Journal of Electrical Engineering and Automation (ISSN: 2582-3051)
www.irojournals.com/iroeea/

Development and Realization of a Multi-Rate FIR Filter Utilizing Distributed Arithmetic on FPGA

Jerome Albert A.[1], Jenifer Gloria Daphne V.[2], Vigneshwaran S.[3], Mariammal K.[4]

Department of Electronics Engineering, Madras Institute of Technology, Chennai, India

Email: [1]jeromealbert10@gmail.com, [2]jeniferdaphne@gmail.com, [3]9111vigneshwaran@gmail.com, [4]mariammalkms@yahoo.co.in

Abstract

This research presents an efficient development of a multi-rate Finite Impulse Response (FIR) filter utilizing Distributed Arithmetic (DA) for digital signal processing (DSP) applications. The motivation behind this work lies in the increasing demand for high-performance filters in various multimedia and communication systems, where low power consumption and area-efficient designs are critical. The proposed FIR filter utilizes DA to minimize the resource requirements typically associated with conventional filter implementations while achieving comparable or improved performance metrics. A Look-Up Table (LUT) based approach is employed to compute filter outputs, allowing for rapid and efficient processing of input samples. Simulation results demonstrate that the multi-rate FIR filter significantly minimizes the computational complexity and enhances speed, Optimizing it for real-time performance in embedded systems. The findings emphasize the significance of integrating Distributed Arithmetic into FIR filter designs., paving the way for more advanced signal-processing solutions in VLSI architectures. This proposed design is implemented using Vivado version 2023 in Zynq UltraScale + MPSoCs family.

Keywords: FIR Filter, Distributed Arithmetic, VLSI, Multi-Rate Processing, Digital Signal Processing (DSP), Look-Up Table (LUT), Computational Efficiency, Real-Time Systems.

DOI: https://doi.org/10.36548/jeea.2024.4.006
Received: 02.01.2025, received in revised form: 24.01.2025, accepted: 05.02.2025, published: 17.02.2025

1. Introduction

Finite Impulse Response (FIR) filters are essential building blocks in digital signal processing (DSP). They are widely used for their stability, linear phase characteristics, and ability to meet various frequency response specifications. These filters play a crucial role in applications such as audio and video processing, telecommunications, and control systems, where precise signal manipulation is essential. The design of FIR filters allows to customize a frequency responses to specific requirements, enabling effective filtering of signals while preserving the integrity of the original data. In many real-time applications, the need for multi-rate systems arises, where signals are processed at different sampling rates. Multi-rate processing offers significant advantages, including reduced computational complexity and improved resource utilization. By employing variable sampling rates, systems can efficiently handle different signal bandwidths, leading to enhanced performance in applications such as data compression, voice coding, and image processing. The integration of multi-rate techniques in FIR filter design can result in significant improvements in system efficiency and effectiveness.

Distributed Arithmetic (DA) has emerged as a powerful technique for implementing FIR filters, offering several advantages over traditional filtering methods. Unlike conventional approaches that rely on multiplications and additions for each input sample, DA utilizes precomputed values stored in Look-Up Tables (LUTs) to simplify calculations. This technique effectively reduces the hardware requirements and increases the speed of operations, making it particularly well-suited for hardware implementations in Very Large Scale Integration (VLSI) designs. By utilizing Distributed Arithmetic, designers can create area-efficient and high-performance FIR filters that are ideal for modern DSP applications, further advancing the capabilities of multi-rate systems.

The growing demand for high-speed data processing in contemporary digital systems has intensified the need for efficient filtering techniques. As the complexity of digital signal applications increases, traditional FIR filter designs face challenges in terms of processing speed and resource utilization. This is particularly evident in systems where high throughput and low power consumption are essential, such as in mobile devices and embedded systems. Multi-rate FIR filters, designed using Distributed Arithmetic, provide a compelling solution by allowing designers to optimize filter performance across different operational modes. By strategically selecting sampling rates and utilizing the benefits of DA, it is possible to achieve substantial reductions in both computational load and memory usage. This adaptability makes multi-rate FIR filters an attractive choice for a wide array of applications, from real-time audio processing to communications systems, ultimately contributing to more efficient and responsive digital signal processing solutions. By employing DA, multi-rate FIR filters can achieve enhanced performance while minimizing the hardware footprint, making them ideal for resource-constrained environments such as embedded systems.

2. Related Works

The implementation of a Multirate Finite Impulse Response (FIR) filter using various radix-based Booth encoding algorithms in the Xilinx System Generator environment was discussed in [1], where different radix Booth multipliers (Radix-2, 4, 8, 16, and 32) were compared using MATLAB Simulink for design and simulation, demonstrating improvements in efficiency and accuracy. Similarly, [2] analyzed the design and performance of multirate FIR filters for signal processing applications, focusing on sampling rate conversion (SRC) through cascaded interpolation and decimation using Altera Quartus-II 10.0, highlighting trade-offs in optimizing performance. The use of distributed arithmetic (DA) in matrix multiplication was explored in [3], where efficiency was improved by precomputing intermediate results and storing them in ROM, which was beneficial for constant vector components. An enhanced FIR filter architecture for ASIC and FPGA implementations was proposed in [4], optimizing parallel DA techniques to reduce resource utilization while maintaining high performance. In [5], an FPGA-based optimized decimator using the DA algorithm for wireless applications was presented, employing polyphase decomposition to reduce hardware complexity and achieve a 6.2–14% improvement, simulated and synthesized using the ISE Simulator Tool. The Multiple Constant Multiplication (MCM) method for processing filter coefficients, which led to higher power consumption, was compared to the DA method in [6], which utilized a Look-Up Table (LUT) to reduce multipliers and significantly lower power consumption. The comparison of MCM and DA in implementing Multiply and Accumulate (MAC) units in FIR filters was conducted in [7], while [8] introduced a Fast Parallel Multiple Access Distributed Arithmetic (FPMA-DA) Reconfigurable FIR Filter, achieving up to 26% better power consumption and 36% better area efficiency over direct FIR implementations. A fixed-point finite impulse response adaptive filter using approximate DA circuits with the radix-8 Booth algorithm and an approximate Wallace tree for efficient partial product generation and accumulation was proposed in [9], reducing hardware costs, delay, area, and power consumption while maintaining similar accuracy. A LUT-less Modified DA architecture for FIR filters was introduced in [10], replacing LUTs with multiplexers and adders to eliminate precomputation and extend the input value range, improving FPGA resource efficiency and performance. The FPGA implementation of a DA-based FIR filter was highlighted in [11], showing its advantages in eliminating direct multiplications through LUTs and shift registers and optimizing signal processing by operating at different sampling rates to minimize redundant computations and hardware complexity. Efficient FIR filter architectures using DA were presented in [12], demonstrating high-speed processing with reduced hardware complexity through bit-serial operations, making them ideal for FPGA implementations in DSP applications. In [13], an efficient approximation LUT-based DA VLSI architecture for FIR filtering was implemented, reducing memory requirements while maintaining computational accuracy, and optimizing DA to enhance efficiency in FPGA and ASIC applications. The optimization of DA-based convolutional algorithms for FIR filters on FPGA was analyzed in [14], focusing on computational efficiency, resource utilization, and speed improvements, with optimizations in LUT-based operations and parallel processing reducing hardware complexity and power consumption. Lastly, [15] presented a high-speed pipeline FIR filter design using critical path analysis, optimizing architecture by reducing propagation delays and enhancing throughput through pipeline processing, Wallace tree structures, and Booth multipliers, making it highly suitable for real-time DSP applications requiring high-speed data throughput.

3. Implementation Framework

(A) Architecture of the Multi-Rate FIR Filter

The structure of the multi-rate FIR filter is crafted to effectively handle input signals at different sampling rates while reducing hardware resource usage and power consumption. The key components of this architecture include the input interface, shift register, Distributed Arithmetic LUT, accumulation logic, and output interface.

Input Interface

The filter obtains input samples x[n] through an input interface. The input is typically a signed 16-bit signal, enabling the processing of a wide range of signal amplitudes.The clock and reset signals control the operation of the filter, ensuring proper timing and initialization.

Shift Register

A shift register with multiple stages (e.g., 16 stages) is employed to store the input samples over time. This register allows the filter to maintain a history of past samples needed for convolution.On each clock cycle, the incoming sample is shifted into the register, and the oldest sample is removed, thus maintaining a continuous flow of data.

Distributed Arithmetic LUT

The core of the filter's architecture is the Distributed Arithmetic LUT, which stores precomputed filter coefficients. This LUT allows for efficient computation of the convolution process without the need for explicit multiplications. Each entry in the LUT corresponds to a specific filter coefficient, enabling fast retrieval based on the input sample bits.

Accumulation Logic

The architecture includes an accumulation unit that sums the contributions from the LUT based on the values stored in the shift register. For each bit in the shift register, the corresponding LUT entry is fetched, and the contributions are accumulated to produce the filtered output. This process significantly reduces the number of multiplications required in traditional FIR filters.

Output Interface

The accumulated result is subsequently scaled and stored in the output register y[n], which is also a signed 16-bit signal. The scaling is essential to maintain the output within the appropriate range. The output can then be interfaced with downstream systems for further processing or analysis.

Multi-Rate Capability

The multi-rate capability of the filter is achieved by adjusting the clock rate or sampling rate according to the application requirements. By selectively processing at different rates, the architecture can adapt to various signal characteristics and application demands. This allows for:

Downsampling: Reducing the sample rate for low-frequency components, conserving processing resources.

Upsampling: Increasing the sample rate for high-frequency components, ensuring fidelity in the output signal.

(B) Utilization of Distributed Arithmetic in FIR Filter Design

Look-Up Tables (LUTs)

Each entry in the LUT corresponds to a specific coefficient, which can be accessed based on the input sample values during the filtering process.

Filling the LUT: The LUT is initialized with coefficients representing the filter's impulse response. For instance, if the filter is designed for specific frequency characteristics, these coefficients are derived through design techniques such as windowing or frequency sampling. The LUT is filled with signed 16-bit values, enabling efficient storage and quick access. This approach reduces the computational complexity associated with multiplying input samples by filter coefficients, as the coefficients can be fetched from the LUT rather than computed on the fly

Shift Register Design for Input Sample Management

A fundamental element of FIR filter implementation is the shift register, responsible for storing recent input samples. In this design, a 16-element array of signed 16-bit registers maintains sequential input values, with new samples inserted at the input while older ones shift towards the output. This ensures continuous availability of the latest data for filtering.

Implementation of the Filtering Algorithm

The filtering algorithm processes stored input samples using coefficients from a Look-Up Table (LUT) to achieve efficient signal processing.

Filtering Process: For each clock cycle, the filter computes results based on the shift register's current state and the LUT values. The algorithm iterates through stored samples, indexing the LUT for corresponding coefficients. These retrieved values are accumulated to compute the weighted sum of the inputs, forming the basis of the FIR filter's output.

Output Calculation: The final filter output is derived from the accumulated sum, which may be scaled or truncated to fit within the 16-bit signed data range. The design ensures proper bit selection to maintain precision while preventing overflow.

(C) Filter Algorithm

The Finite Impulse Response (FIR) filter used in this work follows a multi-rate filtering approach with Distributed Arithmetic (DA) optimization. The filtering process is based on the weighted sum of input samples stored in a shift register and corresponding filter coefficients from a Look-Up Table (LUT). The algorithm is structured as follows:

Interpolation by a Factor of L

The input signal x(n)x(n) is upsampled by a factor LL, inserting L−1L-1 zeros between consecutive samples to increase the sampling rate.

Filtering Process

The interpolated signal passes through two FIR filter stages:

First-stage filter h1(k)h_1(k): Removes spectral images introduced by interpolation.

Second-stage filter h2(k)h_2(k): Further refines the signal to ensure an optimal frequency response.

The FIR filter is mathematically defined as:

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

y(n) is the filtered output, h(i) represents the FIR filter coefficients, x(n-i) are the input samples stored in the shift register, N is the filter order.

Distributed Arithmetic (DA) Optimization

Instead of computing direct multiplications, bit-serial processing is used to access precomputed LUT values. The sum-of-products operation is efficiently computed using LUT-based accumulations, reducing hardware complexity.

Decimation by a Factor of M

The filtered signal is downsampled by a factor MM, retaining every Mth sample to reduce the sampling rate while preserving the signal’s essential spectral characteristics.

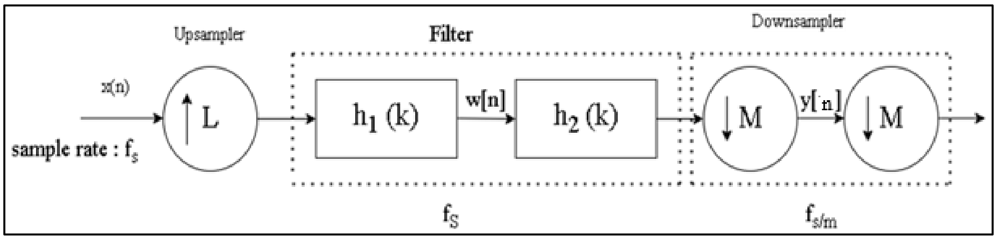

Figure 1. Multi-rate FIR Filter

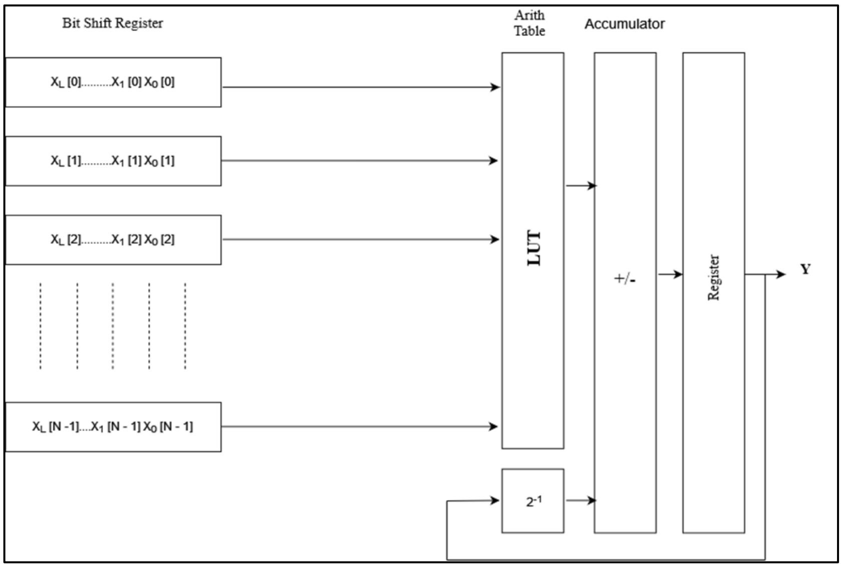
Bit Shift Register
Arith Table
Accumulator
X_L [0].........X_1 [0] X_0 [0]
X_L [1].........X_1 [1] X_0 [1]
X_L [2].........X_1 [2] X_0 [2]
X_L [N -1]....X_1 [N - 1] X_0 [N - 1]
LUT
+/-
Register
Y
2^-1

Figure 2. Distributed Arithmetic Architecture [16]

The above Figures 1 and 2, depicts the architecture of Multi-rate Fir filter and distributed Arithmetic Architecture respectively. The Multi-rate FIR filter optimizes computational efficiency by processing signals at different sampling rates, reducing unnecessary computations. The DA-based FIR filter uses Look-Up Tables (LUTs), shift registers, and accumulators to implement multiplication-free computation, improving area and power efficiency in FPGA-based designs. The DA technique efficiently computes inner products using bit-serial operations, utilizing memory-based precomputed values for reduced hardware complexity. The Distributed Arithmetic (DA) approach is particularly beneficial for high-speed digital signal processing applications due to its efficient resource utilization. By eliminating the need for direct multiplications, DA reduces the critical path delay, leading to faster computation speeds. The LUT stores precomputed partial sums, which are accessed based on the input bit patterns, enabling parallel processing.

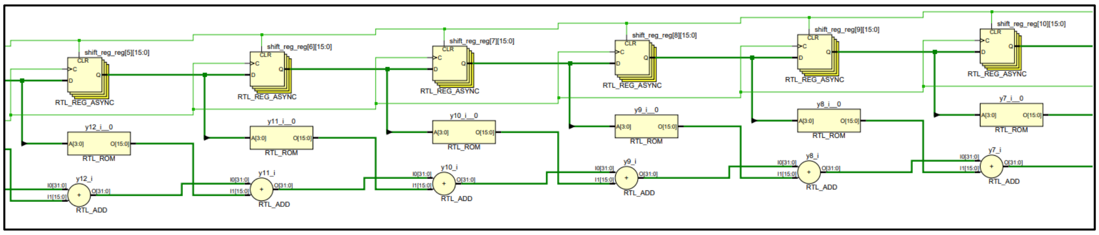

Figure 3. Implemented LUT and Multi-rate Fir Filter

The multi-rate FIR filter Figure 3. was implemented on an FPGA platform utilizing Distributed Arithmetic (DA) with Look-Up Table (LUT) optimization for efficient computation. The filter is designed as a 16-tap FIR filter with a 16-bit signed fixed-point word length, ensuring precise arithmetic operations. The FPGA operates at a clock frequency of 100 MHz, allowing real-time signal processing. The design supports multi-rate processing, enabling operations at different sampling rates through decimation and interpolation techniques. The synthesis report indicates a total of 303 logic cells, 34 I/O ports, and 998 nets, demonstrating an optimized balance between hardware resource utilization and computational performance.

4. Operational Flowchart of Multi-rate FIR Filter

Start: The system is initialized.

Input Signal Received: The system receives an input signal.

Reset Condition: The system checks whether a reset condition has been triggered (like a hardware reset signal).

•**Yes:** If the reset condition is active, the system initializes the shift register and the Look-Up Table (LUT). This prepares the system to process input signals again from a known state.

•**No:** If there’s no reset, the system continues with the next step.

Store Input in Shift Register: The new input sample is stored in the shift register, which holds the most recent samples used for filtering. This is a rectangle indicating a process.

Update Shift Register with New Sample: The shift register is updated to shift previous values, making space for the new input sample. This is necessary for keeping track of the most recent inputs used in the filter calculations.

Access LUT for Coefficients: The system accesses the LUT to retrieve the precomputed coefficients needed for the Distributed Arithmetic calculations. The LUT stores coefficients for efficient computation, reducing the need for multiplication.

Multiply Shift Register Values by LUT Coefficients: The values in the shift register are multiplied with the corresponding coefficients from the LUT. This step involves utilizing the DA algorithm for efficient computation.

Sum Results in Accumulator: The partial products from the previous step are accumulated to form the filtered output. The accumulator sums the results of the DA operations.

Output Filtered Signal: The final result, which is the filtered signal, is produced as the output of the system.

End: The process ends, and the system is ready for the next input signal.

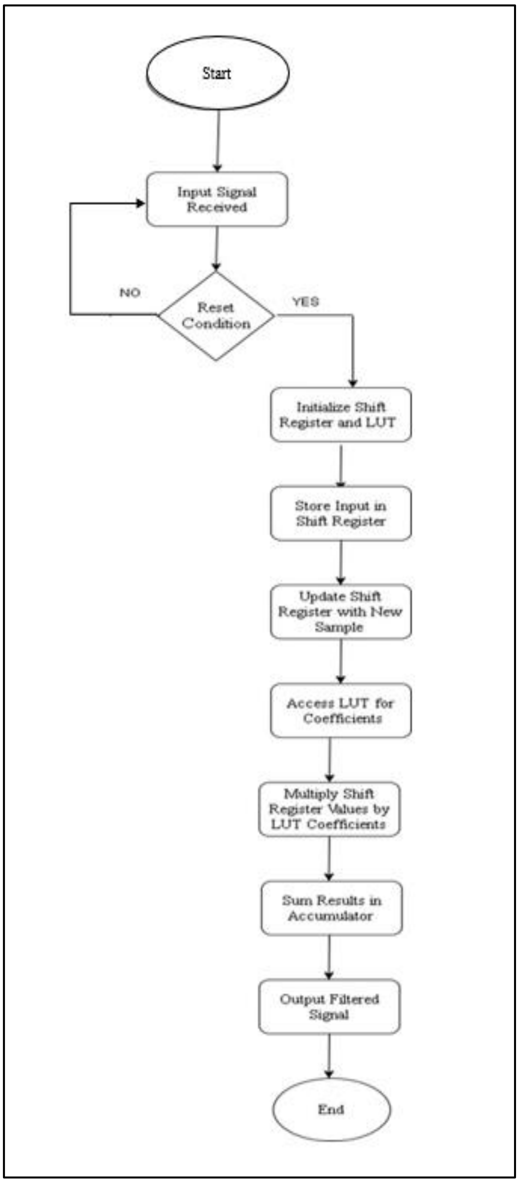
Start
Input Signal Received
NO
Reset Condition
YES
Initialize Shift Register and LUT
Store Input in Shift Register
Update Shift Register with New Sample
Access LUT for Coefficients
Multiply Shift Register Values by LUT Coefficients
Sum Results in Accumulator
Output Filtered Signal
End

Figure 4. Flow Process of multi-rate FIR Filter using Distributed Arithmetic

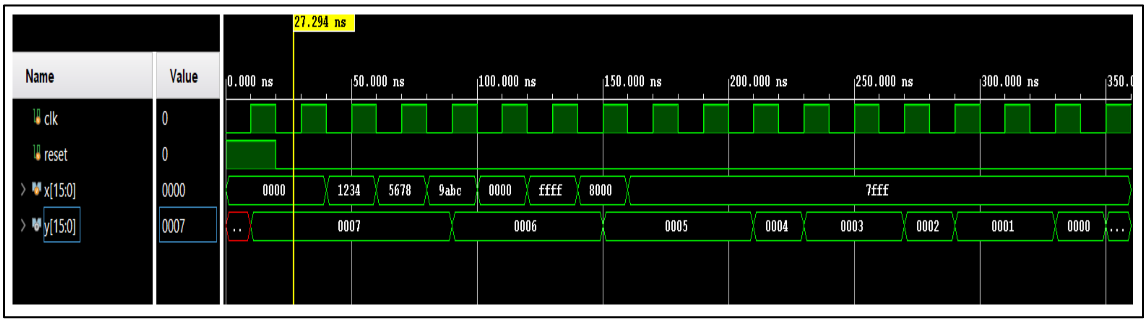

Figure 5. Simulation Waveform for Multi-rate FIR Filter using DA

5. Simulation Results

In the simulation results shown in Figure.5, it showcased the performance and effectiveness of the implemented multi-rate FIR filter utilizing Distributed Arithmetic (DA). The waveform captured the clock signal (clk), reset signal (reset), input values (x), and output values (y). The timing diagram illustrated how the input signal is processed through the filter, showing the sequence of values held in the shift register and how the corresponding output (y) values change over time. The behaviour matched the expected performance, confirming that the input data is correctly processed and filtered using the pre-defined coefficients stored in the LUT (Look-Up Table). The results verified that the multi-rate FIR filter implementation using Distributed Arithmetic is not only functionally correct, but also highly efficient in hardware resource utilization.

Table 1. Resource Utilization of Multi-rate FIR Filter using DA

Parameters	Proposed Filter Utilization
LUT’s	407
FlipFlops	80
IO	22
PS Static Power	0.016 W
PL Static Power	0.713 W
Signals Power	3.002 W
I/O Power	11.806 W

Table.1 shows the minimal percentage of utilized LUTs, FFs and power used.It indicates that the design could be further scaled or integrated with additional functionalities while maintaining resource efficiency.

6. Conclusion and Future Work

This research successfully implemented a multi-rate FIR filter using Distributed Arithmetic (DA) on the Vivado Xilinx platform.The design showcased effective filtering performance while consuming minimal hardware resources, as indicated by the reduced use of Look-Up Tables (LUTs) and Flip-Flops (FFs). The simulation results confirmed the functionality and accuracy of the filter, validating the effectiveness of the DA approach in optimizing computational efficiency and reducing the need for multipliers. The key contribution of this work is the optimized implementation of a multi-rate FIR filter that reduces hardware usage, making it ideal for real-time digital signal processing applications where power efficiency and space limitations are essential. The utilization of DA for coefficient calculation through Look-Up Tables (LUTs) significantly reduces complexity, offering a scalable and efficient solution for various embedded systems and VLSI designs.

Future research could focus on further optimizing the architecture for even lower power consumption or exploring different DA variants to enhance performance in other types of filters. Additionally, extending this design to handle adaptive filtering and real-time dynamic updates of coefficients could broaden its applicability in advanced digital communication and audio processing systems.

References

[1] Ali, Zulfiqar, Sania Syed, Syed Tahir Hussain Shah, Wesam Khalil, and Muhammad Ayaz. "Multirate FIR Filter Using Radix Sort Booth Algorithm In Xilinx System Generator." In 2022 International Conference on Emerging Technologies in Electronics, Computing and Communication (ICETECC), Pakistan, IEEE, 2022. 1-6.

[2] D. Vaithiyanathan, K. Mariammal and B. P. James, "Performance Analysis of Multirate Filter Structures for Signal Processing Applications," 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2023, 1-7,

[3] V.Lesnikov, T. Naumovich and A. Chastikov, "Matrix Multiplication Using Distributed Arithmetic," 2021 23rd International Conference on Digital Signal Processing and its Applications (DSPA), Moscow, Russian Federation, 2021, 1-5.

[4] T. Soni, A. Kumar and M. K. Panda, "Modified Efficient Parallel Distributed Arithmetic based FIR Filter Architecture for ASIC and FPGA," 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2023, 860-865,

[5] L. Singh, Geetanjali and R. Mehra, "FPGA based Optimized Decimator using Distributed Arithmetic Algorithm for Wireless Applications," 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2021, 1-6.

[6] R. Vinay, T. S. V. S. Vijayakumar, L. M. Saini and B. Singh, "Power efficient FIR filter Architecture using Distributed Arithmetic Algorithm," 2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India, 2020, 1-4.

[7] M. Bharathi and Y. J. M. Shirur, "VLSI Synthesis of Multiply and Accumulate Structures Using Distributed Arithmetic," 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India, 2023, 166-169.

[8] A. H. A. Bayoumi, S. A. Ibrahim and H. A. H. Fahmy, "Fast Parallel Multiple Access Distributed Arithmetic (FPMA-DA) Reconfigurable FIR Filter," 2023 International Conference on Microelectronics (ICM), Abu Dhabi, United Arab Emirates, 2023, 1-6.

[9] H. Jiang, L. Liu, P. P. Jonker, D. G. Elliott, F. Lombardi and J. Han, "A High-Performance and Energy-Efficient FIR Adaptive Filter Using Approximate Distributed Arithmetic Circuits," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 1, Jan. 2019, 313-326.

[10] N. S. and J. E. P., "An Efficient Modified Distributed Arithmetic Architecture Suitable for FIR Filter," 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2021, 89-93,

[11] M. M. Basha, P. S. R. Shashank, G. Rushikesh, K. V. Reddy, G. G. Kumar and S. Gundala, "Distributed Arithmetic Based FIR Filter: FPGA Implementation," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, 1-4,

[12] G. Charan Kumar and S. Shiyamala, "Efficient Architectures of FIR Filters using Distributed Arithmetic," 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2024, 129-133,

[13] K. k. Bhadavath and Z. M. Livinsa, "An Efficient Approximation Look Up Table Based Distributed Arithmetic (DA) VLSI Architecture for Finite Impulse Response," 2022

2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, 1849-1852.

[14] C. Chen, V. Romashchenko, M. Brutscheck and I. Chmielewski, "Performance Analysis and Optimization of Distributed Arithmetic-Based Convolutional Algorithms for FIR Filters on FPGA," 2023 34th Irish Signals and Systems Conference (ISSC), Dublin, Ireland, 2023, pp. 1-6.

[15] S. M. Cho, P. K. Meher, L. T. Nhat Trung, H. J. Cho and S. Y. Park, "Design of Very High-Speed Pipeline FIR Filter Through Precise Critical Path Analysis," in IEEE Access, vol. 9, 2021. 34722-34735.

[16] Serrano, Javier. "Introduction to FPGA design." (2008).