



Development and Realization of an FIR Filter Utilizing an Innovative RNS Form with a Dual Modular Set

Mentam Sunaina^(✉) and G. L. Sumalata

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

Abstract. Distributed Arithmetic (DA) serves as a potent tool in DSP for the efficient execution of diverse algorithms. This paper introduces an enhanced implementation of a Finite Impulse Response (FIR) filter based on Distributed Arithmetic (DA) utilizing a new modular set of Residue Number System (RNS) approach. The proposed approach demonstrates notable reductions in computational complexity and improved performance with a reduced delay of 17% when compared to traditional FIR filters. Unlike any other traditional RNS method, this paper offers 2 moduli set of RNS system to make it even more simple to calculate and execute. Experimental findings underscore the effectiveness of this approach in terms of filter performance, resource utilization, and power efficiency. Consequently, this optimized DA-based FIR filter utilizing the RNS method stands as a promising solution for efficient and high-performance filtering applications across various domains, such as digital signal processing and communication systems.

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

1 Introduction

Digital Signal Processing (DSP) systems have brought about significant advancements across a wide range of fields, spanning telecommunications, image and video processing, and numerous others. The Distributed Arithmetic (DA) algorithm is a frequently employed method in realizing Finite Impulse Response (FIR) Filters in DSP systems [1]. This technique enhances computational efficiency by substituting resource-intensive multiplication operations with a blend of addition and look-up table operations [2]. Meanwhile, designing of Distributed Arithmetic based FIR filter using RNS method is a quite known [3].

RNS system has gained its strength over the last decades, however the only hard factor is its conversions and picking the moduli set for the approach [10].

To overcome the complexity of multiple modular set values, this paper proposes a new 2 moduli set RNS form, to design an FIR filter. This paper highlights the simplest form of both forward and reverse conversions of RNS system to design an FIR filter.

The paper is structured as follows: 1. Introduction, 2. Exploring DA-Based FIR Filters using RNS method, 3. The Proposed Methodology, 6. Forward Conversion Process, 7. Reverse Conversion Technique, 8. Analysis of Simulation Results, 9. Conclusion and 10. Reference.

2 Exploring DA-Based FIR Filter Using RNS Method

Distributed Arithmetic based FIR filter using RNS method relies on inner product computations, effectively eliminating the recurring use of multiplication operations [4]. Let’s delve into its mathematical formulation.

$$Y = \left| \sum_{n=0}^{N-1} \left[\sum_{k=0}^{K-1} d_k b_{kn} \right] 2^n \right|_{mi} \tag{1}$$

In this context, *mi* refers to the modular set of the RNS system, which is represented as $mi = [m1, m2, m3, \dots mn]$. The choice of this modular set depends on the specific number of tap filters employed in the system.

Forward Conversion

The forward conversion process involves determining the remainder obtained when the number is divided by the moduli in the selected residue set [9]. The set of the smallest remainders among these divisions constitutes the RNS representation of the given number.

Example. Number: 15.

Module set: {8, 9, 7}.

8) 15 (1	9) 15 (1	7) 15 (2
8	9	14
-----	-----	-----
7	6	1
-----	-----	-----

So, the RNS form of 15 with respect to the module set {8, 9, 7} is {7, 6, 1}.

Reverse Conversion

The Reverse Conversion from RNS form employs a mathematical technique known as the Chinese Remainder Theorem. This theorem offers a solution to a system of congruence, facilitating the conversion process [6].

$$X = \left| \sum_{i=1}^I N_i \cdot N_i^{-1} \cdot y_i \right|_M \tag{2}$$

where, *i* represents the number of moduli values in the set

$$M = m_1 \cdot m_2 \cdot m_3 \tag{3}$$

$$N_i = M \div (m_i) \tag{4}$$

And N_i^{-1} is the modular multiplicative inverse of N_i and can be obtained by satisfying the below equation

$$(N_i \cdot N_i^{-1}) \% m_i = 1 \tag{5}$$

3 Proposed Method

In this paper, Distributed Arithmetic based FIR filter using the RNS moduli set $\{2^n, 2^{pn} - 1\}$ is used, where n is the bit length and p is a natural number. Unlike any other RNS set, this module set only has 2 modules which make it way simpler in calculation and execution purpose. The design has been executed and synthesized within Xilinx Integrated Synthesis Environment using Artix-7 which uses 13% lesser power and 27% lesser delay compared to the existing architectures.

For a 4-tap FIR filter, the input values are taken as $X \{14, 19, 17, 15\}$ and the Filter coefficients denoted as $D \{12, 18, 16, 13\}$. As it's a 4-tap filter, and the natural number p is taken as 2. So, the modular set will be $m \{16, 255\}$.

The initial step involves transforming the input data and filter coefficient data into residue form, employing specific moduli values shown in Tables 1 and 2 respectively. Subsequently, the residues are individually translated into binary representation, and they are then transmitted to a DA filter where the primary design process takes place. Figure no. 1 represents the design of 2 modular RNS set.

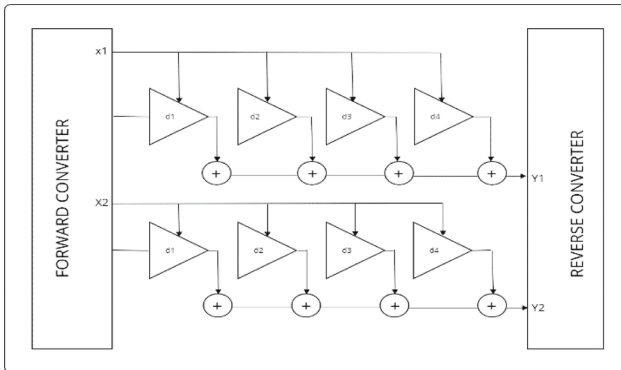


Fig. 1. Proposed FIR filter using two modular RNS set

3.1 Forward Conversion

Table 1. Conversion of input signals into Residue form

Input	m = 16	m = 255
14	14	14
19	3	19
17	1	17
15	15	15

Hence, the RNS input data and the filter coefficients are directed to the DA filter (Fig. 1), the filter coefficients are computed in accordance with the provided residue values, facilitating the design of an FIR filter. This design process utilizes the residue values of both the input data and the filter coefficients.

$$Y = \sum_{k=1}^K d_k x_k \quad (6)$$

$$Y_{16} = |x_1 d_1 + x_2 d_2 + x_3 d_3 + x_4 d_4|_{16} \quad (7)$$

$$Y_{16} = |14(12) + 3(2) + 1(0) + 15(13)|_{16}$$

$$Y_{16} = |369|_{16}$$

$$Y_{16} = 1$$

Similarly, $Y_{255} = 212$

Now the output set can be written as $Y \{1, 212\}$

3.2 Reverse Conversion

The Reverse conversion of an RNS system employs Chinese Remainder Theorem.

$$M = m_1 * m_2$$

$$M = 16 * 255$$

$$M = 4080$$

$$M_1 = \frac{M}{m_1} = \frac{4080}{16} = 255$$

Table 2. LUT entries

t_{1n}	t_{2n}	t_{3n}	t_{4n}	Entry values	m = 16	m = 255
0	0	0	0	0	0	0
0	0	0	1	$d_1 = 12$	12	12
0	0	1	0	$d_2 = 18$	2	18
0	0	1	1	$d_2 + d_1 = 7 + 3 = 10$	10	10
0	1	0	0	$d_3 = 16$	0	16
0	1	0	1	$d_3 + d_1 = 8 + 3 = 11$	11	11
0	1	1	0	$d_3 + d_2 = 8 + 7 = 15$	15	15
0	1	1	1	$d_3 + d_2 + d_1 = 8 + 7 + 3 = 18$	2	18
1	0	0	0	$d_4 = 13$	13	13
1	0	0	1	$d_4 + d_1 = 9 + 3 = 12$	12	12
1	0	1	0	$d_4 + d_2 = 9 + 7 = 16$	0	16
1	0	1	1	$d_4 + d_2 + d_1 = 9 + 7 + 3 = 19$	3	19
1	1	0	0	$d_4 + d_3 = 9 + 8 = 17$	1	17
1	1	0	1	$d_4 + d_3 + d_1 = 9 + 8 + 3 = 20$	4	20
1	1	1	0	$d_4 + d_3 + d_2 = 9 + 8 + 7 = 24$	8	14
1	1	1	1	$d_4 + d_3 + d_2 + d_1 = 9 + 8 + 7 + 3 = 27$	11	17

$$M_2 = \frac{M}{m_2} = \frac{4080}{255} = 16$$

In order to find the inverse value of each module, Hit and Check method is used.

$$(M_i * M_i^{-1}) \% m_i = 1$$

Here, i denotes respective module.

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

$$(255 * 15) \% 16 = 1$$

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

$$(16 * 16) \% 255 = 1$$

$$M_1 = 255; M_1^{-1} = 15$$

$$M_2 = 16; M_2^{-1} = 16$$

The final output can we found as

$$Y = |(M_1 * M_1^{-1} * Y_{16}) + (M_2 * M_2^{-1} * Y_{255})|_M \quad (8)$$

$$Y = |(255 * 15 * 14) + (16 * 16 * 158)|_{4080}$$

$$Y = |(53550 + 40448)|_{4080}$$

$$Y = |93998|_{4080}$$

$$Y = 158$$

4 Simulation Results

Table 3. Performance comparison

Design	Total Delay (ns)	Area (μm^2)	ADP ($mm^2 ns$)	Power(mw)	PDP (pJ)
Mohanty et al. (2017)	15.39	85262	0.131	3.2548	50.2548
Kamal et al. (2014)	15.30	6271.6	0.096	4.629	70.8237
Praveen et al. (2020)	12	41946	0.058	3.2548	45.5672
Proposed design	08.94	3586	0.0320	2.8921	25.8309

The performance and hardware intricacy of the design presented in this paper are assessed through a comparative analysis against existing architectural models. The filter design is executed on the Artix-7 Field Programmable Gate Array, belonging to the Xilinx family. This cost-effective and user-friendly device offers efficient implementation capabilities, benefitting from the 28 nm technology DSP block within the Artix-7 family for high-efficiency algorithm execution.

By implementing the proposed design, it becomes evident that crafting a 4-tap filter is considerably simplified, not only by eliminating the concerns related to carries as found in conventional Binary systems but also by utilizing a more straight forward modular set. As observed in Table 3, the data clearly indicates that the proposed design consumes 43% less power compared to previous work [1] and is 27% less Area than the existing work [3]. Furthermore, the overall delay associated with the proposed approach is notably reduced by 17% when compared to the delays seen in all other architectures detailed in Table 3. The area delay product and Power Delay Product also exhibit notably lower values (Fig. 2, 3 and 4).

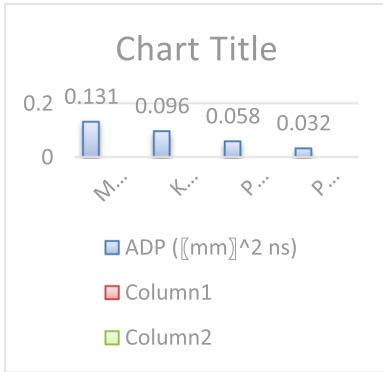


Fig. 2. ADP (mm²ns)

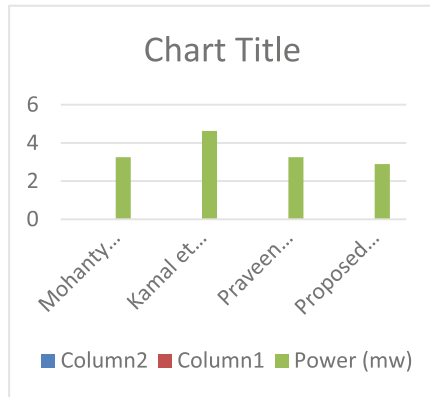


Fig. 3. Power (mw)

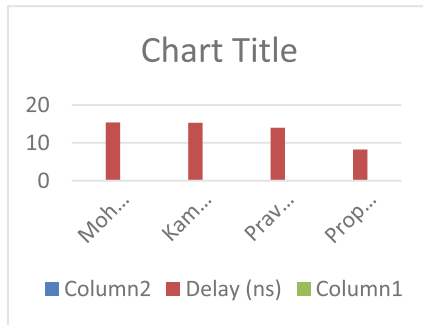


Fig. 4. Critical Path delay (ns)

5 Conclusion

This paper presents a streamlined form of a DA-based FIR Filter, leveraging the RNS method. Instead of employing lengthy binary numbers or intricate modular sets, this approach represents numbers in their simplest RNS form, which is subsequently converted into binary for implementation purposes. The primary objective of reducing overall

delay is effectively realized through this method, with a 17% reduction in delay when compared to existing methods. Given RNS's ability to perform fast and parallel computations, it emerges as a promising alternative to conventional FIR filters. The only drawback of the RNS system is the need for converters, which may introduce some complexity. However, the use of Chinese Remainder Theorem (CRT) simplifies these conversions, making them more accessible. Despite this limitation, RNS is well-suited for many DSP applications due to its capability to minimize power consumption and reduce delay.

References

1. 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)
2. Balaji, M., Padmaja, N.: High-speed DSP pipelining and retiming techniques for distributed-arithmetic RNS-based FIR filter design. *WSEAS Trans. Syst. Control* **17**, 549–556 (2022)
3. Balaji, M., Padmaja, N.: Distributed Arithmetic RNS-based FIR Filter Design using Pipelining and Retiming Methods for High-Speed DSP Systems (2022)
4. 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)
5. Jameii, S.M., Taghipour, S., Azad, M.: Using both binary and residue representations for achieving fast converters in RNS (2011)
6. Cao, B., Chang, C.H., Srikanthan, T.: An efficient reverse converter for the 4-moduli set $2/\sup n/-1, 2/\sup n/, 2/\sup n/+ 1, 2/\sup 2n/+ 1$ based on the new Chinese remainder theorem. *IEEE Trans. Circ. Syst. I: Fund. Theory Appl.* **50**(10), 1296–1303 (2003)
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. Garner, H.L.: The residue number system. In Papers Presented at the March 3–5, 1959, Western Joint Computer Conference, pp. 146–153 (1959)
9. Jenkins, W., Leon, B.: The use of residue number systems in the design of finite impulse response digital filters. *IEEE Trans. Circ. Syst.* **24**(4), 191–201 (1977)
10. 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)
11. Mohan, P.A.: Residue Number Systems: Algorithms and Architectures. Springer Science & Business Media (2021)
12. 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)
13. 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)