Implementation of FIR Filters through Inner product Units and Parallel Accumulations

P.V.Krishnarao[1], Dasari Vandana[1], Bokkisam Venkata Manasa[1], Dasari Pavanim[1], Guntumadugu Pavani Harshitha[1]

[1]Department of Electronics and Communication Engineering, Geethanjali Institute of Science and Technology, Nellore.

Abstract: Finite Impulse Response (FIR) filters are pivotal in digital signal processing, finding applications in diverse fields like audio processing, telecommunications, and biomedical signal analysis. This work presents an enhanced implementation methodology for FIR filters utilizing inner product computation and parallel accumulations. In the existing, FIR filters are typically implemented using convolution techniques, basic adders, and multipliers, which involve sequential processing and intensive computational resources. This method often leads to latency issues and limits real-time applications. Moreover, traditional implementations suffer from inefficiencies in utilizing hardware resources optimally, leading to suboptimal performance. The proposed methodology overcomes these limitations by leveraging inner product computations and parallel accumulation techniques. By exploiting inherent parallelism in the filtering process, the proposed method significantly reduces latency and enhances throughput.

Keywords: Finite Impulse Response Filters, Parallel Adders, Parallel Multipliers, Sequential processing.

1. Introduction

Implementing FIR filters through inner product units and parallel accumulations represents a powerful technique for efficient and high-performance signal processing. In this approach, the input signal is convolved with the filter coefficients using inner product units, which compute the dot product between the input samples and corresponding filter coefficients. These inner product units are typically implemented using dedicated hardware components such as multipliers and adders, allowing for fast and concurrent computation of filter outputs. By leveraging parallelism, multiple inner product units can operate simultaneously on different segments of the input signal, significantly reducing processing time and improving throughput. Parallel accumulation is another key aspect of FIR filter implementation using inner product units. After computing the dot products between input samples and filter coefficients, the resulting products are accumulated in parallel to generate the filtered output samples. This parallel accumulation process involves summing the products from different inner product units in a coordinated manner, often utilizing dedicated adder trees or pipelined structures to achieve high-speed accumulation. Parallel accumulation not only enhances computational efficiency but also facilitates real-time processing of high-frequency signals by minimizing latency and maximizing throughput. Moreover, the scalability of parallel accumulation allows for the implementation of FIR filters with varying tap lengths and processing requirements, making it suitable for a wide range of signal processing applications.

The use of inner product units and parallel accumulations in FIR filter implementation offers several advantages in terms of performance, resource utilization, and flexibility. By distributing the computational workload across multiple processing units, inner product units enable efficient utilization of hardware resources while maintaining high throughput. Furthermore, the parallel nature of accumulation enables seamless integration with pipelined architectures and parallel processing pipelines, enabling scalable and customizable FIR filter designs. However, the design and optimization of inner product units and parallel accumulation architectures require careful consideration of factors such as hardware complexity, timing constraints, and resource constraints. Addressing these challenges involves exploring novel design methodologies, algorithmic optimizations, and hardware-accelerated techniques to realize FIR filters with optimal performance and efficiency. In essence, FIR filter implementation through inner product units and parallel accumulations represents a sophisticated yet versatile approach to signal processing, offering a pathway towards high-speed, real-time, and resource-efficient filtering solutions in diverse application domains.

2. Literature Survey

Yadav Ranjeeta, et.al [1] implemented FIR Filter by using Han-Carlson Adder. To design Filter different blocks required which are Adders, Multipliers, and Delay elements. FIR Filters are easy to design and are less power consuming. Here for designing of multiplier Han-Carlson Adder used and the filter is designed using proposed multiplier and delay element (D-flip flop**).** Syamala Devi, et.al [2] explained that In today’s digital signal processing (DSP) applications, power optimization is one of the most significant design goals. The digital finite duration impulse response (FIR) filter is recognized as one of the most important components of DSP, and as a result, researchers have done numerous significant studied on the power refinement of the filters. Iqbal, et.al [3] In communication system applications, the need for efficient design and implementation of the finite impulse response (FIR) filter is essential. Realization of such a filter with high bit width is a challenging task. The multiplier and accumulator blocks of the FIR filter take more delay, power, and area. In this literature the design of the FIR filter using the efficient multiplier and accumulator blocks is presented. Gayathri, et.al [4] explained in real life applications the signals are continuously captured, monitored, processed and analysed. The pro cessing and analysis of data is easier if it is in the form of digital. The Digital Signal Processing (DSP) finds importance mainly in biomedical devices or wearable devices. Kumar, et.al [5] designed architecture for a Finite Impulse Response (FIR) filter is proposed in this study to efficiently reduce noise in Electrooculography (EOG) signals. Electromagnetic interference (EMI) and muscular activity are two common sources of noise that regularly alter EOG signals, which are utilized to identify eye movements. Shanthi, et.al [6] proposed that Multiplication and Division Operations have been extensively used as basic elements when designing a system for advanced applications. In today's digital Era speed and area are the main constraints while implementing the digital systems. Many processors use the Carry Select Adder (CSA), one of the faster adders.

Rao, et.al [7] explained Fast FIR algorithm (FFA) produces reduced complexity parallel FIR filtering structure. The FFA can reduce the number of multiplications significantly for large value of filter length N. In this paper we have proposed a new approach to design 2-parallel and 3-parallel (i.e. M= 2 and 3) even length FIR filter with poly phase coefficient symmetry based on FFA. Baker, et.al [8] introduced Finite impulse response (FIR) filter structure based on common operation sharing. Stochastic computing (SC) uses streams of pseudo-random bits to perform low-cost and error-tolerant numerical processing for applications like neural networks and digital filtering. A key operation in these domains is the summation of many hundreds of bit-streams, but existing SC adders are inflexible and unpredictable. Balaji, et.al [9] provided design and implementation of a 4-tap, 8-tap, 16-tap, 32-tap, and 64-tap RNS (Residue Number System) based on efficient and excessive-overall performance FIR filter. RNS mathematics is a prized tool for theoretical investigation of the speed limitations of rapid mathematics.

Some suggested solutions also include a few addition operations; however, using conventional adders will slow down operation and add to the amount of logic gates. Biswas, et.al [10] suggested that Finite Impulse Response (FIR) filters are known for their stability and the reason being widely used over Infinite Impulse Response (IIR) filters. Out of many FIR filters, parallel FIR filters are chosen the best over other filters in digital signal processing. Bhagavatula, et.al [11] demonstrated the usage of forest optimization technique for the determination of optimal parameters for finite impulse response (FIR) filter. Inputs are selected as design specifications; an attempt to apply the forest optimization-based algorithm for the complex nonlinear, constrained optimization task of design the filter has been made. Farag, et.al [12] proposed interpretations enable the employment of CLs to develop finite impulse response (FIR) filters, matched filters (MFs), short-time Fourier transform (STFT), discrete-time Fourier transform (DTFT), and continuous wavelet transform (CWT) algorithms. The main idea is to pre-assign the CL kernel weights to implement a specific convolution- or correlation-based DSP algorithm. Such an approach enables building self-contained DNN models in which CLs are utilized for various preprocessing and feature extractions tasks, enhancing the model portability, and cutting down the preprocessing computational cost.

Palau, et.al [13] introduced literature presents a high-throughput hardware design for the Switchable Loop Restoration Filter (SLRF) of the AOM Video 1 (AV1) video format. This hardware includes the two filters defined at the AV1 SLRF: the Separable Symmetric Normalized Wiener Filter (SSNWF) and the Dual Self- Guided Filter (DSGF). The SLRF is the last step in the AV1 loop restoration filters, and it is used to attenuate blurring artifacts, improving the subjective video quality and the coding efficiency. Li, Wenlu, et.al [14] proposed the optical filter with variable waveform shape using on-chip photonic reservoir computing neural network. The on-chip photonic reservoir computing neural network implements a hardware network directly to mitigate the latency and power-consumption. Fornt, et.al [15] identified solution for high-performance, low-power, intensive computing applications. The error acceptability and resilience of such an inexact solution is very dependent on the application field. In this chapter, a set of well-known approximate adder (TrA, SOA, LOA, GeAr) and multipliers (UDM, BAM, AMB, LM) are presented, and their impact is evaluated in the context of two application examples: FIR digital filters and convolutional neural networks for object detection (YOLO).

3. Proposed Methodology

Figure 1 shows the proposed system architecture. The register unit within a FIR filter indeed stands as a fundamental component in the realm of efficient signal processing. Its significance lies in its ability to coordinate a series of intricately designed steps aimed at transforming raw input data into a finely filtered output. At the heart of this process is the storage mechanism provided by D flip-flops. These flip-flops serve as the initial repository for incoming data, allowing for its retention and manipulation as the filtering process unfolds. This storage capability is crucial, as it enables the subsequent operations within the register unit to access and process the data in a controlled manner. Following the storage phase, the register unit orchestrates a sequence of data shifting operations. This involves moving the stored data through various stages within the filter, where it undergoes manipulation and interaction with coefficients. These coefficients are essential in shaping the filtering characteristics of the FIR filter, determining how the input data is modified to produce the desired output. The generation and application of these coefficients represent a pivotal aspect of the register unit's operation, as they define the filter's behavior and response to different input signals.

One of the notable features of the register unit is its ability to handle multiple streams of data simultaneously in parallel. This parallel processing capability allows the unit to operate on different data streams concurrently, each potentially subjected to different levels of delay. This inherent parallelism not only enhances computational efficiency by distributing the processing workload across multiple pathways but also contributes to minimizing processing latency. This reduction in latency is particularly critical in real-time signal processing applications, where timely and responsive filtering is paramount.

Moreover, the integration of D flip-flops, logical operations, and coefficient generation within the register unit represents a harmonious fusion of digital circuit fundamentals and advanced filtering techniques. This integration leverages the strengths of digital circuitry to implement sophisticated signal processing algorithms efficiently. By combining these elements, the register unit is capable of meticulously orchestrating the steps required to create filtered output streams that faithfully represent the input data while adhering to the defined filter characteristics.

In essence, the FIR filter's register unit serves as a testament to the intersection of digital circuitry and signal processing prowess. Its meticulous orchestration of storage, manipulation, and coefficient application enables the realization of precise and responsive filtering in diverse applications ranging from telecommunications to audio processing. As such, the register unit stands as a cornerstone in the architecture of FIR filters, facilitating the transformation of raw input data into refined output signals with unparalleled accuracy and efficiency.

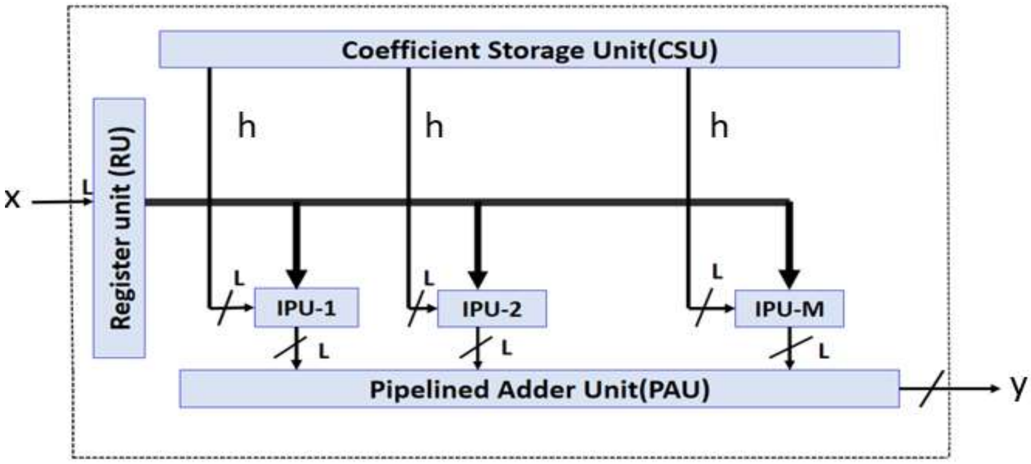


Figure 1. Proposed block diagram.

Step 1: Coefficient Storage Unit with DFFs: The coefficient storage unit with D flip-flops (DFFs) serves as the memory element to store the filter coefficients. Each coefficient represents the weight applied to its corresponding input sample during the filtering process. DFFs are chosen for their simplicity and ability to store binary data.

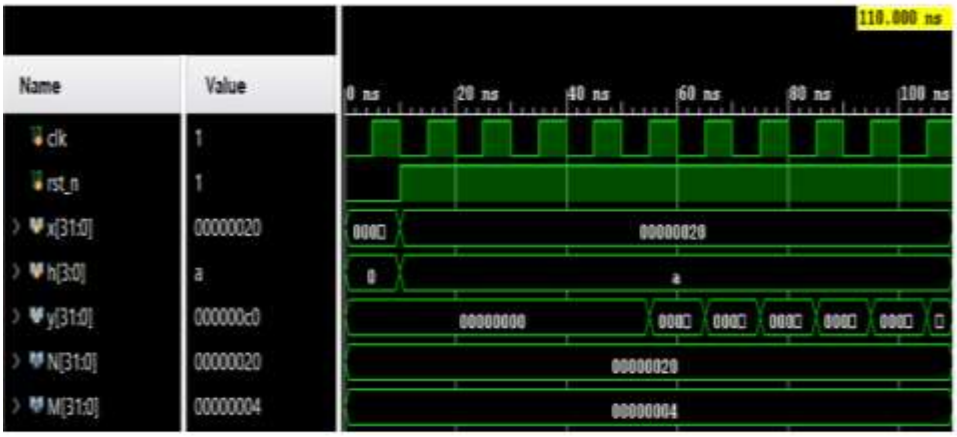
Step 2: Register Unit with DFFs: The register unit with DFFs serves as a buffer to temporarily store input samples before processing. This unit facilitates block processing by accumulating a block of input samples before passing them to the multiplier/accumulator units for filtering. The operational procedure begins with the storage of incoming samples in the DFFs within the register unit.

Step 3: Inner Product Unit with Inner Product Cells: The inner product unit calculates the inner product between the input samples and the filter coefficients. It comprises multiple inner product cells, each responsible for computing the product of a sample and its corresponding coefficient. The operational procedure of the inner product unit involves distributing the input samples and filter coefficients to the inner product cells. Each inner product cell multiplies its assigned sample by the corresponding coefficient, yielding a partial product. These partial products are then accumulated to compute the inner product.

Step 4: Pipelined Adder Unit: The pipelined adder unit performs the accumulation of partial products generated by the inner product unit. It consists of a series of pipelined adders interconnected to efficiently sum the partial products and produce the filtered output.The operational procedure of the pipelined adder unit involves the staged addition of partial products as they propagate through the pipeline. Each stage of the pipeline performs a partial sum, which is then passed to the next stage for further accumulation.

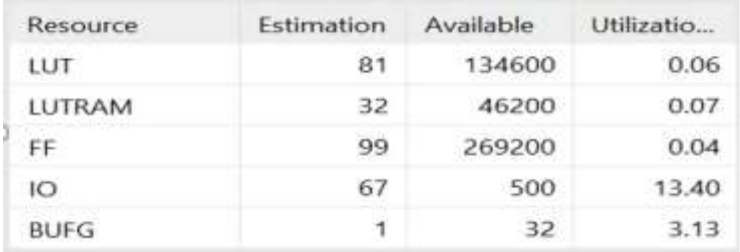
4. Results and Discussion

Figure 2 likely represents the simulation outcome of applying the FIR filter to a signal or set of signals. Figure 3 probably presents a design summary of the FIR filter, detailing its characteristics. Figure 4 likely offers a power summary, which could refer to the power consumption or efficiency metrics of the FIR filter implementation. Figure 5 appears to depict a time summary, potentially showcasing the time-related metrics of the FIR filter, such as processing time per sample or overall processing time for a given signal. Table 1 is mentioned but without detailed contents. However, it's reasonable to assume that it presents a comparison table, likely comparing the performance of the FIR filter with other FIR filter types.



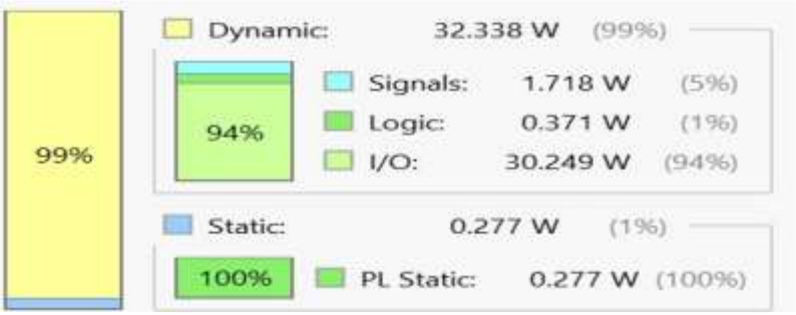
Name	Value
clk	1
rst_n	1
x[31:0]	00000020
h[3:0]	a
y[31:0]	000000c0
N[31:0]	00000020
M[31:0]	00000004

Figure 2. Simulation outcome



Resource	Estimation	Available	Utilizatio...
LUT	81	134600	0.06
LUTRAM	32	46200	0.07
FF	99	269200	0.04
IO	67	500	13.40
BUFG	1	32	3.13

Figure 3. Design summary



Dynamic: 32.338 W (99%)

Signals: 1.718 W (5%)

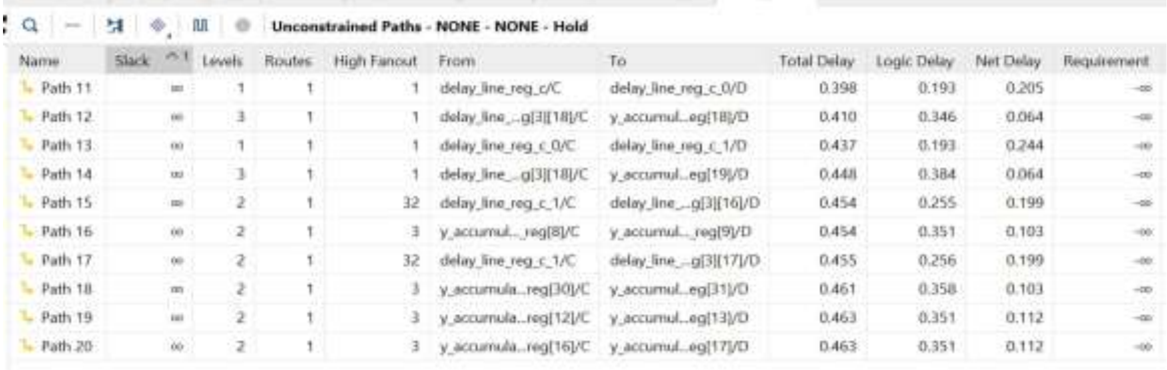
Logic: 0.371 W (1%)

I/O: 30.249 W (94%)

Static: 0.277 W (1%)

PL Static: 0.277 W (100%)

Figure 4. Power summary

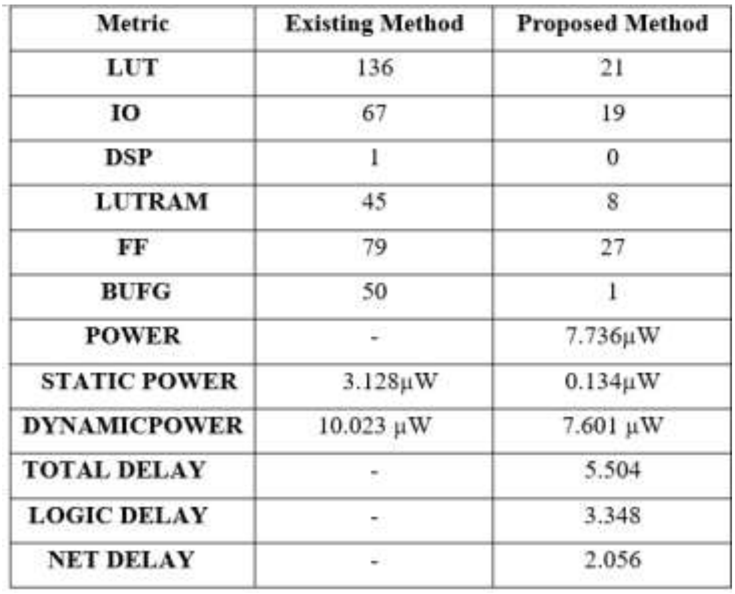


Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 11	∞	1	1	1	delay_line_reg_c/C	delay_line_reg_c_0/D	0.398	0.193	0.205	-∞
Path 12	∞	3	1	1	delay_line_...g[3][18]/C	y_accumul...eg[18]/D	0.410	0.346	0.064	-∞
Path 13	∞	1	1	1	delay_line_reg_c_0/C	delay_line_reg_c_1/D	0.437	0.193	0.244	-∞
Path 14	∞	3	1	1	delay_line_...g[3][18]/C	y_accumul...eg[19]/D	0.448	0.384	0.064	-∞
Path 15	∞	2	1	32	delay_line_reg_c_1/C	delay_line_...g[3][16]/D	0.454	0.255	0.199	-∞
Path 16	∞	2	1	3	y_accumul..._reg[8]/C	y_accumul..._reg[9]/D	0.454	0.351	0.103	-∞
Path 17	∞	2	1	32	delay_line_reg_c_1/C	delay_line_...g[3][17]/D	0.455	0.256	0.199	-∞
Path 18	∞	2	1	3	y_accumula...reg[30]/C	y_accumul...eg[31]/D	0.461	0.358	0.103	-∞
Path 19	∞	2	1	3	y_accumula...reg[12]/C	y_accumul...eg[13]/D	0.463	0.351	0.112	-∞
Path 20	∞	2	1	3	y_accumula...reg[16]/C	y_accumul...eg[17]/D	0.463	0.351	0.112	-∞

Figure 5. Time summary.

Table 1. Comparison Table



Metric	Existing Method	Proposed Method
LUT	136	21
IO	67	19
DSP	1	0
LUTRAM	45	8
FF	79	27
BUFG	50	1
POWER	-	7.736μW
STATIC POWER	3.128μW	0.134μW
DYNAMICPOWER	10.023 μW	7.601 μW
TOTAL DELAY	-	5.504
LOGIC DELAY	-	3.348
NET DELAY	-	2.056

5. Conclusion

In conclusion, the register unit within FIR filter serves as the cornerstone of efficient signal processing, orchestrating a series of meticulously designed steps to transform input data into a filtered output. From the initial storage of input data in D flip-flops to the subsequent data shifting operations and generation of coefficients, each step in the register unit's operation is geared towards achieving precise and tailored filtering. The unit's ability to simultaneously process multiple streams of data in parallel, each subjected to different levels of delay, underscores its role as an adept parallel processing entity within the FIR filter architecture. This parallelism not only enhances computational efficiency but also contributes to minimizing processing latency, a critical aspect in real-time signal processing applications. The integration of D flip-flops, logical operations, and coefficient generation showcases a harmonious blend of digital circuit fundamentals and advanced filtering techniques. In essence, the register unit's meticulous orchestration of these steps facilitates the creation of filtered output streams that faithfully represent the input data while accommodating the defined filter characteristics. Thus, the FIR filter's register unit stands as a testament to the intersection of digital circuitry and signal processing prowess, playing a pivotal role in the realization of precise and responsive filtering in diverse applications ranging from telecommunications to audio processing.

References

[1]. Yadav Ranjeeta, Surekh Ghangas, Krishna Kumar Verma, Divyanshu Joshi, Harsit Yadav, and Anmol Dev. "IMPLEMENTATION OF EFFICIENT FIR FILTER." International Development Planning Review 22, no. 2 (2023): 9-20.

[2]. Syamala Devi, P., D.Vishnupriya, G. Shirisha, Venkata Tharun Reddy Gandham, and Siva Ram Mallela. "Design of High Efficiency FIR Filters by Using Booth Multiplier and Data-Driven Clock Gating and Multibit Flip-Flops." In International Conference on Communications and Cyber Physical Engineering 2018, pp. 319-326. Singapore: Springer Nature Singapore, 2023.

[3]. Iqbal, JL Mazher, G. Narayan, T. Manikandan, M. Meena, and Jose Anand. "Low power and low area multiplier and accumulator block for efficient implementation of FIR filter." In Low Power Designs in Nanodevices and Circuits for Emerging Applications, pp. 267-282. CRC Press.

[4]. Gayathri, S., S. Esha, Challa Bhavya, and Yasha Jyothi M. Shirur. "Design and Implementation of Arithmetic based FIR Filters for DSP Application." In 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), pp. 782-787. IEEE, 2023.

[5]. Kumar, A. Ramesh, Aruru Sai Kumar, K. Hemanth Lakshmi Phani Prasad, B. Sriraj, and P. Raja Rajasri. "High performance FIR Architecture for EOG Signal Noise Supression." In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-6. IEEE, 2023.

[6]. Shanthi, G., Aruru Sai Kumar, Md Masood Hasan, H. Tanuja, and Ch Yashwanth. "An Efficient and High Speed FIR Filter using BEC with MUX Technique." In 2023 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), pp. 256-262. IEEE, 2023.

[7]. Rao, K. Anjali, and Neetesh Purohit. "Hardware Efficient 2-Parallel and 3-Parallel Even Length FIR Filters Using FFA." In 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), pp. 1-5. IEEE, 2023.

[8]. Baker, Timothy, and John P. Hayes. "Design of Large-Scale Stochastic Computing Adders and their Anomalous Behavior." In 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1-6. IEEE, 2023.

[9]. Balaji, M., N. Padmaja, P. Gitanjali, Saif Ali Shaik, and Siva Kumar. "Design of FIR filter with Fast Adders and Fast Multipliers using RNS Algorithm." In 2023 4th International Conference for Emerging Technology (INCET), pp. 1-6. IEEE, 2023.

[10]. Biswas, Neelesh, Supriya Dhabal, and Palaniandavar Venkateswaran. "Analysis of Area Efficient Parallel FIR Filters using FPGA." In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-5. IEEE, 2023.

[11]. Bhagavatula, Venkata Vaibhav, and S. V. N. L. Lalitha. "Optimization of FIR filter parameters using forest optimization algorithm." In AIP Conference Proceedings, vol. 2512, no. 1. AIP Publishing, 2024.

[12]. Farag, Mohammed M. "Design and Analysis of Convolutional Neural Layers: A Signal Processing Perspective." IEEE Access 11 (2023): 27641-27661.

[13]. Palau, Roberta de Carvalho Nobre, Wagner Penny, Ramiro Viana, Jones Goebel, Guilherme Correa, Marcelo Porto, and Luciano Agostini. "High-Throughput Hardware Design for the AV1 Decoder Switchable Loop Restoration Filters." Journal of Integrated Circuits and Systems 18, no. 1 (2023): 1-12.

[14]. U. Penchalaiah and V. S. Kumar, "Design and Implementation of Low Power and Area Efficient Architecture for High Performance ALU", Parallel Processing Letters., vol. 32, no. 01n02, pp. 2150017, 2022.

[15].

[16]. Li, Wenlu, Li Pei, Bing Bai, Jianshuai Wang, and Jingjing Zheng. "Optical filter with on-chip photonic reservoir computing neural network." In 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), pp. 1251-1255. IEEE, 2023.

[17]. Fornt, Jordi, Leixin Jin, Josep Altet, Francesc Moll, and Antonio Rubio. "Evaluation of the Functional Impact of Approximate Arithmetic Circuits on Two Application Examples." In Design and Applications of Emerging Computer Systems, pp. 421-451. Cham: Springer Nature Switzerland, 2024.