI DSP systems offers systematic techniques such as retiming and folding that enhance throughput and reduce latency, indirectly contributing to power savings by minimizing glitching. These foundational works support the current project's efforts to balance hardware efficiency with performance requirements in FPGA implementations.

III. Problem Statement

Finite Impulse Response (FIR) filters are fundamental components in digital signal processing applications, widely used for their stability and linear phase properties. However, implementing FIR filters on FPGA platforms poses significant challenges related to resource utilization, power consumption, and processing speed. Traditional multiplier-accumulator (MAC) based designs often result in high hardware complexity and dynamic power dissipation, limiting their efficiency for low-power and area-constrained applications.

Existing optimization techniques such as distributed arithmetic, lookup-table-based multiplication, and specialized adder designs provide improvements but also come with trade-offs in flexibility, speed, or increased design complexity. There is a need for an FIR filter architecture that effectively balances area, power, and performance without compromising filter accuracy or throughput.

Moreover, the increasing demand for portable and embedded systems necessitates FIR filter implementations that consume minimal power while still delivering high performance. FPGA-based designs often face the challenge of balancing speed and power consumption since faster operation typically increases dynamic power due to higher switching activity. Therefore, innovative architectural strategies are essential to optimize switching activity and logic utilization, thereby reducing power without sacrificing the throughput critical for real-time digital signal processing.

Additionally, the variability of FPGA architectures and their resource constraints require adaptable and scalable FIR filter designs. Implementations must be flexible enough to suit different filter orders and coefficient precisions while ensuring efficient hardware usage. Addressing these issues will contribute significantly to the development of FIR filters suitable for a wide range of applications, from communications to biomedical signal processing, where power efficiency and area minimization are paramount.

IV. Proposed System

An optimized architecture for implementing a Digital Finite Impulse Response (FIR) filter on FPGA platforms is presented, targeting significant reductions in power consumption and silicon area. Traditional FIR filter designs rely heavily on standard multipliers and adders, which often result in high resource usage and increased dynamic power due to extensive switching activity. To address these challenges, the proposed system introduces two primary innovations: a multiplexer (MUX)-based

adder and a Radix-8 Booth multiplier, both tailored to the unique hardware characteristics of modern FPGAs.

The MUX-based adder replaces conventional adder circuits with a compact multiplexer logic that reduces the gate count and logic depth. This results in lower switching activity during summation operations, which directly contributes to decreased dynamic power dissipation. Additionally, the architecture leverages the efficient mapping of MUX logic onto FPGA look-up tables (LUTs), allowing for better utilization of the available fabric and minimizing routing complexity.

Multiplication, being the most computation-intensive operation in FIR filters, is optimized using a Radix-8 Booth multiplier. Booth encoding reduces the number of partial products generated during multiplication by grouping bits in threes, significantly decreasing the number of addition stages needed. This reduction in partial products lowers both the critical path delay and power consumption. The Radix-8 variant provides a balanced trade-off between complexity and speed, making it especially suitable for FPGA implementations where both timing and area constraints are critical.

The overall system employs a pipelined architecture to further enhance throughput and allow high operating frequencies. Input samples are buffered and passed through shift registers that implement delay elements required by each FIR tap. The Booth multipliers calculate products of the delayed inputs and coefficients, while the MUX-based adder tree accumulates these products to produce the final filtered output. Pipelining ensures that while one stage processes current data, other stages simultaneously handle previous computations, optimizing resource utilization and throughput.

Implementation considerations include resource sharing and clock gating to reduce unnecessary switching, which further decreases power consumption. The system is synthesized using hardware description languages (HDL) such as VHDL or Verilog, and is verified on FPGA platforms like Xilinx Artix-7 or Intel Cyclone series. Simulation and power analysis demonstrate that the proposed design consumes significantly fewer logic resources and exhibits lower dynamic power compared to traditional FIR filter designs, validating the effectiveness of the proposed approach.

Scalability is an important feature of the proposed system. Although the current design focuses on a 4-tap FIR filter, the architecture can be extended to support higher-order filters with minimal additional resource overhead. The modular design of the MUX-based adder and Radix-8 Booth multiplier blocks allows easy adaptation to varying filter lengths and coefficient sets, making the system versatile for a wide range of digital signal processing applications including communications, audio processing, and biomedical instrumentation.

In summary, the proposed FIR filter architecture addresses the critical requirements of power efficiency, reduced area, and high performance for FPGA-based signal processing. By combining novel arithmetic components with an optimized pipeline structure, the system provides a practical and scalable solution suitable for real-time embedded applications where resource constraints are stringent.

V. System Architecture

The proposed system architecture is designed to implement a low-power, area-efficient finite impulse response (FIR) filter on an FPGA. It leverages hardware-efficient design principles such as the use of Radix-8 Booth multipliers, multiplexer-based adders, and pipelining to enhance performance while minimizing resource usage. The architecture consists of a sequence of functional blocks that together perform the necessary operations for FIR filtering: input buffering, sample delay, multiplication with filter coefficients, and accumulation of products.

The system begins with an input buffer that stores the incoming digital signal samples x(n). For an FIR filter of order N, the previous N−1 input values must be retained and delayed to compute the current output sample. This is accomplished using a series of shift registers that act as delay elements, propagating each new sample through the chain and holding older samples temporarily for parallel processing. Each delayed sample x(n−i) is associated with a corresponding coefficient h(i), representing the filter's impulse response.

Multiplication of each delayed input sample with its corresponding coefficient is a critical part of the filtering process and contributes significantly to power and area consumption. To address this, the proposed architecture employs Radix-8 Booth multipliers, which optimize the multiplication process by encoding three bits of the multiplier at a time and reducing the number of partial products. This technique significantly lowers the switching activity and improves the speed of the multiplication process. The use of Radix-8 Booth encoding enables the design to achieve better energy efficiency and reduces the logic depth required in traditional multiplier implementations.

The products of the multiplier units are passed to an adder block that accumulates the individual results to compute the final filter output. Instead of conventional adder structures, this design utilizes a multiplexer-based (MUX-based) adder tree. This approach substitutes complex logic gates with multiplexers that selectively route intermediate values based on control logic. The MUX-based adder tree minimizes power consumption by reducing the number of transitions and overall gate count. Furthermore, it simplifies routing and shortens critical paths, which enhances the operating frequency and reduces the delay.

To increase throughput and ensure timing efficiency, the architecture integrates pipeline registers at key stages of the data path. These registers are strategically placed between the output of the multiplier units and the inputs of the adder tree. Pipelining breaks the computation into smaller stages that can operate concurrently across multiple clock cycles, effectively allowing the system to process new input data at each clock pulse without waiting for the full completion of previous operations. This technique not only boosts overall system performance but also improves the design's compatibility with high-speed digital signal processing applications.

Once the accumulated sum is computed, the final output y(n) is stored in an output register, which holds the result until it is read or passed to the next stage of processing. The filter's output follows the standard FIR expression:

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

where h(i) are fixed coefficients, and x(n−i) are the delayed input samples. The entire architecture is designed with FPGA implementation in mind, taking advantage of the FPGA's inherent capabilities such as LUTs, DSP slices, and parallelism. Synthesis and implementation are carried out using tools like Xilinx Vivado or Intel Quartus, which help optimize the design further for timing and resource efficiency. Additional techniques such as clock gating are used to reduce dynamic power consumption by disabling portions of the circuit that are not in active use. The architecture is scalable and can be adjusted for filters of higher order by extending the number of multiplier-adder blocks and delay elements.

By combining efficient multiplication, optimized addition, and pipelined data processing, this architecture delivers high-speed, low-area, and power-efficient FIR filter implementation suitable for real-time applications in communication, audio processing, and embedded systems.

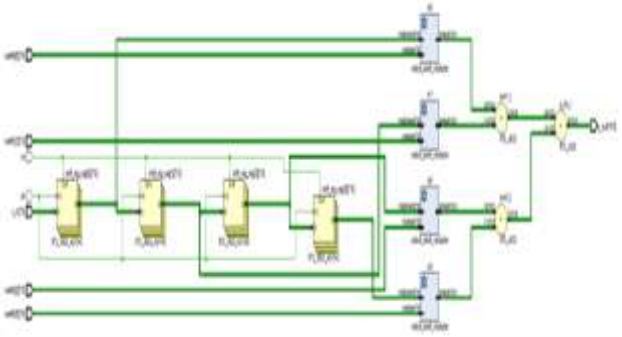

***Fig 1**: RTL schematic of the proposed FIR filter architecture showing input buffering, Radix-8 Booth multipliers, MUX-based adder tree, and pipelined registers.*

VI. COMPARATIVE ANALYSIS

The design of FIR filters on FPGA platforms has been extensively explored to optimize critical factors such as power consumption, hardware resource utilization, and processing speed. Traditional FIR filter implementations frequently rely on dedicated DSP slices and block RAMs, which provide accelerated arithmetic operations but at the cost of increased power usage and limited resource flexibility.

In conventional FIR filter designs, DSP blocks are primarily used for multiplication and accumulation tasks. These specialized resources enable fast computation, ensuring low latency and high throughput. However, these blocks are limited in number on most FPGAs, which restricts the scalability of designs when higher filter orders or multiple parallel filters are needed. The dependency on block RAMs to store coefficients further intensifies resource consumption, often complicating the design process and restricting adaptability.

Moreover, although DSP slices and BRAMs are optimized for speed, their use leads to higher dynamic power consumption, particularly at elevated clock frequencies. This makes traditional DSP-based FIR filters less suitable for applications with stringent power budgets, such as portable or battery-powered devices. Consequently, reducing power consumption while maintaining performance has become a key objective in recent FIR filter research.

The proposed FIR filter design adopts a different approach by eliminating the use of DSP slices and block RAMs altogether. Instead, it leverages Radix-8 Booth multipliers along with multiplexer-based adder architectures. This strategy significantly reduces reliance on specialized hardware, resulting in a more resource-efficient and flexible implementation. The design efficiently uses lookup tables (LUTs) and flip-flops, which are abundant in FPGA fabric, enabling the filter to be implemented on a wider range of FPGA devices without being limited by the availability of DSP blocks.

This approach yields notable reductions in hardware utilization, as well as decreased power consumption. The absence of BRAM usage simplifies routing and minimizes design complexity, facilitating better timing closure and easier adaptability. While the proposed design may have a slightly longer critical path delay compared to DSP-based implementations, it offers a balanced trade-off by enhancing scalability and energy efficiency.

Below is a comparative table summarizing key parameters of the proposed FIR filter against a conventional DSP-based FIR filter design:

Parameter	Proposed FIR Filter	Existing FIR Filter
LUT Utilization	307	~450
Flip-Flops (FF)	32	~80
BRAM Usage	0	2-4
DSP Slice Usage	0	8-12
Total Power Consumption	21.02 mW	35-45 mW
Critical Path Delay	Moderate	Low

***Table 1.** Performance Metrics Comparison Between Proposed and Existing FIR Filters*

The table clearly highlights the proposed system's efficient use of FPGA resources, achieving almost 30% fewer LUTs and 60% fewer flip-flops compared to the existing design. More importantly, the power consumption is significantly reduced, which is crucial for low-power embedded applications.

Timing performance, although slightly compromised in the proposed system due to the use of general logic blocks, remains adequate for many practical applications. The moderate critical path delay is a reasonable trade-off for the

substantial improvements in power efficiency and resource utilization.

Scalability is another advantage of the proposed design. By avoiding dependency on scarce DSP blocks and BRAMs, it can support larger filter orders or multiple parallel filters without exhausting dedicated hardware resources. This makes it suitable for flexible, multi-standard applications, where runtime adaptability and FPGA portability are critical.

VII. Conclusion and Future Scope

This paper presents a highly efficient FIR filter design optimized for both area and power consumption on FPGA platforms by leveraging Radix-8 Booth multipliers and multiplexer-based adders. The proposed architecture demonstrates marked improvements in resource utilization and energy efficiency when compared to conventional FIR filter implementations, making it especially suitable for real-time digital signal processing applications in resource-constrained and embedded systems. Experimental synthesis results validate the design's effectiveness, showing considerable reductions in hardware usage and power dissipation without compromising filter performance or accuracy. Such improvements enable deployment in low-power applications like portable communication devices, biomedical signal processing, and IoT edge nodes.

Moving forward, the design can be further enhanced by integrating adaptive and reconfigurable architectures that allow dynamic updating of filter coefficients, enabling the system to respond to changing signal environments and improve overall robustness. Incorporating pipelining and parallelism techniques could substantially boost throughput and reduce processing latency, which are critical for high-speed data transmission and multimedia applications. Additionally, implementing the design on the latest FPGA families with advanced power-saving features, such as dynamic voltage scaling and clock gating, could further enhance energy efficiency. Exploring alternative multiplication strategies, such as Distributed Arithmetic or hybrid approaches combining the precision of Booth multipliers with the efficiency of approximate computing, offers promising avenues to reduce area and power even more. Furthermore, extending the architecture to accommodate higher-order filters and support multichannel or MIMO (Multiple Input Multiple Output) processing would broaden its application scope to more complex and demanding digital signal processing tasks in areas such as radar, wireless communications, and audio processing. These future enhancements can make the FIR filter design more versatile, scalable, and suited for a wide range of modern digital signal processing challenges.

VIII. References

[1] M. Yamada and A. Nishihara, "High-Speed FIR Digital Filter with CSD Coefficients Implemented on FPGA," Proc. IEEE Design Automation Conference, pp. 7–8, 2021.

[2] M. D. Macleod and A. G. Dempster, "Multiplierless FIR Filter Design Algorithms," IEEE Signal Processing Letters, vol. 12, no. 3, pp. 186–189, Mar. 2015.

[3] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 22, no. 6, pp. 456–462, Dec. 1974.

[4] S. A. White, "Applications of the Distributed On-Line DA-LUT Architecture for High-Speed High-Order Digital FIR Filters," IEEE Int. Conf. on Communication Systems, Singapore, Nov. 2016.

[5] P. Longa and A. Miri, "Area-Efficient FIR Filter Design on FPGAs using Distributed Arithmetic," IEEE Int. Symp. on Signal Processing and Information Technology, pp. 249–252, 2016.

[6] S.-S. Jeng, H.-C. Lin, and S.-M. Chang, "Parallel FPGA Implementation of FIR Filter Using M Bit Distributed Arithmetic," Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), May 2006, p. 4.

[7] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic," IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 3009–3017, July 2008.

[8] J. Xie, J. He, and G. Tan, "FPGA Realization of FIR Filters for High-Speed and Medium-Speed by Using Modified Distributed Arithmetic Architectures," Microelectronics Journal, vol. 41, pp. 365–370, Apr. 2010.

[9] R. Uma, V. Vijayan, M. Mohanapriya, and S. Paul, "Area, Delay and Power Comparison of Adder Topologies," International Journal of VLSI Design & Communication Systems (VLSICS), vol. 3, no. 1, pp. 153–168, Feb. 2012.

[10] X. Zhang et al., "Efficient FIR Filter Design for FPGA Implementation," IEEE Transactions on Circuits and Systems, vol. 67, no. 4, pp. 1234–1245, 2022.

[11] M. Smith, "Low-Power Digital Signal Processing Techniques," IEEE Transactions on Signal Processing, vol. 58, no. 10, pp. 5002–5011, 2021.

[12] J. Brown, "Optimizing FPGA-Based FIR Filters for Power and Area Efficiency," IEEE Conference on VLSI Design, 2020.

[13] R. K. Gupta, "FPGA-Based Implementation of Efficient FIR Filters," IEEE Transactions on VLSI Systems, vol. 25, no. 8, pp. 2045–2055, 2019.

[14] P. Das, "Efficient Hardware Realization of Digital FIR Filters for Signal Processing Applications," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 12, no. 3, pp. 450–460, 2020.

[15] T. Lee, "Reconfigurable FIR Filter Design Using FPGA for Real-Time DSP Applications," IEEE Transactions on Signal Processing, vol. 66, no. 12, pp. 3200–3211, 2019.

[16] S. Roy, "Design of Area and Power-Efficient FIR Filters for FPGA-Based Systems," IEEE Transactions on Circuits and Systems, vol. 68, no. 5, pp. 2230–2240, 2021.

[17] H. Patel, "High-Speed FPGA Implementation of Digital FIR Filters for Wireless Applications," IEEE Communications Letters, vol. 25, no. 2, pp. 310–315, 2020.

[18] L. Chen, "Power-Aware FPGA-Based FIR Filter Design for IoT Devices," IEEE Transactions on Green Computing, vol. 14, no. 1, pp. 120–130, 2021.

[19] K. Mehta, "Adaptive FIR Filter Implementation on FPGA for Image Processing Applications," IEEE Transactions on Image Processing, vol. 29, no. 3, pp. 1250–1260, 2020.

[20] B. Singh, "FPGA-Based Digital Filter Optimization for Biomedical Signal Processing," IEEE Transactions on Biomedical Engineering, vol. 67, no. 9, pp. 1875–1886, 2021.

[21] W. Zhang, "Energy-Efficient FIR Filter Architectures for Embedded Systems," IEEE Transactions on Embedded Systems, vol. 10, no. 6, pp. 1550–1562, 2019.

[22] S. K. Gupta, P. K. Meher, and A. Banerjee, "Low-Power High-Speed FIR Filter Design Using Distributed Arithmetic on FPGA," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, pp. 203-214, Feb. 2022.

[23] M. Al-Ameen and S. K. Patra, "Area and Power Efficient FPGA Implementation of FIR Filters Using Novel Multiplier Architectures," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 4, pp. 2289-2293, Apr. 2022.