

Implementation of Efficient FIR Filter Architectures using Distributed Arithmetic Computation for Digital Channelizer of SDR

Kiran Kumar Bhadavath (✉ kiranbadhavat@gmail.com)

Lords Institute of Engineering and Technology

Z. Mary Livinsa

Sathyambama Institute of Science and Technology: Sathyabama Institute of Science and Technology

Research Article

Keywords: SDR, Digital channelizer, FIR filter, block processing, DA, OBC-DA, ASIC, and Reconfigurable filter

Posted Date: June 8th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1724960/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Implementation of Efficient FIR Filter Architectures using Distributed Arithmetic Computation for Digital Channelizer of SDR

Kiran Kumar Bhadavath^{1*}, Z. Mary Livinsa²

¹Research Scholar, Department of ECE, Sathyabama Institute of Science and Technology, Chennai. *Corresponding author: kiranbadhvat@gmail.com

²Research Supervisor, Department of ECE, Sathyabama Institute of Science and Technology, Chennai,

Abstract: Always an efficient and low complexity reconfigurable filter architecture is required for the channel filters of digital channelizer in Software Defined Radio (SDR). In this paper, a block-based reconfigurable FIR filter is proposed using Distributed Arithmetic (DA) Technique. The complexity of the conventional multiplier is replaced with the DA multiplication process and the throughput of the entire filter is increased by block processing. Memory reuse is also achieved in the proposed direct form systolic FIR filter architecture due to parallel processing. The different bandwidth filters and the corresponding coefficients are stored in the LUTs concerning different channels of the digital channelizer. The proper channel selection is used to choose an appropriate filter and partial products are generated using Offset Binary Coding (OBC). Next, the multiplication process is done by the proposed decomposed LUT-based DA technique in the processing blocks of the filter. The proposed filter is coded by Verilog and synthesized by ASIC-based tools from Cadence in 45 nm CMOS technology. The performance parameters such as area, delay, power consumption, ADP, and PDP are evaluated and compared with the state-of-the-art works. The ADP and PDP values are saved by 44.1% and 30% by the proposed OBC-DA-based filter architecture than the conventional DA-based filter architecture, respectively.

Keywords: SDR, Digital channelizer, FIR filter, block processing, DA, OBC-DA, ASIC, and Reconfigurable filter.

1. Introduction

Software Defined Radio (SDR) is used for multi-standard wireless applications due to its flexibility and ease of adaption. The main important part of the SDR is the digital channelizer. The digital channelizer does the digital down-conversion, sampling rate conversion, and channel filtering. The digital channelizer of SDR mainly consists of a bank of channel filters [1, 2]. These types of channel filters are mostly constructed by Finite Impulse Response (FIR) filters because FIR filters are in linear phase, regular, modular, and stable. Low complexity, low power, and sharp filters are desired for the real-time SDR digital channelizers.

The complexity of the filter is depending on the multiplier, which is used to multiply the filter coefficients and input samples. The SDR wideband receiver architecture, which consists of a digital channelizer is shown in Fig. 1. Digital channelizer is implemented by efficient FIR filters, which are designed for different bandwidths with different filter specifications. In this work, efficient and dedicated low complex reconfigurable filter architectures are implemented for the SDR channelizer. While implementing the dedicated and reconfigurable hardware architectures, the time complexity, area, and power consumption are increased. Hence, optimization is required for the implementation of the dedicated hardware architecture of the filter. In the filter architecture, the multipliers are very complex and power-consuming blocks, which are huge numbers for higher-order filters. It is motivated to implement the low complex multiplierless design-based filter architectures. Distributer Arithmetic (DA) and Canonical Signed Digit (CSD) based Multiplierless designs are mostly used for the filter architecture implementation. CSD is a unique number representation to encode the filter coefficients and multiplication is done by adders and shifters only. The DA multiplication is carried out by accumulator, shifters along with memory or Look-Up Table to store the filter coefficient partial products.

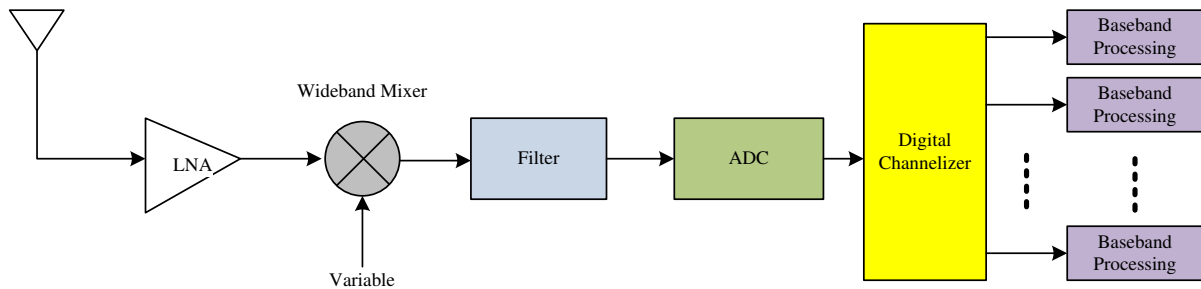


Figure 1: Generic Wideband Receiver.

Many researchers suggested the various DA-based filter architectures and some of them are reported in the related work of this paper.

1.1 Related work

Kalathil et al. [3] proposed the recombination of filter banks for a novel realization of digital channelizers in SDR. Designed the trans-multiplexer and uniform filter bank using cosine modulated filter banks for digital channelizers. The optimization techniques are used for the realization of multiplierless recombination filter banks in CSD space. It reduces the complexity of implementation.

Maresh and Vinod [4] implemented the reconfigurable and low complexity channel filters for SDR using Frequency Response Making (FRM) technique. The reconfigurable

channel filters with low complexity are the major requirements for SDR. The FRM technique offers low complexity by its nature, and this provides reconfigurability at filter level as well as architecture levels. The proposed channel filter produced the synthesis results with a reduced average area of 53.6% and power of 57.6%. The speed of the architecture is improved by 47.6% compared to the reconfigurable filters available in the literature.

Mohanty et al. [5] proposed the design of DA-based reconfigurable block-based FIR filter architecture. This design is applicable for larger block sizes and higher-order filter lengths. Interestingly, the number of registers of the proposed structure does not increase proportionately with the block size. This is a major advantage for area delay and energy-efficient high-throughput implementation of reconfigurable FIR filters of higher block sizes. From the theoretical comparison, the proposed structure for block-size 8 and filter-length 64 involves 60% more flip-flops, 3.5 times more AND-OR gates, 6.2 times more adders, and offers 8 times higher throughput. ASIC synthesis result shows that the proposed structure for block-size 8 and filter-length 64 involves 1.8 times less area-delay product (ADP) and energy per sample (EPS) than the existing design.

Aksoy et al. [6] presented a review on various techniques for the multiplierless realization of 1-D FIR filter architectures. The direct, transposed, and hybrid form structures with constant multiplications using the multiplierless method with shift-add operations are described. The performance metrics, viz., area, delay, and power consumption of various architectures are analyzed using the Field-Programmable Gate Array (FPGA) and ASIC platforms and compared with generic multipliers. It is concluded that the direct form structures with multiplierless designs occupy less area for symmetry filters, hybrid form is preferred for asymmetric filters, and transposed form structure is used for high-speed applications.

Chen and Chiueh [7] implemented the reconfigurable Finite Impulse Response (FIR) filter architecture with less number of functional blocks. An 8-digit reconfigurable FIR filter chip is implemented in single poly quadruple metal 0.35nm CMOS technology. Simulation results show that the fabricated chip operates up to 86 MHz and consumes 16.5mW power from a 2.5V power supply.

Kalaiyarasi and Kalpalatha Reddy [8] proposed an efficient realization of Distributed Arithmetic with Offset Binary Coding for the implementation of the FIR filter. The size of the LUT increases with the coefficient values. For small coefficients, it is easy to realize and becomes complex and takes more storage resources of FPGA for large coefficient values. Proposed the DA algorithm using Offset Binary Coding Algorithm by reducing the LUT size

and delay. The simulation results show that the proposed method utilizes fewer resources of FPGA and increases the speed of the filter compared to conventional filters.

Chitra et al. [9] proposed an efficient DA-based approach with reduced area and power FIR filter implementation for dynamically varying filter coefficients. In order to reduce the complexity of implementing the higher-order filters used a portion of bit serial-based calculation in LUTs. Used the shared LUTs instead of RAM-based LUTs in this work, which reduces the depth of LUTs. The possible partial DA results for the vectors used in shared LUTs are stored in register banks that are shared between multiple DA units. The area is effectively reduced due to shared register banks.

Praveen Sundar et al. [10] proposed a low complex implementation of the FIR filter architecture for hearing aid application. The multiplierless decimation filter in the hearing aid is implemented using Distributed Arithmetic. For higher-order filters, the high-speed hearing aid is realized by using LUTs and a shift-accumulate operator. The implemented architecture occupied 20% less area delay product and 40% less power delay product compared to available architectures.

Park et al. [11] proposed an efficient DA-based scheme for the implementation of high throughput reconfigurable FIR filter with varying filter coefficients at runtime. The shared LUTs are used to realize the DA computation instead of ROM-based LUTs, which are costly for ASIC implementation. The bit slices of different weightage are shared by the LUTs instead of using separate registers to store the inner partial products in DA. A distributed-RAM-based design is also proposed for the FPGA implementation of the reconfigurable FIR filter, which supports up to 91 MHz input sampling frequency.

Mahesh and Vinod [12] proposed two new low complexity reconfigurable architecture FIR filters using the constant shift method and programmable shift method. This proposed FIR filter architecture is compatible with operating for different word length filter coefficients without any overhead in the hardware resources. The common subexpression elimination algorithm is used for implementing efficient dynamically reconfigurable filters. The simulation results show that the proposed architectures produced enhanced results with reduced area and power consumption with improved speed compared to available reconfigurable FIR filter architectures in the literature.

Khan et al. [13] presented an efficient design and implementation of low power and low area acoustic echo canceller. This design applied the block mean square algorithm for adaptive filter (ADF) design using offset binary coding. The proposed approach the ADF is formulated by splitting the matrix-vector multiplication into smaller blocks. Each block is

realized using lookup tables and shift-accumulated units with offset terms. The offset terms in the corresponding lookup tables are updated by an efficient scheme in the proposed method. In addition, a novel optimization is proposed based on the grouping of partial products (PPs) and moving windows. The simulations are performed to evaluate the efficiency of the proposed design. ASIC synthesis show that the proposed design is superior to state-of-the-art architectures.

Ahmed et al. [14], proposed a composite design using Distributed Arithmetic (DA) to replace the multipliers with memory units that store Partial Products (PPs) to compute multiplication. The depth of their memory units is increased as the filter order increases. To overcome this, Half Memory (HM) algorithm and Offset Binary Coding (OBC) are used to modify the structure of PPs to reduce the memory size by a factor of 4 for the same filter order. This proposed design improves the enhanced system throughput, power consumption, critical path delay, and FPGA resource utilization. However, it introduces Latency in both the output and update segments of the LMS algorithm. In order to provide an option between resource utilization and latency, proposed a mechanism to halve the originally produced latency by the parallel processing of input bit stream w.r.t even and odd bits. It is also proposed a method that reduces the latency of the update module at the slight expense of other design attributes.

Prakash and Shaik [15], proposed an efficient implementation of the least mean square adaptive filter. The architecture is based on Distributed Arithmetic (DA), in this the shift and accumulate operation in the filter is done on partial products, which are precomputed and stored in LUTs. The filter coefficients in LUTs need to be updated in adaptive filters. The Offset binary coding technique is applied to update the LUTs. The simulation results indicate that the proposed method consumes less power and very less chip area and operates at high throughput even for higher-order filters.

Khan and Shaik [16] proposed a pipelined DA-based LMS adaptive filter to reduce the complexity of the circuit. This is implemented by OBC and eliminated the errors during initial clock cycles. In addition, proposed a low complexity implementation for the offset term, weight update block, and shift-accumulate unit. The analysis shows that the byte-complexity of proposed structures varies linearly with the order of DA base unit, while their bit-complexity depends on the topology.

Rui and Linda [17] proposed novel DA-based adaptive filters for the implementation of FIR filters. Here the coefficients are used as coefficients to access the sums of delayed and scaled inputs samples in Look Up Tables (LUT). Also proposed two smart LUT updating methods and LMS to update the weights in the adaptive filter to minimize the error. Simulation

results show that proposed designs produced high speed, low computation complexities, and low area cost compared to available structures.

Naga Jyothi and Sridevi [18] Implemented the Decision Feedback Equalizer (DFE) using a novel Memory Less Distributed Arithmetic filter. The conventional memory unit and adders are replaced with multiplexers and enhanced compressor adders in the proposed design. The proposed design occupies a lower area and produces more throughput compared to MAC-based filters and other memory-based DA filter architectures.

Rammohan et al. [19] designed the decimation filter in hearing aids using multiplierless architecture to adjust the sound levels. Proposed a low complexity FIR filter for hearing aid application using approximate 4:2 compressor adders in memoryless DA-based filter structure. In order to reduce the area complexity of DA architectures for higher-order filters, implemented the memoryless DA using compressor adders in this work. The proposed design is implemented in Application Specific Integrated Circuit design and also on FPGA.

NagaJyothi and Sridevi [20] reviewed the available non-reconfigurable FIR filter architectures based on DA, LUT-less based DA and LUT based DA. It summarizes the area and power reports of LUT-based DA and LUT-less DA architectures of available architectures. Presented the comparative results in terms of area, delay, Area Delay Product, and Power Delay Product.

Mohanty and Meher [22] proposed a novel common DA formulation for the convolution and correlation and a new LUT updating technique. The LUT size and adders are reduced by this new approach. A parallel structure of the LMS adaptive FIR filter is implemented using the proposed DA scheme. The proposed design is compared with the best existing filter architectures in terms of area, delay, and power consumption.

Mohanty et al. [23] proposed a low complex LUT update unit and less size LUT for the implementation of a block-based LMS adaptive filter. LUT sharing is also incorporated in the DA process and reconfigurability is achieved by scaling the design for the different filter lengths. The ASIC-based design is considered for the implementation of the proposed filter architecture and it was compared with the state-of-the artworks to analyze the performance.

Khan et al. [24] suggested the optimized block LMS-based ADF using OBC- DA technique for ear-phones applications. Two different splitted LUTs are designed along with a novel LUT updating technique to produce the ADF output. Various higher order filters are implemented for the noise cancellation in headphones and compared with existing works.

Khan et al. [25] proposed the two efficient LUT design approaches for the implementation of block-based LMS ADF architectures. These optimized LUT-based DA

techniques are contributed to reducing the power and area of the filter architecture. The echo cancellation application is considered for the analysis of the proposed filter.

Recently, Odugu et al. [26, 27] proposed a memory-based LUT multiplication process that is used to implement the 2-D FIR filter architectures. In this, block-based symmetry FIR filters are integrated to realize a filter bank using DA multipliers. The even multiples of coefficients or odd multiples of coefficients are only considered to store in LUT to reduce the size of LUT. The parallel processing, symmetry in the coefficients, and DA concept combinations are used to optimize the VLSI design metrics of the FIR filter bank architectures. The same features can be applied to 1-D filters also.

In this paper, a customized dedicated FIR filter architecture is proposed for the applications of SDR channelizer using decomposed Offset Binary Coding (OBC) based DA concept. The desired channel filter can be selected by selection logic and corresponding filter coefficients are fetched and the partial products are generated by the OBC logic. These partial products have not been stored in any memory and are directly selected by multiplexer logic into the processing blocks of the filter.

The direct form structure is considered for the systolic architecture and block processing is introduced. The advantage of the direct form structure is less size memory registers are needed. The size of the memory registers is independent of the intermediate signal word length because all the registers are placed at the input side of the architecture only. The advantage of block processing is high throughput and memory reuse. Memory reuse means that the already produced samples in the past registers can be reused for filter processing to produce the output.

The DA-based multiplierless design reduces the complexity of the conventional multiplier, but the LUT which is used to store the precomputed values of filter coefficients size is increased with the filter length. The size of the LUT can be decreased in half by using OBC. In the OBC, the redundant partial products are removed, which can be generated by using small combinational logic. Even the OBC-based LUT size is also large for very higher-order filters, so decomposition is introduced in the OBC-based LUTs. The fine-grain LUTs are used to store the short vectors of the filter coefficients. Based upon this decomposition of the LUT, the OBC-based partial products are generated by combination logic, and these partial products are directly applied to the processing block of the systolic filter architecture. MUX logic is used to select the appropriate partial product of filter coefficients. The input sample bits are used as selection lines of the multiplexer. The direct form structure, block processing concept, OBC-

based DA, and decomposition are reducing the memory complexity and as well as area, delay, and power consumption of the filter architecture.

The main contributions of the proposed work are as follows:

- The direct form FIR filter is considered for the systolic architecture implementation, whereas the direct form needs a fewer number of registers.
- Block processing is used to improve the throughput of the filter and for memory reuse
- Reconfigurable filter architecture is proposed using Multiplierless design using DA
- OBC based DA to reduce the size of the LUT and further, the LUT is decomposed for reduction of memory size
- Conventional DA based filter and OBC based proposed filter for SDR channelizer are implemented using ASIC design in 45nm technology
- Performance parameters such as area, delay, power, ADP, and PDP are evaluated and compared with the state-of-the artworks.

The rest of the paper is organized as follows: The basic concepts of DA and OBC and corresponding formulations are presented in section 2. Section 3 describes the block processing concept in the proposed filter and OBC formulation in the block-based filter. The main architectures of the proposed FIR filter, coefficients storage block, and sub-blocks of the filter architecture are explained in section 4. Section 5 consists of experimental results and discussions. Finally, the conclusion of the work is presented in section 6.

2. Background:

2.1 DA Concept

In this section, the basic DA-based FIR filter and OBC-DA-based FIR filter architectures are described for understanding the basic difference between conventional DA and OBC-DA.

The basic DA-based FIR filter structure for the filter length of $N = 4$ is shown in Figure 2. In this structure, the $2^N = 16$ partial products of filter coefficients corresponding to the $N = 4$ are precomputed and stored in the ROM LUT. The current input sample and past input sample bits are considered as the address of the LUT memory to fetch the appropriate partial product. Next, the partial products are shifted and accumulated by the proper combination logic for the B-number of clock cycles to produce the final filter output, where B is the number of bits of the input sample. The main advantage of the DA concept is that the conventional complex and power hungry multipliers are completely replaced. Hence, the area, delay, and power consumption of the architecture are decreased. In this architecture, the size of the LUT memory

is the bottleneck for the designers. If the filter length N is increased by one, then the size of the LUT becomes double. For higher filter needs a large memory to store the partial products of filter coefficients.

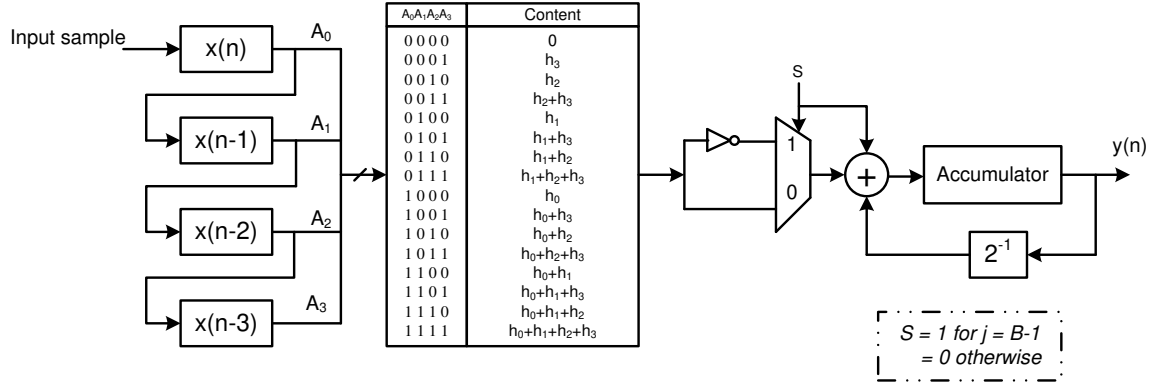


Figure 2: Basic DA-based FIR filter architecture for $N = 4$.

2.2 Formulation of basic DA

The inner product in the DA between two vectors such as input vectors X and h is given by Eq. (1).

$$Y = \sum_{i=0}^{N-1} h_i \cdot x_i \quad (1)$$

Where, h_i coefficient vector with M -bit constants and x_i is input sample with $w - bits$ is represented in 2's complement form as given by Eq. (2).

$$x_i = -x_{i,w-1} + \sum_{j=1}^{B-1} x_{i,w-1-j} 2^{-j} \quad (2)$$

By substituting Eq. (2) in Eq. (1), then the output of the filter is given by Eq. (3) and Eq. (4).

$$Y = \sum_{i=0}^{N-1} h_i (-x_{i,w-1} + \sum_{j=1}^{w-1} x_{i,w-1-j} 2^{-j}) \quad (3)$$

$$Y = -\sum_{i=0}^{N-1} h_i x_{i,w-1} + \sum_{j=1}^{w-1} (\sum_{i=0}^{N-1} h_i x_{i,w-1-j}) 2^{-j} \quad (4)$$

Where some terms are defined as,

$$H_{B-1} = -\sum_{i=0}^{N-1} h_i x_{i,w-1} \text{ and } H_{w-1-j} = \sum_{i=0}^{N-1} h_i x_{i,w-1-j} (j \neq 0)$$

Then the final D- based FIR filter output is given by Eq. (5)

$$Y = \sum_{j=0}^{w-1} H_{w-1-j} 2^{-j} \quad (5)$$

2.3 Offset Binary Coding:

The FIR filter architecture for the filter length of $N = 4$ using DA with an OBC scheme is shown in Fig. 3. The OBC scheme reduces the LUT-ROM size by a factor of 2 to 2^{N-1} [21]. In this structure, the XOR gates are used for the current and past input samples to decode the address of LUT. The Mux with initial gives the initial value to the shift-accumulator. The Multiplexor next to the LUT-Rom is used to inverse the output of the LUT corresponding to the $j = w - 1$. The control signals S_0 and S_1 are used and S_0 is 1 for $j = w - 1$ and 0

otherwise. Similarly, S_1 is 1 for $j = 0$ and 0 otherwise. The formulation of the OBC-DA scheme is described in this section.

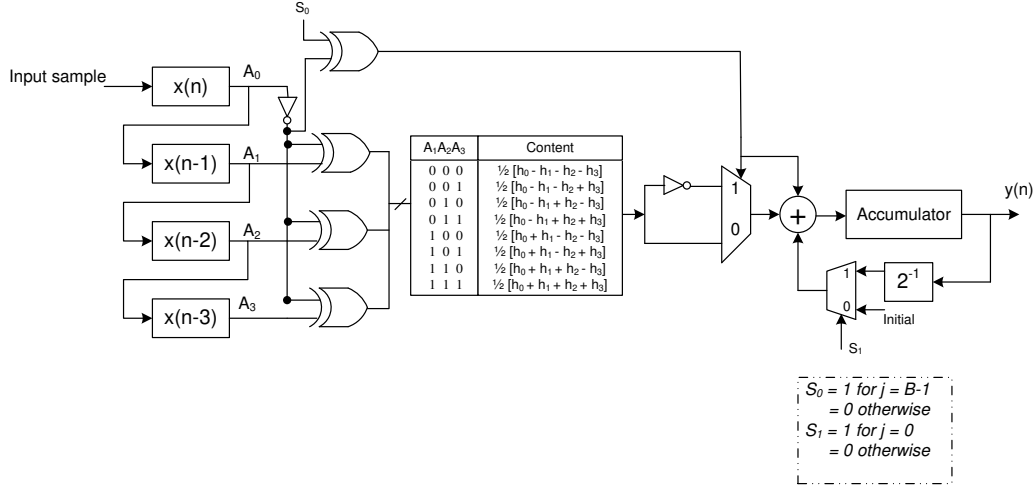


Figure 3: OBC-DA-based Fir filter architectures for $N = 4$.

2.4 Formulation of OBC

The input sample of the filter in the 2's complement form can be written as Eq. (6).

$$x_i = \frac{1}{2} [x_i - (-x_i)]$$

$$x_i = \frac{1}{2} [-(x_{i,w-1} - \overline{x_{i,w-1}}) + \sum_{j=1}^{w-1} (x_{i,w-1-j} - \overline{x_{i,w-1-j}}) 2^{-j} - 2^{-(w-1)}] \quad (6)$$

Where, $x_i = -\overline{x_{i,w-1}} + \sum_{j=1}^{w-1} \overline{x_{i,w-1-j}} 2^{-j} + 2^{-(w-1)}$.

Define the terms, $d_{ij} = \begin{cases} x_{i,j} - \overline{x_{i,j}}, \text{ for } j \neq w-1 \\ -(x_{i,w-1} - \overline{x_{i,w-1}}), \text{ for } j = w-1 \end{cases}$ and $d_{ij} \in [-1, 1]$. Then the Eq. (6) can be expressed as Eq. (7).

$$x_i = \frac{1}{2} [\sum_{j=0}^{w-1} d_{i,w-1-j} 2^{-j} - 2^{-(w-1)}] \quad (7)$$

Finally, the filter output expression $Y = \sum_{i=0}^{N-1} h_i \cdot x_i$ can be written as Eq. (8).

$$Y = \sum_{i=0}^{N-1} \frac{1}{2} h_i [\sum_{j=0}^{w-1} d_{i,w-1-j} 2^{-j} - 2^{-(w-1)}]$$

$$= \sum_{j=0}^{w-1} (\sum_{i=0}^{N-1} \frac{1}{2} h_i d_{i,w-1-j}) 2^{-j} - (\frac{1}{2} \sum_{i=0}^{N-1} h_i) 2^{-(w-1)} \quad (8)$$

Now define $P_j = \sum_{i=0}^{N-1} \frac{1}{2} h_i d_{i,j}$, for $0 \leq j \leq w-1$ and $P_{extra} = -\frac{1}{2} \sum_{i=0}^{N-1} h_i$. Therefore, the

Final filter output using the OBC scheme is given by Eq. (9).

$$Y = \sum_{j=0}^{w-1} P_{w-1-j} 2^{-j} + P_{extra} 2^{-(w-1)} \quad (9)$$

The above equations refer to the OBC concept. The P_j values are mirrored in the LUT table. In other words, the P_j has 2^{N-1} possible values depending on the x_{ij} values. Hence, it is possible to reduce the LUT size by a factor of 2. The new LUT contents can be observed in Fig.3. It can be observed from the basic DA-based filter architecture and OBC-based filter architecture, that the LUT-ROM size of the OBC scheme is reduced to half for the filter length $N = 4$. So the size of the memory is reduced and accessing time to fetch the memory is also reduced by the OBC concept. In this work, the OBC-based DA process is adopted for the proposed block-based reconfigurable FIR filter architecture implementation.

3. Block based concept in the proposed FIR filter

The block-based or Parallel FIR filter processes the L (block size) number of input samples and produces the L number of output samples per one iteration. The q -th block of filter out is given by Eq. (10).

$$y_q = X_q \cdot h \quad (10)$$

Where, X_q is input matrix and h is the coefficient matrix of the FIR filter for length N and it is given by Eq. (11).

$$h = [h(0), h(1), h(2), \dots, h(N-1)]^T \quad (11)$$

If the current input block is $[x(qL), x(qL-1), \dots, x(qL-L+1)]$, then the input matrix is derived from the current input block and it is given by Eq. (12).

$$X_q = \begin{bmatrix} x(qL) & x(qL-1) & \dots & x(qL-N+1) \\ x(qL-1) & x(qL-2) & \dots & x(qL-N) \\ \vdots & \vdots & \dots & \vdots \\ x(qL-L+1) & x(qL-L) & \dots & x(qL-L-N+2) \end{bmatrix} \quad (12)$$

In this proposed work, the input matrix is decomposed into $N/2$ smaller matrices, and each sub-matrix is denoted by R_q^j and matrix size is $(L \times 2)$ and it is given by Eq. (13).

$$R_q^j = \begin{bmatrix} x(qL-2j) & x(qL-2j-1) \\ x(qL-1) & x(qL-2j-2) \\ \vdots & \vdots \\ x(qL-2j-L+1) & x(qL-2j-L) \end{bmatrix} \quad (13)$$

The coefficient matrix is also decomposed into $N/2$ short vector coefficients and each submatrix is denoted by w_j and it consists of 2 coefficients, and it is given by Eq. (14).

$$w_j = [h(2j) \ h(2j+1)]^T \quad (14)$$

Then the output of the filter with submatrices is given by Eq. (15).

$$y_k = \sum_{j=0}^{\left(\frac{N}{2}\right)-1} R_q^j \cdot w_j \quad (15)$$

Each filter output $y(qL - i)$ for $0 \leq i \leq L - 1$ is the sum of the $(N/2)$ inner product terms and it can be written as Eq. (16).

$$y(qL - i) = \sum_{j=0}^{\left(\frac{N}{2}\right)-1} u(i, j) \quad (16)$$

Where, $u(i, j)$ is a 2-point inner product of and the input vector R_q^{ij} and w_j and it is expressed as Eq. (17),

$$u(i, j) = R_q^{ij} \cdot w_j \quad (17)$$

Here, R_q^{ij} is $(i + 1)$ -th row of the R_q^j as given by Eq. (18),

$$R_q^{ij} = [x(qL - 2j - i) \ x(qL - 2j - i - 1)] \quad (18)$$

3.1 OBC-DA formulation for the block-based FIR filter

Let $R_q^{ij}(k)$ be the $(k+1)^{\text{th}}$ components of the input sample and it consists of 2 vectors R_q^{ij} . The number of bits is w and it is expressed in two's complement form for the DA process is given by Eq. (19).

$$R_q^{ij}(k) = -(R_q^{ij}(k))_0 + \sum_{m=1}^{w-1} 2^{-m} (R_q^{ij}(k))_m \quad (19)$$

Alternatively, $R_q^{ij}(k)$ is also be expressed in OBC $R_q^{ij}(k) = \frac{1}{2} [R_q^{ij}(k) - (-R_q^{ij}(k))] = \frac{1}{2} [R_q^{ij}(k) - \overline{R_q^{ij}(k)}]$.

Where, $\overline{R_q^{ij}(k)}$ is two's complement form of $R_q^{ij}(k)$ and Eq. (19) can be modified according to the OBC form.

$$R_q^{ij}(k) = -\frac{1}{2} \left\{ [(R_q^{ij}(k))_0 - \overline{(R_q^{ij}(k))_0}] + \sum_{m=1}^{w-1} 2^{-m} ((R_q^{ij}(k))_m - \overline{(R_q^{ij}(k))_m}) - 2^{-(w-1)} \right\}$$

Assume that $(dR_q^{ij}(k))_m = (R_q^{ij}(k))_m - \overline{(R_q^{ij}(k))_m}$ is the one's compliment difference of coefficient bit slices. $(dR_q^{ij}(k))_m \in \{-1, 1\}$, then block-based filter output $u(i, j)$ can be written as Eq. (20)

$$u(i, j) = \sum_{m=0}^{w-1} \left[\sum_{k=0}^1 \frac{1}{2} h(2j + k) (dR_q^{ij}(k))_m \right] 2^{-m} - \frac{1}{2} \left[\sum_{k=0}^1 h(2j + k) \right] 2^{-(w-1)} \quad (20)$$

Inner summation is referred to as partial products. The order of the summation for the above expression Eq. (20) can be rearranged and it can be modified as Eq. (21).

Now define $P(i, j) = \sum_{k=0}^1 \frac{1}{2} h(2j + k) dR_q^{ij}(k))_m$ and $P(i, j)_0 = - \sum_{k=0}^1 \frac{1}{2} h(2j + k)$

$$u(i, j)_m = \sum_{m=0}^{w-1} P(i, j) 2^{-m} - P(i, j)_0 2^{-(w-1)} \quad (21)$$

4. Architecture of proposed block-based FIR filter

In this section, the proposed block-based reconfigurable FIR filter architecture using OBC based DA method for digital channelizer is explained. Figure 4 represents the view of the SDR digital channelizer, which consists of R-channel filter coefficients with each channel filter length $N = 8$. All the different filter coefficients are stored in the array of N number of LUTs. The channelizer needs various channel filters with different filter specifications and bandwidths. The desired channel filter can be selected by an address line of the channel-selector and the corresponding filter coefficients are fetched, which are stored in the same location of each LUT. The reconfigurability is achieved at the architecture level, by the selection of desired channel filters.

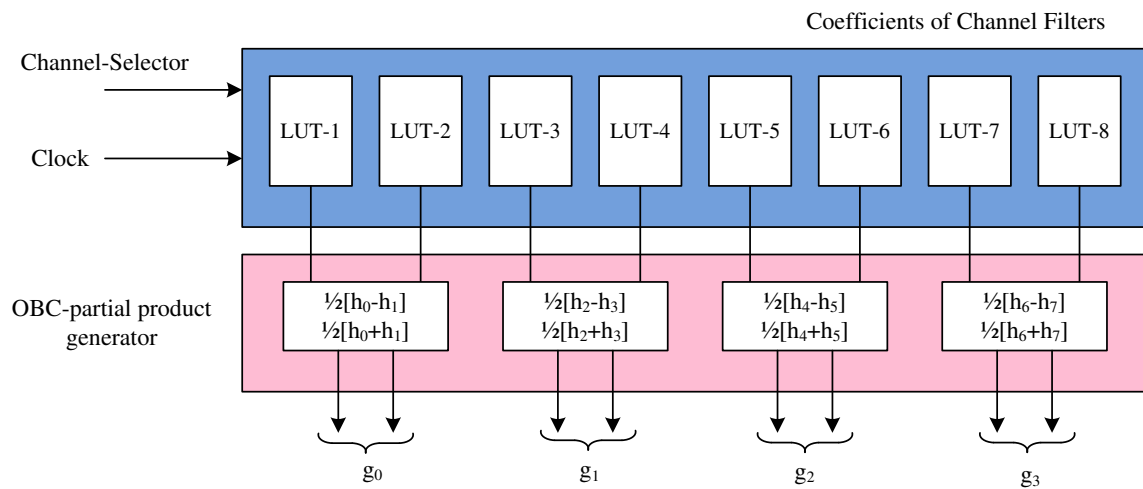


Figure 4: Various filter coefficients storage unit of digital channelizer with R- channel filters and each filter length of $N = 8$.

Next, the filter coefficients are converted into partial products based on the OBC. The OBC-based partial products are generated and applied to the filter processing blocks. Here, the decomposition of OBC-based DA-LUT is considered for the reduction of the memory size to store the OBC contents and to make the process easy. The decomposition factor is $N/2 = 8/2 = 4$. Four filter coefficients are converted into two OBC partial product terms, as shown in Fig 2. The OBC-based partial inner products are generated for every clock cycle by OBC

partial products generation block. In each clock cycle, the filter coefficients h_i are fed to OBC partial product generator and $N/2 = 4$ sets of partial inner products related to OBC are generated and denoted by g_j for $0 \leq j \leq N/2 - 1$. Each set of partial products consists of two non-zero OBC product terms of filter coefficients such as $\frac{1}{2}[h(2j) - h(2j + 1)]$ and $\frac{1}{2}[h(2j) + h(2j + 1)]$ etc.

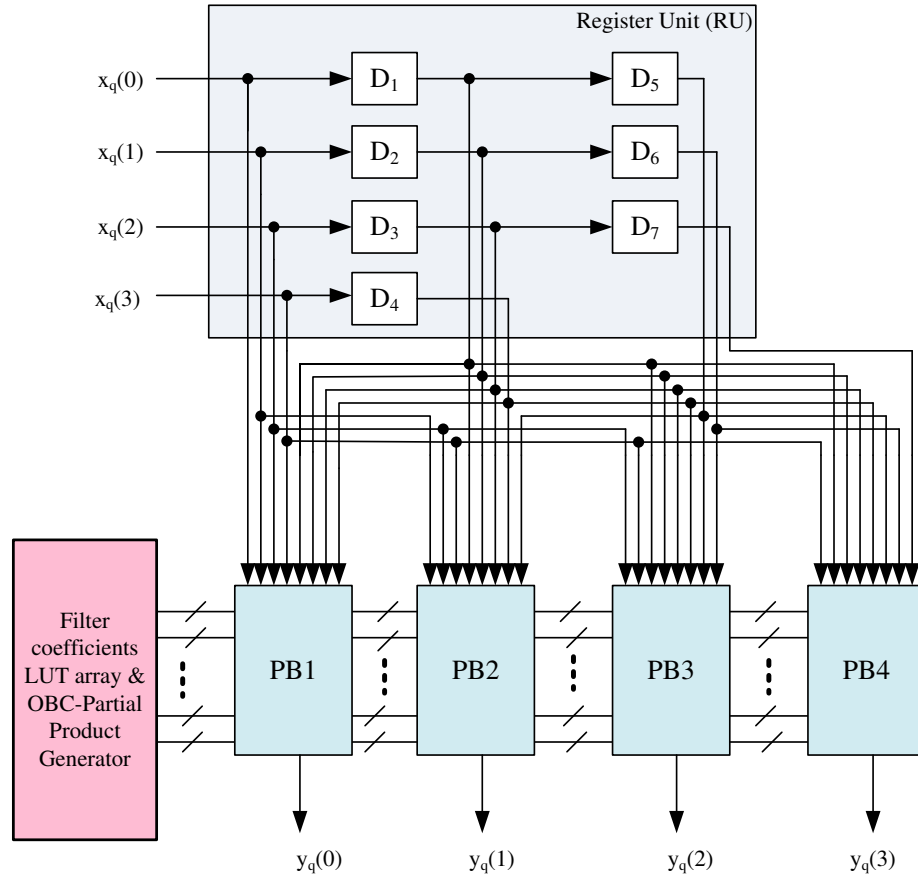


Figure 5: The proposed architecture of block-based reconfigurable FIR filter.

The main architecture of the proposed block-based direct form systolic FIR filter is shown in Fig. 5. The architecture e mainly consists of Register Unit (RU), Filter coefficients LUT array along with OBC partial product generator, which is already explained above, and L number of Processing Blocks (PB). The RU is composed of the block size $L = 4$ and filter length $N = 8$. The RU has $(N-1)$ registers (D Flip-Flops), which produce the combination of past and current input samples corresponding to the block processing. In this process, many redundant samples are generated, which are reused for the filter processing to reduce the memory registers requirement. The RU produces the LN number of samples, which are applied to the L number of PBs. The partial products g_0 to g_3 from the partial product-generator are also applied to the PBs parallelly. Each partial product set consists of two vectors of

coefficients. The L number of PBs processes the L- input samples and filter coefficients and it produces the L number of outputs. The internal structure of the PB is shown in Fig. 6.

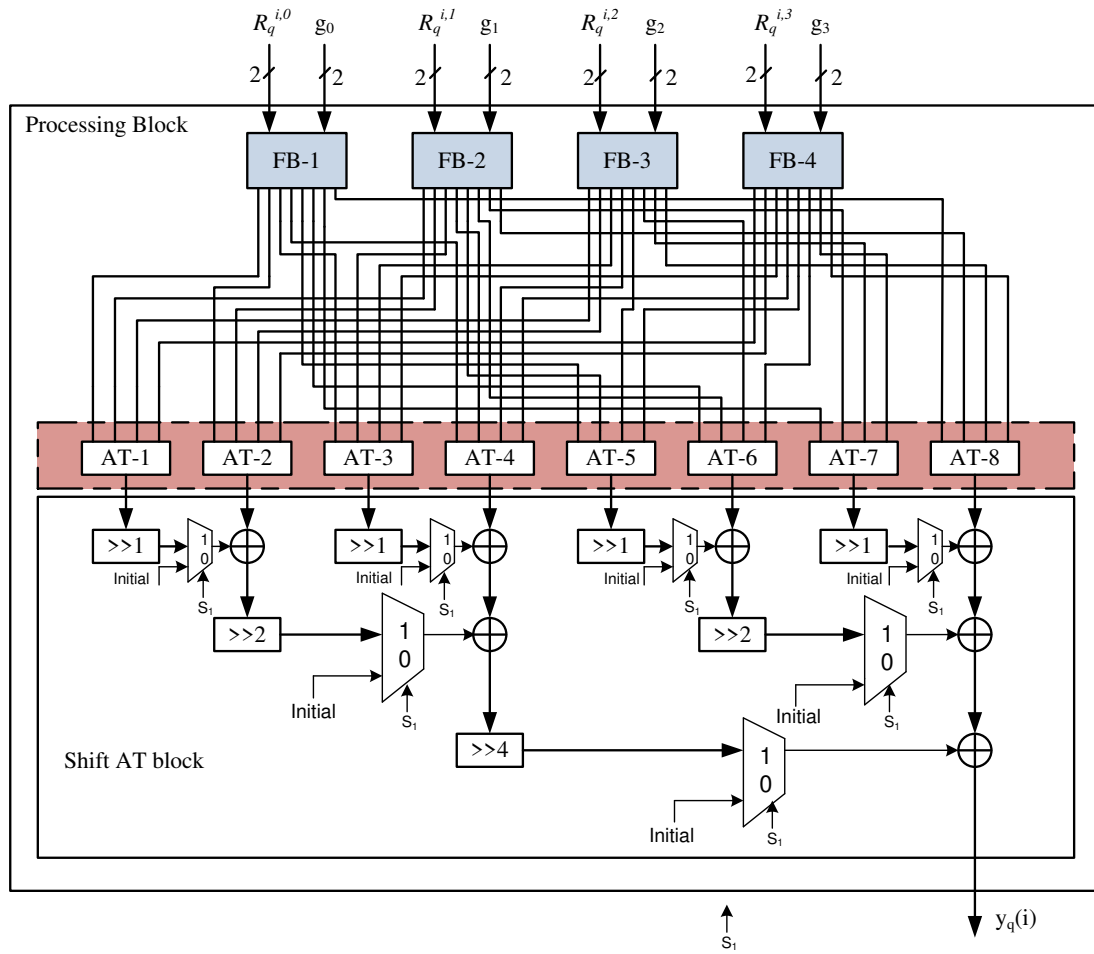


Figure 6: Internal structure of Processing Block of the filter.

The Processing Block (PB) consists of L number of Functional Blocks (FB), the array of Adder Tree (AT) blocks, and shift-accumulators. The decomposed input samples, mean two samples, and two OBC-based partial products are applied to each FB. For $N = 8$ and $L = 4$, four FBs and four partial product sets, and 4 sets of input samples, eight AT blocks and shifters and adders are required.

Each FB process the input samples and filter coefficients to produce eight outputs, the internal structure of the FB is shown in Fig 7. The outputs from each FB are applied to the AT block and next, outputs of the AT are right-shifted and accumulated by corresponding adders and finally give the i-th block output.

The same functionality is carried out in each PB, for example, $L = 4$, four PBs produce Four outputs, each PB consisting of 4 FBs. In each FB, multiplexer logic is used to process the input samples and filter coefficients. The OBC partial products are selected by input sample

bits. Each input sample has B-bits, and two-LSB bits from two input samples are taken as selection lines, and these bits are applied to XOR gates. The first MUX selects the $\frac{1}{2}[h(2j) - h(2j + 1)]$ when the selection bit is '0' otherwise $\frac{1}{2}[h(2j) + h(2j + 1)]$ is selected. The output of the first MUX is inverted by another MUX, which is controlled by the current input sample bit. Here, one control signal S_0 is considered to select the original MUX output or inverted output. Similarly, B-1 combination logics are performed and produce the B number of output lines, which are given to the B number of AT blocks. The same processing is taken place in all the PBs and FBs of the filter architecture.

architecture.

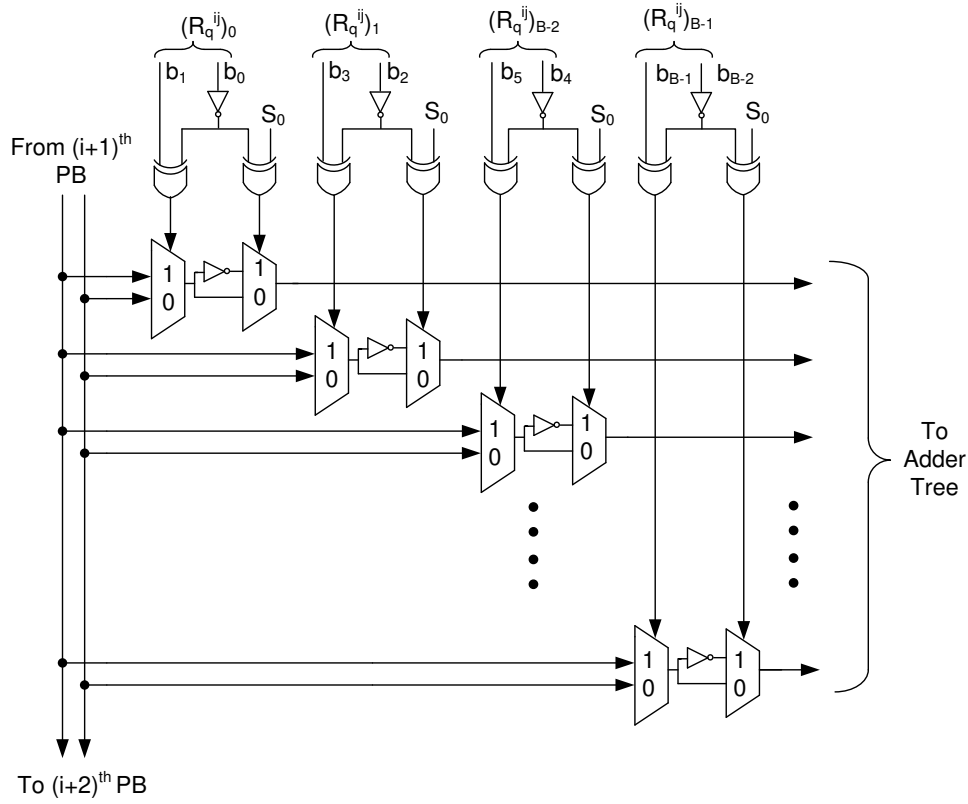


Figure 7: Structure of Function Block (FB) in PB.

5. Implementation and Experimental Results

The proposed block-based reconfigurable DA-based FIR Filter architecture is coded by Verilog HDL and simulated and synthesized using ASIC-based Cadence tools. The entire HDL-based filter architecture is elaborated by appropriate test inputs by the Incisive Enterprise Simulator (IES) simulation tool. Next, the proposed design is synthesized to equivalent hardware of the 45nm CMOS generic library by the Genus synthesis tool provided by Cadence. The area, delay, and power reports are generated without any constraints by the synthesis tool. In this paper, first, the proposed systolic FIR filter architecture is implemented for $N = 8$ and

$L = 4$ using conventional DA and OBC- DA for comparison purposes. Table.1 shows the performance metrics comparison between the conventional DA-based filter architecture and the OBC-based proposed filter

Table 1: Comparison of VLSI design metrics of conventional DA and OBC-DA-based FIR filter architectures for block size $L = 4$.

Filter type	Filter Length (N)	Area (μm^2)	Delay (ns)	Power (mW)	ADP ($\mu\text{m}^2.\mu\text{s}$)	PDP (mW.ns)
Filter using Conventional DA	8	35658.98	1.302	24.69	46.427	32.14638
Filter using OBC-DA	8	22777.23	0.899	10.56	20.476	9.49344

The FIR filter implementation using the OBC-DA-based method produced efficient values for the area, delay, power, Area Delay Product (ADP), and Power Delay Product (PDP) compared to of conventional DA-based method. The graphical presentation of ADP and PDP of the filter implementation using the OBC-DA method in comparison with conventional DA is shown in Fig 8. The proposed OBC-DA-based method produced 44.1% and 30% less ADP and PDP values respectively compared to the conventional DA-based method.

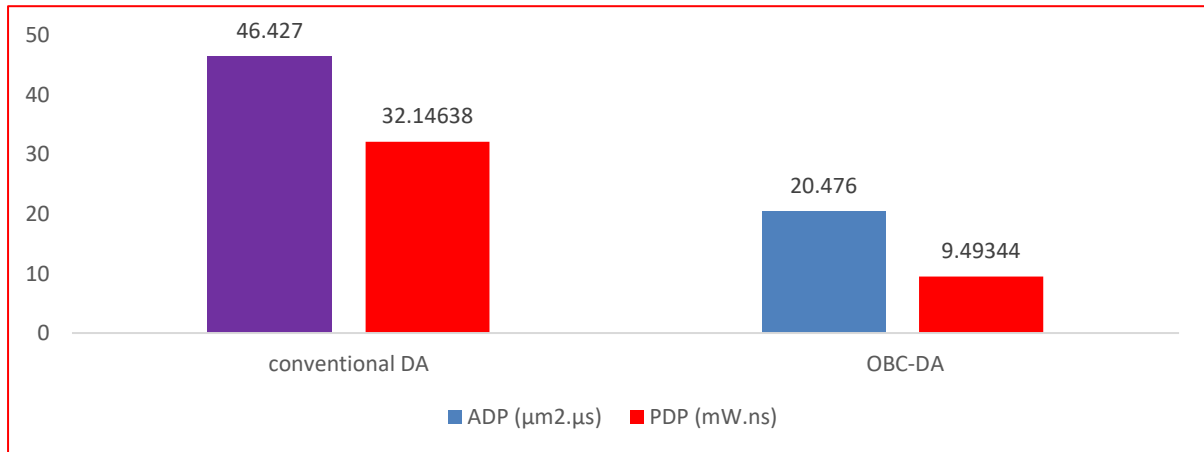


Figure 8: Comparison of ADP and PDP of the proposed filter architecture with conventional DA-based filter architecture.

The Area, delay, power, ADP, and PDP of the FIR filter implementation using the proposed OBC-DA-based method for different filter lengths (N) with various combinations of block sizes (L) are presented in Table 2. The FIR filter implementation uses OBC-DA for filter lengths (N) of 8, 16, and 32 for block sizes (L) of 2 and 4. The values of the area, delay, and power are increased proportionally by increasing the filter length and block size. The graphical

comparison of the ADP and PDP values for $N = 8$, $N = 16$, and $N = 32$ with block size $L = 2$ and $L = 4$ is shown in Fig 9 and 10 respectively.

Table 2: Comparison of proposed filter architecture with various filter lengths and block sizes.

Filter Length (N)	Block Size (L)	Area (μm^2)	Delay (ns)	Power (mW)	ADP ($\mu\text{m}^2.\mu\text{s}$)	PDP (mW.ns)
N = 8	L = 2	18456.96	0.85	8.652	15.688	7.3542
	L = 4	22777.23	0.899	10.56	20.476	9.49344
N = 16	L = 2	38652.12	0.974	14.265	37.647	13.8941
	L = 4	42699.2	0.9899	16.99	42.267	16.8184
N = 32	L = 2	65892.287	0.985	22.64	64.903	22.3004
	L = 4	95672.235	1.012	28.36	96.820	28.7003

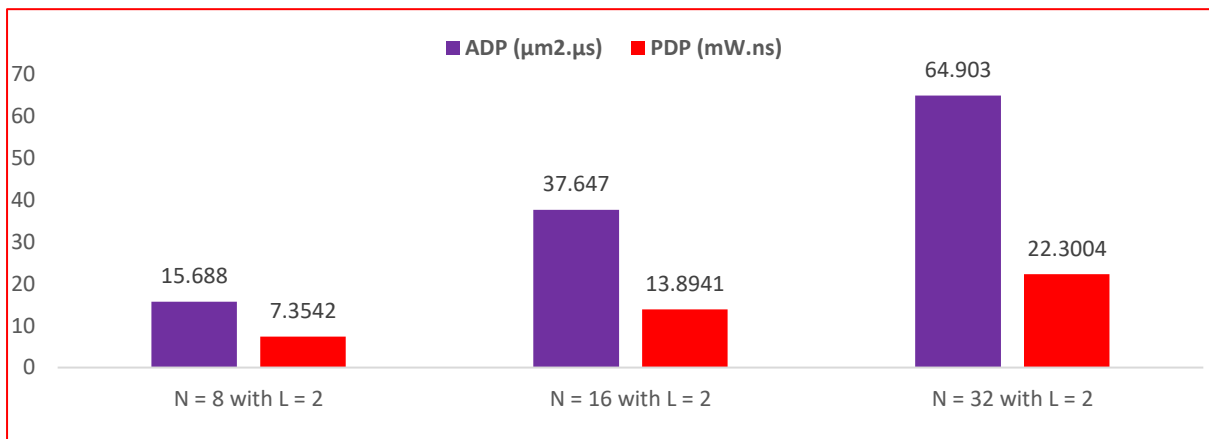


Figure 9: Comparison of ADP and PDP of the proposed filter architecture for different filter lengths and block size 2

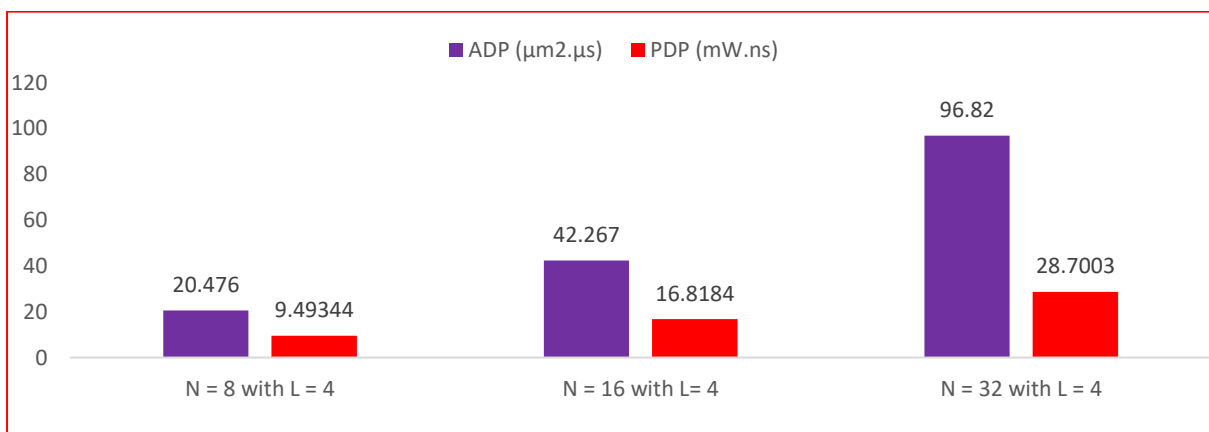


Figure 10: Comparison of ADP and PDP of the proposed filter architecture with different filter lengths and block size of 4.

The proposed FIR filter implementation using the OBC-DA-based method is compared with the available implementation methods in terms of the performance matrices, viz., area,

delay, power, ADP, and PDP for filter length $N = 32$, and block size $L = 4$ is presented in Table 3.

The area occupied by the proposed and available FIR filter architectures is shown in Fig 10. The area of the proposed filter implementation produced fewer values compared to available methods in the literature. The proposed method with OBC-DA has occupied 12.4%, 22.3%, 3.3%, 55.9%, 79.2%, 73.5%, 70.2%, 51.3%, and 20.9% less area compared to filter architectures by Mahesh CSM [12], Mahesh PSM [12], Park et al. [11], Mohanty [5], Mohanty et al. [22], Mohanty et al. [23], M T Khan [24], M T Khan et al. [25], and M T Khan et al. [13] respectively.

Table 3: Comparison of proposed filter architecture with existing DA-based filter architectures in terms of VLSI design metrics.

Filter type	Filter Length (N)	Area (μm^2)	Delay (ns)	Power (mW)	ADP ($\mu\text{m}^2 \cdot \text{ns}$)	PDP (mW.ns)
Mahesh CSM [12]	32	109252	1.48	38.91	161693	57.5868
Mahesh PSM [12]	32	123204	1.71	34.38	210678.8	58.7898
Park et al. [11]	32	92446	1.10	31.20	101690.6	34.32
Mohanty [5]	32	217074	1.09	77.64	236610.7	84.6276
Mohanty et al. [22]	32	462106.08	1.75	180.05	808685.6	315.088
Mohanty et al. [23]	32	362105.36	1.74	135.06	630063.3	235.004
M T Khan [24]	32	321944.05	1.68	116.54	540866	195.787
M T Khan et al. [25]	32	196789.43	1.73	74.12	340445.7	128.228
M T Khan et al. [13]	32	121029.68	2.01	49.21	243269.7	98.9121
Proposed Filter	32	95672.235	1.012	28.36	96820.3	28.7003

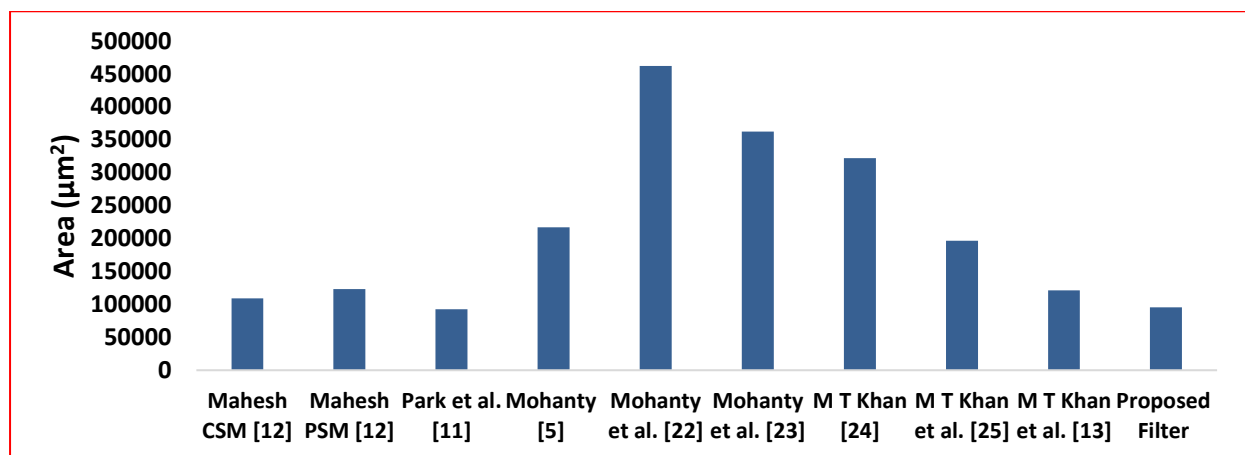


Figure 10: Comparison of Area occupied by the proposed structure with existing filter architecture for the length of the filter is 32.

The power consumed by the proposed FIR filter architecture and the existing architectures in the literature is shown in Fig 11. The power consumed by the proposed FIR filter architecture is reduced by 27.1%, 17.5%, 9.1%, 63.4%, 84.2%, 79%, 75.6%, 61.7%, and 42.3% compared to filter architectures by Mahesh CSM [12], Mahesh PSM [12], Park et al. [11], Mohanty [5], Mohanty et al. [22], Mohanty et al. [23], M T Khan [24], M T Khan et al. [25], and M T Khan et al. [13] respectively.

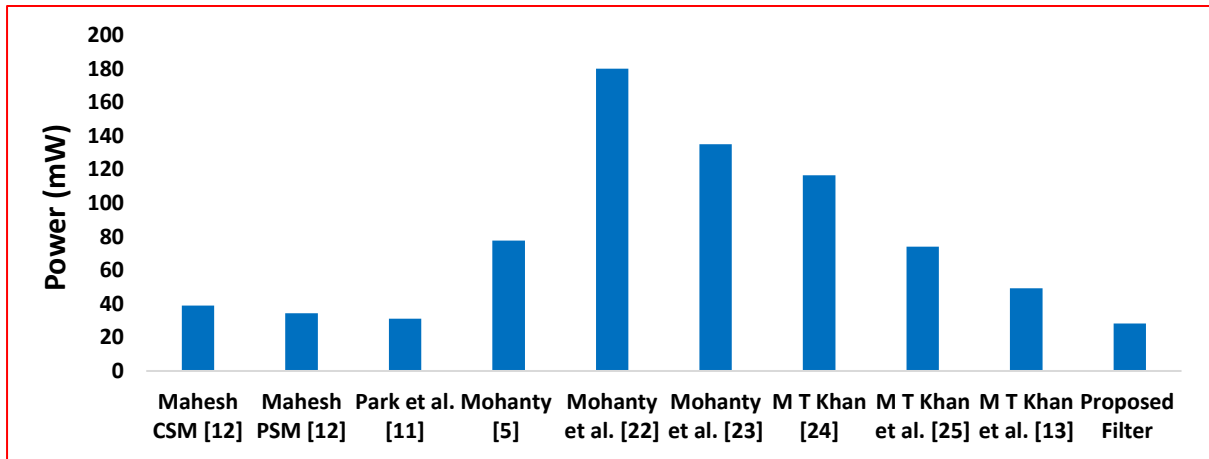


Figure 11: Comparison of power consumption of the proposed structure with existing filter architecture for the length of the filter is 32

The delay generated by the proposed architecture with OBC-DA and existing architectures in the literature is shown in Fig 12.

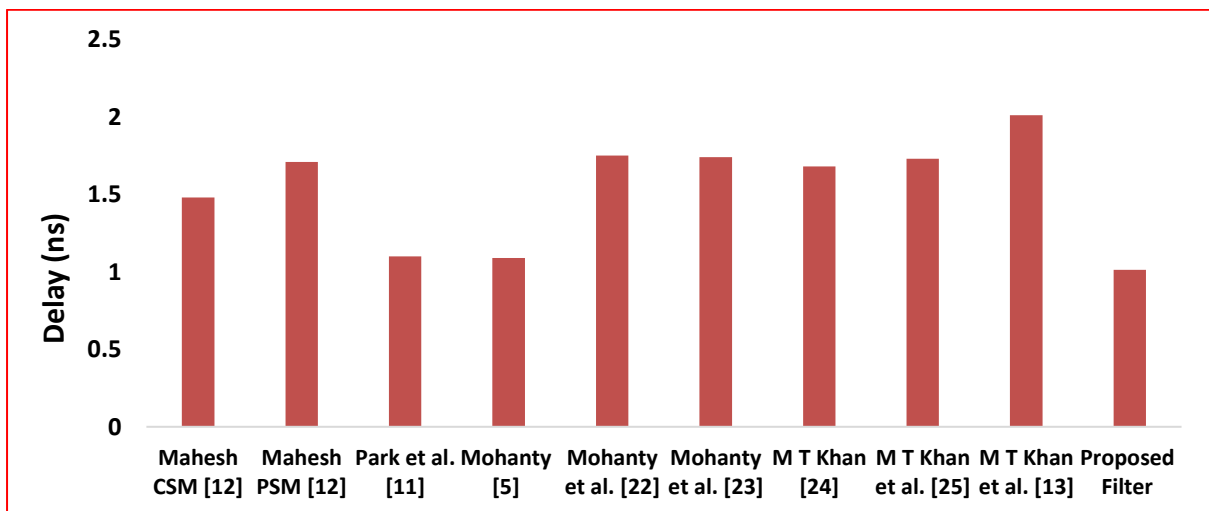


Figure 12: Comparison of Delay of the proposed structure with existing filter architecture for the length of the filter is 32.

The proposed FIR filter architecture with OBC-DA has 31.6%, 40.8%, 8%, 7.15%, 42.1%, 41.8%, 39.7%, 41.5%, and 49.6% less delay compared to filter architectures by Mahesh

CSM [12], Mahesh PSM [12], Park et al. [11], Mohanty [5], Mohanty et al. [22], Mohanty et al. [23], M T Khan [24], M T Khan et al. [25], and M T Khan et al. [13] respectively.

The graphical presentation of the Area Delay Product (ADP) of the proposed FIR filter and existing architectures in the literature is shown in Fig 13.

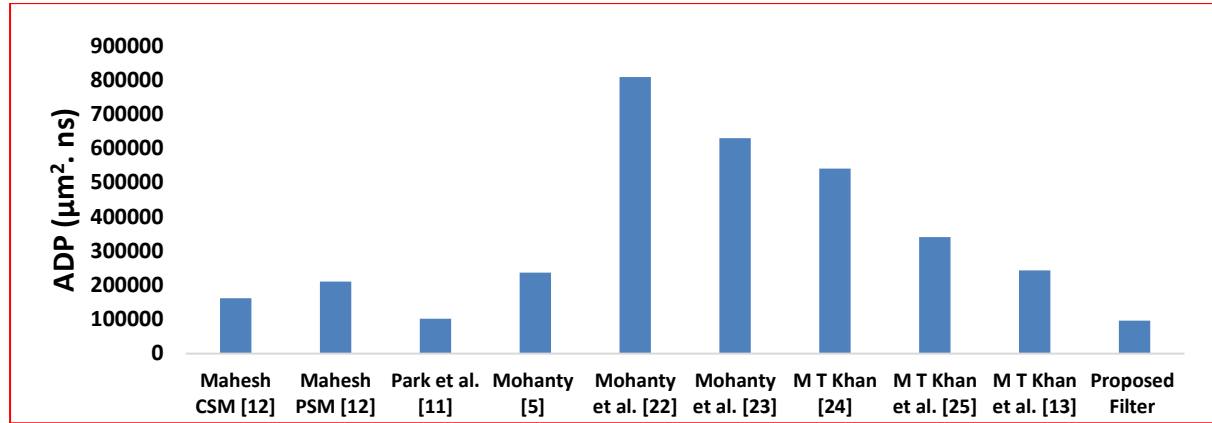


Figure 13: Comparison of ADP of the proposed structure with existing filter architecture for the length of the filter is 32,

The ADP values of the proposed FIR filter architecture are 1.67, 2.17, 1.05, 2.73, 8.35, 6.5, 5.58, 3.51, and 2.51 times less compared to the architectures by Mahesh CSM [12], Mahesh PSM [12], Park et al. [11], Mohanty [5], Mohanty et al. [22], Mohanty et al. [23], M T Khan [24], M T Khan et al. [25], and M T Khan et al. [13] respectively.

The comparative representation of the Power Delay Product (PDP) of the proposed FIR filter architecture with existing architectures in the literature is shown in Fig 14.

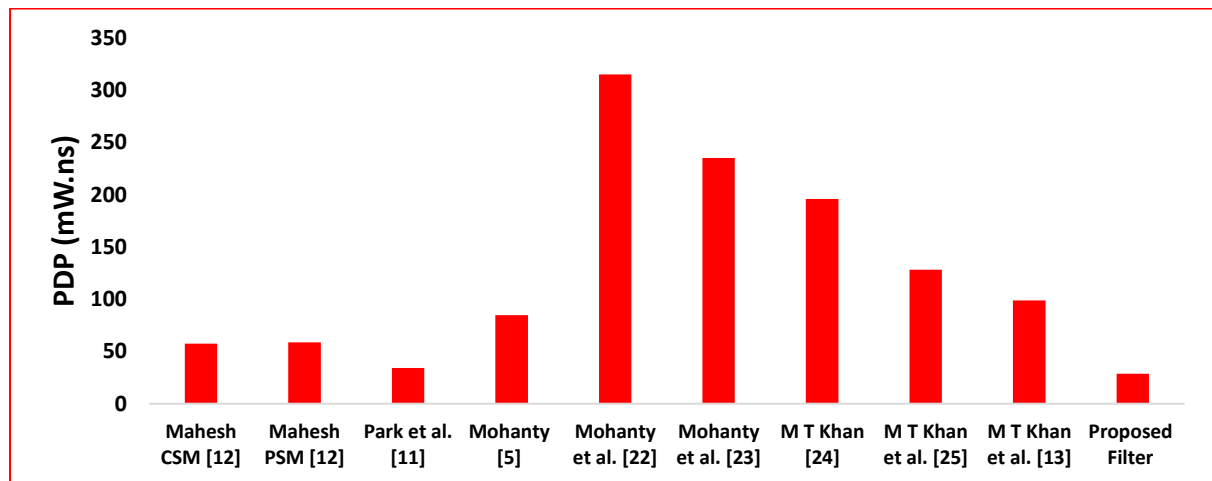


Figure 14: Comparison of PDP of the proposed structure with existing filter architecture for the length of the filter is 32.

The ADP values of the proposed FIR filter architecture are 2, 2.04, 1.19, 2.94, 10.9, 8.18, 6.82, 4.46, and 3.44 times less compared to the architectures of Mahesh CSM [12],

Mahesh PSM [12], Park et al. [11], Mohanty [5], Mohanty et al. [22], Mohanty et al. [23], M T Khan [24], M T Khan et al. [25], and M T Khan et al. [13], respectively.

6. Conclusion

In this paper, an efficient block-based reconfigurable FIR filter architecture using decomposed DA-based OBC scheme for digital channelizer. The SDR channelizer consists of various channel filters and desired channel filter can be selected by the control logic. The selected filter coefficients are used to produce partial products based on the OBC scheme. The systolic filter architectures are implemented using a direct form structure for various filter lengths. The parallel processing concept is used to improve the throughput and reduce the number of memory registers. The multiplierless design is achieved by DA based OBC concept. Due to the OBC scheme, the size of the LUT is reduced to half. Next, the OBC LUT is decomposed into smaller blocks to reduce the accessing time of ROM and complexity. The separate memory is not required for the storage of OBC contents, which are selected from the partial products generator block by mux logic. The proposed filter architectures with different filter lengths and block sizes are simulated and synthesized in ASIC-based tools provided by the Cadence Vendor. All the designs are implemented in 45nm CMOS technology using the Genus synthesis tool and area, delay, and power reports are generated. Finally, the comparative analysis is done with existing works. The proposed filter architecture is superior to the recent DA-based filter architectures in terms of ADP and PDP values. The proposed OBC-DA-based reconfigurable FIR filter architecture obtained 44.1% and 30% less ADP and PDP than the conventional DA-based filter architecture, respectively. In the future, this filter architecture can be implemented for very higher filter lengths with high parallel processing.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Hentschel, Tim, and Gerhard Fettweis. "Software radio receivers." *CDMA techniques for third generation mobile systems*. Springer, Boston, MA, 1999. 257-283.
2. Jondral, Friedrich K. "Software-defined radio—basics and evolution to cognitive radio." *EURASIP journal on wireless communications and networking* 2005.3 (2005): 1-9.

3. Kalathil, Shaeen, Bijili Sravan Kumar, and Elizabeth Elias. "Efficient design of multiplier-less digital channelizers using recombination non-uniform filter banks." *Journal of King Saud University-Engineering Sciences* 30.1 (2018): 31-37.
4. Mahesh, R., and A. Prasad Vinod. "Reconfigurable frequency response masking filters for software radio channelization." *IEEE Transactions on Circuits and Systems II: Express Briefs* 55.3 (2008): 274-278.
5. Mohanty, Basant Kumar, et al. "A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic." *Integration* 54 (2016): 37-46.
6. Aksoy, Levent, Paulo Flores, and José Monteiro. "A tutorial on multiplierless design of FIR filters: algorithms and architectures." *Circuits, Systems, and Signal Processing* 33.6 (2014): 1689-1719.
7. Chen, K-H., and T-D. Chiueh. "A low-power digit-based reconfigurable FIR filter." *IEEE Transactions on Circuits and Systems II: Express Briefs* 53.8 (2006): 617-621.
8. Kalaiyarasi, D., and T. Kalpalatha Reddy. "Area efficient implementation of FIR filter using distributed arithmetic with offset binary coding." *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)* 4.3 (2014): 01-09.
9. Chitra, E., T. Vigneswaran, and S. Malarvizhi. "Analysis and implementation of high performance reconfigurable finite impulse response filter using distributed arithmetic." *Wireless Personal Communications* 102.4 (2018): 3413-3425.
10. Praveen Sundar, P. V., et al. "Low power area-efficient adaptive FIR filter for hearing aids using distributed arithmetic architecture." *International Journal of Speech Technology* 23.2 (2020): 287-296.
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. Mahesh, R., and A. Prasad Vinod. "New reconfigurable architectures for implementing FIR filters with low complexity." *IEEE transactions on computer-aided design of integrated circuits and systems* 29.2 (2010): 275-288.
13. Khan, M. T., Shaik, R. A., & Alhartomi, M. (2021). An Efficient Scheme for Acoustic Echo Canceller Implementation using Offset Binary Coding. *IEEE Transactions on Instrumentation and Measurement*.

14. Ahmad, S., Khawaja, S. G., Amjad, N., & Usman, M. (2021). A Novel Multiplier-Less LMS Adaptive Filter Design Based on Offset Binary Coded Distributed Arithmetic. *IEEE Access*, 9, 78138-78152.
15. Prakash, M. S., & Shaik, R. A. (2013). Low-area and high-throughput architecture for an adaptive filter using distributed arithmetic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 60(11), 781-785.
16. Khan, M. T., & Shaik, R. A. (2018). Optimal complexity architectures for pipelined distributed arithmetic-based LMS adaptive filter. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(2), 630-642.
17. Guo, R., & DeBrunner, L. S. (2011). Two high-performance adaptive filter implementation schemes using distributed arithmetic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 58(9), 600-604.
18. NagaJyothi, G., & Sridevi, S. (2019). High speed and low area decision feedback equalizer with novel memory less distributed arithmetic filter. *Multimedia Tools and Applications*, 78(23), 32679-32693.
19. Rammohan, S. R., Jayashri, N., Bivi, M. A., Nayak, C. K., & Niveditha, V. R. (2020). High-performance hardware design of compressor adder in DA based FIR filters for hearing aids. *International Journal of Speech Technology*, 23(4), 807-814.
20. NagaJyothi, G., & SriDevi, S. (2017, March). Distributed arithmetic architectures for fir filters-a comparative review. In *2017 International conference on wireless communications, signal processing and networking (WiSPNET)* (pp. 2684-2690). IEEE.
21. K. Parhi, VLSI digital signal processing systems: design and implementation. New Delhi, India: Wiley India Pvt Ltd., 2007.
22. B. K. Mohanty and P. K. Meher, "A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 921–932, Feb. 2013, doi: 10.1109/TSP.2012.2226453.
23. B. K. Mohanty, P. K. Meher, and S. K. Patel, "LUT optimization for distributed arithmetic-based block least mean square adaptive filter," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 5, pp. 1926–1935, May 2016, doi: 10.1109/TVLSI.2015.2472964.

24. M. T. Khan and R. A. Shaik, "High-performance hardware design of block LMS adaptive noise canceller for in-ear headphones," *IEEE Consum. Electron. Mag.*, vol. 9, no. 3, pp. 105–113, May 2020, doi: 10.1109/MCE.2020.2976418.
25. M. T. Khan, J. Kumar, S. R. Ahamed, and J. Faridi, "Partial-LUT designs for low-complexity realization of DA-based BLMS adaptive filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 4, pp. 1188–1192, Apr. 2021, doi: 10.1109/TCSII.2020.3035693.
26. Odugu, Venkata Krishna, C. Venkata Narasimhulu, and K. Satya Prasad. "Implementation of Low Power Generic 2D FIR Filter Bank Architecture Using Memory-based Multipliers." *Journal of Mobile Multimedia* (2022): 583-602.
27. Odugu, Venkata Krishna, C. Venkata Narasimhulu, and K. Satya Prasad. "A novel filter-bank architecture of 2D-FIR symmetry filters using LUT based multipliers." *Integration* (2022).