



Enhancing Finite Impulse Response (FIR) Filtering with Distributive Arithmetic (DA) and Residue Number System (RNS) Optimization

Mentam Sunaina^(✉) and G. L. Sumalata

GRIET, ECE, Hyderabad, India
sunainagrace.123@gmail.com

Abstract. Digital Signal Processing (DSP) systems have revolutionized numerous fields such as telecommunications, audio manipulation, image and video treatment and various other domains. Whereas Distributed Arithmetic (DA) is a powerful technique used in DSP area for efficient implementation of various algorithms. This paper presents an optimized Finite Impulse Response (FIR) filter with Distributed Arithmetic (DA) and Residue Number System (RNS). The proposed design leverages the properties of the RNS to achieve efficient and high-performance filtering operations. The input samples and filter coefficients undergo a transformation into residue representation, and the intermediate products are calculated using residue arithmetic. The proposed design has been implemented in Xilinx platform. It uses 15% lesser power demonstrates reduced computational complexity and improved performance has been reduced to compared to conventional FIR filters. The experimental outcomes underscore the efficacy of the suggested method concerning filter efficiency, resource utilization, and energy consumption. The optimized FIR filter using the RNS method presents a promising solution for efficient and high-performance filtering utilized in a range of fields, including digital signal processing and communication systems.

Keywords: Finite Impulse Response Filter · Residue Number System · Distributed Arithmetic · Digital Signal Processing

1 Introduction

Digital Signal Processing Systems uses Digital Finite Impulse Response Filtering operations on a large scale [1]. However, a Digital FIR filter uses Multiply and Accumulate (MAC) where each input is multiplied with every filter coefficient which results in the use of large number of multiplier circuits [2]. This ultimately leads to increased demands in terms of area, delay, and power consumption. In response to this challenge, various algorithms have emerged as viable solutions. Few algorithms were memory-based designs [3], while few were conversion-based designs One such memory-based design is the

Distributed Arithmetic based FIR filter, where the design of FIR filter is carried out using Look-Up-Tables. Look-Up-Table computes the Filter coefficients and the incoming Input signals [2]. The inputs to the DA based FIR filter are given in two's complement form which results in increased computational time. This is because the speed of the computational process is directly related to the length of the input data in bits. This results in computational speed if the bit length of the input data is high.

In order to overcome this, a new approach called RNS based FIR filter is introduced where any complex number is converted into a set of simple residues using the respective moduli numbers [6]. This paper proposes a memoryless FIR filter with traditional DA algorithm using RNS method. This approach uses the simplified conversion process which increases the overall speed and reduces the computational time of the whole design.

The outline of this paper is: 1. Introduction, 2. What is a DA based FIR filter and its design, 3. Why RNS, 4. What is Residue Number System, 4. Proposed method 5. Forward conversion, 6. Reverse conversion, 7. Simulation Result, 8. Conclusion, 9. Reference.

2 DA Based FIR Filter

Distributed Arithmetic (DA) is a widely employed algorithm in the realization of Finite Impulse Response (FIR) Filters [5]. It offers computational efficiency by replacing the expensive multiplication operations with a combination of additions and look-up table operations.

Distributed Arithmetic uses inner product computations, which eliminates the use of repeated multiplication [8]. Below is the mathematical representation of DA algorithm

$$Y = \sum_{k=1}^K d_k x_k \quad (1)$$

where Y is the output, x_k is the input signal, d_k is the filter coefficient and K is the word length. For a K tap FIR filter, there will be K execution cycles. But as the system takes the input signals as binary values, the input signal x_k can be written as

$$x_k = \sum_{n=0}^{N-1} b_{kn} 2^n \quad (2)$$

where b_{kn} is the binary value, which is either 0 or 1 and 2^n is the weight of the corresponding binary bit (see Fig. 1).

$$Y = \sum_{k=1}^K d_k \sum_{n=0}^{N-1} b_{kn} 2^n \quad (3)$$

On rearranging the above equation,

$$Y = \sum_{k=1}^K \left[\sum_{n=0}^{N-1} d_{kn} b_{kn} \right] 2^n \quad (4)$$

$$Y = \left| \sum_{n=0}^{N-1} \left[\sum_{k=0}^{K-1} d_k b_{kn} \right] 2^n \right|_m \quad (5)$$

where, m_i is the modular set of the RNS system, which is $m_i = [m_1, m_2, m_3, \dots, m_n]$. This modular set depends on the number of tap filters that are being used. Where, the inner product can be represented in RNS form as

$$Q_{m_i}(d_k, b_{kn}) = \sum_{n=0}^{N-1} |d_k|_{m_i} |b_{kn}|_{m_i} \tag{6}$$

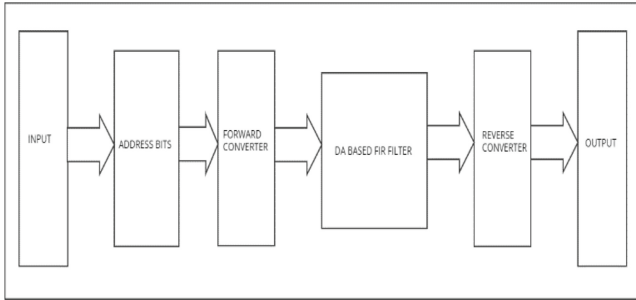


Fig. 1. Block diagram FIR filter using RNS method employing DA algorithm

3 Residue Number System

3.1 What is RNS?

The Residue Number System (RNS) is a mathematical method for representing a sizable number by decomposing it into distinct residue values within a modular set [8]. Each residue value is obtained by dividing the original number to its corresponding modulus value. These modulus vales are co-prime to each other. The main reason for adopting the RNS system in Filter design is because of its capability of reducing Power consumption.

Unlike Binary System, an RNS system represents any large number into a set of small residue values without any carry bits. There are different kinds of residue sets like $\{2^{n-1}, 2^n, 2^{n+1}\}$, $\{2n, 2n + 1, 2n - 1\}$, $\{2^{2n}, 2^n + 1, 2^n - 1\}$ $\{2^{2n}, 2^n + 1, 2^n - 1\}$ etc. [12].The modular set used in this paper is $\{2n, 2n + 1, 2n - 1\}$.

Forward Conversion

The initial step involves transforming the input data and filter coefficient data into residue representation through the utilization of specific moduli values. Later the residues are individually converted into binary form and sent into DA filter where the actual design process is carried out [9].

Example:

Number: 12

Module set: {3, 4, 5}

3) 12 (4	4) 12 (3	5) 12 (2
12	12	10
-----	-----	-----
0	0	2
-----	-----	-----

So, the RNS form of 12 for the moduli set {3, 4, 5} is {0, 0, 5}

Reverse Conversion

To perform the Reverse Conversion from RNS form, a new method known as the Chinese Remainder Theorem is applied. Given a set of pairwise coprime moduli, the theorem states that there exists a unique solution modulo the product of these moduli. This solution can be found by using the Chinese Remainder Theorem algorithm, which involves computing partial remainders and then combining them to get the final solution.

So, the equation for the Reverse conversion is

$$X = \left| \sum_{i=1}^I N_i \cdot N_i^{-1} \cdot y_i \right|_M \tag{7}$$

where, *i* depends on the number of moduli,

$$M = m_1 \cdot m_2 \cdot m_3 \tag{8}$$

$$M_i = M / m_i \tag{9}$$

And N_i^{-1} can be found by satisfying the below equation

$$\left(N_i \cdot N_i^{-1} \right) \% m_i = 1 \tag{10}$$

This can be satisfied by following hit or miss trick

This paper employs the modular set {2n, 2n + 1, 2n-1} for the design of a 4-tap FIR filter. Using Distributed Arithmetic technique. This design has been carried out and synthesized in Xilinx Integrated Synthesis Environment using Spartan3E with 28 nm technology which uses 30% less power compared to existing architectures.

For a 4-tap filter, the moduli set is m {8, 9, 7}. So, to design a 4 tap FIR filter, for the input X {4, 5, 6, 7}, the filter coefficient is taken as D {3, 7, 8, 9}. The forward conversion of the input signals is shown in the below table.

4 Proposed Method

4.1 Forward Conversions of Input Data

Table 1. Conversion of input signals into Residue form

Input	m = 8	m = 9	m = 7
4	4	4	4
6	5	5	5
6	6	6	6
7	7	7	0

The forward conversion of input data into RNS form is shown in Table 1. Whereas the filter coefficients are preserved in a Look-Up Table for storage and subsequent use and later used accordingly during filter design. The filter coefficients $\{d_1, d_2, d_3, d_4\} = \{3, 7, 8, 9\}$. The stored elements in LUT are showed in Table. 2

Consequently, upon sending the input data to the DA filter, the filter coefficients are computed as required, facilitating the design of an FIR filter.

$$Y = \sum_{k=1}^K d_k x_k \tag{11}$$

$$Y_8 = |d_1 x_1 + d_2 x_2 + d_3 x_3 + d_4 x_4|_8 \tag{12}$$

$$Y_8 = |4(3) + 5(7) + 6(0) + 7(1)|_8$$

$$Y_8 = |12 + 35 + 0 + 7|_8$$

$$Y_8 = |54|_8$$

$$Y_8 = 6$$

Similarly, $Y_9 = 5, Y_7 = 4$

So, this can be written as $Y \{Y_8, Y_9, Y_7\} = \{6, 5, 4\}$.

4.2 Reverse Conversion Process

The Chinese Remainder Theorem is a method that can be employed to conduct the reverse conversion of RNS. This can be followed by

$$M = m_1 * m_2 * m_3 \tag{13}$$

Table 2. Contents into LUT

t_{1n}	t_{2n}	t_{3n}	t_{4n}	Entry values	m = 8	m = 9	m = 7
0	0	0	0	0	0	0	0
0	0	0	1	$d_1 = 3$	3	3	3
0	0	1	0	$d_2 = 7$	7	7	0
0	0	1	1	$d_2 + d_1 = 7 + 3 = 10$	2	1	3
0	1	0	0	$d_3 = 8$	0	8	1
0	1	0	1	$d_3 + d_1 = 8 + 3 = 11$	3	2	4
0	1	1	0	$d_3 + d_2 = 8 + 7 = 15$	7	6	1
0	1	1	1	$d_3 + d_2 + d_1 = 8 + 7 + 3 = 18$	2	0	4
1	0	0	0	$d_4 = 9$	1	0	2
1	0	0	1	$d_4 + d_1 = 9 + 3 = 12$	4	3	5
1	0	1	0	$d_4 + d_2 = 9 + 7 = 16$	0	7	2
1	0	1	1	$d_4 + d_2 + d_1 = 9 + 7 + 3 = 19$	3	1	5
1	1	0	0	$d_4 + d_3 = 9 + 8 = 17$	1	8	3
1	1	0	1	$d_4 + d_3 + d_1 = 9 + 8 + 3 = 20$	4	2	6
1	1	1	0	$d_4 + d_3 + d_2 = 9 + 8 + 7 = 24$	0	6	3
1	1	1	1	$d_4 + d_3 + d_2 + d_1 = 9 + 8 + 7 + 3 = 27$	3	0	6

$$M = 8 * 9 * 7$$

$$M = 504$$

$$M_1 = M / m_1 \tag{14}$$

$$M_1 = \frac{504}{8} = 63$$

$$M_2 = M / m_2 \tag{15}$$

$$M_2 = \frac{504}{9} = 56$$

$$M_3 = M / m_3 \tag{16}$$

$$M_3 = \frac{504}{7} = 72$$

To get the inverse value of each module, hit and check method is employed to satisfy the below equation

$$(M_i * M_i^{-1}) \% m_1 = 1 \quad (17)$$

$$(M_1 * M_1^{-1}) \% m_1 = 1$$

$$(63 * 7) \% 8 = 1$$

$$(M_2 * M_2^{-1}) \% m_2 = 1$$

$$(72 * 4) \% 7 = 1$$

So,

$$M_1 = 63; M_1^{-1} = 7$$

$$M_2 = 56; M_2^{-1} = 5$$

$$M_3 = 72; M_3^{-1} = 4$$

where i represents the respective module.

So, the actual output can be found by

$$Y = |(M_1 * M_1^{-1} * Y_8) + (M_2 * M_2^{-1} * Y_9) + (M_3 * M_3^{-1} * Y_7)|_M \quad (18)$$

$$Y = |(63 * 7 * 6) + (56 * 5 * 5) + (72 * 4 * 4)|_{504}$$

$$Y = |(2646 + 1400 + 1152)|_{504}$$

$$Y = |5198|_{504}$$

$$Y = 158$$

5 Simulation Results

The presented design's performance and hardware complexity are evaluated by contrasting them with those of the current architectures. The proposed filter is designed on Spartan 3E Field Programmable Gate Array which is a part of Xilinx family. It is a cost sensitive device which is easy to implement. The Spartan 6 family provides a 45 nm technology DSP block to implement high efficient algorithm.

Table 3. Performance Comparison

Design	Critical Path Delay (ns)	Area (μm^2)	Total delay (ns)	ADP ($mm^2 ns$)	Power (mw)	PDP (pJ)
Chen et al. (2006)	16.33	8357.4	16.33	0.138	5.032	82.13
Kamal et al. (2014)	15.30	6271.6	15.30	0.096	4.629	70.8237
Grande et al. (2020)	14	5432	14	0.076	4.411	61.754
Proposed design	12.24	4812	12.24	0.059	4.249	58.684

Using the proposed design, it is evident the design of 4 tap filter is made easy by eliminating the risk of carries in any regular Binary system. From Table 3, it clearly shows that the proposed design uses 15% lesser power compared to previous work [6] and utilizes 17% lesser area than the existing work [10]. All the other performance metrics are mentioned in Table 3 and the graphical representation of Area, delay and power are mentioned in the Figs. 2, 3 and 4 respectively. Additionally, it’s worth noting that the overall delay of the proposed approach is significantly reduced when compared to the delays associated with all other architectures listed in the Table 3. The area delay product and Power Delay Product are also comparatively very less.

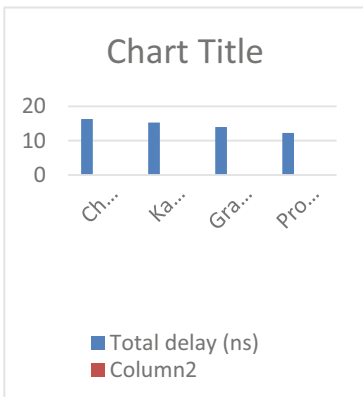


Fig. 2. Total delay

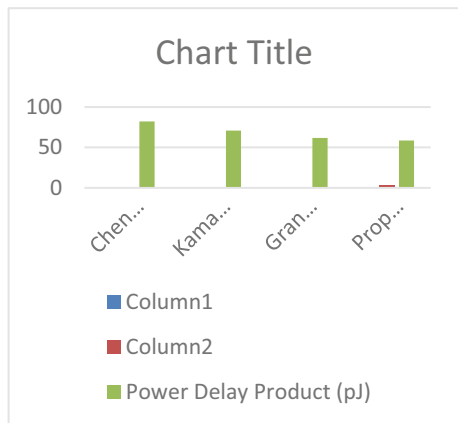


Fig. 3. Power Delay Product (pJ)

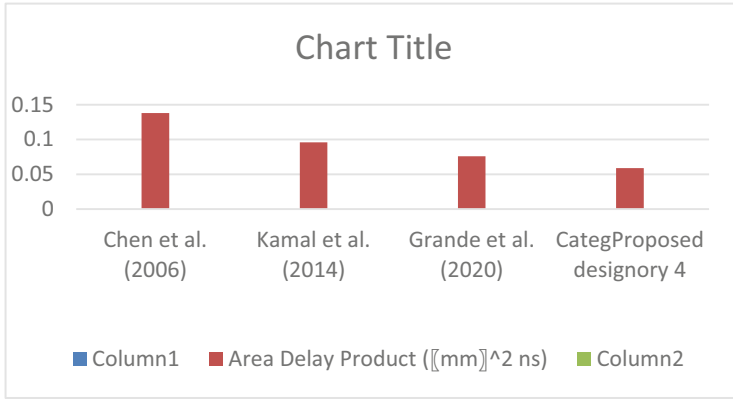


Fig. 4. Area Delay Product (mm^2ns)

6 Conclusion

This paper shows the simplest form of DA based FIR Filter using RNS method. Instead of using the lengthy Binary number, any number is represented as the simplest RNS form which is then converted into Binary form and used for implementation purpose. The actual goal of decreasing the overall delay is achieved using this method. Compared to the existing methods, there is a 17% less delay in the proposed method. As RNS can perform fast and parallel computations, this method better suits as a replacement for a regular FIR filter. The only limitation of using RNS system is the use of converters which makes it look a little complex. However with the use of CRT the conversions have been made simple and easy to compute. But use of RNS system is best suit in many DSP applications because of its capability of less Power consumption and reduced Delay.

References

1. NagaJyothi, G., SriDevi, S.: Distributed arithmetic architectures for fir filters-a comparative review. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 2684–2690. IEEE (2017)
2. 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. **61**(4), 921–932 (2012)
3. Shanthala, S., Kulkarni, S.Y.: VLSI design and implementation of low power MAC unit with block enabling technique. Eur. J. Sci. Res. **30**(4), 620–630 (2009)
4. Hartley, R.I.: Subexpression sharing in filters using canonic signed digit multipliers. IEEE Trans. Circ. Syst. II: Analog Digit. Sig. Process. **43**(10), 677–688 (1996)
5. Chandra, A., Chattopadhyay, S.: A new strategy of image denoising using multiplier-less FIR filter designed with the aid of differential evolution algorithm. Multimedia Tools Appl. **75**, 1079–1098 (2016)
6. Akhter, S., Kumar, S., Bareja, D.: Design and analysis of distributed arithmetic based FIR filter. In: 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 721–726. IEEE (2018)

7. Jamal, K., Chari, K.M., Srihari, P.: Test pattern generation using thermometer code counter in TPC technique for BIST implementation. *Microprocess. Microsyst.* **71**, 102890 (2019)
8. Kamal, R., Chandravanshi, P., Jain, N.: November. Efficient VLSI architecture for FIR filter using DA-RNS. In: 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE), pp. 184–187. IEEE (2014)
9. Jameii, S.M., Taghipour, S., Azad, M.: Using both binary and residue representations for achieving fast converters in RNS (2011)
10. Jamal, K., Srihari, P., Chari, K.M., Sabitha, B.: Low power test pattern generation using test-per-scan technique for BIST implementation. *ARPN J. Eng. Appl. Sci.* **13**(8) (2018)
11. Cao, B., Chang, C.H., Srikanthan, T.: An efficient reverse converter for the 4-moduli set $2/\text{sup } n/-1, 2/\text{sup } n/, 2/\text{sup } n/+ 1, 2/\text{sup } 2n/+ 1$ based on the new Chinese remainder theorem. *IEEE Trans. Circ. Syst. I: Fundam. Theory Appl.* **50**(10), 1296–1303 (2003)
12. Matutino, P.M., Chaves, R., Sousa, L.: December. ROM-less RNS-to-binary converter moduli $\{2 \cdot 2n-1, 2 \cdot 2n+1, 2 \cdot n-3, 2 \cdot n+3\}$. In: 2014 International Symposium on Integrated Circuits (ISIC), pp. 432–435. IEEE (2014)
13. Sridevi, N., Jamal, K., Mannem, K.: Implementation of error correction techniques in memory applications. In: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 586–595. IEEE (2021)
14. Garner, H.L.: The residue number system. In: Papers Presented at the March 3–5, 1959, Western Joint Computer Conference, pp. 146–153 (1959)
15. Albicocco, P., Cardarilli, G.C., Nannarelli, A., Re, M.: Twenty years of research on RNS for DSP: Lessons learned and future perspectives. In: 2014 International Symposium on Integrated Circuits (ISIC), pp. 436–439. IEEE (2014)
16. Meher, P.K., Chandrasekaran, S., Amira, A.: FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. *IEEE Trans. Signal Process.* **56**(7), 3009–3017 (2008)
17. Longa, P., Miri, A.: Area-efficient FIR filter design on FPGAs using distributed arithmetic. In: 2006 IEEE International Symposium on Signal Processing and Information Technology, pp. 248–252. IEEE (2006)
18. Praveen Sundar, P.V., Ranjith, D., Karthikeyan, T., Vinoth Kumar, V., Jeyakumar, B.: Low power area efficient adaptive FIR filter for hearing aids using distributed arithmetic architecture. *Int. J. Speech Technol.* **23**(2), 287–296 (2020)
19. Balaji, M., Padmaja, N., Gitanjali, P., Shaik, S.A., Kumar, S.: Design of FIR filter with fast adders and fast multipliers using RNS algorithm. In: 2023 4th International Conference for Emerging Technology (INCET), pp. 1–6. IEEE (2023)