

Performance Analysis of Reconfigurable Multiplier Unit for FIR Filter Design

Jalaja S

Dept. of VLSI Design and Technology
Bangalore Institute of Technology
Bangalore

Pooja M.V

Dept. of Electronics and Communication Engineering
Bangalore Institute of Technology
Bangalore

Abstract:- The design of Finite Impulse Response (FIR) filter performance is analyzed using Reconfigurable multipliers unit (Dadda, Booth, Wallace, and Shift & Add multipliers) and retimed SQR T CSLA block. The FIR filter is frequently used in digital signal processing technique for a variety of applications including speech processing, loudspeaker equalization, echo cancellation, noise cancellation, arithmetic computations, and image processing. In this paper, the FIR filter takes an input channel and produces multiple output channels by multiplying the input samples with corresponding filter coefficients. The reconfigurable nature of the filter allows for flexibility in selecting the type of multiplier based on specific performance requirements or resource constraints. The specific architecture and interconnections of the components, such as multipliers, adders, and output channels, depend on the chosen multiplier type and the desired property of the filter. The nature of the Control signals is to switch between different multiplier types and adjust the filter accordingly. To optimize the utilization of resources, a resource sharing principle is employed in the proposed FIR filter architecture, regardless of the number of channels and taps. These techniques ensure efficient resource allocation and utilization. The FIR architecture is restructured by incorporating adders and different multipliers in this design. This approach effectively reduces the area occupied by the adders and multiplier blocks, resulting in improved area efficiency and delay. The structure of the FIR filter has the multipliers arranged in a Multiply-Accumulate (MAC) structure, where the multiplication and accumulation operations are performed, and the delay blocks serve as the major building blocks of the filter. The speed of the multiplier is one the component of FIR filter performance, as it determines the critical path in the filter structure. As a result, the proposed architecture power consumption is less compared to existing method [18][19][20][21]. The modified FIR filter coding is implemented using, Verilog Hardware Description Language (HDL). The simulation and synthesis processes are carried out which allows for testing and optimization of the design. The paper introduces a novel approach low power Reconfigurable multiplier unit to design Finite Impulse Response architecture and it shows better efficiency compared to existing architecture [9].

Keywords:- Reconfigurable Multiplier, FIR Filter, Dadda, Booth, Wallace, and Shift & Add Multipliers, Resource Sharing Principle, Retime SQR T CSLA.

I. INTRODUCTION

Infinite Impulse Response (IIR) filters and Finite Impulse Response (FIR) filters are two filters categorized in digital filters. Linear phase response and inherent stability are the two highlights of FIR filters are preferred over IIR filters. The absence of feedback in the equation of FIR filters ensures stability, and their advantage lies in their ability to produce linear phases. FIR filters find extensive use in speech processing, noise cancellation, computer graphics, image processing, telecommunications, and consumer electronics applications. Multipliers play a tedious role in hardware blocks for Digital Signal Processing (DSP) and embedded applications. The speed of multiplication determines the overall processor speed. To achieve high-speed data rates, FIR filters are commonly used due to their stability, linear phase response, and non-feedback nature. FIR filters are stable because they lack feedback, unlike IIR filters. Additionally, their linear phase response makes them highly desirable. The novel design approach for an FIR filter, utilizing the enhanced Squirrel search algorithm (ESSA) and a variable latency carry skip adder (VL-CSKA) based Booth multiplier. The proposed ESSA algorithm optimizes the filter coefficient (FC) selection by minimizing switching activities, taking into account ripple contents, power, and the transition width parameter. This optimization ensures that the FIR filter meets the required specifications in the frequency domain [18]. A new multiplier design that outperforms the Array, Vedic, Booth and Wallace series of multipliers which are the four main categories of parallel digital multipliers. The proposed multiplier is an enhanced version based on combination of Wallace and Dadda multiplier architectures [22]. The simplicity of implementation is another advantage, as FIR calculations can be performed by looping a single instruction on most DSP microprocessors. In FIR filters, the total delay depends on the delay introduced by the adders and multipliers in the filter architecture, based on the number of taps (N) in the filter. Therefore, the design of FIR filters is significantly influenced by the adders and multipliers. For improved performance, it is essential to minimize the delay in the architecture of these components. FIR filters are commonly used in Digital Signal Processing (DSP) systems, and they operate by convolving the input data samples with the desired unit response of the filter. In this project, a 16-tap filter is designed, where the

order of the filter determines the number of multipliers and adders required. The filter output is a weighted sum of the current input signals and previous inputs. A power efficient design for the Wallace tree multiplier incorporating a power efficient 7:3 counter, which is composed of multiplexers and XOR gates. The partial product reduction is identified as the main contributor to the power consumption in the multiplier. To address this the proposed design utilizes a counter based modular Wallace tree (CBMW) multiplier approach [19]. A novel architecture for a low power, low area shift and add multiplier the architecture focuses on reducing the power consumption and area by modifying the conventional design. Key modifications involve minimizing the switching activities of major blocks within the multiplier such as adder and counter [20]. The Radix 8 encoding multiplier utilizes a 4-bit encoding scheme. The Radix-8 Booth multiplier exhibits drawbacks in parameters such as delay and speed when compared to a Radix-4 Booth multiplier. The limitations arise due to the intricate nature of circuit design in the Radix-8 Booth multiplier [21]. The objective of the project is to address the limitations mentioned above by designing a reconfigurable multichannel FIR filter that can dynamically switch between different types of multipliers, such as Dadda, Booth, Wallace, and Shift & Add. This flexibility allows for the selection of the most suitable multiplier type based on specific application requirements, optimizing the trade-offs between area and speed. The primary focus of the project is to reduce the area and delay associated with arithmetic units, specifically adders and multipliers, to enhance the performance of the FIR filter. To achieve this goal, different multiplier types, including Dadda Multiplier, Booth Multiplier, and Shift and Add Multipliers, are explored and evaluated. Achieving balanced logic utilization and efficiency is the primary objective when implementing an architecture utilizing modern FPGAs. Balanced logic utilization refers to effectively utilizing DSP slices, block RAMs (BRAMs), and look-up tables (LUTs) in a proportional manner. [23] For fixed filter coefficients, the option of ROM-based look-up tables (LUTs) is employed, while for reconfigurable coefficients, the option of distributed RAM-based LUTs is utilized [23]. The technique known as common subexpression elimination (CSE) is widely recognized as a popular approach for reducing the number of logic operators (LOs) in a digital filter. This is achieved by eliminating redundant instances of the same bit pattern [24]. The VFF-QRRLS-BC algorithm, based on the QR decomposition (QRD), is a novel approach for system identification in the presence of input noise. It incorporates bias compensation and introduces a new variable forgetting factor scheme, which aims to enhance both the convergence speed and the steady-state mean squares error of the algorithm [25]. To enhance the efficiency and speed of FIR filters while reducing their latency and hardware complexity, focus is placed on the development of high-throughput and energy-efficient designs. One approach involves utilizing transposed FIR filters, which naturally have a shorter critical path when compared to their direct form counterparts. Transposed filters achieve this by employing just a multiplier and an adder in their structure [26]. To address the challenges associated with Conventional adders and multipliers that lead to increased size and power consumption in n-sized repeated filters. The XOR-Mux

adders and Truncation Multipliers. This innovative approach effectively reduces the logic size from $2n$ to n , thereby mitigating the size and power consumption concerns. By avoiding the use of traditional adders and multipliers, the proposed methodology offers a more efficient and streamlined solution for n-size repeated filters [27]. An implementation of Multichannel FIR (Finite Impulse Response) filter using Time Division Multiplexing (TDM) the TDM mechanism allows for the use of a single multiplier and adder, regardless of the number of taps and channels in the filter. By applying the principle of resource sharing and increasing the operating frequency of the filter, the aim is to optimize the resource complexity of the multiplier. To achieve these two schemes are implemented Output Product Coding (OPC) and Dual Port Schematic architectures [1]. The fundamental operation of the filter can be achieved by employing either Multiply and Accumulate (MAC) blocks or shift and add algorithms. In the case of MAC, each stage of the filter necessitates an 'n' bit multiplier and an 'n-1' bit adder [2]. The Distributed Arithmetic (DA) algorithm aims to handle the products of sums in various filtering applications and frequency transfer function. This algorithm utilizes a Look Up Table (LUT) that stores the fixed coefficients of the FIR filter. By employing LUT based DA Algorithm instead of Multiply and Accumulate (MAC) units, several advantages can be achieved including increased efficiency, reduced area usage, improved speed, lower Power Consumption and decrease hardware complexity [3]. The carry By Pass Adder (CBA) is employed as a replacement for the conventional adder. By utilizing the CBA Adder, the RFIR architecture demonstrates enhanced performance in terms of reduced area, Power consumption and delay [4]. The design Space Encompasses seven hardware implementations. Initially the basic architectures of 16 tap direct form (DF) Transposed form (TF), Direct Form 2 (DF2), Transposed Form 2 (TF2) FIR filters are implemented. Following that a polyphase 16 tap structure using the DF filter referred to as DFPOLY, is implemented. Finally, the DF and TF2 filter structures are pipelined in the polyphase architecture resulting in the implementations of DFPOLYPIPE and TF2POLYPIPE [5]. In the RNS based system a collection of moduli is employed to convert the binary system into the RNS system. This conversion enables efficient implementation of arithmetic operations by minimizing carry propagation. By breaking down an operation into smaller operations, the RNS system reduces the impact of carry propagation, leading to improved efficiency in arithmetic computations [6]. By primarily concentrating on the development, analysis, and enhancement of Dynamic Partial Reconfiguration (DPR) systems, with a specific focus on the dynamic reconfiguration rate achievable on modern devices. As a result, a distributed arithmetic (DA) implementation that enables efficient implementations with compact hardware footprints on contemporary FPGA devices [7]. An alternative approach for performing calculations using variable partition hybrid type structures, with the aim of designing an advanced FIR channel. The operation of Multiply and Accumulate (MAC) play a crucial role in the FIR channel structure, where a coefficient is duplicated and compared with the delayed information test before the results are aggregated [8]. The Vedic Multiplier carry out multiplication operations within IPU of the transpose form

FIR filter. This specific multiplier is well suited for conducting parallel multiplication of a significant number of bits [10]. The Digital Up Converter (DUC) plays a vital role in digital front-end (DFE) circuits used in various RF systems such as communications, sensing, and imaging. Its primary function is to convert one or multiple channels of data from baseband to passband signal. This passband signal consists of modulated carriers at specific radio or intermediate frequencies (RF or IF), as defined by the predefined set of frequencies [11]. Adders and Multipliers are significant components in the design of FIR filters due to their impact on the overall delay. The total delay of an FIR filter is determined by the combined delay introduced by adders and multipliers within its architecture. The number of adders and multipliers present in the filter’s architecture depends on value depends on N, which represents the number of taps in the N-tap filter [12]. The DUC technique involves filtering and converting input signal into higher sampling rate. By employing multiple sampling rates in digital signal processing, a software defined radio (SDR) can benefit from increased flexibility. SDR refers to a single device that supports various standards accommodating, different sample rates, channel bandwidths, and carrier to noise ratios [13]. Discrete time (DT) domain circuits are becoming increasingly popular in various architectures, including N-path filters, sampling mixers and analog FIR/IIR/ FFT filters within this particular environment. Implementing discrete-time analog signal processing (DT-ASP) before an ADC brings significant advantages such as relaxed requirements for the ADC through flexible filtering, the potential to improved dynamic range performance, and enhanced robustness in the face of digital CMOS scaling [14]. As the number of additions and multiplications increases the computational complexity also grows. It explores the various implementation techniques for FIR filters. This FIR filter aims to minimize the number of arithmetic operations needed for inner product calculations to be a predetermined value. One such technique involves utilizing a look up table (LUT) design where pre-computed results are stored simplifying the overall process [15]. The fundamental of frequency sampling method for designing Finite Impulse Response in communication systems. It emphasizes the FIR architecture optimization technique for prototyping and hardware base 2D FIR filter models [16].

II. RECONFIGURABLE MULTIPLIER UNIT

Conventional method of Dadda, Booth, Wallace, and Shift & Add multipliers pseudo code is written below.

➤ *Wallace Tree Multiplier*

The Wallace tree algorithm combines adjacent partial products using a series of reduction layers. Here are the equations used in each reduction layer of the Wallace tree algorithm:

- **Initial Reduction Layer:** In this layer, adjacent partial products are added together in groups of three. The resulting sum and carry-out are computed as follows:

$$S[i] = P[i][0] + P[i][1] + P[i][2] \text{ (bit-wise addition)}$$

$$C[i] = P[i][0] \& P[i][1] | P[i][1] \& P[i][2] | P[i][0] \& P[i][2] \text{ (bit-wise carry-out)}$$

The S[i] represents the sum of the three partial products, and C[i] represents the carry-out generated from the addition.

- **Intermediate Reduction Layers:** In subsequent reduction layers, the sum and carry-out values obtained from the previous layer are combined in groups of three. This process continues until there are no more carry-outs remaining.

$$S[i] = S[i-1] + S[i] + C[i-1] \text{ (bit-wise addition)}$$

$$C[i] = C[i-1] \& S[i-1] \text{ (bit-wise carry-out)}$$

Here, S[i] represents the sum of the three inputs (previous sum, current sum, and carry-in), and C[i] represents the carry-out generated from the addition.

Final Reduction Layer: The final reduction layer combines the remaining sum values and carry-outs to obtain the final product. The last reduction layer may not have three inputs, depending on the number of bits in the operands.

$$S_final = S[n-2] + S[n-1] + C[n-2] \text{ (bit-wise addition)}$$

$$C_final = C[n-2] \& S[n-2] \text{ (bit-wise carry-out)}$$

Here, S_final represents the final sum, and C_final represents the final carry-out.

➤ *Booth Multiplier*

- **Initializations:**

- ✓ Initialize two variables: M (multiplicand) and Q (multiplier).
- ✓ Initialize an extra bit, Q(-1), initially set to 0.

- **Iterations:**

Repeat the following steps for each group of Radix-4 digits in the Multiplier.

- ✓ Check the pattern formed by the least significant two Radix-4 digits of Q and Q(-1).
- ✓ Based on the pattern, perform a specific operation:
 - If the pattern is 01, add M to an accumulator.
 - If the pattern is 10, subtract M from the accumulator.
 - If the pattern is 00 or 11, no operation is performed.
- ✓ Right-shift Q and Q(-1) by 2 Radix-4 positions, discarding the least significant Radix-4 digit of Q and assigning the previous least significant Radix-4 digit of Q to Q(-1).

• *Final Result:*

- ✓ The final result is obtained by considering the accumulated value in the accumulator after iterating through all the groups of Radix-4 digits in the multiplier.

➤ *Shift and Add Multiplier*

• *Initialization:*

- ✓ Initialize two variables: M (multiplicand) and Q (multiplier).
- ✓ Initialize a product register, initially set to 0.

• *Iterations:*

Repeat the following steps for each bit in the multiplier, starting from the least significant bit:

- ✓ If the least significant bit of Q is 1, add the multiplicand M to the product register.
- ✓ Right-shift the product register by 1 bit.

• *Final Result:*

The final result is obtained from the value stored in the product register after iterating through all the bits of the multiplier.

➤ *Dadda Multiplier*

The Dadda multiplier algorithm involves several equations to perform partial product generation and reduction. Let's go through the equations step by step:

- **Partial Product Generation:** For each pair of bits (A[i], B[j]), where A is the multiplicand and B is the multiplier:
- ✓ Generate a partial product P[i,j] by multiplying the two bits: $P[i,j] = A[i] * B[j]$
- **Partial Product Reduction:** The partial products generated in step 1 are then reduced using a carry-save adder (CSA) structure and a reduction tree. The CSA combines three partial products and produces two outputs: the sum (S) and the carry (C). The reduction tree operates in a cascading manner, where the carries from one level are propagated to the next level.

The equations for the reduction tree can be represented as follows:

- ✓ $S [0,0] = P [0,0]$
- ✓ $S [0,1] = P [0,1] + P [0,2] + C [0,0]$
- ✓ $S [0,2] = P [0,3] + P [0,4] + C [0,1]$
- ✓ $S [1,0] = P [1,0] + P [2,0] + C [0,0]$
- ✓ $S [1,1] = P [1,1] + P [2,1] + P [3,0] + C [0,1] + C [1,0]$
- ✓ $S [1,2] = P [1,2] + P [2,2] + P [3,1] + P [4,0] + C [0,2] + C [1,1]$

The carry outputs (C) from each level are fed into the next level to propagate the carries.

- **Final Product Calculation:** After the reduction process, the final product is obtained by adding all the sums from the reduction tree, taking into account the carries from the last level:

$$\text{Final Product} = S [0, N-1] + C[N-1,0] + C[N-1,1] + \dots + C [N-1, M-1]$$

Where N is the number of bits in the multiplicand and M is the number of bits in the multiplier.

III. PROPOSED METHOD

Consider the N-tap FIR filter mathematical expression or equation is given by below

$$y[n] = b_0x[n] + b_1x[n - 1] + \dots + b_Nx[n - N]$$

$$= \sum_{i=0}^N b_i x[n - i]$$

Where x(n) = input value

Y(n) = output value

bi = filter coefficient

N = filter order

In above equation Y(n) composed of sum and product units. In proposed method retimed CSLA is used to reduce the carry propagation delay. For multiplier unit novel based reconfigurable multiplier unit used to reduce the power consumption of design.

The Square root- Carry select adder block increases the longest path delay of the final output addition. In figure (1) Group 1 represents 2 bit ripple carry adder similarly group2, group 3, group 4 and group 5 indicate 4 bit, 8 bit, 13 bit and 19 bit ripple carry adder block respectively. The proposed architecture, the C0 block multiplexer is retimed itself to reduce the critical path delay. In conventional method linear CSLA and SQRT CSLA is directly proportional to addition speed and number of bit length N. In proposed adder retimed flipflops are placed to reduce delay for carry propagation. CSLA consists of large combinational blocks the global retiming is applied for full design and registers is moved across each critical path logic structure. In figure (1) the final addition and carry are selected by multiplexer each cutset introduce delay that breaks the path delay. The same procedure is applied to entire CSLA block to reduce final critical path.

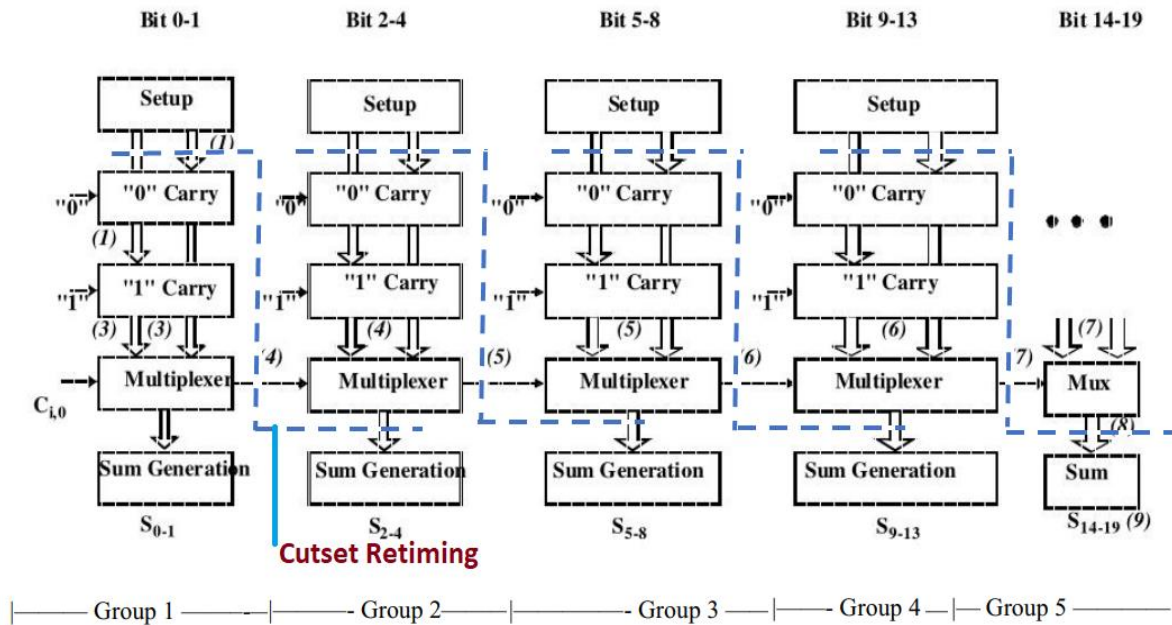


Fig 1 Cutset Retime SQRT-CSLA

In a reconfigurable multiplier as shown in figure (2) the coefficient values and input bits can be dynamically programmed into different types of multipliers. The idea behind this reconfigurable multiplier is to improve the performance of FIR filter output in an efficient way. The reconfigurable FIR filter with multichannel multiplier data flow graph is as shown in figure (3). The input values obtained from the multiplier mux are then directed to an adder. The adder performs the addition operation on these inputs. The purpose of this operation may vary depending on the specific requirements of the system or application. The output of the adder can be further processed or utilized in subsequent stages of the system for various purposes the optimization of resource consumption in a system that involves multiple filters. Specifically, it suggests reducing the number of filters by utilizing the programmability of Finite Impulse Response (FIR) filters. However, reusing the same filter block for every sample in all 40 filters can introduce a notable increase in latency. To address this issue, one option is to increase the clock frequency by a factor of 40. This increase in clock frequency would allow for faster processing of the samples, reducing the overall latency of the algorithm.

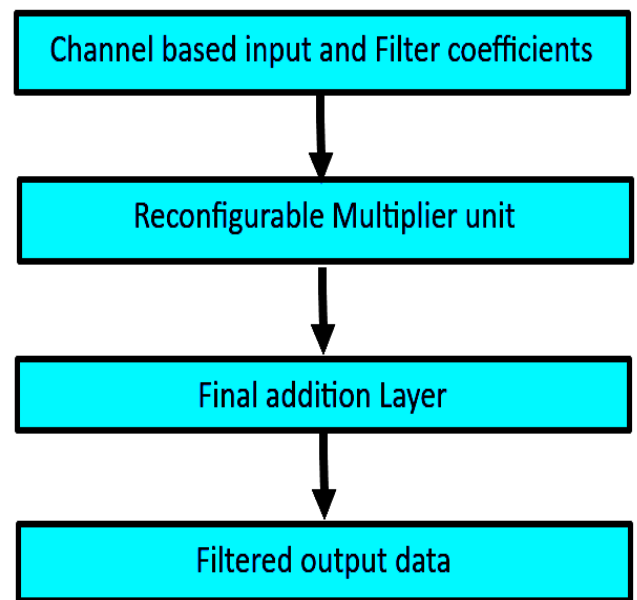


Fig 3 Data Flow Graph of Proposed Filter

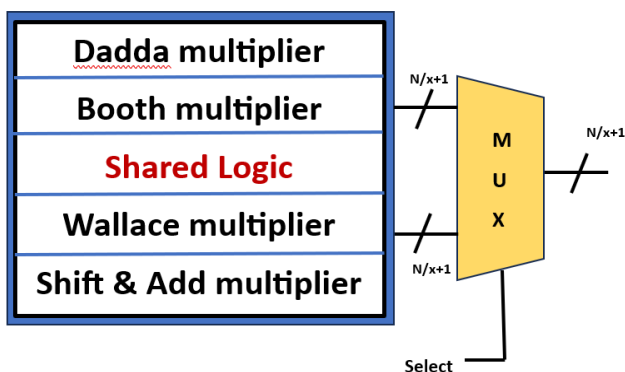


Fig 2 Proposed Reconfigurable Multiplier Unit

IV. RESULTS AND COMAPRISION

The figure (4) show simulation results of Wallace-tree Multiplier. The coefficients from H(0) to H(15) and inputs from CH1 to CH8 feed to Dadda, Booth, Wallace, and Shift & Add multipliers. Based on channel selection multiplier type is selected using multiplexer. The snapshot of 16-tap FIR Filter using Booth Multiplier filter is shown in figure (5). The multiplier mux output is feed to the adder unit to get final filtered data is illustrated in figure (6). The figure (7) shows, LUT's, & Slices from Technological Schematic.

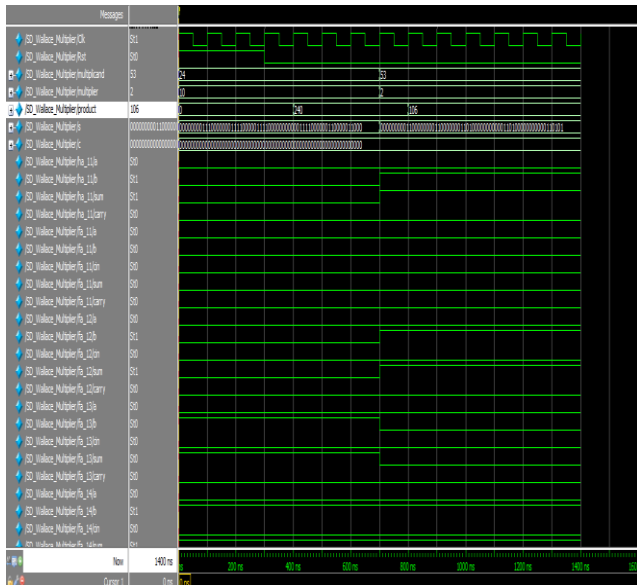


Fig 4 Simulation Results of Wallace-Tree Multiplier Using Model Sim

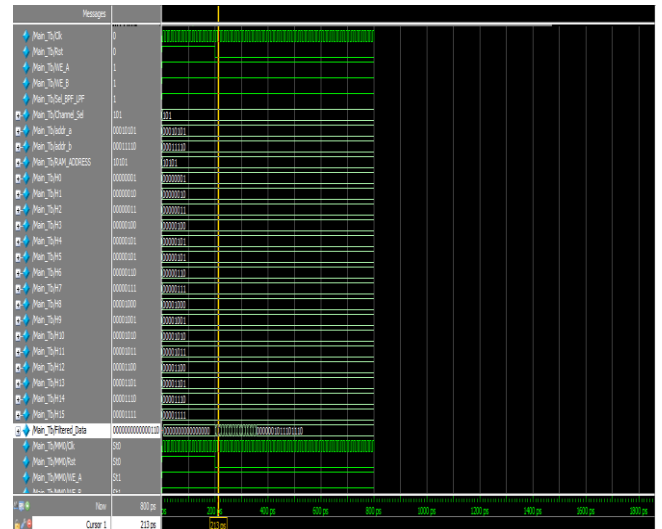


Fig 6 Simulation Results of 16-tap FIR Filter Final Output

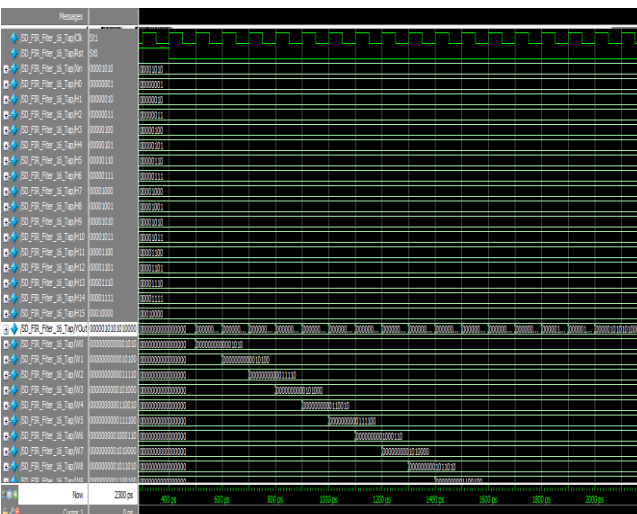


Fig 5 Simulation Results of 8bit 16-tap FIR Filter for Shift & Add Multiplier Using Model Sim FIR Filter

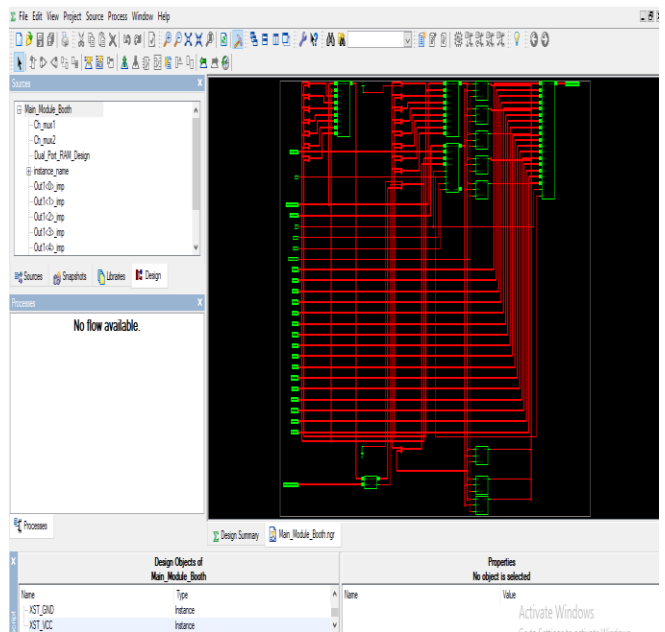


Fig 7 LUT's, & Slices from Technological Schematic

The Table 1 and 2 shows the synthesis report of 16-bit retimed Sqrt CSLA and modified 16-bit multiplier design Summary. It shows the Gate Count, delay, Slices and area comparison with existing method.

Table 1 Delay and Area Count of 16-bit Sqrt CSLA Groups

Type	ARCA Sqrt CSLA [15]		Modified Sqrt CSLA	
	Delay [ps]	Area [µm2]	Delay [ps]	Area [µm2]
Group 1	6	19	4	17
Group 2	10	44	9	38
Group 3	12	74	11	63
Group 4	14	104	12	94
Group 5	16	134	15	102

Table 2 Delay and Area Count of 16-bit Multiplier

Method Name	Proposed multiplier using modified Sqrt CSLA			Existing System	
	Slices	Gate	Overall Delay(ns)	Slices	Delay (ns)
Wallace-Tree Multiplier [18]	3523	8995	38.02	7474	72.47
Booth Multiplier[21]	1567	8819	12.5	1690	13.993
Shift & Add Multiplier[19]	778	8663	39.238	662	48.812
Dadda Multiplier[20]	2677	8621	2.277	3262	2.64

The performance of FIR filter using proposed reconfigurable multiplier unit synthesized using 180nm technology. The power, area and delay of 8bit 7-tap and 16-Tap FIR filter design performance is compared with existing design [9] is compared in Table 3. As a result, the proposed design shown much better result in terms of area-delay-product and power-delay-product is as shown in Table 4.

Table 3 180 nm Technology Area, Power and Delay Performance of 8-bit RFIR Design

Architectures	Bits and Taps	Area [μm^2]	Power [nW]	Dealy [ps]
RFIR–R2–LCSLA [9]	8B and 7T	2,14,781	19,84,548	258
RFIR–R2–LCSLA[9]	8B and 7T	2,14,781	19,84,548	258
RFIR–APC–OMS[9]	8B and 7T	1,92,962	11,40,187	130
Proposed RFIR	8B and 7T	1,98,453	10,32,091	102
Proposed RFIR	8B and 16T	1,99,268	10,57,080	114

Table 4 180 nm Technology Area Power Product and Area Delay Product Performance of 8-bit RFIR Design

Architectures	Bits and Taps	APP [$\mu\text{m}^2 * \text{nW}$]	ADP [$\mu\text{m}^2 * \text{ps}$]
RFIR–R2–LCSLA[9]	8B and 7T	4,26,24,32,03,988	5,54,13,498
RFIR–R2–LCSLA[9]	8B and 7T	4,26,24,32,03,988	5,54,13,498
RFIR–APC–OMS[9]	8B and 7T	2,20,01,27,63,894	2,50,85,060
Proposed RFIR	8B and 7T	2,04,82,15,55,223	2,02,42,206
Proposed RFIR	8B and 16T	2,10,64,22,17,440	2,27,16,552

V. CONCLUSION

Advanced multiplication techniques offer to optimize the area and delay utilization to enhance the performance of the design. Different multiplier types, including Wallace tree, Shift & Add, Booth, and Dadda, offer varying trade-offs in terms of complexity and performance. The proposed structure of a reconfigurable multiplier unit for FIR filter implementation provides opportunities for future developments in digital signal processing. It demonstrates the novel architecture flexibility and adaptability for filter design to process multiple channels of data simultaneously. By focusing on aspects such as filter structure, configurable taps and performance optimization, the proposed design lays a strong foundation for future advancements. It paves the way for further improvements and innovations in the field of multichannel FIR filtering.

REFERENCES

- [1]. J. Britto Pari and D. Vaithyanathan. An Efficient Multichannel FIR Filter Architecture for FPGA and ASIC Realizations. Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India. & Department of Electronics and Communication Engineering, College of Engineering Guindy, Anna University, Chennai, India. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 10 (2017) pp. 2209-2220.
- [2]. S. N. Raju Kalidindi, Sudheer Kumar Terlapu, M. Vamshi Krishna. Implementation of efficient reconfigurable FIR filter with control logic for 5G applications. The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021.
- [3]. Pradnya D. Shahare, Samrat S. Thorat. A Review: FPGA Implementation of Reconfigurable Digital FIR Filter. International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Volume 5 Issue 3, March 2016.
- [4]. Kasarla Satish Reddy, Hosahally Narayangowda Suresh. A Low Power VLSI Implementation of Reconfigurable FIR Filter Using Carry Bypass Adder. International Journal of Intelligent Engineering and Systems, Vol.11, No.2, 2018.
- [5]. Subhankar Bhattacharjee, Sanjib Silb Amlan Chakrabarti. Evaluation of Power Efficient FIR Filter for FPGA Based DSP Applications. International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013.
- [6]. J. Britto pari & S P Joy Vasantha Rani. Reconfigurable architecture of RNS based high speed FIR Filters. International Journal of Engineering & Materials Sciences Vol. 21 April 2014, PP. 233-240.
- [7]. Daniel Llamocca, Marios Pattichis, and G. Alonzo Vera. Partial Reconfigurable FIR Filtering System Using Distributed Arithmetic. Hindawi Publishing Corporation International Journal of Reconfigurable Computing Volume 2010, Article ID 357978, 14 pages doi:10.1155/2010/357978.
- [8]. A Murali, K Hari Kishore. FPGA Implementation of Proficient 16-Tap FIR Filter Design Using Decision Tree Algorithm. Turkish Journal of Computer and Mathematics Education Vol.12 No.3(2021), 3064-3075.
- [9]. Kasarla Satish Reddy, Sowmya Madhavan, Przemyslaw Falkowski-Gilski, Parameshachari Bidare Divakarachari and Arun Mathiyalagan "Efficient FPGA Implementation of an RFIR Filter Using the APC–OMS Technique with WTM for High-Throughput Signal Processing " Electronics MDPI 2022.

- [10]. S.Keerthana and J.Julie Antony Roselin. Optimized Design of FIR Filter using Vedic Multiplier for Reconfigurable Applications. *International Journal of Scientific & Engineering Research*, Volume 8, Issue 2, February-2017 ISSN 2229-5518.
- [11]. R. Sailakshmi, S. Padmapriya. Design and Implementation of Pulse Shaping RRC FIR Filter in Digital Up Converter. *International Journal of Research in Electronics & Communication Technology* Volume 4, Issue 3, May- June, 2016, pp. 01-09 ISSN Online: 2347-6109 Print: 2348-0017, DOA: 21053201 © IASTER 2016, www.iaster.com.
- [12]. S. Subathradevi, and C. Vennila. Delay Optimized Novel Architecture of FIR Filter using Clustered-Retimed MAC unit Cell for DSP Applications. Department of ECE, Anna University BIT Campus, Tiruchirappalli, India & 2 Department of ECE, Saranathan College of Engineering, Tiruchirappalli, India *Appl. Math. Inf. Sci.* 11, No. 4, , 1199-1205, 2017.
- [13]. G. Swathi ,M. Revathy. Design of a Multi-Standard DUC Based FIR Filter Using VLSI Architecture. *International Journal of Scientific Engineering and Research (IJSER)* Volume 3 Issue 11, November 2015.
- [14]. Sanjay Raman, Chair, Yang Cindy Yi, Dong. S. Ha, Jeffrey H. Reed, Rafael Davalos. Reconfigurable Discrete-time Analog FIR filters for Wideband Analog Signal Processing. Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering January 14, 2018.
- [15]. R. Jothin, P. Sreelatha, A. Ahilan & M. Peer Mohamed “High-Performance Carry Select Adders”, *Springer Link Circuits, Systems, and Signal Processing* volume 40, 2021.
- [16]. R. Prameela Jyothi, K. Sangeet Kumar and Mahesh K. Singh. Finite Impulse Response Filter Growth and Applications. *Advanced Production and Industrial Engineering* R.M. Singari and P.K. Kankar (Eds.) © 2022 The authors and IOS Press.
- [17]. Kamal Hossain, Roni Ahmed, Md. Asadul Haque, Muahmmad Towfiqur Rahman. *International Journal of Science and Research (IJSR)*. ISSN:2319-7064 SJIF (2019) :7.583. Volume 10 Issue 2, February 2021. A Review of Digital FIR filter Design in Communication Systems.
- [18]. Aditya Mandloi, Santhosh Pawar. *ELSEVIER Microprocessors and Microsystems* Volume 86, October 2021, 104333. Power and delay efficient FIR filter design using ESSA and VL-CSKA based on Booth Multiplier.
- [19]. Vaibhavi Solanki, A. D. Darji, Harikrishna Singhapuri. Design of Low-Power Wallace Tree Multiplier Architecture using Modular Approach. *Electronics Engineering Department, Sardar Vallabhbai National Institute of Technology Surat 395-007, India.*
- [20]. C. N. Marimuthu, Dr. P. Thangaraj, Aswathy Ramesan. Low Power Shift and Add Multiplier Design. *International Journal of Computer Science and Information Technology*, Volume 2, Number 3, June 2010.
- [21]. Vikas Kaushik and Himanshi Saini. A Review on Comparative Performance Analysis of Different Digital Multipliers. *Advances in Computational Sciences and Technology* ISSN 0973-6107 Volume 10, Number 5, 2017, PP. 1257-1272.
- [22]. Jasbir Kaur, Sumit Kumar. Performance Comparison of Higher Radix Booth Multiplier Using 45nm technology. *International Journal Innovative Research of Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization)* Vol. 5, Issue 1, January 2016.
- [23]. Maamoun, M.; Hassani, A.; Dahmani, S.; Ait Saadi, H.; Zerari, G.; Chabini, N.; Beguenane, R. Efficient FPGA based architecture for high-order FIR filtering using simultaneous DSP and LUT reduced utilization. *IET Circuits Devices Syst.* 2021, 15, 475–484. [CrossRef].
- [24]. Roy, S.; Chandra, A. A triangular common subexpression elimination algorithm with reduced logic operators in FIR Filter. *IEEE Trans. Circuits Syst. II Express Briefs* 2020, 67, 3527–3531.
- [25]. Tan, H.J.; Chan, S.C.; Lin, J.Q.; Sun, X. A new variable forgetting factor-based bias-compensated RLS algorithm for identification of FIR systems with input noise and its hardware implementation. *IEEE Trans. Circuits Syst. I Regul. Pap.* 2019, 67, 198–211.
- [26]. Patali, P.; Kassim, S.T. High throughput and energy efficient FIR filter architectures using retiming and twolevel pipelining. *Procedia Comput. Sci.* 2020, 171, 617–626.
- [27]. Radhakrishnan, P.; Themozhi, G. FPGA implementation of XOR-MUX full adderbased DWT for signal processing applications. *Microprocess. Microsyst.* 2020, 73, 102961.