Design and Performance Analysis of RNS-Based Reconfigurable FIR Filter for Noise Removal in Speech Signals Applications

MANJUNATH P. S.[1*], REVANNA C. R.[2], KUSUMA M. S.[3], PONDURI SIVAPRASAD[4], UPPALA RAMAKRISHNA[4]
[1]Department of Electronics and Tele Communication Engineering, BMSCE, Bangalore, INDIA
[2]Department of ECE, Government SKSJTI, K R Circle, Bangalore, Karnataka, INDIA
[3]Department of ECE, Govt. S.K.S.J. Technology Institute, Bangalore, INDIA
[4]RVR&JC College of Engineering, Andhra Pradesh, INDIA

[*]*Corresponding Author*

Abstract: - In DSP solutions, the Residual Number System with Two's Complement systems is the most commonly utilized system for building low-power and high-throughput programmable Finite Impulse Response filters. It would be done by creating FIR filters in the Residual Number organization and 2's Enhance scheme by comparing the results to the current assert. The RNS based on FIR filter architecture reduces power consumption while allowing the device to operate at 150 MHz without increasing its size significantly. In case of memory and latency reduction, the implementations of the Residual Number System and 2's Complement System must be able to obtain and decode signals with fewer physical servers for every clock signal. The principal idea of this proposed model is to provide data bits with larger sizes for RNS-based multiplier and delayed wavelet LMS (DWLMS) that operates at speed high with premised reconfigurable FIR via forward and reverse conversions that don't produce as much power output and size as reflective thinking. The Application Specific Integrated Circuit will be designed and integrated for 32 nm technology. The proposed design addresses the four essential parameter optimization, such as power, area, and timing, using the Residual Number System, which is superior to Two's Complement System. According to the findings, there is a 13 percent reduction in power, a 21 % enhancement in area, and a 13 % enhance in throughput.

Key-Words: - FIR, RNS, DWLMS, Signal processing, FPGA, and Noise removal

Received: November 15, 2022. Revised: April 22, 2023. Accepted: May 16, 2023. Published: June 16, 2023.

1 Introduction

The output voltage is one of the most crucial limiting elements in designing future Application Specific Integrated Circuits. Low power consumption improves the ASIC's flexibility by decreasing the device's price, complexity, and bulk. DSP blocks are a significant source of output power in today's ASICs. In digital signal processing applications, the excess information has long been recommended as an energy alternative to the traditional 2's Enhance scheme line, [1]. Using FIR separators in the RNS, the 2's Enhance scheme has been shown to minimize power usage in some tests. FIR filters are one of the most basic DSP components. A basic overview of how Residual Number System simulations can be carried out, [2]. The Chinese scientist Sun Tzu, who existed in the third century AD, used the residue number system for the first time in his Arithmetic Classic of Sun Tzu. A finite impulse response filter is a form of a digital filter capable of simulating almost any frequency response, [3]. The study, [4], presents a couple of additional sequential algorithms, whereas these algorithms do not produce a single result every clock cycle, thus, they won't be investigated further. The output of a finite impulse response filter is often created using a succession of latencies, multipliers, and adders. FIR filters are one of the most significant structure fragments of many digital signal transmission algorithms, [5]. Demand for reconfigurable data transmission that may function in various standards has recently increased due to software-defined radio applications, [6]. Digital filters are typically implemented using a DSP; however, Digital Signal Processing-based solutions cannot meet the high-speed demands in some scenarios, [7]. Since of their serial construction and programmable logic, Field Programmable Gate Array-based systems can attain high speed, giving them additional flexibility and dependability throughout establishment and growth. The digital filters that are available in [8], are FIR as well as IIR filters. Many digital signal processing applications use FIR filters, as these filters can offer regular periods and design implementation, [9].

2 Problem Formulation

The Finite Impulse Response architecture comprises a series of multiplication addition and units that utilize pricey N Field Programmable Gate Arrays multiply-and-accumulate blocks. Distributed Arithmetic, when compared to traditional direct arithmetic, can save you a lot of money on hardware by replacing MAC units with a Look-Up Table, [10]. Compared to standard RNS (where the synthesis tool decides the adders and multipliers to use), specific TCS adders and multipliers demand more excellent hardware resources in terms of LUTs, FFs, and memory, [11]. The research focuses on RNS-specific approaches rather than low-power TCS or FIR filter techniques, [12]. The RNS adders and multipliers concentrate on the design rather than the actual design of the standard binary adders and binary multipliers utilized in the design. RNS-based R-FIR is suggested to reduce the number of computing processes, and the constructed R-FIR filter is tested using noisy EEG data for noise reduction, [13].

3 Proposed Methodology

The transmission of speech signals with a lot of noise required more bandwidth and more power consumption in 5G communications. Therefore, in this work, we are concentrating on the following:

i. Optimization of power consumption
ii. Minimization of noise to save bandwidth
iii. Improvement of throughput and speed

Based on the literature survey, power, and area consumption mainly depend on multipliers and adders due to more partial product generation. We have proposed an RNS-based multiplier and parallel prefix adder (PPA) to minimize power and area. For the noise removal in signals, the proposed 64-tap FIR filter has been used by incorporating an RNS multiplier and PPA adder, and the results are validated on real-time FPGA. An equalizer is a real-time computer that tries to explain the relationship between two signals. The proposed work is a reconfigurable FIR filter with NLMS; here, reconfigurable means parameterization of step size and input sample size; these can be 8bit, 16bit, and 32bits. The single tap FIR filter is shown in Fig.1. It contains (+) as an adder, (x) as a multiplier, step size, and delay elements. By changing parameter values, the entire design can reconfigure to any 8bit, 16bit, or 32bits. As a result, we'll concentrate on the scientific methods of adaptive filters rather than their specific implementations in hardware and software. The adaptive filter model uses a number and determines the type of characteristics to be changed.

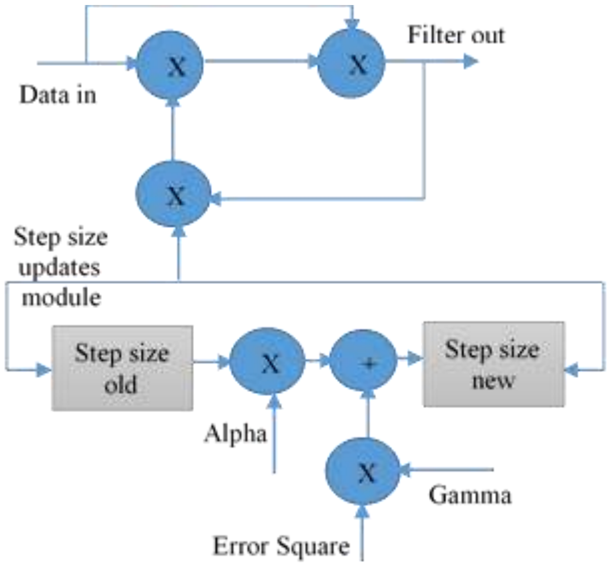

Fig. 1: Architectural diagram of 1-tap Coefficients used in FIR filter

The proposed technique for updating the system's parameter values can take a variety of shapes. Still, it's usually created as an optimization strategy that minimizes an illusionistic ally parameter. This section introduces the general adaptive filtering issue and the mathematical language for describing the adaptive filter's shape and operation. Then we'll go through a few distinct structures demonstrated to be beneficial in real-world situations. The LMS method uses a predefined step-size parameter for each iteration, which is one of its significant flaws. Before beginning the adaptive filtering procedure, you must first grasp the statistics of the input signal. In practice, this is quite unusual. Even if we assume that the adaptive echo cancellation system would receive voice as input, various parameters of sensory supply, for example, strength and amplitude, will influence its presentation. The normalized LMS algorithm is an extension of the least mean square method that avoids the issue by choosing a new step size value indicated by *(n)* for every progression, as shown in equation (1). The reciprocal of the entire estimated signals and their energy *(E)* estimated coefficients at every instantaneous value for any given input signal x can determine the step size *(n)*. An auto-correlation matrix (R) between the input vector dot product with itself, as well as input vectors, is analogous to the addition of energies that are expected for the input signal (x).

$$equal\ t_r[R] = \sum_{i=0}^{N-1} E[x^2(n-1)] \qquad (1)$$

3.1 Employment of the Normalized LMS Algorithm

The LMS algorithms use normalized concepts to implement and design using Verilog HDL and it's synthesized in the Xilinx ISE design suite their results and shown in Table 1. The step size has been derived from present inputs and previous output values and the LMS method is substantially more stable along with unknown signals. The proposed LMS is more suitable for real-time adaptive echo cancellations due to its convergence speed and its simplicity in design. Its process is completely iterative as per equation (2).

$$w(n+1) = w(n) + \mu(n)e(n)x(n) \qquad (2)$$

Variable step size used in NLMS: As per equation (2), in every iteration for each tap, the weight is processed with input signals and a predetermined step size value. The step size for each iteration is given as a vector in the Least Mean Square (NLMS) method for every sample *(x)*. Each vector member and size are corresponding to a particular step size value in the filter tap weight vector *w(n)*. Upper and lower values limit the allowed values for each element in the step size to avoid the step size parameters from becoming extremely large, leading to instability, or extremely small, resulting in delayed responsiveness to changes in the intended impulse response. As with the traditional LMS method, previous knowledge of signal statistics is required to ensure that the adaptive filter performs optimally.

4 Delayed Wavelet MPLMS Algorithm

Under correlated input circumstances, wavelet domain LMS is projected for scant adapting filters. However, unlike the techniques of PNLMS and MPNLMS, there has been no challenge in building WMPNLMS in hardware so far in the literature. To develop WMPNLMS in hardware, we initially encompass all reformulations to the wavelet domain to construct DWMPLMS and also suggest additional architectural improvements to lower the computational difficulty of wavelet implementation, as shown in Fig. 2. At every iterative of every tap it explores the de-correlating features of several wavelet transformations and their VLSI implementation aspects.

4.1 Sliding Wavelet Transform

At each new iteration, the statistics path *u(n)* is well-defined as $u(n) = [u(n), u(n-1), \dots, u((N+1))]$ and T is made efficient by allowing a single information model to first stay another to leave $uT(n)$ i.e., $uT(n) = Tn(n)$ signifies the transformed input vector, where T signifies the extraneous compress with sub-bands. The flowing flora of the input may be utilized to take advantage of severances in the computation of *u(n)* and *u(n + 2)* running wavelet transformations here *u(n + 2) = [u(n + 2), u(n + 1),..., u(n N + 3) T*. Let T_8 be an 8-coefficients of Symlet of vanishing moment 4 with four coefficients and four extremely high-frequencies such as g0, g1, g2, and g3 coefficients. This makes wavelet transform calculation more complex, making it incompatible with the DWMPLMS technique, which requires at least three stages of decomposition to achieve acceptable de-correlating qualities, [14]. The redundancies persist across numerous stages of decay in the U-HAAR ripple, and their constants are one's complement and may be exploited using a steady assembly. For instance, consider an 8-point U-HAAR wavelet matrix with two fading stages.

$$8 - point\ wavelet\ matrix$$

$$= \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ g_2 & g_3 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{pmatrix} \begin{pmatrix} u(n+2) \\ u(n+1) \\ u(n) \\ u(n-1) \\ u(n-2) \\ u(n-3) \\ u(n-4) \\ u(n-5) \end{pmatrix}$$

The elevated signal is routed and concludes a series of records timed at clk/2. As indicated, this tapped delay line yields L/2 elevated ripple apparatuses, [15]. The moderate components proceed over the second stage of breakdown, which follows the same structure as the first. Valid outputs are created once per four clocks for the second level's high-frequency components. As a result, in between legitimate outcomes, we'll need two registers that run on clk/2. Also observed is that intermediate results are lost when the design is run at clk/4, therefore running at clk/2 is the best suitable frequency and losses are very less. To capture all of the intermediate outcomes at the 3rd equal, we have essential 4 registers operating on clk/2 between legitimate outputs. For the difference in the 1st near, we'll require (L2)/2 registers. Similarly, we'll require (L 4)/2 registers in the second stage, then (L 8)/2 high-frequency registers apparatus, and one more (L 8)/2 register for low-frequency works in the final stage.

$8-point\ wavelet\ matrix$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & \end{pmatrix} \begin{pmatrix} u(n+2) \\ u(n+1) \\ u(n) \\ u(n-1) \\ u(n-2) \\ u(n-3) \\ u(n-4) \\ u(n-5) \end{pmatrix}$$

4.2 Design of RNS-Based Multiplier for LMS and FIR Filters Design

The RNS, adders, and multipliers are used to develop low latency and high throughput systems. Certain adders and multipliers based on the Residual Number System are compared to modified Residual Number System-based multipliers and adders. This study looks at how to use adders and integrators of RNS and what all combinations consume low power. RNS's main processes are frontward and backward alterations, which multiply circuit source and filter constants.

4.3 RNS Arithmetic

RNS arithmetic is based on the algebraic symmetry connection. Assume, the integers, as well as b, are proved to be consistent with modulo m if a-b is accurately divisible by m. It is commonly written as a ≡ b (mod m) in technical situations. The quantity m is known as a modulus. The remainder of the modulus m is indicated by r, the division of the number, and q is the quotient a, a = q · m + r. From the provided definition, we can derive the accompanying congruence ≡ r (mod m). The number r denotes a's residue around m, which is represented as $=|a|_m$. We assume that one of the least non-negative residues modulo m, r ∈ {0, 1, 2, ..., m − 1}. Consider $\{m_1\ m_2,..., m_N\}$ as a series of N non-negative, pair-wise approximation moduli. It implies that for all I as well as j I ≠ j, the moduli mi, and m_j in the moduli set have no common divisor of more than 1. The input impedance of the RNS moduli set can now be specified as M. According to the equation, the product of the moduli set can be used to determine M (1).

$$M=\prod_{n=1}^{N} m_n \tag{3}$$

For each moduli set, an integer X < M has an autonomous module consisting of N members. The continuity equation can be used to calculate it: {xi = $|X|m_i$: 1 ≤ i ≤ N}. However, one representation is that of $(x_1, x_2, ..., x_N)$.

Example 1: Take the moduli-set {3, 5, 7}, then m1 = 3, m2 = 5 and m3 = 7. The moduli-dynamic format's spectrum would be

$$M=\prod_{n=1}^{3} m_n = m_1.m_2.m_3=3.5.7=105$$

Now let X = 10. Then (x_1,x_2 , x_3)can be calculated as follows

$x_1 = |X|m_1 = |10|_3 = |3\,.\,3 + 1|_3 = |3\,.\,3|_3 + |1|_3 = 1$

$x_2 = |X|m_2 = |10|_5 = |2.5|_5 = 0$

$x_3 = |X|m_3 = |10|_7 = |1.7 + 3|_7 = |1.7|_7 + |3|_7 = 3.$

So X = 10 can be represented as (1, 0, 3) in the RNS moduli-set {3, 5, 7}. Either signed or unsigned integers can be represented using a residue number system. The residual Number System can express unverified integers in the range 0 ≤ X ≤ M - 1 for unsigned numbers. RNS can collect data that satisfy one of the given equations for signed numbers:

$\frac{-M-1}{2} \le X \le \frac{M-1}{2}$ For odd

$\frac{-M}{2} \le X \le \frac{M}{2}$ -1 For even

Table 1. An example of RNS representation for signed and unsigned numbers

(X_1, X_2)	Unsigned	Signed
(0,0)	0	0
(1,1)	1	1
(0,2)	2	2
(1,0)	3	-3
(0,1)	4	-2
(1,2)	5	-1

Used this moduli-set as an instance of signed and unsigned expressions $\{m_1,m_2\}$ = {2, 3}

4.4 Onward Transfer

Advanced interpretation is usually more straightforward than backward interpretation. Even though residue number systems can represent a certain bit width, the input is typically displayed with a considerably smaller bit size. Naturally, by executing in this way, the complexity can be reduced. The following well-known equation is used to determine a TCS amount to solve the forward conversion issue:$a_{n-1}{-}2^{n-1} + \sum_{i=0}^{i=n-2} a_i\, 2^i$

To find the total of the numbers, the simplest method is to utilize RNS adders rather than TCS adders $a_i 2^i$. It can even accept negatives if the answer on line 64, [4], is significantly altered.

Modulus periodic properties can be leveraged to improve this strategy. The cyclic features are obtained by taking into account the residue of every 2^i mod m. However, a LUT is possible as it needs to store all probable combinations of input at n_input bits equivalent to RNS values of n_rns ≥ n_input input bits. As a consequence, future investigations into this approach can be ruled out. It provides an innovative periodic multiplications method. Unfortunately, it is quite complex, making parametrized implementation for arbitrary modulo and input bit width quite hard. Moduli conversion: The input operands X are related to the considered modules {m1, m2,…mp} and their corresponding residues {r1,r2,…rp} as shown in the equations below:

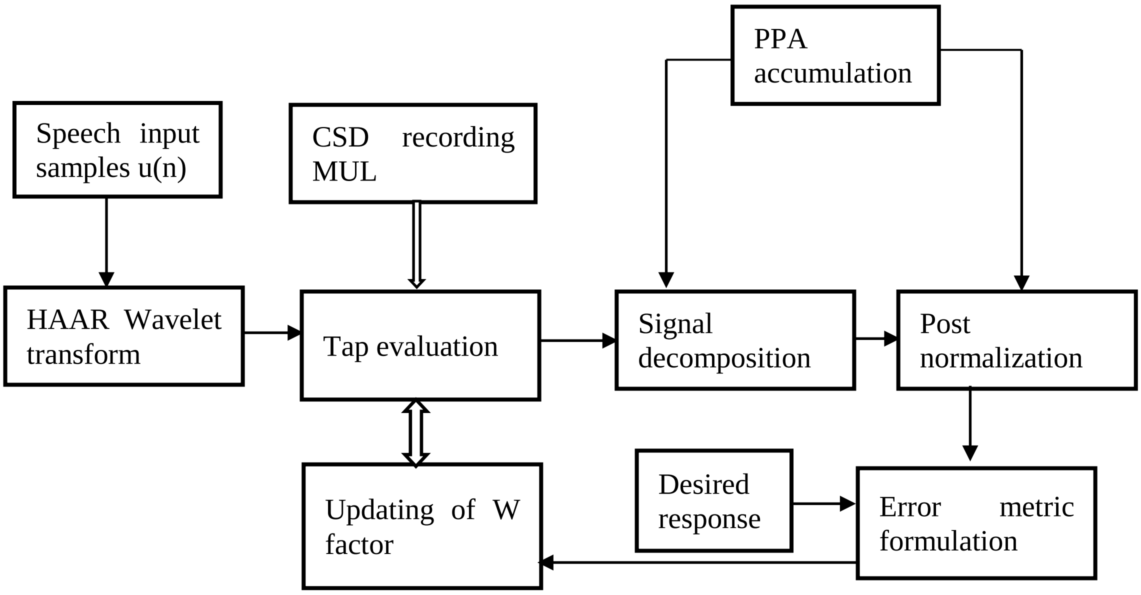

Fig. 2: Proposed wavelet transform domain used in NLMS architecture

$$X = \sum_{j=0}^{4p-1} X_j 2^j = M_3 2^{3p} + M_2 2^{2p} + M_1 2^p + M_0 \quad (3)$$

Where $M_3 = \{X_{2p+k-1} \ldots . X_{3p}\}$ $for\ 1 \leq k \leq p$ $and\ M_3 = 0\ for\ k = 0$The formula (3) is a shape numeral from double to Residual Number System forward conversion, X. It is an integer ranging from [0, N-1], as well as the residuals, are $2^{n-k}, 2^n, 2^{n+k}$ of RNS set $\{x_1, x_2, x_3\}$ for moduli set $\{m_1, m_2, m_3\}$ and here the value of N= m_1*m_2* m_3.

4.5 Reverse Conversion

Backward recognition is transferring the RNS to the 2's Signaling Pathway. The Chinese Remainder Theorem and the Mixed-Radix Conversion are the two standard techniques used for backward interpretation. The majority of additional procedures are derived from these two, [4]. The easiest of these alternatives is currently CRT. MRC employs "mixed-radix" approaches, which would demand significant additional work. Pseudo-SRT division or the core function are two choices (as described in [4]). A fascinating way is to use a Look-Up Table. Unfortunately, the generated Look-Up Table is too large for a synthesis tool to handle.

Choosing a moduli-set: Prior studies, [8], [5], [7], have shown that a considerable portion of energy dissipation happens during routine operations rather than in the forward or reverse transformation when the number of taps in a Finite Impulse Response filter is high.

Consequently, the initial hypothesis about which moduli to use is based on comparing the output power of a simple one-tap Finite Impulse Response filter element without interpretation. Before optimum arrangement was found, these essential components were built in several ways.

Carry parallel-prefix adder in the end: The Carry parallel-prefix adder in the end-around is designed to only function for modulo 2n1, with the advantage that it uses roughly the same hardware as a regular parallel-prefix adder by using the end-around carry. The parallel-prefix adder was created by converting the RNS adder in [16], from VHDL to System Verilog. With an end-around carry, it employs a Sklansky parallel-prefix structure. As seen in Fig. 4, the adder uses a variety of logic operators. The equation describes the exact behavior of the logic processors.

4.6 FIR Filter Design

The finite-duration Impulse Response, filter is by far the most popular digital filter. Formula (4) can be used to calculate a signal's filtered output, and an FIR filter is based on the general theory of continuous multiplication.

$$Out[n] = \sum_{j=0}^{N} h[j]in[n-1] \quad (4)$$

The output is represented by out[i], input is indicated by in[i] as well as the parameters in solution (2) are specified by h[i]. The order of the filter is determined by N, and the filter will have N + 1 tap. The simple mathematical processes of a Finite Impulse Response filter are additive and multiply. These operations can be carried out in a variety of ways using the residue number system. One of the most fundamental issues with the RNS is modulo excess, which occurs whenever the output is more than the modulo. The outcome of a modulo's operations mi must always range from *{0,..., m_i-1}*. So the outcome of addition ranges from *{0,...,2(m_i-1)}*., and only one subtraction with mi will be necessary to persist in the correct range. Reproduction, on the other hand, yields an outcome ranging from *{0,...,*$(m_i - 1)^2$}. To identify the techniques that are best suited for addition as well as multiplication, the overall output voltages and analyses of specific adders, as well as multipliers for all given moduli, are carried out. The study, [14], discusses three approaches to unrestricted modulo addition development. Look-Up Table, two binary adders, and a hybrid of the two. The three solutions will be superlative regarding the area and time for a particular modulus, [14]. The study, [15], shows how to use a parallel-prefix adder to execute addition in the unusual modulo set in a novel method *{ -1, + 1}*. , [16], provides a little more detailed description. Because of its basic level, this technique, [17], can be used as a starting deployment concept. The Verilog language and generation engine support the built-in Verilog operators "+," addition, "%," percent, and modulus. While evaluating the other methods, a naive baseline strategy that only uses these approaches will be helpful. Modulo will also be done with the easy way of using a common adder. The implementation of additions is carried out by

1. RNS adder that is based on LUT
2. Two binary adversary
3. A hybrid between zero as well as one
4. Modified parallel-prefix adder employing Modulo $2^n - 1$
5. Modulo $2^n + 1$ is determined by the application of diminished-one number representation, and by using built-in functions in Verilog “+” as well as “%”
6. Modulo adder of the usual type 2^n

Multiplication: RNS-designed multiplications can be stated in a selection of ways. A modulo-m product splitting multiplier using Read-Only Memory as a potential function is shown in [18]. This technique is more practicable than multiplication by equal modulus, as stated in [19], because it uses three multipliers rather than only two. The unique are agreed increases regarding time, travel, and power may be made. *2n -1,2n, 2n + 1*, [20], proposes a parallel modulo-m multiplier for 2n + 1 that does not necessitate special acceleration techniques. The above method may be interesting, especially if n is small. The study, [21], describes a Booth-8 encoding implementation for 2n + 1. This strategy outperforms previous solutions for n 32, while it can be adapted admirably for lower n. If this wasn't the case, a Booth-4 encoding was utilized. Booth encoding is well-known in situations other than RNS. So, it won't be talked about further in this paper. Another fascinating way can be found in [5], which uses an isomorphic approach to replace multiplication with addition and a look-up table. It'd be amazing to see how this might be put into practice.

4.7 Application of R-FIR Design for Noise Removal in Speech Signals

A steadily flowing spectral envelope characterizes the conversation. Individuals translate this spectral envelope into speech and its associated content.

Speech recognition attempts to replicate the effort of projecting the spectral envelope into a series of sentences. There are various faults with this procedure. Have different speaking patterns. The spectral envelope will fluctuate depending on local dialects and unique features, such as whether the individual is male or female and their height. A sentence can be delivered in various ways by a single speaker. It might be the result of speaker stress, such as while yelling, or it could be the result of the speaker purposely emphasizing words to change their meaning. Individuals have no difficulty coping with these factors, but developing an automated system to imitate this process is a significant task. Most current voice assistants use analytical methods to deal with the numerous fluctuations. These processing algorithms' dependability has continuously increased over time. They can achieve over 90% effective advertising on an infinite phrase speaker-independent challenge and above 95% on weaker keyword tests. The gadgets are functional at this level of performance, but most of them are taught and validated in the same calm environment. In practice, it is typically quiet, and the aural environment is frequently unregulated. Background noise, such as fans turning on or automobiles passing by, and received signals, such as the microphone or phone line stream in use, may differ. As the ambient noise and channel society change, the speech spectra alter, and the device's performance declines, which is frequently dramatic.

RNS-based R-FIR STRUCTURE**:** A finite impulse response filter is a digital filter capable of simulating almost every resonant event. A sequence of delays, multipliers, and adders is commonly used to generate the output of an FIR filter. Fig. 3 depicts the basic block diagram for an N-length FIR filter. Previous input samples are used because of delays. The HK values represent the multiplying parameters, and the result obtained at time n is the total postponed data multiplied by the suitable parameters.

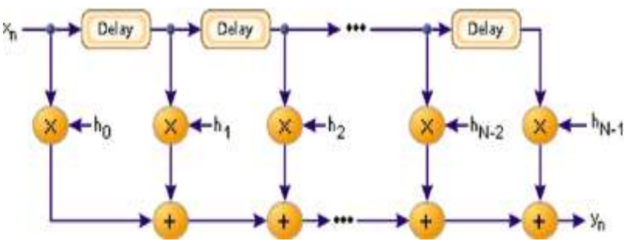

Fig. 3: A Finite Impulse Response filter's major infrastructure

The method of determining a filter's length and properties is known as filter planning. The goal is to set such values so that when the filter is applied, it returns desired stop and passband values. In C, listing one shows how to accomplish it. This code must be executed on a system with a multiply-and-accumulate command to perform many taps. When the Distributed method is used to generate the linear time-invariant structure, the approach outlined in previous works is applied to optimize it. As a result, the fixing amount for the above part of the LUT storage reminiscence location will be the inverse of the bottom half, resulting in a Look-Up Table cut in half using concurrency. The report creator circuit generates the Look Up Table address. The only part of the statement pulls up the necessary restoration value. Using the Ctrl control-adding-decrease tool, the destination completes the non-negative and non-positive conversion between the upper and lower halves of the restored quantity. The Look-Up Table is split into two four-input tables resulting from innovations and discoveries. The network for generating addresses divides the input signals into four categories using the 4-input LUT. The data buffer may be built up based on the order of the filters. The serial data collected can be transmitted to the 20-bit serial-in parallel-out shift register, separated, and given to the LUT in order because the needed filter is a 16th-order. Because the coefficient is amplified 216 times, the output circuit decreases the resultant value. The function of the basic FIR filter is expressed as follows:

The filter's length is represented as k, and in this research work, it varies between 0 and 63. Computation (6) specifies a straightforward FIR filter approach with significantly few registers, as shown in Fig. 5. The Fig. 4 depicts a portion of RFIR based on RNS.

$$out(x)\sum_{j=0}^{N-1} in(j)h^{-j} = \sum_{j=0}^{N-1} in(n)h(n-k) \qquad (6)$$

The standard FIR filter design uses a binary number strategy to model adders and coefficients. It thereby results in higher diffusion and net weights and limits the speed. To overcome the abovementioned drawback, the RNS-based FIR filter design in Eq. (6) uses a significantly improved PPA to eliminate the subcarriers. The outcomes of the existing Parallel Prefix Adder and the modified PPA are compared in Table 2.

Table 2. Performance trade-off comparison over input word length

word length	Moduli set($^{2n+1}$, 2^n,2^{n-1})	PPA		RNS	
		Area (LEs)	Fmax MHz	Area(LEs)	Fmax MHz
8-bit	(7,8,9)	1187	77.3	273	75.48
16-bit	(31,32,33)	1672	77.46	1189	83.56

Optimization of Memory: Backward translation is used to transform the residue number to an integer only after the completion of conventional arithmetic of basis estimation. Depending on the size of every module, the statistics with a minimal standard error are estimated using this technique. All possible alternatives are computed to ensure and accommodate it as a readily accessible block in storage for the inverse mutation operator. The hardware intricacy of the bidirectional switching unit will be minimized as these memory units will be transformed into customized onboard block RAMs during the manufacturing of hardware. By utilizing the shortest lead time, as demonstrated in Fig. 4, the SoC, which is cache, minimizes the cost and time compared to other conventional methods.

5 Results and Discussion

Empirical data from multiple moduli corroborate the output metrics of the theoretical Residual Number System Finite Impulse Response architecture. For propagation, position & route, the computational plan is applied in Verilog hardware description language, and its compliance is verified using the Model Sim. The propagation review and system applications are provided in Table 3 and Table 4, and the measurements are investigated with Artix-7 Prototype FPGA hardware. As per the results of the tests, Distributed Arithmetic design hacks reduce route latency while limiting economic efficiency. They are examining the computational constraints as an academic unit is employed during hardware transmission to indicate the significance of the theoretical DA-based RNS framework in various features of the dynamic FIR filter.

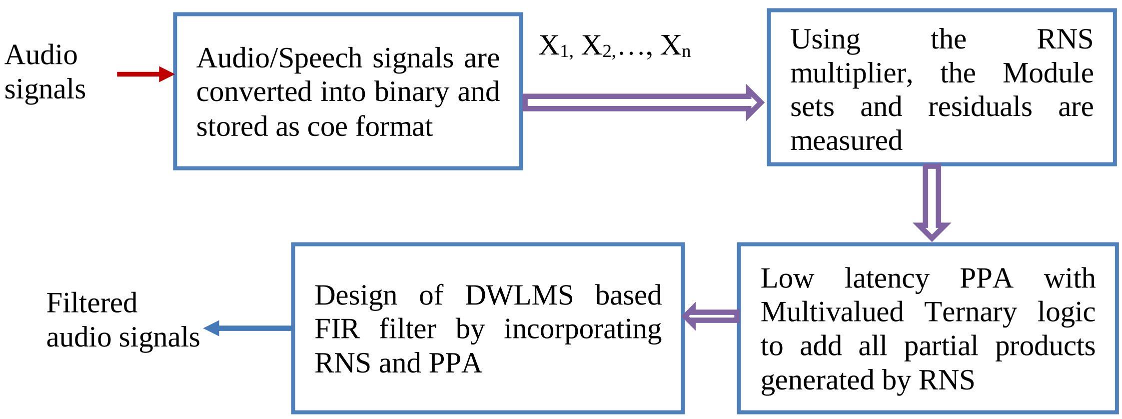

Fig. 4: Overall block diagram of FIR filter using Residual Number System filter with Parallel Prefix Adder for ECG signal applications

The RNS-based FIR channel is suggested. The selection of the moduli set has the benefit of an add & shift strategy. A previously proposed reconfigurable RNS FIR channel adaptation is contrasted with the suggested flow pattern. The circuits are combined using a Vivado design suite in the RTL compiler at the architecture level. The channels' displays are evaluated regarding the region, energy, and Suspension. FPGA DSP Builder is also used to test the proposed technique in practice. As demonstrated in Fig. 4, Finite Impulse Response stations are often used to implement modern electronic indicator preparation bases.

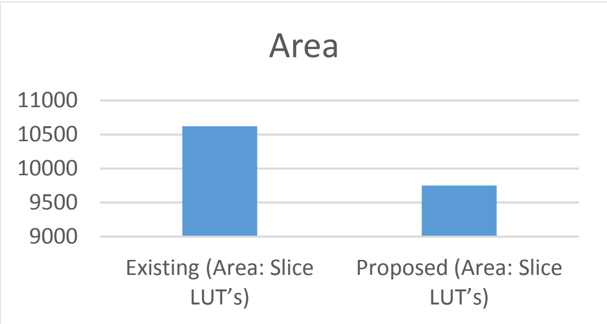

Fig. 4: The issue of continuous research and development is how to implement them efficiently using cost-effective Very-large-scale integration equipment.

The efficient execution of the multi-operand modulo adders was given special attention. For a 32-tap FIR channel, replacing a standard modulo snake tree with a paired viper with increased accuracy occurred by a single modulo diminution phase 10% reduction in space needs. On the other hand, the FIR channel list math QRNS, on themed a 3-repeater per-tap two-C channel by up to 60% while utilizing fewer LEs for channels with a total of eight taps. A twenty-two-tap channel, in particular, necessitated twenty-four percent LEs, which was not the typical plan as illustrated in Fig. 6.

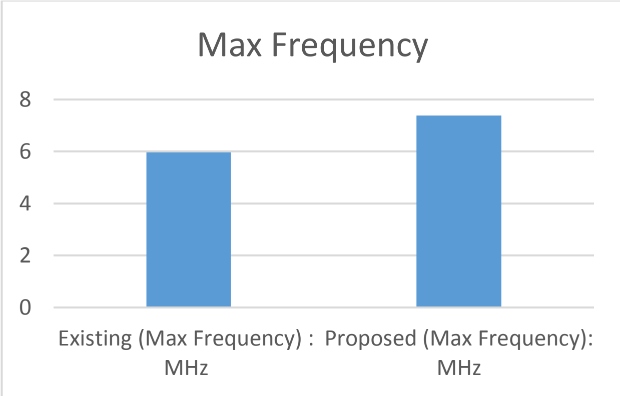

Fig. 5: Improved Area and frequency concert exchange evaluation of Distributed Arithmetic-based arithmetic in Residue Number System Over Finite Impulse Response length.

The complete design is synthesized using Xilinx Vivado Design Suite 2018.1 and a summary of the report which is in terms of LUTs, flip-flops, delay, power, and throughput are shown in Table 3. The proposed design is well optimized compared to existing work in terms of the above-mentioned parameters and their values are plotted in Fig.6, area and power are more optimized parameters compared to others. The obtained results are validated in MATLAB with the help of MATLAB-generated FIR coefficients through the Filter Design Analysis tool.

Table 3. Power, energy, place value, lag, as well as power estimates of a hypothetical RNS-FIR design (A=Area, T=Delay, and P=Power).

RNS based Multiplier	Logic Elements	LUT	Latency [ns]	Power [mW]	Area* delay	Time* power	Product of A*T*P
Radix-8 based 32 bits, [1], [17], multiplier	1671	---	19.56	78.3	6581×10^{-6}	126.31×10^{-9}	19.2×10^{-9}
Booth multiplier, [15], 16 bits	1200	---	27.2	0.85	19.06×10^{-6}	51.1×10^{-9}	09.2×10^{-6}
Multiplier-based Shift & Add 32 bits, [19]	2107		20.51	0.1	21.07	25.1	52.8
Proposed RNS-based FIR design	1240	7241	14.8	0.088	8.30×10^{-6}	21.4×10^{-9}	8.1×10^{-6}

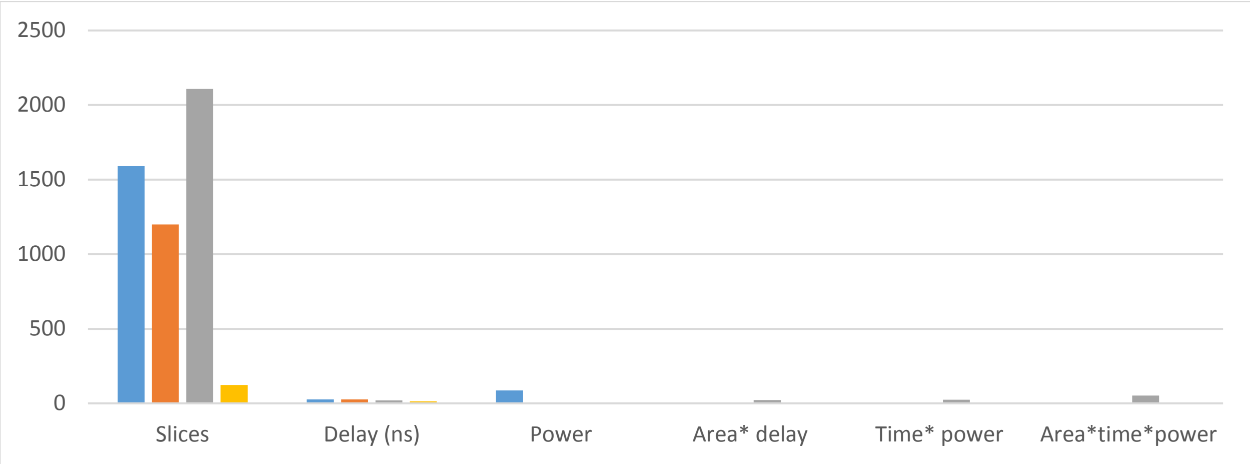

Fig. 6: Improved tool consumption analysis for SDR workloads between new and existing ternary-RNS-based FIR filter designs

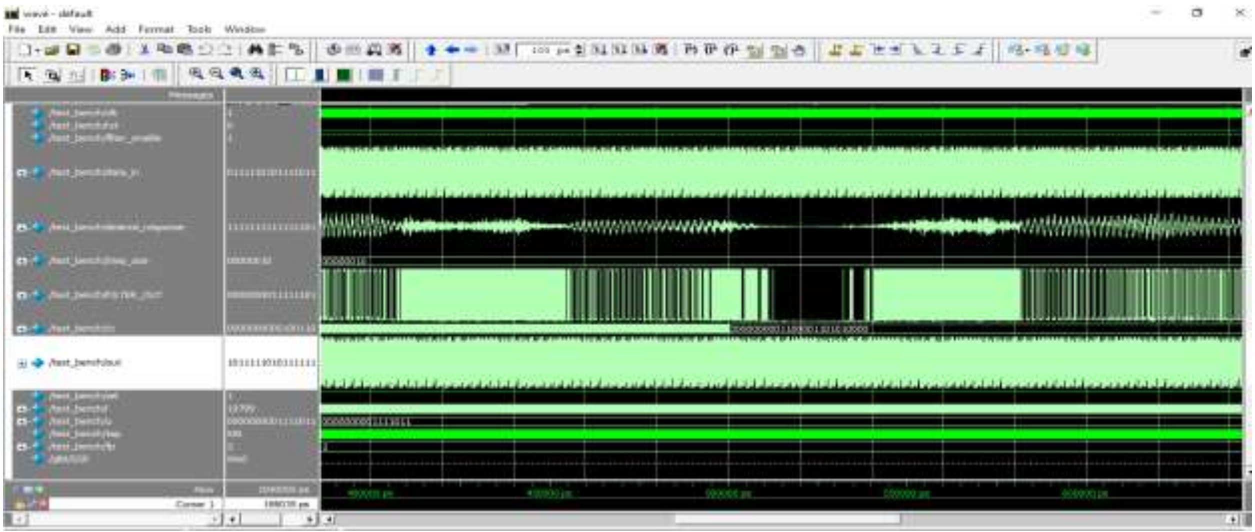

Fig. 7: Simulated results of proposed FIR and NLMS filters for noisy audio signals and filtered signals

Fig. 7 is the simulated results of the proposed RNS-based FIR filter design in the Modelsim simulator and input speech signals are exported from MATLAB in the form of a text file that contains samples of speech signals. These signals real-time recorded speech from an audio device and converted into samples in MATLAB and stored in a text file. This text file is read in the top module of Verilog design and stored on Block Memory Generator. In Fig.7, data_in is a variable that shows sample speech signals read from memory and applied to FIR filter design and Filter_out is another variable used to show filtered signals which are noise removal signals.

6 Conclusion

In this study article, the Booth algorithm increases the competence of 2^n-1 modulo sets by confining the PPs to $1/4^{th}$ of Booth-programmed parameters. Given the presence of a solid manifold, this multiplier is ineffective to dynamic lower ranges, yet it is further along with energy and geography for more extensive considering multiple. In contrast to the modulo converter discussed above, disused scrambling is created to address the transmission of intra-carry computation in RNS because it is suitable for a wider band. Continued studies on 2n-1 modulo factors are, though, still needed. The company uses an RNS-based multiplier that is faster and has better connectivity than just a higher-rate Booth-encoded translator. The repetitive residue number approach can also produce an essential modulo converter in DSP growth. To assess and remove noise signals in audio signals, the researchers used high-end FIR filters with upgraded RNS units. The outcome of the apparatus improvement detailed in this study revealed that each level of modified RNS is a direct effect on the h/w speed & reliability of FIR filter design. By postponing the direct minimization in the RNS technique and reducing the competence consequence break in Finite Impulse Response filter design, the results are compared with Random Access Memory-based hypothetical overturn alteration and Distributed Arithmetic-based excess evaluation. This effort restores trustworthy system efficiency by increasing the Filter bank switch and introducing an upgraded RNS and a unit with a memory-efficient backward converter. When contrasted to DA-based FIR and radix/booth multipliers-based FIR filters, the results show a 13 percent improvement in energy efficiency, a 21 percent decrease in area, and a 13 percent increase in capacity.

References:

[1] C. Srinivasa Murthy, et.al, "Design and Implementation of Hybrid Techniques and DA-based Reconfigurable FIR Filter Design for Noise Removal in EEG Signals on FPGA", WSEAS TRANSACTIONS On SYSTEMS And CONTROL, E-ISSN: 2224-2856, Volume 17, 2022, DOI: 10.37394/23203.2022.17.37.

[2] M. D. Felder, J.C. Mason, B.L. Evans, Efficient dual-tone multifrequency detection using the nonuniform discrete Fourier transform, *IEEE Signal Processing Letters*, **5**(7): 160–163, 1998, doi: 10.1109/97.700916

[3] R. Beck, A.G. Dempster, I. Kale, Finite-precision Goertzel filters used for signal tone detection, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **48**(7): 691–700, 2001, doi: 10.1109/82.958339.

[4] V. Ramakrishna, T.A. Kumar, Low Power VLSI Implementation of Adaptive Noise Canceller Based on Least Mean Square Algorithm, *2013 4th International Conference on Intelligent Systems, Modelling, and Simulation*, pp. 276–279, Bangkok, Thailand, January 29–31, 2013, doi: 10.1109/ISMS.2013.84

[5] D. Souris, K. Sgouropoulos, K. Tatas, V. Pavlidis, A. Thanailakis, A methodology for implementing FIR filters and CAD tool development for designing RNS-based systems, Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS'03), pp. V–V, Bangkok, Thailand, May 25–28, 2003, DOI: 10.1109/ISCAS.2003.1206208.

[6] S. Pontarelli, G. C. Cardarelli, M. Re, and A. Salsano, "Totally Fault-Tole RNS Based FIR Filters," 2008 14th IEEE International On-Line Testing Symposium, 2008, pp. 192-194, DOI: 10.1109/IOLTS.2008.14.

[7] R. Kamal, P. Chandravanshi, N. Jain, and Rajkumar, "Efficient VLSI architecture for FIR filter using DA-RNS," 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE), 2014, pp. 184-187, DOI: 10.1109/ICECCE.2014.7086656

[8] I. Kouretas and V. Paliouras, "Delay-variation-tolerant FIR filter architectures based on the Residue Number System," 2013 IEEE International Symposium on Circuits and Systems (ISCAS), 2013, pp. 2223-2226, DOI: 10.1109/ISCAS.2013.6572318.

[9] S. R. Kotha, S. Bajaj, and S. S. Kumar, "A LUT based RNS FIR filter implementation for reconfigurable applications," 18th International Symposium on VLSI Design and Test, 2014, pp. 1-6, DOI: 10.1109/ISVDAT.2014.6881047.

[10] M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram, BZ-FAD: A low-power low area multiplier based on shift-and-add architecture, IEEE Trans. Very Large Scale Integration (VLSI) Systems. 17 (2009) 302–306.

[11] S. R. Kotha, S. Bajaj, and S. S. Kumar, "An RNS-based reconfigurable FIR filter design using shift and add approach," 2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), 2014, pp. 640-645, DOI: 10.1109/CSNDSP.2014.6923906.

[12] Cong Liu, Jie Han, and Fabrizio Lombardi, A low-power, high-performance approximate multiplier with configurable partial error recovery, in Proc. IEEE Design, Automation and Test in Europe Conf. and Exhibition (DATE), (2014), pp. 1–4.

[13] J. Chen and J. Hu, "Energy-Efficient Digital Signal Processing via Voltage-Overscaling-Based Residue Number System," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 7, pp. 1322-1332, July 2013, DOI: 10.1109/TVLSI.2012.2205953.

[14] BotanyBotang Shao and Peng Li, Array-based approximate arithmetic computing: A general model and applications to the multiplier and squarer design, IEEE Trans. Circuits and Systems-I: Regular Papers. 62 (2015) 1081–1090.

[15] Tallapragada, V. V. Satyanarayana, et al. "Design and Optimization of Fuzzy-Based FIR Filters for Noise Reduction in ECG Signals Using Neural Network." *IJFSA* vol.11, no.3 2022: pp.1-16. http://doi.org/10.4018/IJFSA.312215.

[16] S. Bose, A. De and I. Chakrabarti, "Area-Delay-Power Efficient VLSI Architecture of FIR Filter for Processing Seismic Signal," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 11, pp. 3451-3455, Nov. 2021, doi: 10.1109/TCSII.2021.3081257.

[17] X. X. Zheng, J. Yang, S. Y. Yang, W. Chen, L. Y. Huang and X. Y. Zhang, "Synthesis of Linear-Phase FIR Filters With a Complex Exponential Impulse Response," in IEEE Transactions on Signal Processing, vol. 69, pp. 6101-6115, 2021, doi: 10.1109/TSP.2021.3115352.

[18] X. Xi and Y. Lou, "Sparse FIR Filter Design With k-Max Sparsity and Peak Error Constraints," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 4, pp. 1497-1501, April 2021, doi: 10.1109/TCSII.2020.3027704.

[19] P. Shukl and B. Singh, "Combined IIR and FIR Filter for Improved Power Quality of PV Interfaced Utility Grid," in IEEE Transactions on Industry Applications, vol. 57, no. 1, pp. 774-783, Jan.-Feb. 2021, doi: 10.1109/TIA.2020.3031875.

[20] Wu, T. High-Speed Fault-Tolerant Finite Impulse Response Digital Filter on Field Programmable Gate Array. J. Shanghai Jiaotong Univ. (Sci.) 26, 554–558 (2021). https://doi.org/10.1007/s12204-020-2214-z

[21] B. R. S. Rao and B. B. T. Sundari, "An efficient reconfigurable FIR filter for dynamic filter order variation", Proc. Int. Conf. Commun. Electron. Syst. (ICCES), pp. 1724-1728, 2019.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Manjunath P.S. and Revanna.C. R have identified problems in existing works in the field of filter design and its solutions. Kusuma M.S. and Ponduri Sivaprasad have carried out the design in Verilog using Vivado Design Suite 2018.1 and its simulation and also optimization. Manjunath and Uppala Ramakrishna are written a manuscript. Kusuma M. S was responsible for the Statistics.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflict of interest to declare.