

High-Speed and High-Performance FIR filter design for Wireless embedded Systems

Charles Rajesh Kumar J.

Research Scholar, Bir Tikendrajit University

Dr. Raghavendra. D. Kulkarni,

Research Supervisor, Bir Tikendrajit University

Article Info

Page Number: 13293 – 13298

Publication Issue:

Vol 71 No. 4 (2022)

Abstract: In the present paper, we study the growth properties of entire functions of several complex variables. The characterizations of generalized lower order of entire functions of several complex variables have been obtained in terms of their Taylor's series coefficients. Also, we have obtained the characterization of generalized lower order of entire functions of several complex variables in terms of approximation and interpolation errors.

Article History

Article Received: 25 August 2022

Revised: 30 September 2022

Accepted: 15 October 2022

Keywords and phrases: Entire function, Maximum term, Central index, generalized lower order, Approximation errors, Interpolation errors

Abstract

Today, immeasurable analytics is used in signal-processing and communication applications. The FIR filter is commonly used to improve signal quality by filtering. The speed of the multiplier module engaged in the function of DSP determines the system's performance. Because multipliers serve as essential components of the FIR filtering scheme, their effectiveness plays a role in defining the implementation of the FIR filter scheme. The rapid expansion of portable mobile communication systems and multimedia has raised the demand for rapid-speed signal processing systems with small footprints and low power usage. FIR filters are widely employed in image processing, mobile communication, signal processing, voice and video signal processing, noise filtering, healthcare electronics, and other applications. The fundamental components of multipliers and adders determine the total effectiveness of a signal processing system using an FIR filter. The FIR filter may function at a high sampling rate for a particular purpose and at an acceptable sampling rate with little power consumption for another. DSP systems employ Software Defined Radio (SDR) to substitute analog calculation in wireless communication. The channelizer is a computing significant component of an SDR receiver that needs to consume less power and operate at a higher sampling rate. As a result, the adders and multiplier utilized in the layout of the FIR filter have to be swift and effective. The growing popularity of laptops and portable devices in wireless networks has fueled studies on low-power microelectronics. There are more portable applications than ever before that require little energy along with substantial throughput. As a result, low-energy system architecture has emerged as an essential performance aim. As a result, this work faces additional constraints: fast speed high throughput, while consuming as little power as possible. The FIR Filter is an essential element in developing an effective DSP system. As a result, an attempt is made in this work to construct an energy efficient and high-speed FIR filter. When the fundamental framework of an FIR filter is

examined, it is discovered to be a mixture of multipliers and delays that are, in turn a combination of adders. This report describes the successful execution along with outcomes analysis. Adder and multiplier assessment to reduce energy use during addition and multiplication approach to increase performance by comparing different types of Multipliers and adders, accordingly. Utilizing the power comparison results of multipliers and adders, a low-power multiplier and adder are chosen to create an efficient FIR filter. Models' software utilizing Verilog code has executed the suggested FIR filter design. The FIR filter was built using this approach, utilizing a control unit, RAM, coefficient RAM, and an accumulator circuit. The FIR design was constructed without a multiplier, resulting in less time and area. For 32-tap, power, area, and latency have all been considered. The suggested approach improves Flip-flops, slices, and LUT in FPGA implementations. The primary concept behind the proposed filter algorithm is to substitute adders and multipliers with adders and shifters to minimize hardware costs. Only shifters and adders are required throughout the FIR filter implementation, with no multipliers.

Keywords: -FIR Filter, Wireless Embedded System, DSP, Adder, Multiplier, Shifter

1. Introduction

FIR filtering utilizes convolution and correlation to execute its action in overall DSP applications [1]. Filtering is a method of extracting valuable data from a signal. This is accomplished by conducting weighted sums on the specified input signals for filtering, digital audio, and video signals during filter transmission. After removing the undesirable movement known as noise, the signs with beneficial data were saved with only their vital details [2]. Several mathematical calculations are performed on a sampled discrete-time signal for any alterations to condition the signal. The convolution approach determines the filter's response for the provided signal $k(n)$. Several multiplication techniques make use of adders to perform their computations. Two architectures are presented in [3] in this paper. To analyze delay and critical path, sequential and parallel microprogram FIR filters with Vedic multipliers and Wallace tree multipliers (WTM) are utilized to evaluate their performance based on ASIC implementation results. This research demonstrates that the WTM outperforms FIR systems.[4] suggested a high-performance and low-power FIR filter solution for assigning FIR filters with customizable coefficients. Computation sharing underpins these virtualized resources. This study aims to reduce duplicate computations to accomplish high-performance Filtration processes. By recognizing familiar calculations, the computation-sharing strategy helps reduce the surplus component involved in filtering. [5] proposed combining FIR filters with allocation of resources algorithms for greater efficiency and equivalent energy consumption with pre-emptive operations using FIR filters designed using WTM and CSA multiplier, which are also used in DSP for accelerating up activities and adaptive filtering applications. [6] improved the system's performance by developing an FIR filter based on the Urdhava- Tiryagbhyam algorithms. He devised the task to cut calculation time better than the built-in MATLAB function while outperforming the other solutions. Among the several FIR framework techniques, linear convolution is the most fundamental.FIR filters are critical in DSP systems because their feed-forward and linear-phase implementation features render them extremely important in the development of robust exceptionally well filters. **Fig. 1(a) and (b)** correspondingly show the transposed direct-form and direct-form FIR filter implementations. Though the two designs have comparable hardware complexities, the transposed version is often favored due to its greater power and performance

efficiency. Because a large number of constant multiplications are required, the multiplication element of the transposed direct-form digital FIR filter, in which the expansion of the filter coefficients with the filter input is accomplished, significantly impacts the overall complexity and effectiveness. This is known as the multiple-constant-multiplications (MCM) operations, and it is a critical function and performance constraint in various DSP applications, including rapid error-correcting codes, discrete cosine transforms (DCTs), and Fourier transforms. Despite power-efficient, area, and delay multiplier designs, such as Wallace [7] and improved Booth [8], which have been suggested, complete multiplier flexibility is not necessary for constant multiplications because filter coefficients are fixed and defined in advance by algorithms for DSP [9].

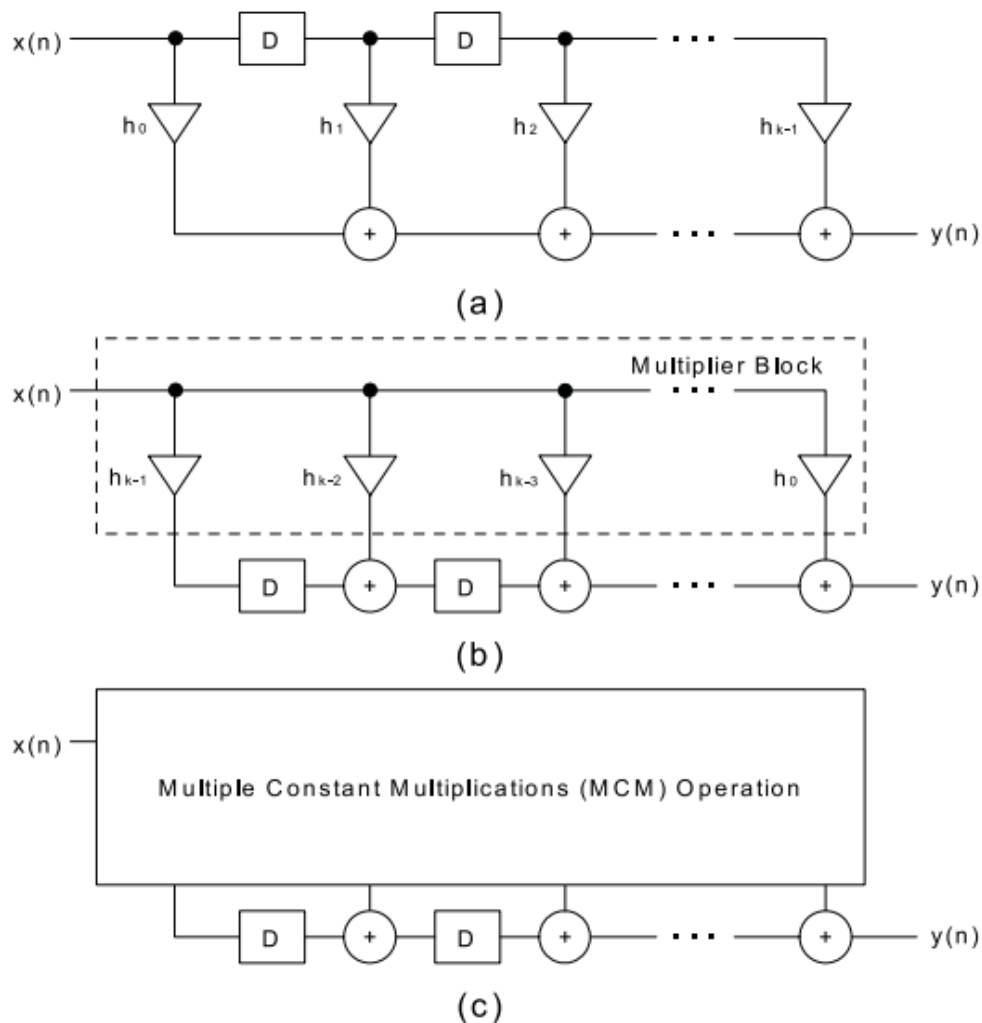


Fig.1. (a) direct form of FIR filter and (b) Transposed direct-form of FIR Filter.

As a result, filtering coefficient expansions using input information is often accomplished using an addition and shifting architecture [10] involving every constant multiplication realized using subtraction/addition and shift processes in an MCM procedure [Fig. 1(c)]. FIR filters are essential components in several wireless hand-held systems for various video and image processing communication applications to minimize noise while improving certain features. The specialized

filter was developed to fit the applications while having the least redundancy. The previous implementations of the technical filter design had several flaws. The cost of subexpression sharing [10, 11] is a complex process that resembles a chaotic adder tree. Substructure sharing will cause registers to develop rapidly to maintain proper timing. Furthermore, the folded design cannot take benefit of the fixed coefficients [12, 13]. As a result, the folded design will have larger area and large energy utilization. The coefficients of filter are commonly signified by the canonical signed digit (CSD) in the direct and transposed forms to reduce the non-zero numbers of the constant multipliers. Simultaneously, Rajeev et al.[14] and Laskowski[15] helped to eliminate the MSB sign extension duplication. On the other hand, the structural symmetry in the linear-phase frequency response can't be implemented in transposed direct-form filter architectures. Since multipliers are often more expensive regarding area and power use than adders, several earlier efforts have concentrated on constructing FIR filters using area-efficient multipliers. B Since certain specific to the application FIRs have predetermined coefficients, many constant-multiplier-based designs (CM) have been suggested instead of the costlier general multipliers [16, 17, 18, 19, 30,21]. These CM-based FIRs, on the other hand, are specific to the application and only work with a limited set of coefficients; thus, they are not suited for real-time reconfiguration devices with programmable coefficients, including signal equalization and adaptive pulse shaping.

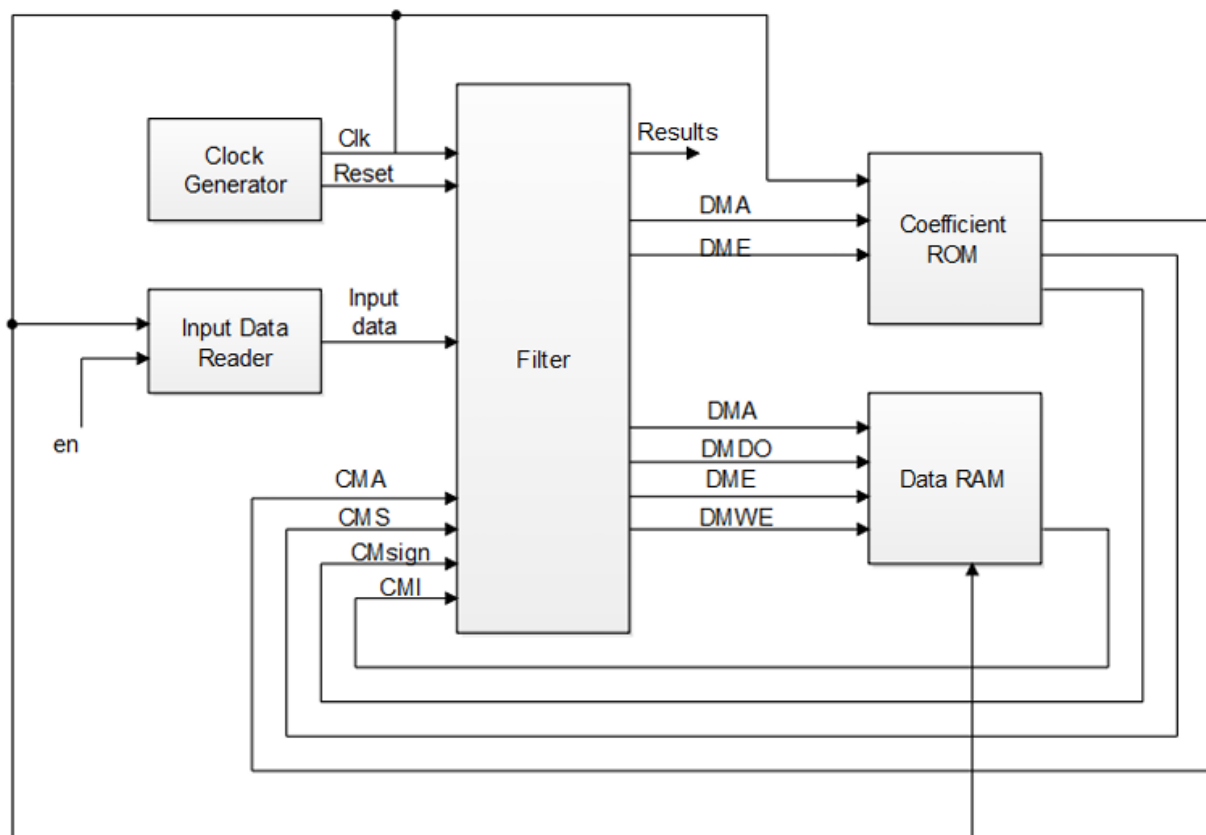


Fig. 2 The traditional structure of FIR filter architecture

On the contrary, FIR filters are widely employed in cellular wireless communication systems and high-throughput multimedia signal processing. Therefore, there are various parallel FIR

implementations, including fast FIR algorithms (FFA) [22] and block FIRs [23]. The core idea behind FFA is to use polyphase breakdown to divide an FIR filter into multiple sub-filters that can run simultaneously, resulting in lower computing complexity. The necessary amount of multipliers in an FFA-based FIR filter layout can be considerably decreased at the expense of a rise in adders for additional post and preprocessing. Symmetric FFA-based systems using symmetric convolutions have also been suggested in [24, 25]. While FFA-based approaches can reduce multipliers significantly, they can only be efficient for parallel FIR filters with modest parallelism. Alternatively, the increasing adders will provide substantial overhead space as the architecture's complexities increase. Block FIRs, on the other hand, have been suggested in [26, 27, 28] for high-throughput signal processing. Block FIRs may be integrated with CM-based techniques for a given coefficient set, unlike FFA-based systems, yet the associated level of parallelism increases the hardware resource dramatically. Consequently, the space and energy savings of modern parallel FIRs still need to be addressed. The traditional structure of FIR filter architecture is illustrated in Fig. 2. The typical FIR filter block comprises a filter, a clock generator, an input data reader, data RAM, and a coefficient ROM. The input data reader provides the value of the input data. The coefficient ROM stores the coefficient values. The clock generator generates the clock signal. The input data is sent to the data RAM for storage. CMI (Coefficient memory input) filtering requires C_{sign}, C_{MS}, and C_{MA}. C is the coefficient, MA is the memory address, M_{sign} is the memory sign, and MI is the memory input. CMI is used to determine whether to add or subtract. C_{MS} is employed in the shifting operation. The current method saves the co-efficient value in the ROM, which takes up more space. In addition, the standard adder is employed to complete the addition function. The suggested technique stores shift data only in ROM while using a low area carry select adder to save space, power, and delay.

2. Literature Review

DSP circuitry is essential in computer and communication systems. One application of DSP is a FIR filter. The fundamental purpose of this research is to provide a method for updating the architecture of a FIR digital filter from software development to the hardware stages. It involves settling on an architectural strategy, structure, and the most cost-effective hardware. Given the accessibility of a well-defined equation, the practical and theoretical findings from the FIR band pass filter show that the window designing technique is reasonably simple and straightforward. The most significant objectives for DSP processor development and execution are area optimization and energy use minimization. The FIR Filter is the essential building element for designing and implementing the DSP processor. The FIR Filter comprises three fundamental modules: multiplier, flip-flops, and adder unit. The multiplier, the slowest block of everything, dramatically influences the FIR Filter's effectiveness [29]. For DSP applications, digital filters are effective structures, signal evaluation, and estimation [30]. The number of operations has increased as technology has advanced. With VLSI-based technological advances, the time required to create digital filters has significantly decreased, leading to the creation of on-chip VLSI-oriented design for DSP applications. The two basic types of digital filters are IIR and FIR filters depending on their impulse response. FIR filters outperform IIR filters in terms of stability and assured linear phase features, and they are more straightforward to set up. FIR filters are also more computationally efficient, lowering the number of computations. FIR filters have the drawback of using more significant amounts of memory compared to IIR filters. Multiplication is the essential function of FIR filters, requiring increased hardware in terms of

delaying elements, speed, and area, as well as an increase in energy usage, resulting in poor filter design. As a result, these parameters and the computation conducted in terms of multiplier must be minimized. The RNS filter incorporates optimized adder and multiplier designs to minimize the filter's size and latency. Using optimized multipliers and adders in RNS FIR filters may decrease hardware complexities, increase efficiency, and reduce energy use. The fundamental action of a FIR filter is to multiply the input samples by the filter coefficients and then add the results. Optimized multipliers, such as WTM, LUT multipliers, and Dadda multipliers, may decrease the number of partial products (PP) needed, lowering the filter's overall hardware complexity. Consequently, energy use and circuitry area may be reduced [31]. Similarly, by lowering the propagation latency of carry signals and minimizing the total amount of levels of logic gates in the adder circuitry, optimized adders such as CLA adders, KSA adders, and suggested adders can increase the filter's effectiveness. As an outcome, the filter will run faster and consume less power. Optimizing multipliers and adders in FIR filters may improve performance and reduce power use and circuit space, making them a critical design factor for practical DSP applications [32]. DSP knowledge is in high demand because of the critical importance of its jobs. Several multipliers and adders are frequently found in advanced DSP systems. Sophisticated signal processing methods with well-designed multipliers and adders can produce superior outcomes. Adders are essential for numerous situations, such as processor chips and controllers. Adders can be identified in a variety of networks and structures. The time it takes for a "carry" to propagate through a digital adder limits its addition rate. The sum for every bit location is created sequentially in a typical ripple adder after adding up all of the preceding bit locations and "carry" transfers into the subsequent bit position [33]. MIMO applications heavily use the numerous parallel methods available for high throughput devices. The "parallel" design enhances system performance, but the L-parallel filter layout exponentially raises hardware expense and energy use. Due to this limitation of parallel structure, various fast multipliers were created using a mix of adders [34]. This approach uses less space than typical "parallel" architecture, decreasing hardware requirements in half. There would only be a lot in an FIR filter with the multiplier and adder. The filter as an entire thing can achieve the specified processing speed and energy dissipation if the adder and multiplier blocks work correctly. Numerous efficient adder and multiplier designs have been developed [35]. Compared to previous designs, the Dadda and Wallace multiplier produced favorable outcomes in terms of delay. The implementation findings show that the suggested configuration outperforms the standard one in terms of astounding velocity and decreased dissipation of power [36]. The PP-reducing step in multiplication processes is well recognized for its high consumption of energy and silicon area use. As a result, three significant methodologies are commonly used to construct approximate multipliers. When creating PP, the first strategy employs approximations. Following that, the PP tree is truncated. The final approach approximates the compressors and partial derivative adders. As a result, approximation computation was created to decrease energy use. They apply probabilistic pruning, an approximation technique described by [37], in their investigation. [38] Introduced input reordered 4:2 compressors, error-compensated approximation multiplier, the multiplier using OR gates, and produced a low-energy FIR filter. By rearranging the information, the compressor can work with only two of the four inputs, making it simpler and requiring a smaller amount gates. The suggested method provides 99.3 percent accuracy while utilizing 44.7 percent less power and 31.7 percent less area than the best-known techniques. [39] suggested 8X8 approximation multipliers utilizing greater-order approximation compressors (GOAC). Utilizing separate compressors for various weights, you can accumulate

product terms while minimizing energy usage with a few mistakes. GOAC, such as 8X2 compressors, are employed for the intermediary significance weights to streamline the 'carry chain' logic. [40] developed a rounding method-based approximations multiplier for generating an error-efficient structure, which requires rounding up the input of the operands to the subsequent power of two. The modified inputs are processed by a computation unit comprised of subtractor, adder, and shifter units. The size of the input operands can range from 8-bits to 32-bits. According to the results of simulation, the lag time is around 22 percent, and the amount of energy consumption is about 57 percent, representing gains over similar approximation multipliers. [41] suggested a rounding technique that could be changed on the spot and acts as an approximation multiplier to alleviate this difficulty. The multipliers proposed are appealing since they decrease implementing complications while increasing power consumption. The suggested approach consumes 32.5 percent less energy, occupies a 50.8 percent lower area, and has 54.7 percent less delay than filters that employ current multipliers. [42] proposed adding an approximate compressor with just one gate to generate an approximation multiplier. The proposed device occupies 52% less space and consumes 61% less energy than existing techniques [43] proposed an approximation multiplier for un-signed integers because of its high configurability; It attempts to reduce all hardware parameters while retaining exceptional precision. It offers a variety of alternatives to decrease energy consumption by 35-85%, allowing it to be used in various scenarios without breaking the bank. [44] proposed a truncating multiplier as an approximation multiplier. The final value was calculated by trimming the intermediary outcomes and using scientific-binary descriptions of the operands. Compared to the identical multiplier, this one saves 89.2% more power while taking up 74.9% less space on average. This paper describes a unique approximation adder that can be utilized with a high-performance, energy-efficient, approximation multiplier uses a simple tree of approximate adders for PP accumulation. To prevent 'carry propagation,' the suggested approximation adder produces an error vector and a preliminary total. M2 and M1 are two separate architectures for approximate 8X8 multipliers provided by the approximation adder-based reduction of error approaches using OR gates [45]. The recommended approximation multipliers have been shown to use less energy than the speed-optimized accurate Wallace multiplier. The suggested multipliers provide great precision due to their small error margins. As an added benefit, simulations reveal that M2, while having a greater delay and consuming more significant amounts of energy, is more precise than M1. The suggested multipliers are more exact than those employed in previous approximation models. Compared with previous designs that prioritized latency and energy efficiency but had inconsistencies in precision, the suggested alternatives were able to save much while preserving an excellent level of accuracy.

3. Proposed Methods

Figure 3 illustrates the proposed design of FIR filter. This structure contains a filter, clock generator-CG, and accumulator-AC, address generator-AG, control unit, RAM and ROM. The CG generates the clock signal. Shifting information may be kept in ROM, and input data values can be saved in RAM. The filter utilizes the clock signal from CU to compute the filter result, and the reset signal serves to reset the registers in the filtering unit. The AG will create an address that can be utilized for reading ROM data to compute the input data and filter coefficient. The AC stores the outcome of the filter results. N dividing sections with the shifting count is needed to accomplish the filtering process. The procedure is performed after the data reader unit gives the input data. Compute the memory address and activate data RAM for storing

the input data in the initial phase. The subsequent step allows the coefficient ROM to read each coefficient individually to obtain the filter output. Lastly, by the details, activate data RAM. In the design High-speed and energy efficient carry-select adder (CSLA) is employed. The carry select adder architecture is shown in **Fig.4**. The CSLA adder and shifter aid to reduce hardware costs. The difference equation of an n th order FIR filter is written as shown in **Equation (1)**. The transfer function $H(z)$ is given as shown in **Equation (2)**.

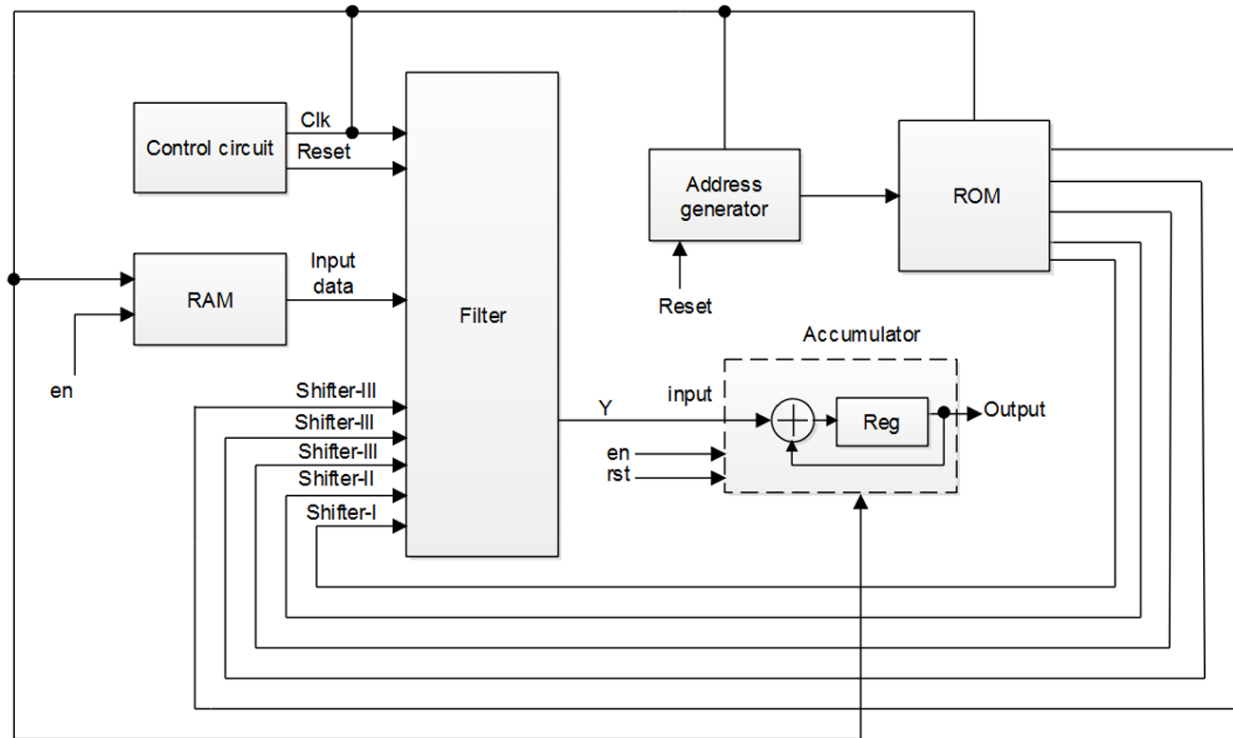


Fig.3 Proposed architecture of FIR filter

$$y(n) = \sum_{k=0}^{N-1} h(n)x(n-k) = \sum_{k=0}^{N-1} b_k x(n-k) \quad (1)$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^N a_i z^{-i} \quad (2)$$

The input value is usually convoluted with a co-efficient value. We'll need N multipliers and $N-1$ adders to do this. This required more area for calculating the outcomes. The suggested approach reduces the power, area, and latency further. The primary concept behind the proposed filter algorithm is to substitute adders and multipliers with adders and shifters to minimize hardware costs. Only shifters and adders are required throughout the FIR filter implementation, with no multipliers. Left and Right shifters are an effective multiplication and division operation method. If an unsigned or signed digital number has been shifted left by n -bits, we may multiply the input value by 2^n . The input value is divided by 2^n in the right-shifting technique. So, logic shifters can be used for all division and multiplication operations. The total expense and hardware

utilization will be reduced if the design uses fewer multipliers and adders. Using the help of the adder and logic-shifter, the suggested approach reduces overall energy use and area.

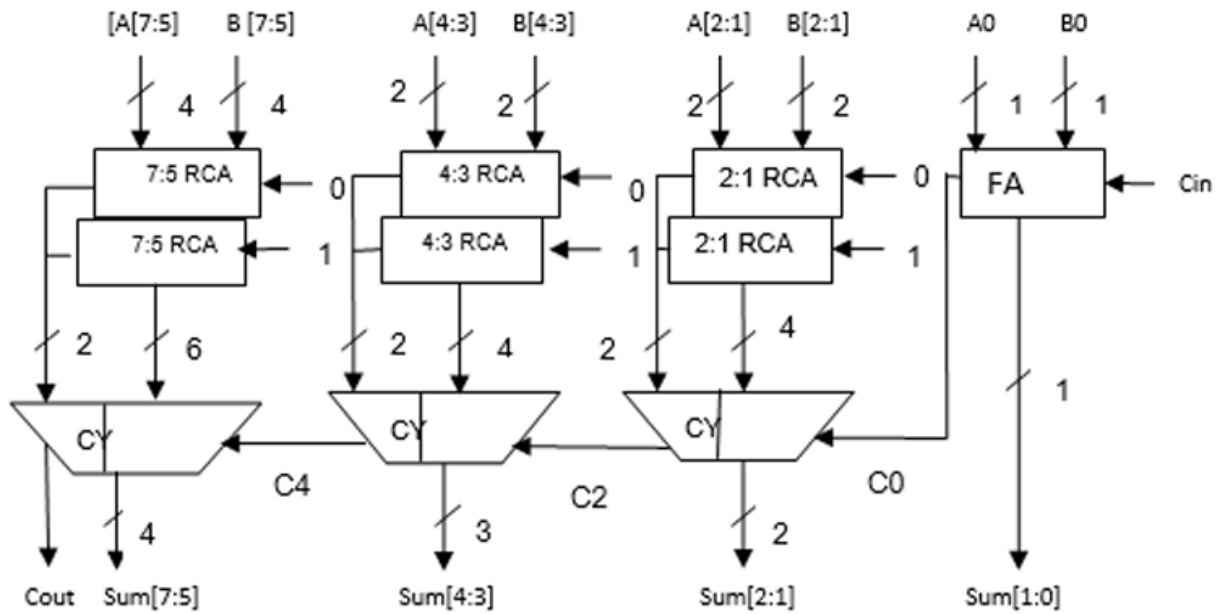


Fig.4 8-bit Carry Select Adder (CSLA) using RCA architecture. Here MUX is shifter-I

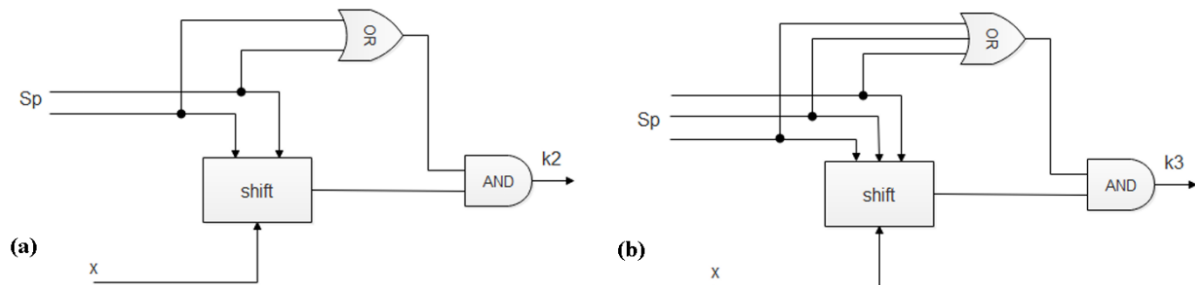


Fig.5. (a) Shifter-I and (b) Shifter -II

The amount of time needed to propagate a carry via a digital adder limits the speed of additions. In a basic adder, the total for every bit location is created sequentially only when the preceding bit location has been added, and a carry propagates into the following position. The CSA is a middle-of-the-road adder in terms of area and speed. Several computational frameworks use the CSLA to solve the carry delay in propagation by separately creating several carries and then selecting one to produce the sum. The CSLA, on the other hand, is not area effective since it employs several pairs of Ripple Carry Adders (RCA) to produce the partial carry and sum by taking into carry input $C_{in} = 1$ and $C_{in} = 0$, after which MUX picks the final carry and sum. The central concept behind this work is to employ a Binary to Excess-1 Converter (BEC) rather than an RCA with $C_{in} = 1$ in a standard CSLA to reduce power and area usage. The key benefit of this BEC logic architecture is that it has fewer logic gates than the n-bit Full Adder (FA) architecture. A CSLA is typically made up of MUX and two RCA. Adding two n-bit values with a CSLA requires two adders (hence two RCA) to perform the computation twice, once with the assumption of the carry = 0 and carry=1. After calculating the two outcomes, the proper sum and

the appropriate carry are chosen using the MUX after the proper carry has been determined. Here MUX is the Shifter-I and Shifter II and Shifter-III are shown in **Fig.5**. One of the most essential building pieces in FIR filter construction is the Processing Element (PE). According to the PE, the rest of the block is performing well is shown in **Fig. 6**. Instead of multipliers, the shifter is employed instead. Shifter-III stands for three bits, Shifter-II stands for two bits, and Shifter-I stands for a single bit. This shifting data is saved in the ROM. The input X is stored in RAM, and every iteration processes a sample. Rather than utilizing a standard adder following every processing step, samples are added with the help of a power and area-efficient CSLA adder. The adder unit's returns are both signed and unsigned; to obtain the natural values of the signed numbers, take a complement of the outputs. The suggested FIR filter approach does not save the coefficient in the ROM; instead, it simply moves information recorded in the ROM. As a result, the recommended process uses fewer ROMs than the traditional approach. The structure of BEC is depicted in **Fig.7**. The modified CSLA is illustrated in **Fig.8**.

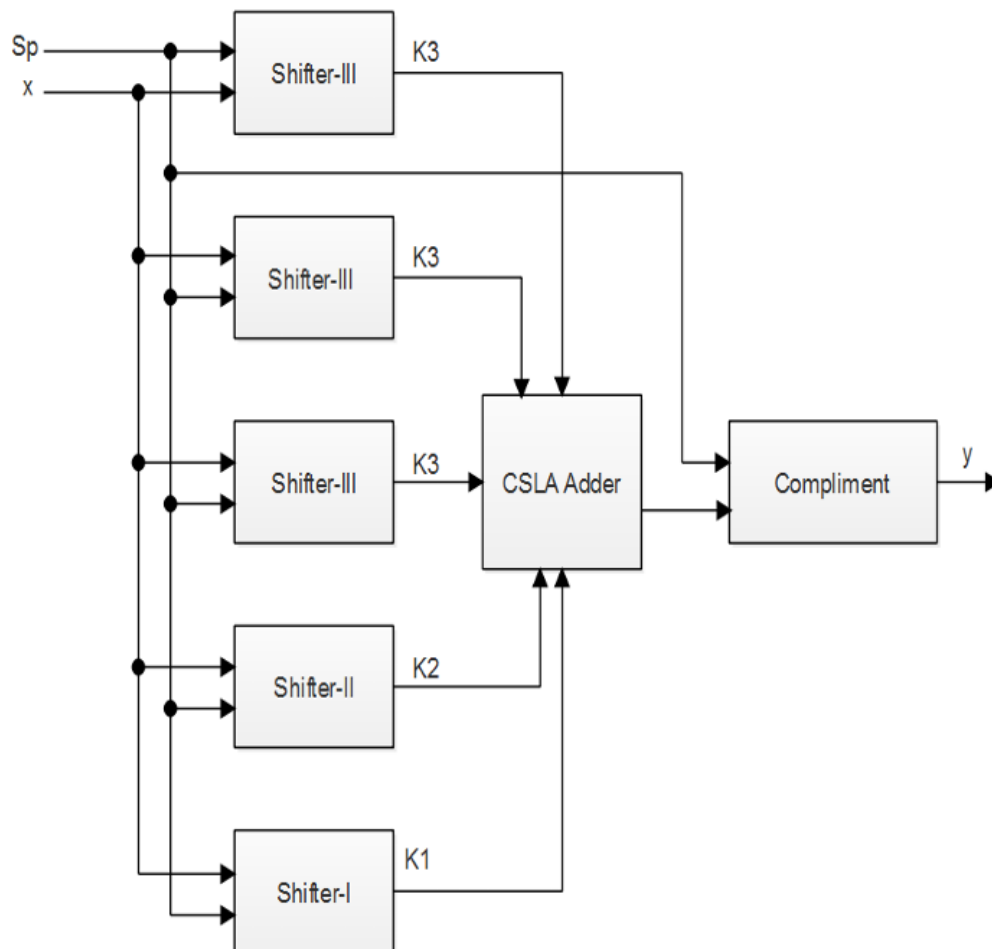


Fig.6 The Processing Element (PE) of the proposed method.

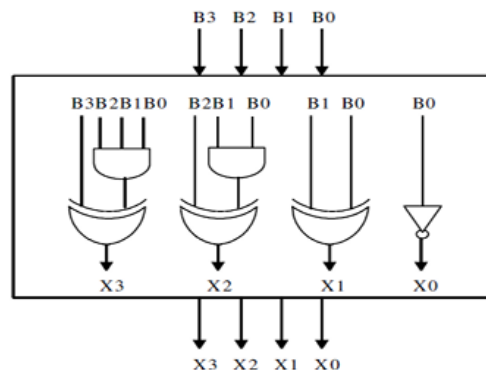


Fig.7The structure of BEC

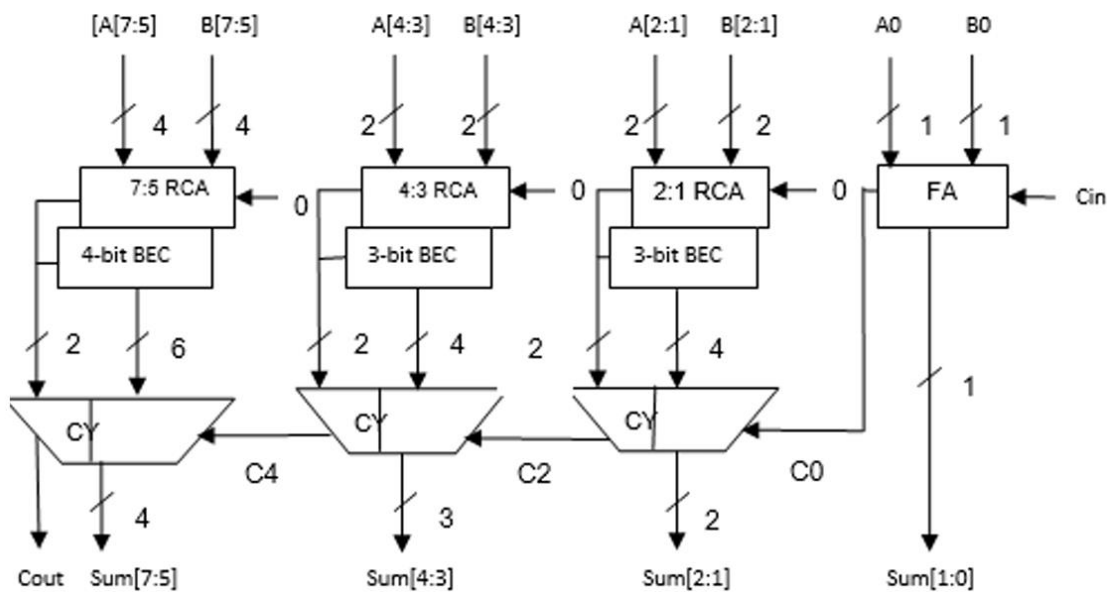


Fig.8.The modified CSLA

CSLA is typically formed by a single multiplexer and two RCA, with RCA formed by cascading FA blocks in series. In RCA, the carry output of the previous stage is sent straight to the carry input of the subsequent step. Even though RCA is the most straightforward and commonly used to add integers of any length, it could be more efficient when many bits are employed. One of the main drawbacks of this adder is that as the bit length grows, so does the delay. The delay increases as a carry signal conversion ripples through every phase of the adder chain from the significant bit to the MSB. CSLA is built on the notion of computing sums by assuming input carry from the prior phase. One adder computes the sum assuming a 0 input carry, while the other computes the sum assuming a 1 input carry. The actual carry then activates a multiplexer, which determines the appropriate sum. The main disadvantage of the normal CSLA is that it requires a large area due to the several pairs of ripple carry adders. Another disadvantage is that CSLA operates slowly since RCA creates additional delay. Because of the dual RCA in the CSA, additional space is required, and the carry out at every step must ripple. One RCA is substituted

by BEC ($C_{in}=1$) to reduce the area and eliminate the delay that results from one of the RCA ($C_{in}=1$). To substitute for the N-bit RCA, a N+1 bit BEC is required, which means that the number of bits required for BEC logic is one bit greater than the number of bits required for RCA. A BEC circuit is used to add 1 to the input bits. One input of the MUX in this circuit is B3, B2, B1, and B0, while an additional input of the MUX is the BEC output. The LSB are added via RCA, and the subsequent units are added in parallel with the supplied incremented. After the interim carries and sums are computed, multiplexers with reduced delay are used for calculating the final sum. The multiplexer unit receives the two sets of input and selects the final sum based on the choose input from the previous phase. Thus, combining BEC with MUX results in faster increasing action with fewer gates. This provides a significant gain in terms of area reduction and total power utilisation. As a result, the final result shows that Carry Select Adder with BEC outperforms CLSA with RCA in terms of power and area.

4. Results and Discussions

This ASIC synthesizing is carried out in the Cadence tool for several technologies, such as 180nm. This tool calculates metrics for performance, such as delay, power, and area. Portable device development has resulted in smaller battery sizes and, as a result, less power using systems. Energy efficiency has become an essential criterion for many designers. By lowering the system size, ASIC could accommodate maximal functionality in the smallest space. The designer will give an area constraint, and the Cadence tool will optimize the area's performance. The area is optimized by using fewer cells and replacing several cells with a single cell that performs both functions. **Table 1** shows the comparison of proposed method with existing methods. **Table 2** compares the LUT, Flip-flop, Slice, RAM, ROM, and Frequency of proposed method with existing methods using Virtex4 xc4vfx12.

Table 1 The comparison of proposed method with existing methods.

Approaches	Delay $\mu(ps)$	Power (nW)	Area(μm^2)
Proposed method	7691.1	3736800	84663
[14]	9842.6	4257915.2	98745
[6]	11444	4674727.1	105490
[13]	13547	5124798.3	124587
Length of the filter is 32-tap			

Table 2 Comparison of LUT, Flip-flop, Slice, RAM, ROM, and Frequency of proposed method with existing methods using Virtex4 xc4vfx12.

Approaches	Frequency	ROM	RAM	Slice	Flip flop	LUT
Proposed method	64.875	1	1	230/5472	40/10944	445/10944
[14]	63.54	2	1	248/5472	41/10944	524/10944
[6]	78.539	2	1	288/5472	42/10944	553/10944
[13]	42.21	2	1	294/5472	46/10944	564/10944
Length of the filter is 32-tap						

5. Conclusion

The decrease of power, area, and delay characteristics in VLSI circuit layouts continues to rise as the level of sophistication of applications increases. Presently, more real-time applications require high throughput with less power than ever. In this case, the FIR filter is utilized to create an effective DSP system. Multipliers and Adders are essential parts of FIR filters for reducing latency and area. As a result, the suggested approach faced additional constraints: high speed, high throughput, and consuming as little power as feasible. The FIR Filter is frequently employed in DSP Applications like image processing, arithmetic computations, noise cancellation, echo cancellation, loudspeaker equalization, and voice processing. This study describes the development and execution of a reconfigurable FIR filter that is both area and energy-efficient. This paper introduces new FIR filter topologies for wireless systems that may substantially decrease hardware costs and energy consumption. Multipliers account for the majority of hardware consumption in constructing FIR filters. Yet, multipliers are no longer needed in the suggested FIR structures; all we require are shifters and adders, allowing us to save many multipliers and adders. In general, we have offered new FIR structures in this study that are superior to standard FIR architectures regarding hardware expenses and power consumption, making it more appropriate for developing ASICs for sensor nodes. Model sim software utilizing Verilog code has executed the suggested FIR filter design. The FIR filter was built using this approach, utilizing a control unit, RAM, coefficient RAM, and an accumulator circuit. The FIR design was constructed without a multiplier, resulting in less time and area. For 32-tap, power, area, and latency have all been considered. The suggested approach improves Flip-flops, slices, and LUT in FPGA implementations. The primary concept behind the proposed filter algorithm is to substitute adders and multipliers with adders and shifters to minimize hardware costs. Only shifters and adders are required throughout the FIR filter implementation, with no multipliers.

Funding

This study got no specific financing from any government, commercial, or non-profit organization.

Acknowledgement

The information, perceptions, and encouragement provided by the Faculty of Engineering at Bir Tikendrajit University in India were critical in supporting us in completing this project. The knowledge and ideas of India's Khader Memorial College of Engineering and Technology substantially affected the direction and focus of our research. We also thank the Department of Electrical and Computer Engineering at Effat University in Saudi Arabia and the Department of Electrical and Computer Engineering at Oakland University in the United States of America for the help we needed to complete this project.

References

1. Itawadiya, A.K., Mahle, R., Patel, V. and Kumar, D. (2013) Design a DSP Operation Using Vedic Mathematics. 2013 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, 3-5 April 2013, 897-902.
2. Proakis, J.G. and Manolakis, D.K. (1996) Digital Signal Processing. Prentice Hall Inc., Upper Saddle River, New Jersey, 82.

3. Abhilash, R., Dubey, S. and Chinnaiah, M.C. (2015) High Performance and Area Efficient Signed Baugh-Wooley Multiplier with Wallace Tree Using Compressors. International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, 24-25 January 2015, 1-4.
4. Park, J., Jeong, W., Choo, H., Mahmoodi-Meimand, H., Wang, Y. and Roy, K. (2002) High Performance and Low Power FIR Filter Design Based on Sharing Multiplication. Proceedings of the 2002 International Symposium on Low Power Electronics and Design, 14 August 2002, 295-300.
5. Park, J., Muhammad, K. and Roy, K. (2003) High-Performance FIR Filter Design Based on Sharing Multiplier. IEEE Transactions on Very Large Scale Integration VLSI System, 11, 244-253.
6. L. Wanhammar, DSP Integrated Circuits. New York: Academic, 1999. [2] C. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., vol. 13, no. 1, pp. 14-17, Feb. 1964.
7. W. Gallagher and E. Swartzlander, "High radix booth multipliers using reduced area adder trees," in Proc. Asilomar Conf. Signals, Syst. Comput., vol. 1. Pacific Grove, CA, Oct.-Nov. 1994, pp. 545-549.
8. J. McClellan, T. Parks, and L. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," IEEE Trans. Audio Electroacoust., vol. 21, no. 6, pp. 506-526, Dec. 1973.
9. H. Nguyen and A. Chatterjee, "Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 8, no. 4, pp. 419-424, Aug. 2000
10. G. Wacey and D. R. Bull, "POFGEN: A Design Automation System for VLSI Digital Filters with Invariant Transfer Function," IEEE International Symposium on Circuits and Systems, ISCAS, vol. 1, 1993, pp. 631 - 634.
11. Mohammed Abo-Zahhad and Sabah Mohamed Ahmed, "Filter Designer: A Complete Design and Synthesis Program for Lumped, Wave-Digital, FIR and IIR Filters," Proceedings of the Thirteenth National Radio Science Conference, March 19-21, 1996, Cairo, Egypt, pp. C24.1 - C24.15.
12. Varun Verma and Charles Chien, "A VHDL based Functional Compiler for Optimum Architecture Generation of FIR Filters," IEEE International Symposium on Circuits and Systems, ISCAS 1996, vol. 4, pp. 564 - 567.
13. Wolfgang Wilhelm and Tobias G. Noll, "A New Mapping Technique for Automated Design of Highly Efficient Multiplexed FIR Digital filters," Proceedings of 1997 IEEE International Symposium on Circuits and Systems, ISCAS 1997, vol. 4, pp. 2252 - 2255.
14. Rajeev Jain, Paul T. Yang, and Toshiaki Yoshino, "FIRGEN: A Computer-Aided Design System for High Performance FIR Filter Integrated Circuits," IEEE Transactions on Signal Processing, vol. 39, no. 7, July
15. Laskowski, J. and Samueli, H., "A 150-MHz 43-Tap Half-Band FIR Digital Filter in 1.2-um CMOS Generated by Silicon Compiler," Proceedings of the Custom Integrated Circuits Conference, 1992, pp. 11.4.1-11.4.4.
16. Dempster, A.G.; Macleod, M.D. Use of minimum-adder multiplier blocks in FIR digital filters. IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process. 1995, 42, 569-577.

17. Mahesh, R.; Vinod, A.P. A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 2008, 27, 217–229.
18. Lou, X.; Yu, Y.J.; Meher, P.K. Fine-grained critical path analysis and optimization for area-time efficient realization of multiple constant multiplications. *IEEE Trans. Circuits Syst. I Regul. Pap.* 2015, 62, 863–872.
19. Meidani, M.; Mashoufi, B. Introducing new algorithms for realizing an FIR filter with less hardware in order to eliminate power line interference from the ECG signal. *IET J. Signal Process.* 2016, 10, 709–716.
20. Ye, J.; Togawa, N.; Yanagisawa, M.; Shi, Y. A low cost and high speed CSD-based symmetric transpose block FIR implementation. In *Proceedings of the IEEE International Conference on ASIC (ASICON)*, Guiyang, China, 25–28 October 2017.
21. Ye, J.; Togawa, N.; Yanagisawa, M.; Shi, Y. Static error analysis and optimization of faithfully truncated adders for area-power efficient FIR designs. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Sapporo, Japan, 26–29 May 2019.
22. Park, J.; Jeong, W.; Meimand, H.M.; Wang, Y.; Choo, H.; Roy, K. Computation sharing programmable FIR filter for low-power and high-performance applications. *IEEE J. Solid-State Circuits* 2004, 39, 348–357.
23. Parker, D.A.; Parhi, K.K. Low-area/power parallel FIR digital filter implementations. *J. VLSI Signal Process. Syst.* 1997, 17, 75–92.
24. Tsao, Y.; Choi, K. Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm. *IEEE Trans. Very Large Scale Integr. Syst.* 2012, 20, 366–371.
25. Tsao, Y.; Choi, K. Area-efficient VLSI implementation for parallel linear-phase FIR digital filters of odd length based on fast FIR algorithm. *IEEE Trans. Circuits Syst. II Express Briefs.* 2012, 59, 371–375.
26. Mohanty, B.K.; Meher, P.K. A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm. *IEEE Trans. Signal Process.* 2013, 61, 921–932.
27. Mohanty, B.K.; Meher, P.K.; Al-Maadeed, S.; Amira, A. Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters. *IEEE Trans. Circuits Syst. I Regul. Pap.* 2014, 61, 120–133.
28. Mohanty, B.K.; Meher, P.K. A high performance FIR filter architecture for fixed and reconfigurable applications. *IEEE Trans. Very Large Scale Integr. Syst.* 2016, 24, 444–452.
29. S. Nagaria, A. Singh and V. Niranjana, "Efficient FIR Filter Design using Booth Multiplier for VLSI Applications," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2018, pp. 581-584.
30. Anurag Aggarwal, AsthaSatija, TusharNagpal, "FIR filter designing using Xilinx system generator," *International Journal of Computer Application*, vol. 68, no.11, Aprl. 2013.
31. Grande Naga Jyothi, Kishore Sanapala, and A. Vijayalakshmi, "ASIC Implementation of Distributed Arithmetic Based FIR Filter Using RNS for High-Speed DSP Systems," *International Journal of Speech Technology*, vol. 23, no. 2, pp. 259-264, 2020.
32. Shaheen Khan, and ZainulAbdinJaffery, "Modified High-Speed FIR Filter Using DA-RNS Architecture," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 554-570, 2020.

33. G. Reddy Hemantha, S. Varadarajan, and M.N. Giri Prasad, "FPGA Implementation of Speculative Prefix Accumulation-Driven RNS for High-Performance FIR Filter," *Innovations in Electronics and Communication Engineering*, pp. 365-375, 2019.
34. Burhan Khurshid, and RoohieNaaz Mir, "An Efficient FIR Filter Structure Based on Technology-Optimized Multiply-Adder Unit Targeting LUT-Based FPGAs," *Circuits System and Signal Processing*, vol. 36, pp. 600-639, 2017.
35. E. Chitra, T. Vigneswaran, and S. Malarvizhi, "Analysis and Implementation of High Performance Reconfigurable Finite Impulse Response Filter Using Distributed Arithmetic," *Wireless Personal Communications*, vol. 102, no. 4, pp. 3413-3425, 2018.
36. LavanyaMaddiseti, Ranjan K. Senapati, and J.V.R. Ravindra, Image Multiplication with a Power-Efficient Approximate Multiplier Using A 4:2 Compressor, *Advances in Image and Data Processing Using VLSI Design, Smart Vision Systems*, 13th ed., IOP Publishing Ltd, pp. 13-15, 2021
37. Piotr Patronik, and Stanisław J. Piestrak, "Hardware/Software Approach to Designing Low-Power RNS-Enhanced Arithmetic Units," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 5, pp. 1031-1039, 2017
38. Yufeng Xu, Yi Guo, and Shinji Kimura, "Approximate Multiplier Using Reordered 4–2 Compressor with OR-Based Error Compensation," 2019 IEEE 13th International Conference on ASIC (ASICON), Chongqing, China, pp. 1-4, 2019.
39. Raj Kamal et al., "Efficient VLSI Architecture for FIR Filter Using DA-RNS," 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE), Hosur, India, pp. 184- 187, 2014.
40. E. Jagadeeswara Rao, and P. Samundiswary, "Error-Efficient Approximate Multiplier Design Using Rounding Based Approach for Image Smoothing Application," *Journal of Electronic Testing*, vol. 37, pp. 623-631, 2021
41. Bharat Garg, and Sujit Patel, "Reconfigurable Rounding Based Approximate Multiplier for Energy Efficient Multimedia Applications," *Wireless Personal Communications*, vol. 118, pp. 919-931, 2021.
42. Seyed Amir Hossein Ejtahed, and SomayehTimarch, "Efficient Approximate Multiplier Based on a New 1-Gate Approximate Compressor," *Circuits Systems and Signal Processing*, vol. 41, pp. 2699-2718, 2022.
43. Mostafa Abbasmollaei et al., "A Power Constrained Approximate Multiplier with a High Level of Configurability," *Microprocessors and Microsystems*, vol. 90, 2022.
44. ShaghayeghVahdat et al., "LETAM: A Low Energy Truncation-Based Approximate Multiplier," *Computers & Electrical Engineering*, vol. 63, pp. 1-17, 2017.
45. T.K. Shahana et al., "Performance Analysis of FIR Digital Filter Design: RNS Versus Traditional," 2007 International Symposium on Communications and Information Technologies, Sydney, NSW, Australia, pp. 1-5, 2007.