

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

Hari Krishnan S*, and Mr. S. Sadiqvali

Abstract—To meet the requirements of the wireless communication industry, digital communication systems require increasingly advanced coding and modulation technologies. Software-Defined Radio enables these advanced ideas to be easily adopted by such systems. The Finite Impulse Response filter is frequently used in wireless communication to pre-process detected signals to reduce noise by utilizing delay elements, multipliers, and adders. Traditional multiplier-based finite impulse response filter designs result in hardware-intensive multipliers that use a lot of space and energy and pose poor calculation speeds and low performance in throughput and latency. To overcome the existing issues, a novel Distributed Arithmetic with a Look Up Table-based FIR filter is proposed, which reduces the Bit Error Rate and latency and improves throughput by optimizing the channel equalizer as a crucial part of Software Defined Radio applications. Further, a key feature named the decimation factor is incorporated to dynamically alter the filter's output frequency response without altering the filter coefficients. Moreover, the worst-case critical route latency of partial product accumulation is reduced using a highly adaptable Parallel Prefix Adder. Additionally, the finite impulse response filters are integrated to decrease the number of Look-Up Tables, thereby saving time and memory. It also investigates the filter efficiency using faster multipliers and adders and validates it on an Artix-7 FPGA. As a result, the proposed model improved the filter's performance over the other existing designs by achieving an operating speed of 260 MHz, delay of 190 ps, power dissipation of 1 mW and throughput of 938.12 Mbps with the number of Look-Up Tables being 16504.

Index Terms—Software-Defined Radio, Noise Removal, Finite Impulse Response, Digital Signal Processing, Field-Programmable Gate Array, and Wireless Communication.

I. INTRODUCTION

Interference with symbols was one of the biggest problems with the digital structure. This suggests that in digital communication systems such as Software Defined Radio (SDR) applications, SDR technology offers the flexibility to implement adaptive filtering and real-time signal processing, allowing for more effective interference management. This capability enhances overall communication reliability, making SDR a valuable tool in modern digital systems.

*Hari Krishnan S / Sanskrithi School of Engineering, Behind Sri Sathya Sai Super Speciality Hospital, Puttaparthi, Sathya Sai District, Andhra Pradesh 515134 (e-mail: hariprofssse@gmail.com)

Mr. S. Sadiqvali / Sanskrithi School of Engineering, Behind Sri Sathya Sai Super Speciality Hospital, Puttaparthi, Sathya Sai District, Andhra Pradesh 515134 (e-mail: syedb6933@gmail.com)

DOI: 10.36244/ICJ.2024.4.3

However, interference with symbols can lead to distorted channels, which may result in errors in data transmission. Interference can occur due to various factors such as noise, signal degradation, or electromagnetic interference. Distorted channels can result in errors in data transmission, making it essential to employ techniques to mitigate interference and restore the integrity of the transmitted symbols [1-3]. As the speed of data transfer systems goes up, Digital Signal Processing (DSP) needs high-speed communication systems. In DSP computers, the speed couldn't go beyond 1 GHz and it plays a crucial role in enhancing the quality of communication by processing and analyzing signals in real-time. The limitation mentioned (1 GHz) likely pertains to a specific context or technology, as modern DSP systems can operate at much higher frequencies, depending on the application [4]. To support multiple bit rates in communication systems, it's necessary to develop new designs. Different bit rates require different modulation schemes, coding techniques, and signal processing methods. Adapting to multiple-bit rates allows for more flexible and efficient data transmission in modern communication systems [5]. Pipelining and parallel processing are techniques used to improve the speed and efficiency of computers in optical transmission systems. Pipelining involves breaking down tasks into stages, allowing for parallel execution of different stages simultaneously. Parallel processing utilizes multiple processing units to perform tasks in parallel, increasing overall system throughput. These techniques are essential for handling the high data rates in optical communication systems [6].

Equalizers are devices or algorithms used on the receiving end of a communication system to reduce distortions like Inter-Symbol Interference (ISI). ISI occurs when symbols interfere with each other in a digital communication signal due to channel characteristics. Equalizers help in recovering the original symbols by compensating for the distortion caused by ISI, thus improving the overall data reception quality [7]. Equalizing Decision Feedback (EDF) is a nonlinear equalization technique used to mitigate ISI in communication systems. It works by making decisions about received symbols and then using these decisions to feedback information to the equalizer to compensate for post-cursor ISI. By doing so, EDF helps in reducing errors caused by

ISI and, consequently, improves the SNR. This technique is particularly effective in scenarios where ISI is a significant challenge.

Maximum Likelihood Sequence Detection (MLSD) is another technique used for ISI mitigation [8]. It considers all possible symbol sequences and selects the one with the highest likelihood, thereby reducing errors caused by ISI. MLSD is especially effective in situations with severe ISI or complex modulation schemes. In a digital communication signal, post-cursor ISI that is, ISI that appears after the primary symbol is eliminated in part by both MLSD and EDF. Through their efficient handling of post-cursor ISI, these methods help to enhance the quality of signals that are received. Combining the various access strategies is probably going to produce the greatest results in terms of enhancing security, increasing data transfer speed, and reducing ISI [9]. These techniques can also address problems related to the way noise spreads, especially in the presence of spectral nulls. Spectral nulls are frequencies where the signal has little or no energy, making them susceptible to noise interference. EDF and MLSD can help mitigate the impact of noise, including noise related to spectral nulls, thus improving overall signal quality [10]. In addition to mitigating ISI and improving SNR, Speed- testing techniques are used. It could refer to the computational complexity and processing speed required for implementing EDF and MLSD algorithms. These algorithms can be computationally intensive, so optimizing their speed is crucial for real-time applications [11-12]. The improvement of Speed and Design Technology is necessary to keep up with the demands of modern high-speed communication systems, where fast and efficient equalization is crucial for reliable data transmission [13].

Distributed Arithmetic (DA) architecture is becoming more popular in DSP. This architecture is chosen due to its simplicity in design and its utilization of Look-Up Tables (LUTs) and transfer build blocks for obtaining partial products. DA architecture offers advantages in terms of efficient hardware implementation, making it a suitable choice for various DSP applications [14]. DA architecture uses two-way binary code complements or offsets for representing the filter coefficients and input values. Binary code complements involve representing numbers as positive and negative complements, which simplifies arithmetic operations. Using offsets can also simplify operations by shifting the input values within a certain range. These techniques contribute to the efficiency and simplicity of DA-based DSP algorithms [15]. To reduce the amount of memory required by a DA Finite Impulse Response (FIR) filter, various strategies are suggested. Memory Divisions refer to dividing the memory resources into smaller blocks or segments to reduce the overall memory footprint, which can help optimize memory usage while still performing the filtering operations efficiently. In addition, Different Memory Bank Approaches are utilized with varying access speeds or capacities to efficiently store and access data. This approach

can be particularly useful when dealing with large datasets and complex FIR filter structures [16-17].

The Look-Up Table (LUT) decomposition scheme is used to simplify the LUT structures in FIR filters based on the DA architecture. However, it may come at the cost of using a few extra filters. The trade-off between complexity and resource usage is common in DSP design. Over the past few decades, there have been efforts to improve the performance and efficiency of filters, including those based on DA architecture. Advances in filter design, algorithm optimization, and hardware capabilities have led to more efficient DSP solutions. These improvements help narrow the field of DA architecture as a powerful and viable design choice for various filter applications [18-19]. The DA architecture is particularly well-suited for FIR filters that are based on Decision Feedback Equalizers (DFEs). DFEs are used to mitigate ISI (Inter-Symbol Interference) in communication systems. The DA architecture can offer an efficient and effective way to implement FIR filters within DFEs, contributing to the overall performance and reliability of communication systems [20]. Hence there is a need to design a novel FIR filter for improving the quality of wireless communication applications.

The Major contributions in this paper are given as follows:

- To design an FIR filter based on DA-LUT multiplier for quicker multiplier and faster adder to improve the speed of filter operation and to create a channel equalizer as part of an SDR application and apply it to FIR for validation.
- To use a decimation factor that dynamically modifies the output frequency response of the filter, a highly adaptive parallel prefix adder (PPA) to lower the worst-case critical route latency and verify the filter efficiency on Artix-7 FPGA.

The content of the paper is organized as section 2 describes the literature survey, section 3 describes the proposed design and its working process, and section 4 discusses the proposed design simulation, performance, and comparative analysis. Finally, section 5 concludes the paper.

II. LITERATURE SURVEY

Kumar et al [21] developed a new architecture for a 2-D block FIR filter by using the DA algorithm, which was renowned for its effective design of the multiply and accumulate block. The DA-LUT has a hardware-based architecture that enables the 2-D FIR filter's architecture to be changed. Additionally, sharing occurs among DA-LUTs at different levels as a result of block processing. In order to simplify the hardware complexity of DA-LUT, a common DA-LUT was created for block inputs. Additionally, the systolic architectures in the suggested design were decreased over the designs that already exist thanks to memory overlapping. By separating the internal block into parallel and small blocks for higher-order 2-D FIR filters, the complexity of DA-LUT was decreased. However, building hardware for DSP applications was more challenging, and it requires specialized knowledge and resources.

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

Amrita Rai [22] proposed a 4-bit FIR filter used in Digital Signal Processing (DSP) employing completely adiabatic technology (PAL) to decrease all parametric performance and power consumption. The designs of a completely adiabatic, low-power, high-speed FIR filter to that of CMOS filters were compared. Reversible logic was used in the design of the PAL FIR filter, and CADENCE digital lab was used to simulate and synthesise it for a variety of parameters, including changes in supply voltage, load capacitance, and transition frequency. This architecture used a logarithmic multiplier to lower the hardware needs and adiabatic technology to provide low power dissipation. However, achieving low area efficiency was more challenging.

Prashanth et al [23] discussed the design of the DA- FIR filter system construct, which was built on an architecture with tightly coupled co-processor-based data processing units. The designed DA-based FIR filter was implemented on field programmable gate array (FPGA) using a series of LUT accesses to simulate multiply and accumulate processes. The proposed filter was implemented using the very high-speed integrated circuit hardware description language (VHDL), and the design is confirmed via simulation. In this study, two optimization strategies were discussed, and the improvements produced were applied to the LUT layer and architecture extractions. The suggested approach provides an optimized design in the form of average LUT minimizations, populated slice reductions, and gate minimization for a discrete impulse response filter. However, combining digital and analog components was challenging, since they have different design constraints, voltage levels, and noise considerations. Hence ensuring seamless integration between the two is crucial.

Maamoun et al [24] proposed an effective high-order FIR filter structure with simultaneous DSP and LUT decreased utilization for FPGA based applications. Also considered was the real-time update of the filter coefficients. Both the speed and the structure of the FPGA were effectively utilized to accomplish these goals. In order to achieve more computation sequences, the difference between the needed input sampling frequency and the FPGA's permitted maximum frequency was handled. Furthermore, the pipelining and selection of the input samples make full use of the unique FPGA Look-up-table Shift-Register (LUT-SR) architecture and internal connections. Reconfigurable filter coefficients were handled by FPGA Block RAMs (BRAMs), and FPGA DSP slices were used to compute the output data of the BRAMs and multiplexers. A single unit was employed for simultaneous control to synchronize the LUT multiplexer selection with the BRAM unit addressing. However, meeting real-time processing requirements is more challenging.

Shrivastava et al [25] proposed an efficient architecture for the DA algorithm-based two-dimensional (2-D) adaptive FIR filter. Practically all DA-based filter topologies demand LUT. The structure that creates the LUT value corresponding to the input, based on adders and logic gates, replaces the

RAM- or ROM-based LUT in the proposed filter architecture. As a result, in DA-based realization, the MAC unit needs fewer logic gates and adders. Additionally, the architecture's memory-sharing idea lessens the latency components. Furthermore, the parallel division of the internal MAC block for the DA decomposition, which provides a greater level of flexibility and parallelism in the proposed design, reduces the complexity of the LUT hardware of higher-order filters. The filter coefficient weights were updated using the 2-D delayed Least Mean Square (LMS) algorithm. However, processing two-dimensional signals introduces challenges related to data handling, such as memory organization and data flow management.

Lakshmaiah et al [26] proposed a modified version of the delayed μ -law proportionate normalised least mean square (DMPNLMS) method. This method is an adapted form of the μ -law proportionate normalised least mean square (MPNLMS). To minimise the silicon area, the technique was implemented through the use of a parallel prefix logarithmic adder of the Ladner-Fischer type. VLSI architecture was implemented and simulated using MATLAB, the Vivado suite, and Cadence RTL and Genus Compiler for complementary metal-oxide-semiconductor (CMOS) 90 nm technology nodes. The DMPNLMS approach showed increased stability, a faster rate of convergence, and a decrease in mean square error. However, the proposed LMS algorithm was sensitive to input noise and outliers and hence ensuring the filter remains robust in noisy environments is more complex.

Khan et al [27] designed an LMS algorithm based on the steepest descent technique presented with a potential expansion to its power-normalized LMS version and examined its convergence features. The design and development of non-pipelined ADF systems was accomplished by transforming the coefficient update equation of the LMS algorithm via TC DA and OBC DA. The LUT pre-decomposition approach was utilised by the suggested architectures to improve throughput performance. It allowed the deconstructed LUTs to be updated concurrently using the same mapping approach. Additionally included was an effective fixed-point quantization model for assessing suggested structures from a practical standpoint. However, minimizing power consumption while maximizing throughput is a constant challenge.

Murthy et al [28] presented multiple methods for designing reconfigurable finite impulse response (RFIR) filters. Software-defined radio (SDR) applications were appropriate for programmable FIR filter designs based on DA. Reusing registers, multipliers, and adders as well as optimizing other factors including area, power consumption, speed, throughput, latency, and flip-flop and slice hardware utilizations were the key contributions of reconfiguration. In light of the aforementioned factors, the efficient design of the building blocks was optimized for the RFIR filter. However, achieving high filter performance, such as sharp roll-off and minimal distortion, is challenging, especially when trying to optimize for reconfigurability.

Rammohan et al [29] presented the decimation filter's hardware implementation and architecture for use in hearing aids. Using Matlab Simulink, the CIC, half band filter, and corrector FIR filters were created and tested for real-time implementation. When compared to a basic decimation filter, the suggested decimation filter architecture uses a compressor adder-based DA FIR filter, which reduces the amount of hardware needed by 69% and reduces power consumption by 83%. FIR filters were used in decimation filters for audio applications because they make it simple to establish a linear phase. However, in the future, a linear phase across the entire band is needed.

Uma et al [30] focused on applying DA based FIR filters to remove baseline drift and muscle artifact noise. An area-efficient modified DA-based FIR filter was used to filter out noise and has no LUTs. The modified DA-based FIR filter's performance was contrasted with that of the traditional DA FIR filter. Baseline Wander noise, Muscle Artefact noises, and an arbitrary real-time ECG record are all extracted from the MIT-BIH noise stress test database. Signal to Noise Ratio (SNR) and Mean Square Error (MSE) output metrics were used to assess both filters' performance. The redesigned DA-based FIR filter yields good output SNR and low MSE for baseline wander noise reduction. However, a filter designed for a stationary noise model was not as effective in removing non-stationary noise components.

Nirmala et al [31] proposed a shared LUT updating system for a reconfigurable offset-binary code (OBC) DA-based FIR filter. With each additional filter, the LUTs in DA grow exponentially larger. A way to lessen this significant memory consumption for higher-order filters was a shared LUT-based DA structure. The shared LUT updating method that was being suggested makes use of LUT partitioning, which divides coefficients into small length vectors and significantly reduces the size of LUTs. CMOS 90 nm technology was used to synthesize the suggested DA filter with Synapsis ASIC Design Compiler. When compared to prior designs, the suggested design delivers high speed at a smaller ADP. However, high-speed, low-area OBC-based decimation filters were quite complex to design and implement, especially when dealing with high-order delta-sigma modulators.

Șorecău et al [32] introduced the SDR measurement system for real-time spectrum monitoring. It enabled channel power and complementary cumulative distribution function measurements. It was validated against a high-performance spectrum analyser (SA) in a laboratory setting and successfully captured signals from modern communication standards. The results demonstrated the SDR system's capability to perform real-time measurements and provided valuable insights into signal behaviour, highlighting its potential for advanced spectrum analysis. However, achieving optimal performance across diverse conditions remains a challenge.

Radu et al [33] proposed a system for identifying the modulation of complex radio signals using an artificial intelligence model integrated with a cloud-based platform. The

implementation controls a software-defined radio platform to generate and receive real modulated signals, demonstrating the viability of cloud computing for signal processing tasks. The results indicate a high degree of success in identifying certain modulation types, allowing users to access the system from anywhere with an internet connection. However, a significant limitation is the challenge of improving model accuracy under varying signal-to-noise ratios.

From the analysis, [21] building hardware for DSP applications was more challenging, [22] does not attain low power and area efficiency, [23] challenges in combining digital and analog components, [24] does not meet real-time processing requirements, [25] data handling problem obtained, [26] need to ensure the filter in noisy environments, [27] minimizing power consumption is a constant challenge, [28] does not achieve high filter performance, [29] linear phase over the entire band is required, [30] not effective in removing non-stationary noise components, and [31] quite complex in high-speed, low-area OBC-based decimation filters. For [32] it is difficult to perform under various circumstances and [33] indicates that increasing model accuracy at different signal-to-noise ratios is a challenge. Hence, to overcome the aforementioned issues and to enhance the performance of DA-FIR filter, a new novel approach has to be proposed.

III. FIR FILTERS IN SOFTWARE DEFINED RADIO WIRELESS COMMUNICATION SYSTEMS

FIR filters are commonly used in wireless communication systems for various purposes, including signal processing and noise reduction. These filters are used to shape or modify the frequency response of signals to improve communication quality. In SDR systems, the Finite Impulse Response filters play a crucial role in the channelization process, which involves extracting narrowband channels from a wideband signal. These FIR filters must be designed to operate at high sampling rates and handle large-order filters to meet stringent adjacent channel attenuation specifications. The design of FIR filters for SDR applications often focuses on achieving a balance between reconfigurability, high-speed operation, and low power consumption, which are essential for next-generation wireless communications. Advanced FIR filter designs utilize techniques like Distributed Arithmetic and Residue Number Systems (RNS) to improve performance. For instance, DA-based FIR filters can offer significant area delay and energy efficiency improvements, making them suitable for high-throughput implementations. Similarly, RNS-based FIR channel filters can be reconfigured to adapt to various channel filtering specifications, providing speed improvements and complexity reduction compared to traditional methods. These innovations in FIR filter design contribute to the versatility and efficiency of SDR systems, enabling them to support multi-standard wireless communication protocols.

A. Problem Statement and Motivation for the Research

In a traditional FIR filter design, the filter coefficients are multiplied with delayed versions of the input signal, and

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

the results are summed to produce the filtered output. This operation requires multipliers, delay elements, and adders. Multiplier-based FIR filters typically require dedicated hardware multipliers, which is expensive in terms of both space and energy consumption. This is a significant drawback, particularly in applications where hardware resources are constrained. The use of dedicated multipliers leads to slower calculation speeds, especially for high-speed signal processing. Multipliers tend to be relatively slow compared to other operations, which limits the filter's performance in applications that require real-time processing. Due to their hardware-intensive nature and slow speed potential, traditional multiplier-based FIR filters suffer from low throughput and high latency. This is problematic in applications where the timely processing of signals is crucial. Hence, a novel FIR filter design that mitigates these issues is imperative for enhancing the efficiency and responsiveness of SDR systems.

B. Proposed Design of Novel FIR Filter for an Effective SDR System

To address the limitations of traditional multiplier-based FIR filters, a new innovative approach called **Distributed Arithmetic (DA) with Look Up Table-based FIR filter (DA-LUT-FIR)** is proposed, in which enhanced efficiency of FIR filters is typically achieved by optimizing the computational components of the filter, such as the multipliers and adders thereby significantly speeding up the filter's operation. This study utilizes an Artix-7 FPGA to implement and test an optimized FIR filter design. The Artix-7 FPGA provides high throughput and low latency, crucial for SDR systems, along with robust DSP resources that efficiently handle complex DA operations without relying on traditional multipliers, thus eliminating common bottlenecks. Its optimized power efficiency reduces energy consumption, making it ideal for energy-sensitive applications. Additionally, the Artix-7's high memory bandwidth and extensive LUT resources enhance processing speed and conserve resources, supporting the filter's high-performance demands. Notably, the filter's output frequency response can be dynamically adjusted through a decimation factor, all while keeping the filter coefficients unchanged. A highly adaptive parallel prefix adder is used to lower the worst-case critical route latency of partial product accumulation. FIR filters reduce the amount of LUTs, thereby conserving memory and processing time. To boost performance while cutting down on processing time, this study also suggests limiting the number of coefficients read in parallel for FIR filter operations.

In this proposed design, DA optimizes multiply-and-accumulate (MAC) operations in the FIR filters where instead of directly multiplying filter coefficients with input samples, it precomputes partial products and stores them in memory (LUTs or RAMs). During filter operation, it efficiently combines these precomputed values to compute the final output thereby significantly reducing the need for multipliers, which are resource-intensive in FPGA implementations. Traditional FIR filters rely on multipliers and adders to compute the convolution of input samples with filter coefficients but the LUT-based FIR filters used in this proposed approach

replace multipliers with precomputed LUT entries, which store the results of coefficient multiplication thereby avoiding expensive multiplication operations and achieving area and power savings. The decimation factor used in this design dynamically adjusts the output frequency response of the filter and hence, if the original filter operates at a higher sample rate, decimation reduces it to match the desired output rate. The parallel prefix adder efficiently computes the sum of partial products and limits the number of coefficients read in parallel during filter operations. By distributing the addition process across multiple stages, this PPA carefully manages the parallelism and reduces critical path delays thereby striking a balance between throughput and resource utilization and improving performance.

The combination of these technical innovations, including the use of Distributed Arithmetic, dynamic decimation factor, parallel prefix adder, reduced LUT utilization, and coefficient parallelization optimization, collectively improve the efficiency and performance of FIR filters in the proposed DA-LUT-FIR approach. These enhancements enable more efficient and high-performance FIR filtering solutions, particularly for applications where resource constraints and real-time processing requirements are critical, such as in SDR systems.

a. Block Level Diagram of the proposed FIR Filter

Figure 1 depicts the overall block-level diagram for the proposed approach. The architecture of a proposed DA-LUT-based FIR filter typically involves several key components and stages. The filter receives input data, which is the signal to be filtered and it is typically in the form of discrete samples.

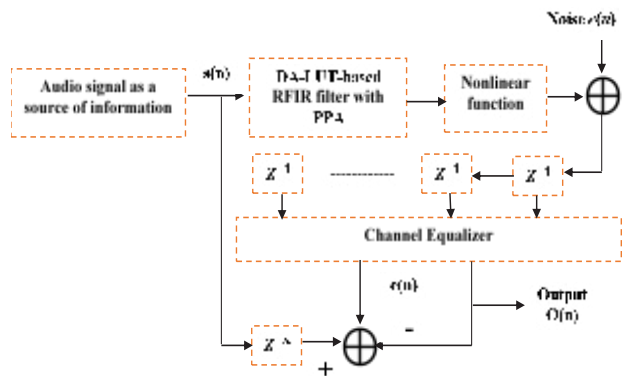


Figure 1: Proposed Block Level Diagram

The FIR filter uses a set of filter coefficients (taps) that determine the filter's behavior. These coefficients are usually constants and define the filter's impulse response. The core of the DA-LUT-based FIR architecture is the use of Look-Up Tables (LUTs), which store precomputed values i.e., the result of multiplying each possible input value by each filter coefficient. The number of LUTs is typically minimized for efficiency. A multiplexer is used to select the appropriate LUT entry based on the current input data value and it effectively "looks up" the precomputed result for the current data value and coefficient. Instead of using traditional multipliers, the DA-LUT-based FIR filter uses multiplier-less multiplication.

The selected LUT entry is treated as a partial product and then the accumulator sums up the partial products obtained from the multiplier less multiplication. This accumulation process continues for multiple data samples, producing the filtered output. The final output of the filter is the result of the accumulation process and it represents the filtered version of the input signal. Depending on the proposed design and application, a decimation stage is added to reduce the output data rate, and is often used in cases where the filter output does not need to retain all the input data points. To enhance performance and throughput, the architecture incorporates parallel processing, which involves the processing of multiple data points and coefficients simultaneously, further improving filter speed. The architecture is highly customizable, allowing for adjustments such as filter length, word length, and the number of LUT entries to be tailored to specific application requirements. The number of LUTs required for a DA-LUT- FIR filter scales with the filter length, particularly if all coefficient multiplications are independently handled. Overall, the DA-LUT-based FIR architecture is designed to efficiently perform filtering operations by utilizing precomputed values stored in LUTs and minimizing the need for traditional multiplication hardware. This results in an efficient and hardware-friendly FIR filter suitable for SDR applications.

C. DA-LUT-FIR filter Formulation

Typically, DA is a well-known FIR filter method, which focuses especially on the computation of the sum of products, often known as the vector dot product that includes several crucial DSP filtering and frequency-shifting operations prompted by the possibilities of the Artix-7 FPGA look-up table architecture. To determine the total number of products needed for FIR filters, DA effectively uses LUTs, shifters, and adders. The DA-LUT-FIR filter Formulation for analysis is discussed in further sections below.

a. Distributed Arithmetic (DA) Computation

When the coefficients of the FIR filter are known, the DA resolves the computation of the internal product, and the output of an FIR filter is given by the convolution sum in equation (1):

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

Where,

- Y[n] is the output signal at time n
- h[k] are the filter coefficients
- x[n-k] is the input signal at time n-k
- K is the number of filter coefficients (filter length)

DA shifts the computation from the traditional method of directly calculating the product of the input signal and filter coefficients to a method that relies on bit-level manipulations. This is especially efficient in FPGAs where LUTs can store precomputed values. The step-by-step DA process is given in the following equations.

Decompose the input data x[n-k] into its binary representation. For simplicity, each input sample is represented by B bits as given in equation (2):

$$x[n - k] = \sum_{b=0}^{B-1} x_b[n - k] \quad (2)$$

Here, $x_b[n-k]$ represents the b^{th} bit of the input sample $x[n-k]$.

Then precompute all possible values of the partial products $h[k] \cdot x_b[n - k]$ for each bit position and store them in LUTs. This reduces the real-time computation to simple LUT lookups and bit-shifting operations. The precomputed partial products for each bit position are accumulated across all filter taps which is given in equation (3):

$$Y = \sum_{k=0}^{K-1} \sum_{b=0}^{B-1} h[k] \cdot x_b[n - k] \cdot 2^b \quad (3)$$

This step involves shifting the precomputed values according to their bit positions and summing them up using adders. Finally, the DA-based FIR filter output is represented by equation (4) as given below:

$$Y = \sum_{k=0}^{K-1} h[k] \cdot x'[k] \quad (4)$$

Here, $x'[k]$ is a function of the bit-level decomposition of the input data $x[n-k]$.

Moreover, in the context of FIR filters, particularly with the proposed DA-LUT-FIR approach, representing input data using a two's complement B-bit binary format is crucial. This representation accommodates both positive and negative values and provides a precision level determined by the number of bits B. The formulation of this representation is provided in the below section.

b. Distributed Arithmetic (DA) for FIR Filters with Two's Complement Representation

In the proposed DA-LUT-FIR filter, the input data $x[k]$ is represented using a two's complement BBB-bit binary representation, allowing for an accurate representation of both positive and negative values. The two's complement representation of $x[k]$ is crucial for handling negative numbers in FIR filter calculations which is given in equation (5);

$$x'[k] = -2^B X_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^b \quad (5)$$

Where, $x_b[k]$ is b^{th} bit of $x[k]$, $X_B[k] \in \{0,1\}$.

Substituting equation (5) in equation (4),

$$\begin{aligned} Y &= \sum_{k=0}^{K-1} h[k] \left(-2^B X_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^b \right) \\ &= -2^B \sum_{k=0}^{K-1} h[k] X_B[k] + \sum_{b=0}^{B-1} 2^b \sum_{k=0}^{K-1} h[k] x_b[k] \\ Y &= -2^B f(h[k] \cdot X_B[k]) + \sum_{b=0}^{B-1} 2^b f(h[k] x_b[k]) \end{aligned}$$

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

The final output of the FIR filter with two's complement representation is given in equation (6); Thus, from equation (6), the simplified input data of B-Bit binary data is given below in equation (7):

$$f(h[k]x_b[k]) = \sum_{k=0}^{K-1} h[k]x_b[k] \quad (7)$$

Therefore, the filter coefficient is further stored at LUT and it is addressed by $X_b[k]$. This reduces entry and summation with LUT of the MAC blocks of FIR filters. The digital filters are made with the use of registers, memory resources, and a scale accumulator to perform this arithmetic. One of the key features of the proposed implementation is the use of a small number of LUTs which is discussed in the following section.

c. Minimizing the LUT size

This proposed approach suggests that the design optimizes the LUTs' usage, potentially by reusing or sharing LUT resources for multiple coefficients to minimize the LUT size while maintaining accuracy. Reducing the number of LUTs leads to savings of hardware resources and power consumption. To execute FIR filter operations efficiently, the proposed approach allows parallel access to multiple coefficients in LUTs. This means that multiple coefficients are accessed simultaneously to perform filter calculations. Parallelism in coefficient access leads to a significant reduction in the processing delay and also enhances the filter's throughput, making it suitable for real-time applications. By minimizing the number of LUTs and enabling parallel access to coefficients, the filter processes the data with lower latency, which is essential for real-time processing.

d. Decimation and Parallel Prefix Adder

Additionally, the frequency response of the filter output is dynamically altered using the decimation factor. Decimation is a process in DSP where the sampling rate of a signal is reduced. By changing the decimation factor, the effective bandwidth and characteristics of the filter output are adjusted without modifying the filter coefficients. It's emphasized that the filter coefficients remain unchanged while altering the frequency response through decimation. This suggests that the filter is designed to be flexible in its application, allowing for real-time adjustments without the need for recalculating or modifying the filter coefficients. To reduce the worst-case critical path time during partial product accumulation, a highly customizable parallel prefix adder is implemented. It is a type of digital adder that efficiently adds multiple numbers of LUT in parallel. By customizing this adder, the design optimizes its performance for the specific requirements of the FIR filter design.

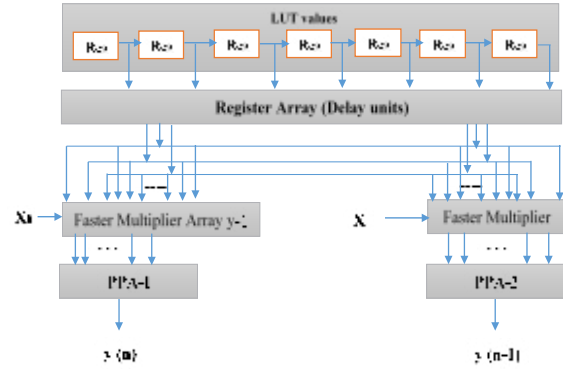


Figure 2: DA-LUT-based RFIR filter with PPA

Figure 2 shows the DA-LUT-based Reconfigurable Finite Impulse Response (RFIR) filter with PPA. The DA with LUT-based RFIR filter, when combined with Power, Performance, and Area (PPA) considerations, offers a versatile and efficient approach to DSP. In order to achieve optimal power efficiency, high performance and minimal hardware footprint, the DA-LUT-based RFIR filter is provided, in which the RFIR filter is a powerful tool for processing digital signals. It allows for adaptability, making it ideal for a wide range of applications in wireless communication systems. This not only reduces power consumption but also accelerates the processing speed of the DA-RFIR filter. The LUT stores precomputed products of filter coefficients and input data, thereby effectively transforming more multipliers into simple LUT. The optimization problem formulation for channel equalizer in terms of the objective function and system constraints is discussed below.

e. Optimization Problem Formulation

The primary objective of this research is to minimize the Bit Error Rate (BER) and latency while maximizing the throughput of the FIR filter system, which is integral to the performance of SDR applications. This section clearly outlines the optimization objectives and constraints associated with the channel equalizer in SDR applications.

Objective Function:

The goal is to minimize latency and BER while maximizing throughput. This can be mathematically represented as:

Objective: min BER (H, X, C), min Latency (H, X, C), and max Throughput (H, X, C)

Where,

- H represents the filter coefficients.
- X denotes the input data.
- C symbolizes the system configurations, including the decimation factor and hardware resources.

Constraints:

Hardware Resource Constraint: The total number of Look-Up Tables (LUTs) and slices used should not exceed the available resources on the Artix-7 FPGA.

$$\begin{aligned} \text{LUTs}(H) &\leq 17,000 \\ \text{Slices}(H) &\leq 10,000 \end{aligned}$$

Power Consumption Constraint: The power dissipation should be within acceptable limits for SDR applications.

$$\text{Power}(H, X, C) \leq 100 \text{ mW}$$

Latency Constraint: The latency must be minimized while ensuring it supports real-time data processing.

$$\text{Latency}(H, X, C) \leq 20 \text{ ns}$$

Throughput Constraint: The filter must maintain a high throughput to handle real-time data processing.

$$\text{Throughput}(H, X, C) \geq 900 \text{ Mbps}$$

Optimization Approach:

Filter Coefficient Optimization: Use DA to precompute possible outcomes for each filter coefficient, reducing the need for real-time multipliers and thus decreasing latency and power consumption.

Parallel Processing: Implement parallel prefix adders to handle partial product accumulations efficiently, enhancing throughput.

Dynamic Decimation: Adjust the decimation factor dynamically to balance the trade-off between processing speed and frequency response.

Adaptive Channel Equalization: Optimize the channel equalizer settings to minimize BER by dynamically adjusting the filter coefficients in response to changing channel conditions.

This approach is not only space-efficient but also reduces the need for resource-intensive multiplication hardware. By carefully analyzing and optimizing the DA-RFIR filter's architecture, the best trade-offs between power efficiency, high performance, and a minimal hardware footprint are achieved. The DA-LUT-based RFIR filter is fine-tuned with PPA, this ensures that signal processing applications operate at peak efficiency, and deliver results that meet the most demanding requirements. With the DA-LUT-based RFIR filter, the power of PPA is unlocked, thereby making it possible to process digital signals with unparalleled efficiency. To perform a performance analysis and optimization of a LUT layer, the proposed model follows a systematic process involving the LUT, identifying bottlenecks, and implementing optimizations. Thereby, the existing technique's drawbacks are overcome by this proposed method. In the next section, the performance and comparison of the proposed method are discussed.

IV. RESULT AND DISCUSSION

In this section, the results for the proposed DA-LUT-FIR filter are presented and engaged in a thorough discussion on

its performance and efficiency. The filter was designed and implemented to address the challenges associated with finite impulse response filtering while harnessing the power of distributed arithmetic and lookup tables for optimized multiplication.

A. Experimental Setup

The simulation results are discussed below. This work has been implemented in the MATLAB working platform using the following system specifications.

Software	: MATLAB
OS	: Windows 10 (64-bit)
Processor	: Intel i5
RAM	: 8GB RAM

B. Simulated output of the proposed method

The proposed structure has been added to Xilinx System Generator and Matlab Simulink. For the execution of the proposed design, a string of channel impulses that have been BPSK message modulated for implementation is considered. The signal was transmitted to adaptive DFE for ISI error correction and noise removal. The algorithm is programmed directly into the FPGA integrated within the SDR. This allows for efficient processing and real-time performance, utilizing the FPGA's parallel processing capabilities while minimizing latency and maximizing throughput.

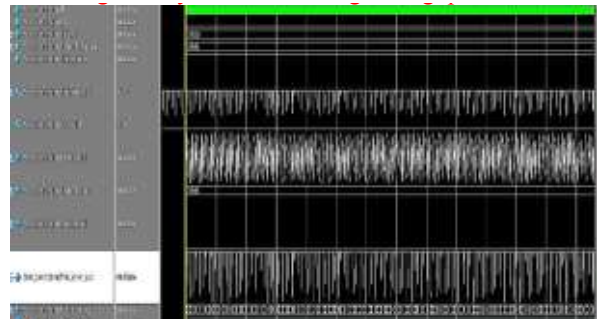


Figure 3: Verilog output of FIR

Figure 3 depicts the Verilog output of the FIR filter. This FIR input module is responsible for receiving the incoming digital data stream and buffering it for processing. It feeds the data into the filter's main processing engine. Multiplier and Accumulator components perform the core filtering operation. The multiplier module multiplies each data sample by the corresponding coefficient, and the accumulator sums up these products to produce the filter's output. The coefficients used by the filter are stored in a memory module. This memory is accessed based on the current position of the sliding window. To slide the window over the input data, there is a control module that manages the window's position and ensures the correct samples are selected for multiplication. Finally, the filtered output data is sent to the output module, which makes it available for further processing.

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

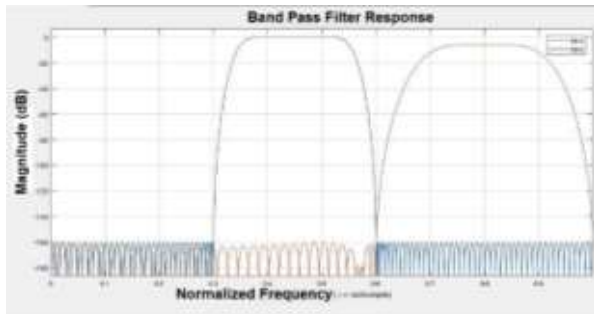


Figure 4: FIR Output Response

Figure 4 depicts the output response for the FIR filter which is characterized by its ability to effectively filter and modify the input signals in a precise and controlled manner. As the input signal progresses through the filter, it encounters each tap and undergoes a series of multipliers and adders. At each tap, the input is multiplied by the corresponding coefficient, and the results are summed together. It represents the frequency response of two bandpass filters where the blue trace (M=1) illustrates a filter that allows a lower range of frequencies to pass through, effectively filtering out frequencies outside this range. Conversely, the orange trace (M=2) demonstrates a filter with a passband at higher frequencies. The graph clearly outlines the effective frequency ranges for each filter, with the passbands being the regions where the magnitude does not exhibit significant attenuation.

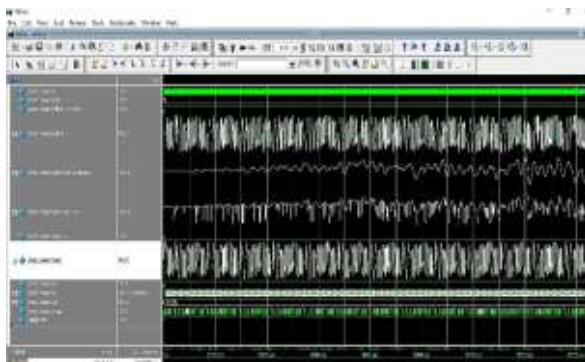


Figure 5: LMS filter output

Figure 5 shows the Least Mean Square (LMS) filter output for the proposed approach. In this figure, the top waveform, labelled “test_bench/data_in” represents the gradient of the input error signal for the first test bench. This is crucial as it indicates how the LMS filter’s predictions deviate from the desired outcome. The waveform below, labelled “test_bench/desired_response” is the target or reference signal that the LMS filter aims to replicate or predict accurately. Then the next waveform, labelled “test_bench/FILTER_OUT,” shows the output of the filter applied to the first test bench’s data. This output is what the LMS filter has produced as its prediction or filtered signal. As the input signal flows through the LMS filter, a remarkable transformation takes place and also this filter armed with its adaptive capabilities meticulously analyses the incoming data in real time. It constantly refines its internal coefficients to minimize the error between the desired signal and the filtered output.

C. Performance metrics of the proposed methodology

The performance metrics collectively provide a comprehensive evaluation of the proposed DA-LUT-FIR filter, helping to assess its efficiency, effectiveness, and suitability for specific applications. The effectiveness of the proposed design is discussed in this section by analysing the performance parameters such as delay, power consumption, number of slice registers, AND/OR gates, LUTs, frequency and the number of adders used.

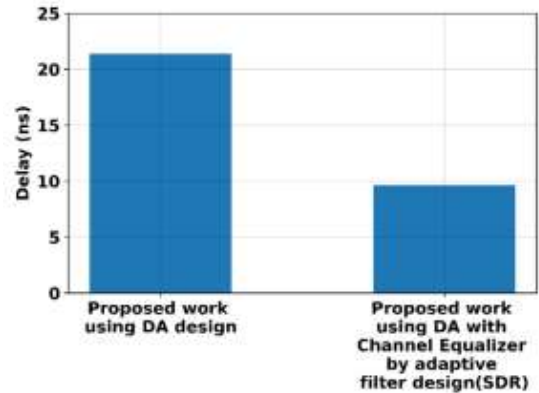


Figure 6: Delay of the proposed design

The graph in figure 6 compares the delay between two FIR filter designs for SDR systems. The first design, using traditional Distributed Arithmetic (DA), has a delay of about 21.41 ns. The second design, which incorporates DA with an adaptive Channel Equalizer, shows a significantly reduced delay of approximately 9.627 ns. This improvement is due to more efficient MAC operations, dynamic adjustment of filter coefficients, and optimized coefficient management. Consequently, the adaptive design ensures faster data processing and lower latency, making it ideal for real-time applications.

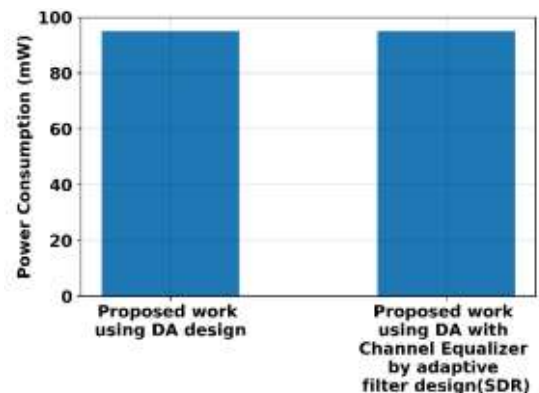


Figure 7: Power consumed by the proposed design

The graph in figure 7 compares the power consumption of the proposed FIR filter design. Both designs show similar power consumption, close to 95 mW. However, the proposed work using DA with an adaptive Channel Equalizer achieves this power efficiency while also significantly reducing delay, as seen in the previous graph. This is due to the optimized use of resources, efficient coefficient management, and reduced need for multipliers, which all contribute to maintaining power

consumption at a low level without sacrificing performance. This balance ensures that the adaptive design is suitable for real-time applications, offering both speed and energy efficiency.

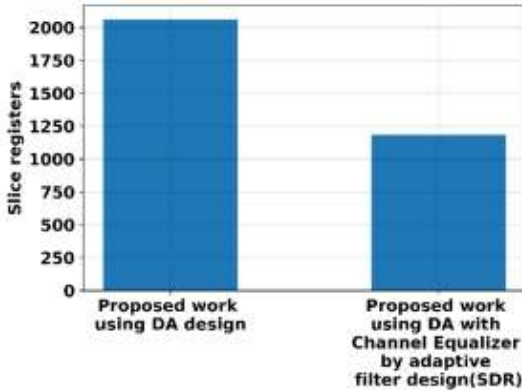


Figure 8: Number of slice registers used in the proposed design

The figure 8 illustrates the number of slice registers used in the proposed work in which using DA design requires 2062 slice registers, whereas the proposed work using DA with an adaptive Channel Equalizer significantly reduces this number to 1182. The reduction in slice registers is by the efficient design of the adaptive filter, which optimizes resource usage by minimizing redundant or unnecessary computations. This leads to a more streamlined architecture, reducing the hardware complexity and enhancing overall efficiency without compromising performance.

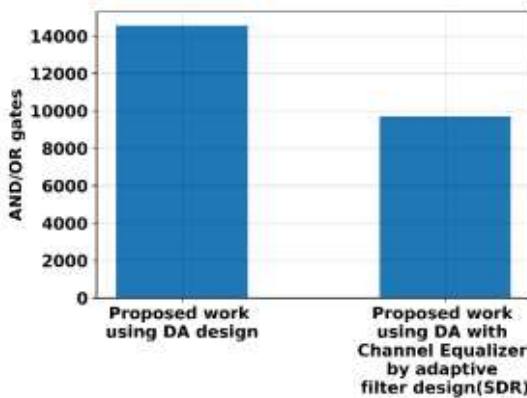


Figure 9: Number of gates used in the proposed approach

The figure 9 shows the number of AND/OR gates used in two different FIR filter designs. The proposed work using DA design requires 14,568 gates, while the proposed work using DA with an adaptive Channel Equalizer uses significantly fewer gates, totalling 9,715. This reduction in the number of gates is because of the adaptive Channel Equalizer's ability to streamline the logic design, minimizing the need for excess logical operations. As a result, the overall gate count is reduced, leading to a more efficient and compact design that maintains functionality while lowering the hardware complexity.

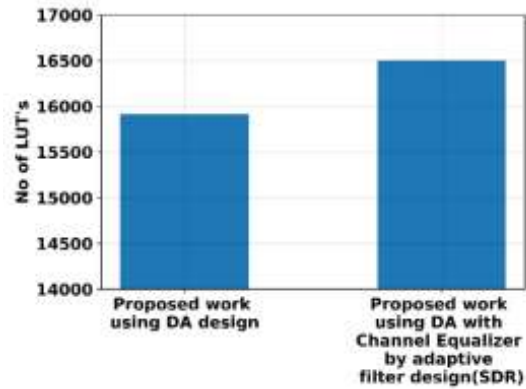


Figure 10: Number of LUTs used in the proposed design

The graph in figure 10 illustrates the number of LUTs used in two different FIR filter designs where the proposed work using DA design requires 15,914 LUTs, while the proposed work using DA with an adaptive Channel Equalizer uses slightly more, totalling 16,504 LUTs. This slight increase in the number of LUTs is by the additional complexity introduced by the adaptive Channel Equalizer, which enhances the system's ability to dynamically adjust to varying channel conditions. Although there is a small increase in the number of LUTs, this trade-off results in improved performance and adaptability, making the design more robust and efficient in handling diverse signal environments.

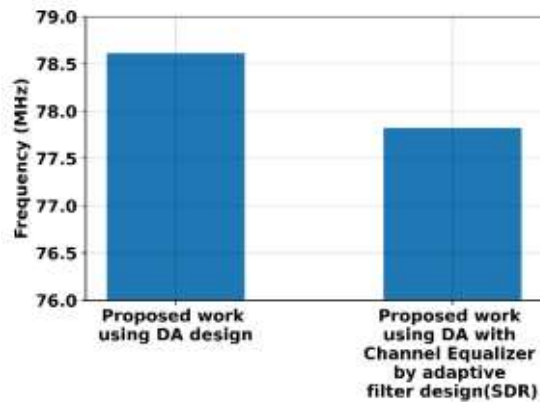


Figure 11: Frequency of the proposed design

The figure 11 compares the operating frequencies of filters in two different designs. The proposed work using DA design attains a frequency of 78.617 MHz, while the proposed work using DA with an adaptive Channel Equalizer operates at a slightly reduced frequency of 77.825 MHz. This minor decrease in frequency is due to the added complexity and functionality of the adaptive Channel Equalizer, which allows the system to better adapt to varying channel conditions. Despite the small reduction in frequency, the enhanced adaptability and performance benefits of the adaptive equalizer outweigh this trade-off, resulting in a more robust and versatile system.

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

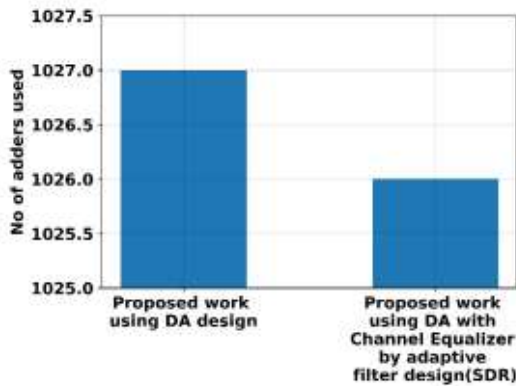


Figure 12: Number of adders used in the proposed filter design

Figure 12 shows the comparison between two proposed designs for digital adders, one employing a conventional DA design and the other integrating an adaptive Channel Equalizer (CE). Interestingly, the latter design, featuring the DA with an adaptive CE, demonstrates a slight reduction in the number of adders required compared to the former, with counts of 1026 and 1027, respectively. This marginal decrease is attributed to the enhanced efficiency achieved through the adaptive CE, which dynamically adjusts to channel variations, optimizing the performance and reducing the demand for additional adders. Hence, while both designs offer competitive functionality, the incorporation of adaptive CE showcases a subtle but notable improvement in resource utilization.

D. Comparison of the proposed methodology

This section highlights the proposed method’s performance by comparing it to the outcomes of existing approaches and showing their results based on various metrics. The performance of the existing approaches such as conventional DA-based filter, LUT-Less 2, Separated LUT-DA, and DA-LUT using buffer [34], GBoost Classifier, Light GBM and Gradient Boosting [33] GFSK, GMSK and BPSK OFDM [35], are compared to that of the proposed DA-based LUT-FIR filter.

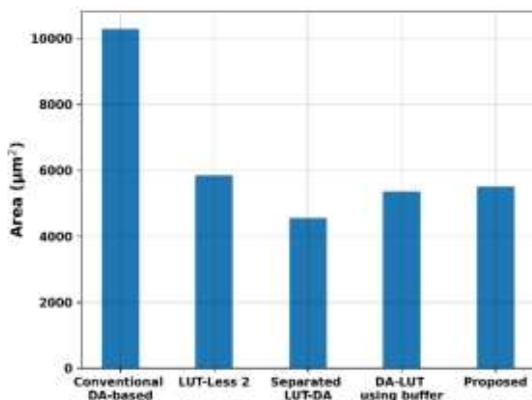


Figure 13: Comparison of area of various filter designs

Figure 13 compares the area efficiency of proposed DA-LUT-FIR filter model with existing approaches, including the conventional DA-based filter, LUT-Less 2, Separated LUT-DA, and DA-LUT using buffer. The conventional DA-based

filter occupies 10279 µm², while LUT-Less 2, Separated LUT-DA, and DA-LUT using buffer require 5854 µm², 4554 µm², and 5356 µm², respectively. In contrast, the proposed model achieves a compact area of 5500 µm² by optimally combining precomputed products of filter coefficients with input data, thereby simplifying complex multiplication operations. With streamlined footprint, the DA-LUT-FIR filter offers a significant advancement for efficient signal filtering.

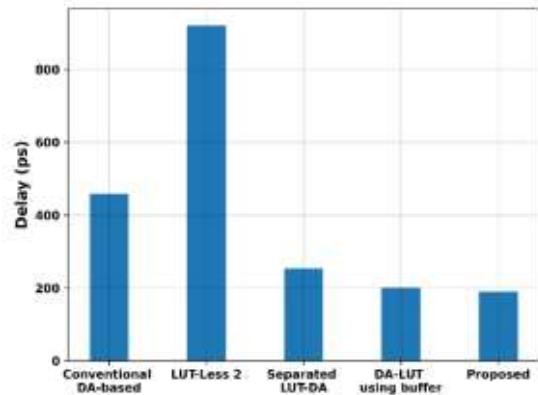


Figure 14: Comparison of Delay

Figure 14 compares the time delay of the proposed DA-LUT-FIR filter model with existing techniques, including the conventional DA-based filter, LUT-Less 2, Separated LUT-DA, and DA-LUT using buffer. The conventional DA-based filter experiences a delay of 459 ps, while LUT-Less 2, Separated LUT-DA, and DA-LUT using buffer have delays of 920 ps, 254 ps, and 201 ps, respectively. In contrast, the proposed approach achieves an impressive delay of just 190 ps. By using the DA-LUT architecture, this innovative filter minimizes the delays typically associated with more resource-intensive FIR filter implementations, showcasing its potential for enhancing performance in time-sensitive scenarios.

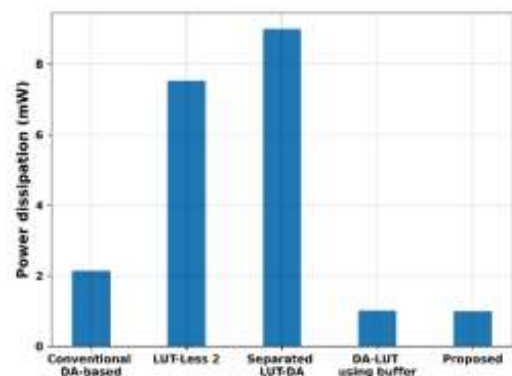


Figure 15: Comparison of Power dissipation

Figure 15 illustrates the power dissipation of the proposed DA-LUT-FIR filter, which is particularly lower than that of traditional FIR filters, making it an appealing option for SDR applications. Existing approaches, including the conventional DA-based filter, LUT-Less 2, Separated LUT-DA, and DA-LUT using buffer, show power dissipation values of 2.14 mW, 7.52 mW, 8.99 mW, and 1.02 mW, respectively. In contrast, the proposed model achieves a power dissipation of just

1 mW. By integrating DA and LUT technologie, the proposed filter minimizes power consumption, making it ideal for power-sensitive environments in SDR applications.

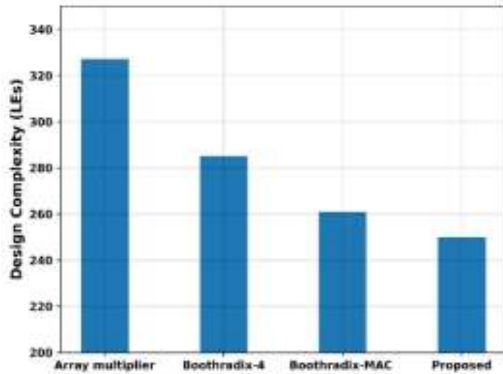


Figure 16: Comparison of Design Complexity

Figure 16 compares the design complexity of the proposed DA-LUT-FIR filter model with existing approaches, including the array multiplier, booth radix-4, and booth radix-MAC unit, which exhibit design complexities of 327 LE, 285 LE, and 261 LE, respectively. Compare to this, the proposed model achieves a design complexity of just 250 LE. This reduction demonstrates the innovative nature of the DA-LUT-FIR filter, significantly decreasing the inherent complexity typically associated with conventional FIR filters and highlighting its efficiency in filter design.

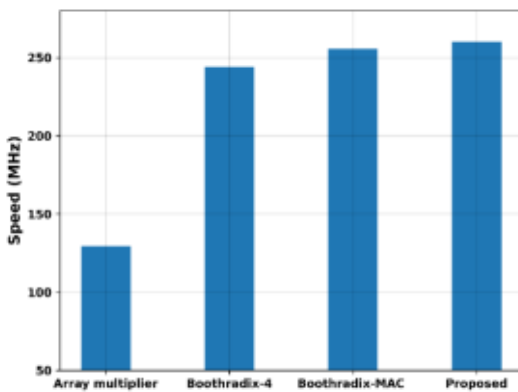


Figure 17: Comparison of processing speed

Figure 17 compares speed of the proposed DA-LUT-FIR filter model with existing models, including the array multiplier, booth radix-4, and booth radix-MAC unit, which achieve speeds of 129.57 MHz, 244.02 MHz, and 255.43 MHz, respectively. The proposed model reaches a speed of 260 MHz. By integrating DA with the LUT approach, it accelerates multiplication operations through precomputed values stored in its LUT, eliminating the need for resource-intensive multipliers.

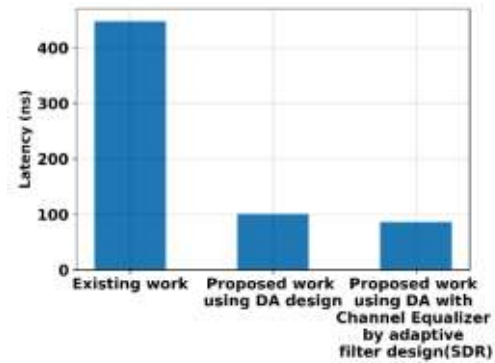


Figure 18: Comparison of latency

Figure 18 provides a comparative analysis of latency in nanoseconds across different FIR filter designs. The baseline latency for 'Existing work' is approximately 448 ns. In contrast, the 'Proposed work using DA with Channel Equalizer by adaptive filter design (SDR)' achieves a remarkable reduction in latency to around 86.126 ns. The 'Proposed work using DA design' shows a slight increase in latency, yet it remains significantly lower than the existing work, with a latency of approximately 101 ns. The proposed designs have significantly reduced latency, making them crucial for applications requiring quick response times.

TABLE I
OVERALL TABLE FOR PERFORMANCE ANALYSIS AND COMPARISON

Parameter	Existing work	Proposed work using DA design	Proposed work using DA with Channel Equalizer by adaptive filter design (SDR)
Block Size	8	8	8
Filter Length	64	64	64
FF	1656	752	952
Delay	56 ns	21.41 ns	9.627 ns
Area (Slices)	839936	6503	8421
Power (Vdd = 1.8V)	251.2 mW	95 mW	95mW
Slice Registers	6144	2062	1182
No of LUT	16,129	15,914	16504
AND/OR gates	190464	14568	9715
Throughput	142.4 Mbps	633.062 Mbps	938.12 Mbps
Frequency	65.7 MHz	78.617 MHz	77.825 MHz
Latency	56ns*8=448 ns	Product of delay and size of data= 12.637ns*8=101.096 ns	Product of delay and size of data=08.527ns*8=86.126 ns
No of adders used	2077	1027	1026
Area-delay product	839936*56ns = 47036416 ns	4145*21.42 ns= 88785.9 ns	8421*9.627 ns = 81068.967 ns
Power-delay product	251.20 mW *56 ns= 14067.2 mW/ns	0.095*21.42 mW = 2.0349 W/ns	0.095*9.627 ns= 0.914565 ns

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

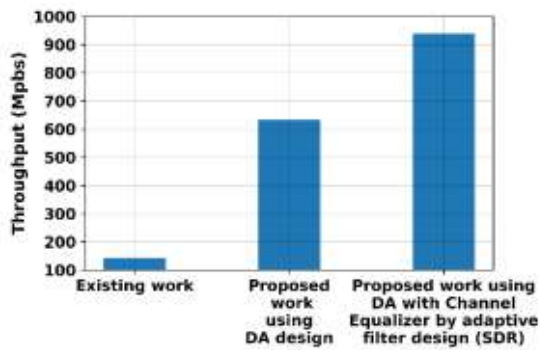


Figure 19: Comparison of Throughput

Figure 19 presents a comparative analysis of throughput performance across three different designs. The Existing work demonstrates modest throughput, significantly below 142.4 Mbps. The 'Proposed work using DA design' shows a substantial improvement, achieving a throughput of 633.062 Mbps. Furthermore, the 'Proposed work using DA with Channel Equalizer by adaptive filter design (SDR)' showcases an impressive throughput close to 938.12 Mbps. The proposed designs significantly improve throughput by integrating an adaptive filter and channel equalization in software-defined radio, doubling the previous models' throughput.

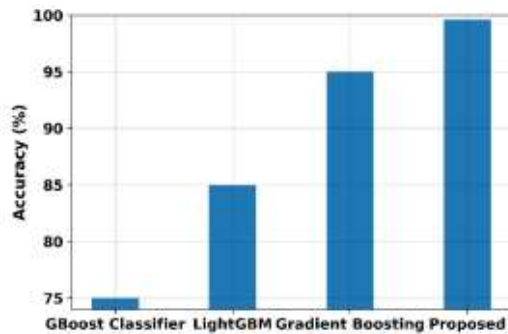


Figure 20: Comparison of Accuracy

Figure 20 compares the accuracy of the proposed DA-LUT-FIR filter model with existing models, including GBoost Classifier, Light GBM and Gradient Boosting, which exhibit accuracy rates of 75%, 85%, and 95%, respectively. The proposed model achieves a significantly higher accuracy of 98%. This improvement underscores the innovative design of the DA-LUT-FIR filter, which not only enhances performance but also minimizes the errors typically associated with traditional filtering methods, highlighting its effectiveness in digital signal processing applications.

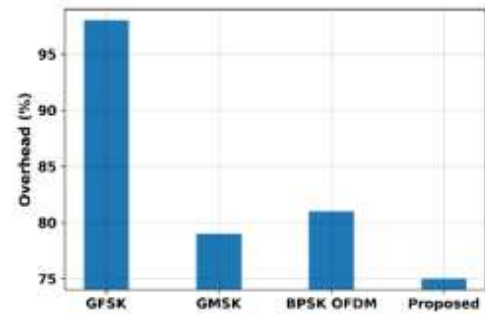


Figure 21: Comparison of Overhead

Figure 21 compares the design complexity of the proposed DA-LUT-FIR filter model with existing models, including GFSK, GMSK and BPSK OFDM, which exhibit design complexities of 98%, 77% and 82% respectively. In contrast, the proposed model achieves a significantly lower overhead of 74%. This reduction demonstrates the innovative nature of the Enhanced Intellectual PMU Controller, significantly decreasing the overhead typically associated with traditional methods and highlighting its efficiency in electric drive applications.

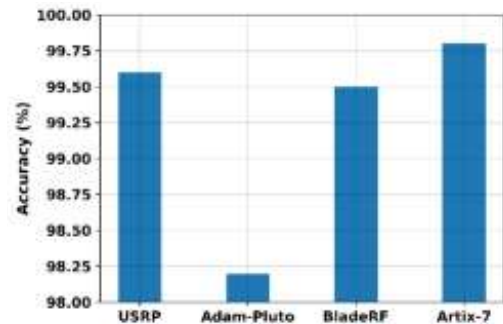


Figure 22: Comparison of accuracy using different FPGA Models

Figure 22 presents an accuracy comparison of various SDR platforms utilizing different FPGA models, including USRP [36], Adalm Pluto [37], and BladeRF [38], which achieve accuracy rates of 99.6%, 98.2%, and 99.5%, respectively. In contrast, the proposed model using the Artix-7 FPGA demonstrates a significantly higher accuracy of 99.8%. This substantial improvement highlights the advanced capabilities of the Artix-7 FPGA in enhancing the performance of SDR applications, effectively minimizing errors associated with traditional models and showcasing its potential in delivering superior digital communication outcomes.

Overall, the proposed models' performance is analyzed and compared with the existing approaches such as conventional DA-based filter, LUT-Less 2, Separated LUT-DA, DA-LUT using buffer array multiplier, boothradix-4, and boothradix-MAC, GBoost Classifier, Light GBM, Gradient Boosting, GFSK, GMSK and BPSK OFD. While comparing the proposed approach with existing models, the proposed approach achieves the best result of delay 190ps, the power dissipation of 1mW, the design complexity attains the value of 250 LE, the processing speed of 260MHz, reduced latency

and overhead of 86 ns and 74%, increased throughput and accuracy of 938.12 Mbps and 99%. Hence the proposed method effectively reduces the noise from SDR applications and enhances the performance of throughput and latency in DA-LUT-based FIR filters.

E. Overall Performance Analysis

The overall performance analysis of the proposed work and its comparison with other existing works is summarized in table 1.

V. CONCLUSION

This study presents a comprehensive evaluation of a novel FIR filter architecture based on Distributed Arithmetic and Look-Up Tables, implemented on an Artix-7 FPGA. This DA-LUT-FIR filter design addresses several key limitations of traditional multiplier-based FIR filters, which often suffer from high hardware complexity, significant power consumption, and slower processing speeds. This proposed filter was implemented with quicker multipliers and adders thereby decreasing bit error rate and latency which in turn helps to boost the throughput of data given in bits. Additionally, the decimation factor frequently changes the FIR filter coefficients, allowing filters to vary their frequency response. According to the experimental findings, fewer LUT for FIR filter coefficients result in less memory usage and latency. The employment of a highly adaptable parallel prefix adder during partial product accumulation was another factor that contributed to the decreased latency. The use of DA and LUTs in the architecture proves to be a powerful combination, delivering remarkable performance improvements and making the filter highly suitable for real-time digital signal processing tasks. The numerical findings from this study—such as the operating speed of 260 MHz, power dissipation of 1 mW, delay of 190 ps, and throughput of 938.12 Mbps—demonstrate substantial improvements over existing methods. These results make the DA-LUT-FIR filter a highly suitable choice for real-time digital signal processing tasks, contributing significantly to the advancement of FIR filter design for future SDR systems.

REFERENCES

- [1] Abraham, J., Venusamy, K., Judice, A., Shaik, H. and Suriyan, K., Research, challenges and opportunities in software define radio technologies. *Int J Reconfigurable & Embedded Syst ISSN, 2089(4864)*, p. 4864. **DOI:** 10.11591/ijres.v12.i2.pp260-268
- [2] Mori, S., Mizutani, K. and Harada, H., 2023. Software-defined radio-based 5G physical layer experimental platform for highly mobile environments. *IEEE Open Journal of Vehicular Technology*, 4, pp. 230–240. **DOI:** 10.1109/OJVT.2023.3237390
- [3] P. P. Sundar, D. Ranjith, T. Karthikeyan, V. V. Kumar, & B. Jeyakumar, "Low power area efficient adaptive FIR filter for hearing aids using distributed arithmetic architecture," *International Journal of Speech Technology*, vol. 23 no. 2, pp. 287–296, 2020. **DOI:** 10.1007/s10772-020-09686-y
- [4] N. J. Grande, & S. Sridevi, "ASIC implementation of shared LUT-based distributed arithmetic in FIR Filter," In *2017 International Conference on microelectronic devices, Circuits and Systems (ICMDCS)*. *IEEE*, pp. 1–4, 2017, August. **DOI:** 10.1109/ICMDCS.2017.8211705
- [5] G. N. Jyothi, K. Sanapala, & A. Vijayalakshmi, "ASIC implementation of distributed arithmetic-based FIR filter using RNS for high-speed DSP systems," *International Journal of Speech Technology*, pp. 1–6, 2020. **DOI:** 10.1007/s10772-020-09683-1
- [6] K. Vijetha, & B. R. Naik, "High-performance area-efficient DA-based FIR filter for concurrent decision feedback equalizer," *International Journal of Speech Technology*, vol. 23, no. 2, pp. 297–303, 2020. **DOI:** 10.1016/j.eswa.2024.123488
- [7] P. Kumar, P. C. Shrivastava, M. Tiwari, & G. R. Mishra, "High-throughput, the area-efficient architecture of 2-D block FIR filter using the distributed arithmetic algorithm," *Circuits, systems, and signal processing*, vol. 38, no. 3, pp. 1099–1113, 2019. **DOI:** 10.1007/s00034-018-0897-2
- [8] M. Sumalatha, P. V. Naganjaneyulu, & K. S. Prasad, "Low power and low area VLSI implementation of Vedic design FIR filter for ECG signal de-noising," *Microprocessors and Microsystems*, vol. 71, p. 102 883, 2019. **DOI:** 10.1016/j.micpro.2019.102883
- [9] K. Bagadi, C. V. Ravikumar, K. Sathish, M. Alibakshikenari, B. S. Virdee, L. Kouhalvandi, K. N. Olan-Núñez, G. Pau, C. H. See, I. Dayou, and P. Livreri, "Detection of signals in MC-CDMA using a novel iterative block decision feedback equalizer," *IEEE Access*, vol. 10, pp. 105 674–105 684, 2022. **DOI:** 10.1109/ACCESS.2022.3211392
- [10] S. R. Rammohan, N. Jayashri, M. A. Bivi, C. K. Nayak, & V. R. Niveditha, "High-performance hardware design of compressor adder in DA-based FIR filters for hearing aids," *International Journal of Speech Technology*, vol. 23, no. 4, pp. 807–814, 2020. **DOI:** 10.1007/s10772-020-09759-y
- [11] S. F. Ghamkhari, & M. B. Ghaznavi-Ghoushchi, "A New Low Power Schema for Stream Processors Front-End with Power-Aware DA-Based FIR Filters by Investigation of Image Transitions Sparsity," *Circuits, Systems, and Signal Processing*, pp. 1–23, 2021. **DOI:** 10.1007/s00034-020-01632-2
- [12] M. E. Meybodi, H. Gomez, Y. C. Lu, H. Shakiba, and A. Sheikholeslami, "Design and implementation of an on-demand maximum-likelihood sequence estimation (MLSE)," *IEEE Open Journal of Circuits and Systems*, vol. 3, pp. 97–108, 2022. **DOI:** 10.1109/OJCS.2022.3173686
- [13] T. V. Padmavathy, S. Saravanan, & M. N. Vimalkumar, "Partial product addition in Vedic design-ripple carry adder design for filter architecture for electrocardiogram (ECG) signal de-noising application," *Microprocessors and Microsystems*, vol. 76, p. 103 113, 2020. **DOI:** 10.1016/j.micpro.2020.103113
- [14] B. Pandey, N. Pandey, A. Kaur, D. A. Hussain, B. Das, & G. S. Tomar, "Scaling of output load in energy efficient FIR filter for green communication on ultra-scale FPGA," *Wireless Personal Communications*, vol. 106, no. 4, pp. 1813–1826, 2019. **DOI:** 10.1007/s11277-018-5717-2
- [15] S. Yergaliyev, and M. T. Akhtar, "A Systematic Review on Distributed Arithmetic-Based Hardware Implementation of Adaptive Digital Filters," *IEEE Access*, 2023. **DOI:** 10.1109/ACCESS.2023.3304234
- [16] W. M. Salama, M. H. Aly, and E. S. Amer, "Underwater optical wireless communication system: Deep learning cnn with noma-based performance analysis," *Optical and Quantum Electronics*, vol. 55, no. 5, p. 436, 2023. **DOI:** 10.1007/s11082-023-04638-7
- [17] P. Chowdari Ch, and J. B. Seventline, "Implementation of distributed arithmetic-based symmetrical 2-D block finite impulse response filter architectures," *F1000Research*, vol. 12, p. 1182, 2023. **DOI:** 10.12688/f1000research.126067.1
- [18] B. U. V. Prashanth, M. R. Ahmed, & M. R. Kounte, "Design and implementation of DA FIR filter for bio-inspired computing architecture," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, p. 1709, 2021. **DOI:** 10.11591/ijece.v11i2.pp1709-1718
- [19] S. Sridevi, & R. Dhuli, "ASIC Implementation of Linear Periodically Time Varying Filter by Thread Decomposition," In *International Conference on Advances in Electrical and Computer Technologies. Singapore: Springer Nature Singapore*, pp. 775–788, 2020, October. **DOI:** 10.1007/978-981-15-9019-1_67
- [20] A. Gorantla, & T. Kudithi, "ASIC Implementation of Linear Equalizer Using Adaptive FIR Filter," *International Journal of e-Collaboration*, vol. 16, no. 4, 2020. **DOI:** 10.1007/978-981-15-9019-1_67

Enhancing Signal Processing Efficiency in Software-Defined Radio Using Distributed Arithmetic and Look-Up Table-Based FIR Filters

[21] P. Kumar, P. C. Shrivastava, M. Tiwari, & G. R. Mishra, "High-throughput, the area-efficient architecture of 2-D block FIR filter using the distributed arithmetic algorithm," *Circuits, systems, and signal processing*, vol. 38, no. 3, pp. 1099–1113, 2019. **doi:** 10.1007/s00034-018-0897-2

[22] A. Rai, "An optimization of low power 4-bit PAL FIR filter using adiabatic techniques," *Sadhana*, vol. 48, no. 2, p. 84, 2023. **doi:** 10.1007/s12046-023-02132-0S

[23] B. U. V. Prashanth, M. R. Ahmed, & M. R. Kounte, "Design and implementation of DA FIR filter for bio-inspired computing architecture," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, p. 1709, 2021. **doi:** 10.11591/ijece.v11i2.pp1709-1718

[24] M. Maamoun, A. Hassani, S. Dahmani, H. Ait Saadi, G. Zerari, N. Chabini, & R. Beguenane, "Efficient FPGA-based architecture for high-order FIR filtering using simultaneous DSP and LUT reduced utilization," *IET Circuits, Devices & Systems*, 2021. **doi:** 10.1049/cds2.12043

[25] P. C. Shrivastava, P. Kumar, M. Tiwari, & A. Dhawan, "Efficient Architecture for the Realization of 2-D Adaptive FIR Filter Using Distributed Arithmetic," *Circuits, Systems, and Signal Processing*, vol. 40, no. 3, pp. 1458–1478, 2021. **doi:** 10.1007/s00034-020-01539-y

[26] G. S. Lakshmaiah, C. K. Narayanappa, L. Shrinivasan, and D. M. Narasimhaiah, "Efficient very large-scale integration architecture design of proportionate-type least mean square adaptive filters," *Int J Reconfigurable & Embedded Syst*, vol. 13, no. 1, pp. 69–75, 2024. **doi:** 10.11591/ijres.v13.i1.pp69-75

[27] M. T. Khan, M. A. Alhartomi, S. Alzahrani, R. A. Shaik, and R. Alsulami, "Two distributed arithmetic based high throughput architectures of non-pipelined LMS adaptive filters," *IEEE Access*, vol. 10, pp. 76 693–76 706, 2022. **doi:** 10.1109/ACCESS.2022.3192619

[28] C. S. Murthy, & K. Sridevi, "Optimized DA-reconfigurable FIR filters for software defined radio channelizer applications," *Circuit World*, vol. 47, no. 3, pp. 252–261, 2021. **doi:** 10.1108/CW-11-2020-0332

[29] S. R. Rammohan, N. Jayashri, M. A. Bivi, C. K. Nayak, & V. R. Niveditha, "High performance hardware design of compressor adder in DA based FIR filters for hearing aids," *International Journal of Speech Technology*, vol. 23, pp. 807–814, 2020. **doi:** 10.1007/s10772-020-09759-y

[30] A. Uma, & P. Kalpana, "ECG Noise Removal Using Modified Distributed Arithmetic Based Finite Impulse Response Filter," *Journal of Medical Imaging and Health Informatics*, vol. 11, no. 5, pp. 1444–1452, 2021. **doi:** 10.1166/jmih.2021.3770

[31] Y. Nirmala, B. Bhaygya, & B. Saimani, "High speed low area OBC DA based decimation filter for hearing aids application". **doi:** 10.1007/s10772-019-09660-3

[32] Șorecău, M., Șorecău, E., Sarbu, A. and Bechet, P., 2023. Real-time statistical measurement of wideband signals based on software defined radio technology. *Electronics*, 12(13), p. 2920. **doi:** 10.3390/electronics12132920

[33] Radu, F., Cotfas, P. A., Alexandru, M., Bălan, T. C., Popescu, V. and Cotfas, D. T., 2023. Signals Intelligence System with Software-Defined Radio. *Applied Sciences*, 13(8), p. 5199. **doi:** 10.3390/app13085199

[34] E. Chitra, & T. Vigneswaran, "An efficient low power and high speed distributed arithmetic design for FIR filter," *Indian Journal of Science and Technology*, vol. 9, no. 4, pp. 1–5, 2016. **doi:** 10.17485/ijst/2016/v9i4/79055

[35] Molla, D. M., Badis, H., George, L. and Berbineau, M., 2022. Software defined radio platforms for wireless technologies. *IEEE Access*, 10, pp. 26 203–26 229. **doi:** 10.1109/ACCESS.2022.3154364

[36] Alashqar, A., Mesleh, R. and Alshawaqfeh, M., 2023, June. Digital Communication Software-Defined Radio-Transceiver Implementation Using MATLAB and USRP. In *2023 International Wireless Communications and Mobile Computing (IWCMC)* (pp. 929–934). IEEE. **doi:** 10.1109/IWCMC58020.2023.10182939

[37] Üngüder, Ö. Ö., 2023. *Real-Time Chat Application with ADALM-PLUTO Software Defined Radio* (Doctoral dissertation, Hochschule Rhein-Waal). <https://opus4.kobv.de/opus4-rhein-waal/frontdoor/index/index/docId/1852>

[38] Terris-Gallego, R., Fernandez-Hernandez, I., López-Salcedo, J. A. and Seco-Granados, G., 2023. E1-E6 SDR Platform Based on BladeRF for Testing Galileo-Assisted Commercial Authentication Service. *Engineering Proceedings*, 54(1), p. 29. **doi:** 10.3390/ENC2023-15428



S. Hari Krishnan obtained his B.E Degree in Electronics & Communication Engineering from Mepco Schlenk Engineering College, Sivakasi, Tamilnadu and M.E Degree in VLSI DESIGN from Karpagam University, Coimbatore, Tamilnadu. His Field of interest includes Digital Signal Processing and VLSI Design. He has organized and attended more number of Conferences, seminars, Workshops, Symposiums, Faculty development programmes, Project expos.. He has published several articles in international and national conferences and also he guided various UG projects. Currently, he is an Associate Professor in the Department of ECE at Sanskrithi School Of Engineering, Puttaparthi, Andra Pradesh, India.



Syed Sadiq Vali has received Bachelor of technology degree (B.Tech) in Electronics and Communication Engineering(ECE) from Sri Krishna Devaraya University, Aanantapur, Andhra Pradesh, in 2017, M.Tech in VLSI system design from JNTU Anantapur in 2021 and pursuing Ph.D degree from mohan Babu University, Tirupati. Currently, he is an Assistant Professor in the department of Electronics and Communication Engineering at Sanskrithi School of Engineering, Puttaparthi, Andra Pradesh, India. Research Interests includes: System-on-Chip Architecture design and Low-Power VLSI systems.