E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

CVR Journal of Science and Technology, Volume 21, December 2021
DOI: 10.32377/cvrjst2115

A High-Performance FIR Filter Architecture using Symmetry and Distributed Arithmetic Logic

Venkata Krishna Odugu[1], Janardhana Rao Bitra[2] , and Satish Bojjawar[3]
[1]Assoc. Professor, CVR College of Engineering/ECE Department, Hyderabad, India
Email: venkatakrishna.odugu@gmail.com
[2]Assoc. Professor, CVR College of Engineering/ECE Department, Hyderabad, India
Email: janardhan.bitra@gmail.com
[3]Assoc. Professor, CVR College of Engineering/EIE Department, Hyderabad, India
Email: satishbojjawar@gmail.com

Abstract: **In this paper, the symmetry type Finite Impulse Response (FIR) filters are described using distributed arithmetic (DA) logic. The direct-form type filter architecture is considered for the implementation. Direct-form is requires a fewer number of registers than transposed form architecture. The symmetry in the filter coefficients decreases the number of multipliers. Due to the reduction of multipliers, the area and power parameters are reduced by 50%, because multipliers complex block in terms of area and power. Further, the multipliers are completely replaced with shift and add operations to reduce the complexity using DA. In this work, the higher-order filters are decomposed into smaller filter blocks to reduce the Look-up-table (LUT) complexity. The proposed DA-based symmetry FIR filter is validated using FPGA and synthesized using Genus tools from Cadence in 90nm CMOS technology. The physical design is done using Innovus tools from Cadence and the layout of the proposed architecture is generated. The utilization of hardware blocks, delay, area, and power consumption parameters are compared with existing FIR filter architectures.**

Index Terms: **FIR filter, Distributed Arithmetic, symmetry filters, FPGA, and LUT.**

I. Introduction

Digital filters are two types as Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The FIR filter is mostly used in many signal processing applications due to the stability, ease of design. The FIR filters can be implemented in direct type or transposed type architectures. Any filter architecture consists of delay elements, adders, and multipliers. From these blocks, the multiplier is a more power-consuming block and complex block. Hence, multiplier blocks are replaced with adders and shifters using Distributed Arithmetic (DA) logic. The DA logic-based multiplication process in the filter decreases the area, power, and delay.

In the work [1], a memoryless DA-based FIR filter is proposed for hearing aid applications using compressor circuits. An approximate 4:2 compressor is incorporated in the filter and mux-based logic is used to reduce the memory, area, and power consumption. In [2], Application Specific Integrated Circuit (ASIC) implementation of FIR filters is described using shared Look-Up Table (LUT) based DA logic to improve the VLSI performance metrics. The shared LUT concept reduces the hardware complexity and memory.

Park et al. [3] suggested Field Programmable Gate Array (FPGA) and ASIC-based architectures for reconfigurable FIR filters using DA logic. In this work, the throughput is improved using block processing and power consumption is reduced by the DA concept. The parallel DA-based low power The FIR filter architecture is proposed by Khan et al. in [4]. The bit-level parallel processing is used in DA-based filters to improve the speed, and the proposed filter architecture is compared with existing (Multiply and Accumulate) MAC-based filter architectures.

In the research paper [5], a novel DA-based FIR filter for efficient decision feedback equalizer is proposed. In this work, an approximate compressor is used to reduce the area and mux-based logic is considered in the filter to improve the speed. In the paper [6], two DA-based adaptive FIR filters are explained using smart modified LUTs to store the filter coefficients to improve the area, delay, and power parameter

A transposed type FIR filter is demonstrated in [7]. In this work, the conventional LUT is split into several ROM-LUTs to provide parallelism in the DA-based filter. Prakash et al. [8] proposed an efficient least mean square adaptive filter using the DA technique. In this paper, the coefficient storage unit is updated using offset binary coding (OBC) to improve the speed and throughput. In [9], a block-based FIR filter using DA logic is described for software-defined radio applications. The throughput and VLSI design metrics are improved by this proposed method. Park et al. [10] proposed a reconfigurable FIR filter based on the DA method. A shared LUT and distributed RAM-based techniques are incorporated in the proposed filter to improve the performance metrics.

The symmetry-based implementation of the FIR filter using DA techniques is proposed in this paper. The higher-order filter implementation using DA increases the complexity of the LUT. To overcome this, the filter is divided into filter blocks of L, and DA is applied to each filter block and the individual outputs are determined in parallel and the final output is computed by adding all filter blocks outputs using an adder tree along with appropriate shifters.

The rest of the paper is organized as follows: Section 2 describes the concept of DA logic and corresponding equations. The proposed architecture implementation is demonstrated in section 3. Section 4 explains the analysis of results. The paper is concluded in section 5.

Received on 28-03-2021, Revised on 04-05-2021, Accepted on 17-06-2021.

CVR Journal of Science and Technology, Volume 21, December 2021
DOI: 10.32377/cvrjst2115

A. Background work

The algebraic sum of convoluted impulse response coefficients and input samples is FIR filter output. The basic FIR filter output is expressed by (1).

$$y = \sum_{i=0}^{N-1} C_i X_i \tag{1}$$

Here, C_i is filter known coefficients and X_i is the input sample of the filter. The above equation is represented in the block diagram and shown in Fig. 1. A linear phase FIR filter is symmetric about the center value. This symmetry in the FIR filter reduces the number of power hunger multipliers required for the realization of the filter architecture. The symmetry type 10-tap linear phase FIR filter architecture is shown in Fig. 2.

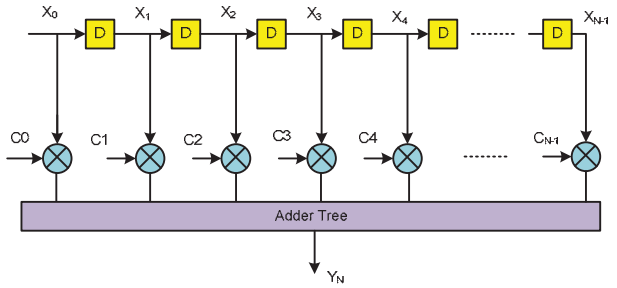

Figure 1. The general architecture of the N-tap FIR filter.

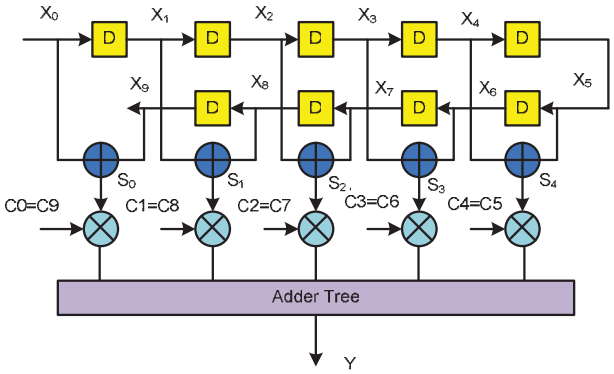

Figure 2. 10-tap symmetry type FIR filter

In the Symmetry filter, the input samples to be multiplied with the same coefficients are pre-summed before multiplication. Next, all the inner products are summed by the adder tree and produce the final filter output. The expanded filter output is represented in (2). In the architecture, the outputs of the adders are denoted by S_0, S_1, S_2, S_3, and S_4.

$$Y = C_0\,S_0 + C_1\,S_1 + C_2\,S_2 + C_3\,S_3 + C_4\,S_4 \tag{2}$$

II. Distributed Arithmetic Concept

The basic filter equation of the FIR filter is modified corresponding to the DA technique. From the input samples or coefficients, one of them is considered and modified into bits representation as shown in (3). The input sample is Xi is represented as,

$$Xi = \sum_{b=0}^{B-1} X_i\,[b] \times 2^b \tag{3}$$

Where $X_i\,[b] \in \{0,1\}$ and denotes the b[th] bit of input sample Xi. The input may consist of B-bits. Then the filter output expression (1) is modified as (4). The rearrangement is applied in (4) then the DA-based FIR filter output is shown in (5).

$$y = \sum_{i=0}^{N-1} C_i \times \sum_{b=0}^{B-1} X_i\,[b] \times 2^b \tag{4}$$

$$y = \sum_{b=0}^{B-1} 2^b \times \sum_{i=0}^{N-1} X_i\,[b] \times C_i \tag{5}$$

The simple DA-based FIR filter architecture corresponding to the above equation is represented in Fig. 3.

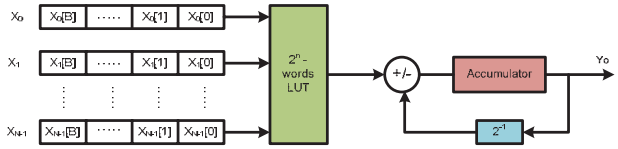

Figure 3. Basic DA based FIR filter

The DA-based filter architecture consists of shift registers, LUT, shifter, accumulator, and adder/subtractors. The parallel in serial out shift registers are required to feed the input sample bit-by-bit to the coefficient storage unit is LUT.

TABLE I.
PRECOMPUTED COEFFICIENTS STORED IN THE LUT FOR N = 4

S. No	Address Input bits	The stored value in LUT
1	0000	0
2	0001	C0
3	0010	C1
4	0011	C1+C0
5	0100	C2
6	0101	C2+C0
7	0110	C2+C1
8	0111	C2+C1+C0
9	1000	C3
10	1001	C3+C0
11	1010	C3+C1
12	1011	C3+C1+C0
13	1100	C3+C2
14	1101	C3+C2+C0
15	1110	C3+C2+C1
16	1111	C3+C2+C1+C0

CVR Journal of Science and Technology, Volume 21, December 2021
DOI: 10.32377/cvrjst2115

The LUT is a memory block, which is used to store the precomputed values of filter coefficients. The size of the LUT depends on the order of the filter N. The LUT storage elements for N = 4 are shown in Table I. The input bits from each sample are considered as the address of the LUT and the corresponding data stored in the LUT is fetched and given as the partial product. The partial product output is fed to the adder, which is added by the previous accumulated output. The previous output is generated by the shifter and is denoted by 2^{-1}.

III. Proposed FIR Filter Architecture using DA

In this section, the symmetry-based FIR filter architecture using the DA technique is described. If an order of the filter N is high, then the size of LUT becomes 2^N - words. For example, N =16, then 65536 words are required and it is very complex to implement the LUT. Hence, the N-th order filter is divided into L filter blocks. The individual filter blocks are transformed into DA form and filter block outputs are determined and the final output is computed by adding up all filter block outputs. The DA-based filter output equation is modified as (6).

$$y = \sum_{i=0}^{L-1} C_i X_i + \sum_{i=L}^{2L-1} C_i X_i + \sum_{i=2L}^{3L-1} C_i X_i + .. + \sum_{i=(\frac{N}{L}-1)}^{N-1} C_i X_i \quad (6)$$

The proposed symmetry FIR filter after decomposition into filter blocks is shown in Fig. 4. In the symmetry FIR filter after the addition of the same coefficients, the signals are denoted by $\{S_0, S_1, S_2, ... S_{N-1}\}$. For higher-order filters, these samples are decomposed into L blocks. The samples $\{S_0, S_1, \ldots S_L\}$ are fed to one LUT, which consists of 2^L words. Here the LUT size is reduced from 2^N to 2^L words. Table II compares the size of the filters with and without division of filter.

TABLE II.
. Comparison of LUT size for FIR filters

DA Filter	LUT size	Total Memory Space
Normal filter with no division	2^N words	2^N words
Filter with division into N/L blocks	2^L words	2^L x N/L words

The proposed FIR filter architecture consists of L number of filter blocks. Each decomposed filter block has an L number of inputs. The L number of input samples is given as inputs to the filter block. The internal structure of the filter block is shown in Fig. 5. The iteration process of DA filter logic in the filter block is represented in this figure. The L number of samples and each sample is represented by B bits. From L samples, bit-by-bit concatenated data is considered as the address and corresponding partial product stored in the LUT. The output of the LUT is shifted by the multiplication of 2^B, 2^{B-1}, and so on. Finally, all the individual outputs are accumulated and filter block output is computed. Similarly, all the individual filter blocks are generated and summated by the adder tree. If the order of the filter is large, then the complexity of the adder tree also increases.

In the adder tree, all the shifter's outputs need to be added. The shifted values in each level increase the number of bits in the outputs and the complexity of the adder tree is high and each summand requires large and different word lengths. To overcome this complexity, the shifters are embedded along with the adder tree levels. The shifters placing in each level of the adder tree are shown in Fig. 6.

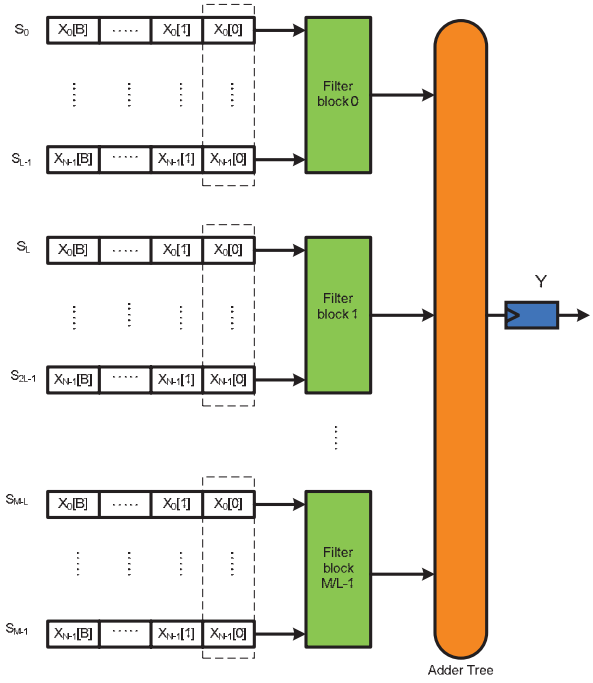

Figure 4. Higher-order Symmetry type FIR filter using DA

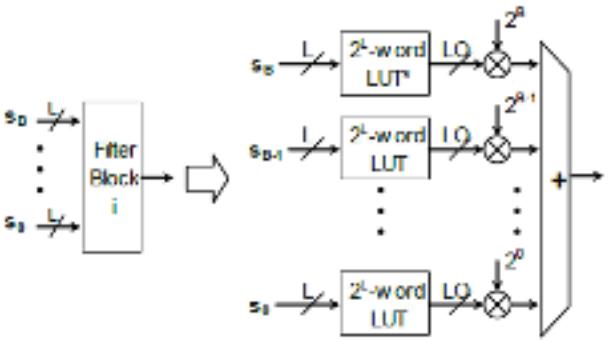

Figure 5. The internal structure of the filter block

IV. Results

The complexity of the filter architecture depends on the parameters of the filter, such as the number of taps, number of bits for filter coefficients, number of bits for input sample, and input size of LUT. In this work, FIR filter architectures are implemented using the proposed method for the order of N = 8, 16, and 32. The number of input samples and coefficients are denoted by 8-bits for all orders of the filters. These filters are decomposed with a factor or block size of L = 4, then the input size of the LUT is 4.

CVR Journal of Science and Technology, Volume 21, December 2021
DOI: 10.32377/cvrjst2115

The developed FIR filters are coded in Verilog HDL. The code is validated and synthesized employing FPGA as the target device in Xilinx tools. The utilization summary of all these filters is analyzed and compared in table III in terms of slice LUTs, slice registers, LUT-FF pairs, and adders. It can be noted that the filter architecture with symmetry required a fewer number of hardware blocks than filters without symmetry.

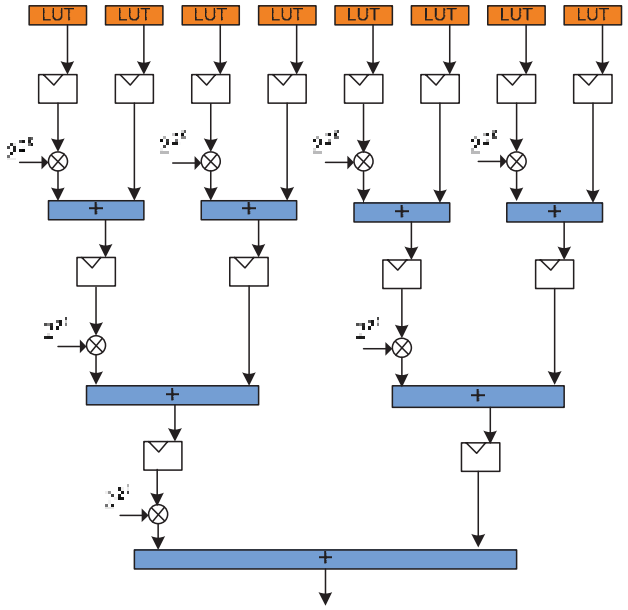

Figure 6. Adder tree and embed shifters in each level.

Next, the same Verilog codes are synthesized on the platform of ASIC design. The Genus tools are used to synthesize the design in CMOS 90nm library, and area, delay, and power reports are generated. The physical design is carried out by Innovus tools from Cadence. The final physical design layout of the chip is generated. The comparison of area, delay, and power values computed by synthesis tool for the proposed architecture with various order of the filters are presented in Table IV.

TABLE III.
FPGA UTILIZATION SUMMARY

Type of filter	Order	Slice Reg.	Slice LUT	No of LUT-FF pairs	Adder
Filter without symmetry	8	2582	98	34	67
	16	3562	146	58	132
	32	8759	193	62	215
Proposed filter with symmetry	8	2038	85	26	77
	16	3088	132	43	148
	32	7238	178	59	230

TABLE IV.
COMPARISON OF VLSI DESIGN METRICS

	order	Area (μm^2)	Delay (ns)	Power (mW)	ADP ($\mu m^2.\mu s$)
Proposed filter with symmetry	8	5965	8.256	1.956	11.66
	16	7986	8.986	2.489	19.87
	32	11255	9.012	3.089	34.76

The graphical analysis of delay and power consumption values of developed architecture with different values of N is shown in Fig. 7. The area comparison in the form of a bar chart of the proposed filter for N = 8, 16, and 32 is shown in Fig. 8.

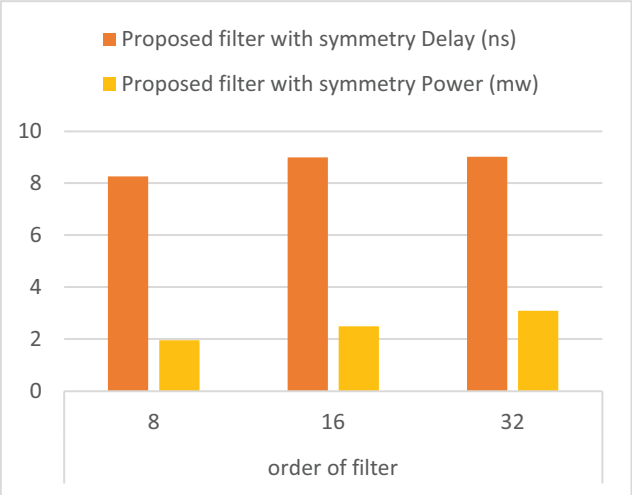

Figure 7. Delay and power values comparison with different order of the filter

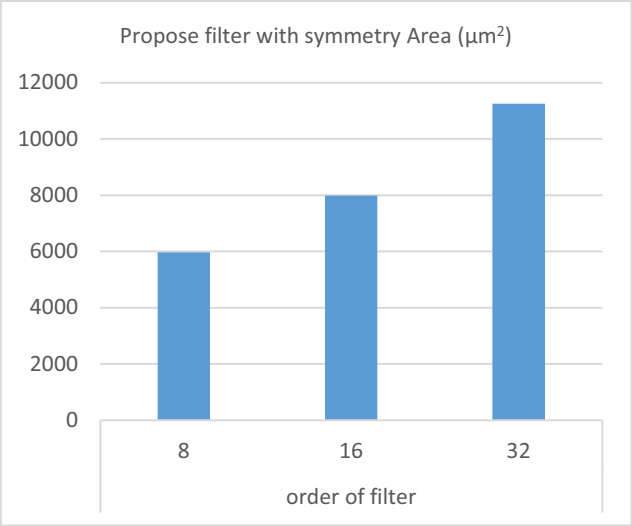

Figure 8. Area comparison with different order of the filter

CVR Journal of Science and Technology, Volume 21, December 2021
DOI: 10.32377/cvrjst2115

The VLSI design metrics of the proposed FIR filter architecture reported by ASIC design tools are compared with state-of-artworks in Table V. The graphical comparison analysis of delay-power, and area of proposed architecture with existing architectures are shown in Fig. 9 and Fig. 10 respectively. The final physical design layout is generated by Innovus tools presented in Fig.11.

TABLE V.
COMPARISON OF AREA, DELAY, AND POWER OF PROPOSED FILTER WITH STATE-OF-ARTWORKS.

Various works	order	Area (μm^2)	Delay (ns)	Power (mW)
Naga Jyothi et al. [9]	16	18956	7.45	16.1
Naga Jyothi et al. [10]	16	16688	1.38	9.42
Meher et al. [11]	8	14855	5.20	6.81
	16	29667	5.60	12.08
	32	59244	5.60	24.00
Proposed filter	8	9965	4.256	3.956
	16	12986	4.986	6.489
	32	17125	5.012	11.089

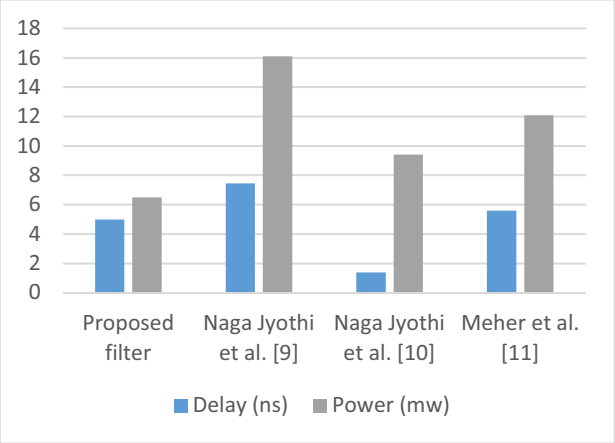

Figure 9. Delay and power comparison of the proposed filter with existing filters.

Figure 10. Area comparison with existing filter architectures

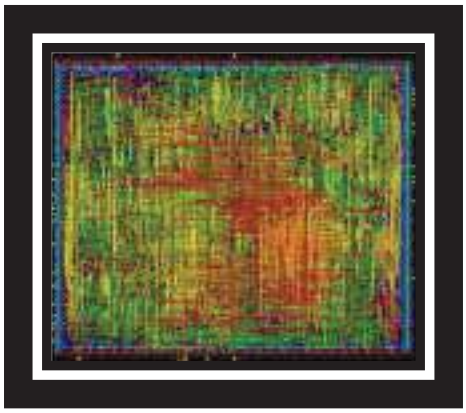

Figure 11. Physical design layout of proposed FIR filter architecture in 90 nm technology.

V. CONCLUSIONS

In this paper, an area, delay, and power-efficient DA-based FIR filters are implemented in the ASIC platform. The symmetry in the filter coefficients reduces the number of multipliers required for the filter architecture. Further, the multiplier complexity is decreased using the DA technique. In the DA-based filters, the multiplication is carried out by shift and add operations only using memory element LUT.

The Large size LUT is required for higher-order filters. This complexity of the LUT is overcome by the decomposition of the filter into an L number of filter blocks. In each filter block, DA formation is applied and the final output is generated using an optimized adder tree. The FIR filter architecture is implemented for N = 8, 16, and 32 using ASIC-based tools Genus and Innovus from Cadence. The area, delay, and power parameters of the proposed architecture are compared with state-of–artworks. The same design is validated and synthesized using Xilinx tools for the target device is FPGA. The utilization summary of HDL synthesis is also compared.

REFERENCES

[1] Rammohan, S. Radha, et al. "High-performance hardware design of compressor adder in DA based FIR filters for hearing aids." *International Journal of Speech Technology* 23.4, 807-814, 2020.

[2] Grande, N.J., Sridevi, S. Asic implementation of shared LUT based distributed arithmetic in fir filter. *In 2017 International Conference on Microelectronic Devices, Circuits and Systems* (ICMDCS), pp. 1–4. IEEE, 2017.

[3] Park, S. Y., & Meher, P. K. (2014). Efficient FPGA and ASIC realizations of a da-based reconfigurable fir digital filter. *IEEE Transactions on Circuits and Systems II*, *61*(7), 511–515.

[4] Khan, Shaheen, and Zainul Abdin Jaffery. "Low power FIR filter implementation on FPGA using parallel Distributed Arithmetic." *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015.

[5] Naga Jyothi, Grande, and Sriadibhatla Sridevi. "High speed and low area decision feed-back equalizer with novel memory less distributed arithmetic filter." *Multimedia Tools and Applications* 78.23 (2019): 32679-32693.

[6] Guo, Rui, and Linda S. DeBrunner. "Two high-performance adaptive filter implementation schemes using distributed

arithmetic." *IEEE Transactions on Circuits and Systems II: Express Briefs* 58.9 (2011): 600-604.

[7] Upadhyay, Ankit, and Uday Panwar. "High-Performance VLSI Architecture for Transpose Form FIR Filter using Integrated Module." *2018 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2018.

[8] Prakash, M. Surya, and Rafi Ahamed Shaik. "Low-area and high-throughput architecture for an adaptive filter using distributed arithmetic." *IEEE Transactions on Circuits and Systems II: Express Briefs* 60.11 (2013): 781-785.

[9] Jyothi, Grande Naga, Anusha Gorantla, and Thirumalesu Kudithi. "Asic implementation of linear equalizer using adaptive fir filter." *International Journal of e-Collaboration (IJeC)* 16.4 (2020): 59-71.

[10] Mohanty, Basant Kumar, et al. "A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic." *Integration* 54 (2016): 37-46.

[11] Park, Sang Yoon, and Pramod Kumar Meher. "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter." IEEE Transactions on Circuits and Systems II: Express Briefs 61.7 (2014): 511-515.

[12] NagaJyothi, Grande, and Sriadibhatla Sridevi. "High speed low area OBC DA based decimation filter for hearing aids application." *International Journal of Speech Technology* 23.1 (2020): 111-121.

[13] Meher, P. K. (2006). Hardware-efficient systolization of DA-based calculation of finite digital convolution. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *53*(8), 707–711.

[14] Meher, P. K., & Park, S. Y. (2011). High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic. *In VLSI and System-on-Chip (VLSI-SoC),* 2011 *IEEE/IFIP 19th International Conference on*, pp. 428–433. IEEE.