

# Design and Implementation of RNS Filter using Modular-Multipliers

P. Bhargavi, K. Srinivasa Reddy, M. Durga Prakash.

**Abstract:** In this paper, an efficient RNS based multiply-accumulate (MAC) unit is proposed to implement residue number system (RNS) based finite impulse response filter (FIR). The proposed MAC (PMAC) approach reduces the number of adders in critical path delay. In this work, a FIR filter with PMAC approach is implemented using structural Verilog HDL language. The United Microelectronics Corporation 90 nm technology library has been used for synthesis. The performance metrics such as area, power and delay are obtained using Cadence RTL compiler. The synthesis results shows that RNS filter with PMAC improves clock frequency and reduces delay and area when compared to conventional MAC (CMAC). To compare the performance of the filters power delay product (PDP) is also considered. The PMAC architecture has improved PDP gain by 30.63% when compared to CMAC.

**Index Terms:** FIR filter, residue number system, Multiply-accumulate unit.

## I. INTRODUCTION

Digital signal processing (DSP) systems such as IOT devices, biomedical devices and traction converters uses finite impulse response (FIR) filters as the building blocks which allows an AC components and blocks DC components. FIR filters are stable and have linear phase as compared to infinite impulse response (IIR) filters. The transposed direct form FIR filter structure is suitable for higher clock frequencies than the direct form FIR filter structure [1]. Previously, to improve clock frequency RNS based FIR filters are used which performs arithmetic operations in parallel with small size residue numbers [2]. It also provides delay advantage as compared to the conventional binary method. Further, the implementation of proposed work also uses RNS based FIR filters. A binary number can be represented by a set of numbers  $\{m_1, m_2, m_3 \dots, m_q\}$  called moduli in RNS method [3]. Here, the relatively prime values are the number of modulus represented as  $q$ . The dynamic range is defined as  $[0, M-1]$  and  $M$  is given by,

$$M = \prod_{i=1}^q m_i \quad (1)$$

Where,  $m_i$  means modulus. In some of the popular moduli sets such as  $\{2^k-1, 2^k, 2^{k+1}\}$  and  $\{2^k-1, 2^k, 2^{k+1}-1\}$   $m_i$  can be

represented by powers of 2 [2-4]. The RNS moduli set for  $q = 3$  is defined as  $\{m_1, m_2, m_3\}$ . The residues of two inputs  $A$  and  $B$  with respect to  $(m_1, m_2, m_3)$  moduli set are represented as  $(a_1, a_2, a_3)$  and  $(b_1, b_2, b_3)$ , respectively. The residues are defined as  $a_1 = |A|_{m_1}$ ,  $a_2 = |A|_{m_2}$ ,  $a_3 = |A|_{m_3}$  and  $b_1 = |B|_{m_1}$ ;  $b_2 = |B|_{m_2}$ ;  $b_3 = |B|_{m_3}$ . They can be represented as

$$a * b = \begin{cases} |a_1 * b_1|_{m_1} \\ |a_1 * b_1|_{m_2} \\ |a * b_1|_{m_3} \end{cases} \quad (2)$$

From equation (2), it is clear that the modulus arithmetic operations are performed in parallel in RNS. Therefore, RNS based systems are faster than the binary number systems. The RNS-based FIR filters are implemented using forward converter (i.e. binary to residue conversion), arithmetic circuits such as modulo multipliers and modulo adders for each modulus and a reverse converter (i.e. residue to binary conversion). In present work, modulo set  $M1 \{2^n-1, 2^n, 2^{n+1}-1\}$  is selected. There are some previous papers who worked on  $M1$  moduli set. Mohan had implemented the design of residue number system (RNS) to binary converters for a new powers-of-two related three-moduli set  $\{2^{n+1}-1, 2^n, 2^n-1\}$  is considered [3]. Kotha & Sahoo had used an efficient new modular multiplication method for  $\{2^{k-1}, 2^k, 2^{k+1}-1\}$  moduli set is proposed to implement a residue number system (RNS)-based fixed coefficient finite impulse response filter [5]. Wang et.al [6] had first taken a new formulation of the Chinese remainder theorem is proposed that reduces the size of modulo operation. Zivalijevic et.al [7] had designed and implemented for modulo set  $2^n+1$  for arithmetic operations. The fixed coefficient FIR filter implementation, PMAC is proposed using the  $M1$  moduli set. In this work, the input binary number is split into groups with the length of  $K$  bits each. The partial products are selected using  $K$  bit number of a particular filter coefficient. The filter uses a 3:2 compressor and ripple carry adder (RCA) instead of two mod adders [8]. When compared to conventional CMAC hardware cost of a filter is reduced. The paper consists of four sections. The proposed PMAC approach with the  $M1$  moduli set is explained in section 2. In section 3, the implementation methods of filter, synthesis results have been discussed and concluded in section 4.

Revised Manuscript Received on July 05, 2019.

**P. Bhargavi**, Electronics and Communication Engineering, Vr. Siddhartha Engineering College Vijayawada, India.

**K. Srinivasa Reddy**, Electronics and Communication Engineering, Vr. Siddhartha Engineering College Vijayawada, India.

**M. Durga Prakash**, Electronics and Communication Engineering, Vr. Siddhartha Engineering College Vijayawada, India.



II. PMAC DESIGN

In this section, FIR filter architectures and their functionality has been discussed. A FIR filter of order N is given by a transfer function Y (n).

$$Y(n) = \sum_{l=0}^{N-1} h_l * X(n-l) \tag{3}$$

A. Conventional MAC

A conventional multiply-accumulate RNS filter architecture (CMAC) is implemented with conventional modulo-multiplier CMM and a modulo-adder (MA) as

shown in figure 1. The RNS representation of A and hl are (a1, a2, a3) and (h1, h2, h3), respectively. The residues of each coefficient are represented as {h1,1,h1,2,h1,3}, where h1,1=|h1|2<sup>k-1</sup>, h1,2=|h1|2<sup>k</sup>, h1,3=|h1|2<sup>k+1-1</sup>. Both the numbers have bit size of {n,n,n+1}. In fig. 1, CMAC consists of partial products generation, CSA stages and two mod-adders (MA) for adding the sum and carry bits generated from CSA stages. For n=8, when modulo set M1 is {2<sup>n</sup>-1, 2<sup>n</sup>, 2<sup>n+1</sup>-1}, the modular multiplication of A \* hl consists of {8, 8, 9} partial products are generated. The moduli set M1 requires six CSA stages and two modulo adders for each modulus. The critical path that takes high computation time is forward converter, CMM and MA as shown in figure.

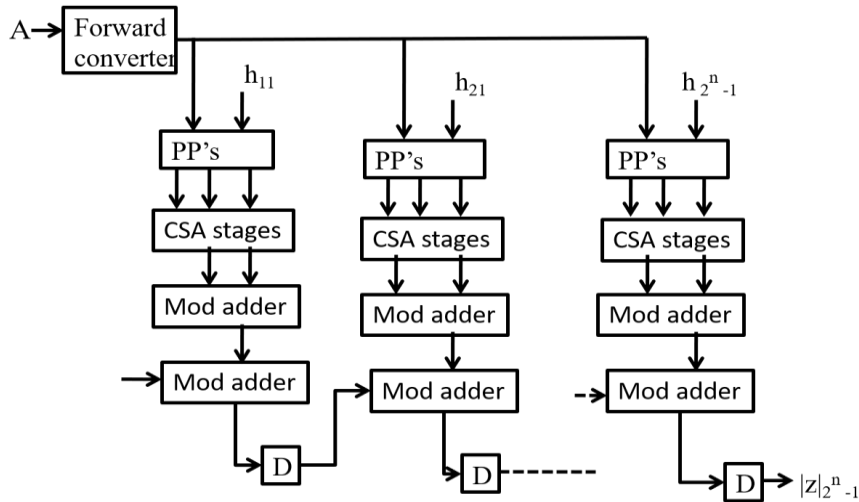


Fig. 1: A Conventional RNS filter multiply-accumulate architecture (CMAC) [4]

B. PMAC Implementation

In the proposed work, RNS based multiply-accumulate architecture (PMAC) is implemented. In this work, partial products generation, CSA stages, 3:2 compressor and ripple carry adder are proposed for fixed coefficients as shown in fig. 2. The proposed architecture increases the clock frequency and provides implementation of delay and area efficient RNS filter. In the proposed filter, i bit input A is multiplied with n bit filter coefficient hl using binary multiplier. For example, consider i=8 and n=5, then the

modulo-set is {2<sup>n</sup> -1, 2<sup>n</sup>, 2<sup>n+1</sup> -1} i.e. {31, 32, 63}. Now, assume that A=100 and hl=25 for 2<sup>n</sup> -1 modulo-set, the residues of each modulus A (n) =|100|<sub>2<sup>n</sup>-1}</sub> and hl (n) =|25|<sub>2<sup>n</sup>-1}</sub> are computed. When these residues are multiplied, partial products PP (n) are generated which can be expressed as follows:

$$PP(n) = A(n) * hl(n) \tag{4}$$

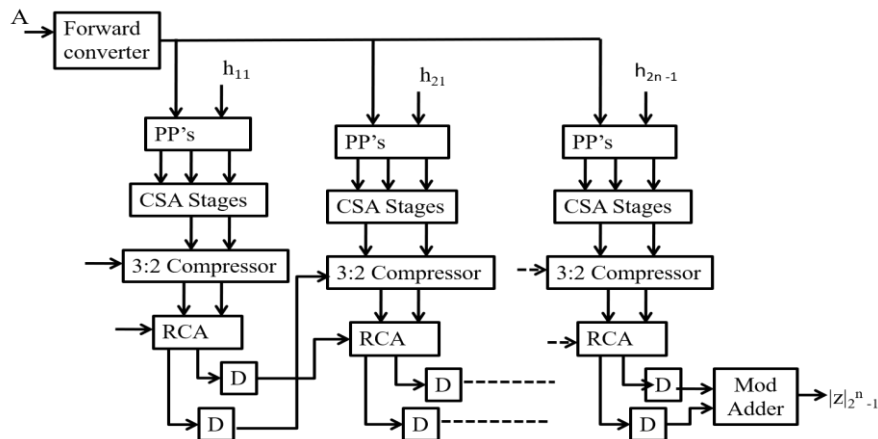


Fig. 2: Proposed RNS filter multiply-accumulate architecture (PMAC)

These partial products are accumulated at every tap by using three CSA stages (carry



save adder), where they are adjusted in same bit size by shifting the MSB bits to LSB bits. The output sum and carry bits along with previous sum bit are given to 3:2 compressor which is followed by one ripple carry adder (RCA). The output sum and carry bits of RCA are stored in the delay registers at every single tap of the filter. The delay elements passes the values delaying them by certain amount of time so that the signal values of the previous steps are multiplied with the corresponding coefficients. Hence the sum and carry bits in the registers are added at the last tap by using modulo adder (MA) before the reverse converter implementation. The filter output in the RNS form  $|Y(n)|_{2^{n-1}}$  just before the reverse converter is:

$$|Y(n)|_{2^{n-1}} = |COE(n) * 2 * X(n)|_{2^{n-1}} \quad (5)$$

Where,  $X(n)$  and  $Y(n)$  are the residue representations of input and output values. The filter output can be obtained in binary form using the reverse converter block. The typical blocks such as RCA and MA used in the proposed filter are discussed in the following subsections.

### C. Ripple carry adder (RCA)

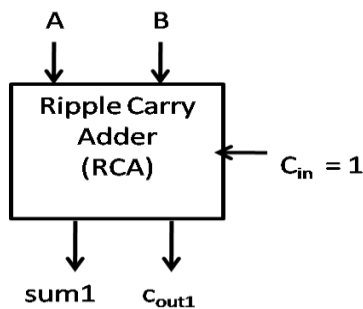


Fig. 3: Ripple carry adder (RCA) [4]

This block does the addition operation of carry from the previous tap of delay register. It is used for accumulating the product values at each tap. In fig. 3, RCA block adds two n-bit inputs A, B with 1-bit  $C_{in}$  which generates n-bit sum and 1-bit  $C_{out}$ . Here the critical path goes from  $c_{in}$  to  $c_{out}$ . In RNS for  $2^n - 1$  and  $2^{n+1} - 1$  modulo addition,  $C_{out}$  and sum will be added using another adder at the last tap of filter. To reduce the adder's cost,  $C_{out}$  will be assigned to  $C_{in}$  of the next tap. However, for  $2^n$  modulus  $C_{out}$  will be neglected at each tap.

### D. 3:2 Compressor

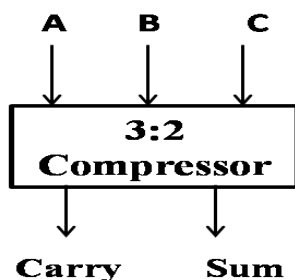


Fig. 4: 3:2 Compressor [8]

The block adds the outputs obtained from the CSA stages along with the previous tap sum stored in register. It gives sum

and carry as a result. The logic inside the compressor is similar to full adder operation. These outputs are given to RCA for the final sum.

### E. Modulo adder (MA)

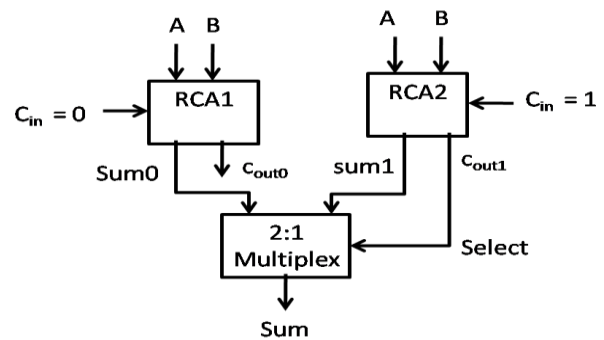


Fig. 5: Modulo adder (MA) for Modulo set [4]

This block is used in PMAC architecture for addition of sum and carry bits at the last tap of filter as shown in figure. The MA shown in figure has two inputs A and B. They are added in parallel with  $C_{in}$  as '0' and '1', which results in Sum0,  $C_{out0}$  and sum1,  $C_{out1}$ , respectively. As shown in figure a 2:1 multiplexer is used to obtain sum with select signal  $C_{out1}$ .

## III. RESULTS AND DISCUSSIONS

In this work, five filters denoted as S2, Y1, G1, L3 and S1 of orders of the filter as 60, 30, 16, 36 and 24 respectively are implemented and used for comparison. The filter specifications and their coefficients used in this work are same as given in Shi and Yu et.al. [9].

Every filter is implemented with 16-bit input X. PMAC and CMAC filter architectures are implemented using gate-level Verilog hardware description language (HDL). For a16-bit input, the proposed filter PMAC is implemented with the filter structure as shown in fig. 2. So, in PMAC filter, the register and RCA blocks are equal to the number of partial products (PP) values. The registers are realized using D-flip-flops in the filter architectures.

Five filters Y1, G1, L3 and S1 are implemented with the proposed PMAC and the conventional CMAC architectures. The UMC 90nm technology library and Cadence RTL compiler have been used for synthesizing the implemented filters. The synthesis results such as area, delay and power values are listed in Table 1. From fig. 1, the critical path delay of CMAC filter consists of partial products (PP), CSA stages and two mod adders (MA). From fig. 2, the critical path delay of PMAC filter consists of partial products (PP), CSA stages, one CSA and one mod adder (MA). The typical filter Y1 is implemented for 16bit operand size as input. Hence it requires the dynamic range of n is 9. Similarly, for S2, G1, L3 and S1 filters, the n values chosen as per their dynamic ranges are 10, 8, 8 and 8.

Table 1: Synthesis Results of PMAC filter

Filter	N	K	Method	Delay(ns)	Area(um <sup>2</sup> )	Power(mW)	PDP	PDP gain (%)
S2	16-bit	10	CMAC	13.214	502367	140.688667	1859.06005	9.421
			PMAC	10.712	570218	165.544188	1683.91548	
Y1		9	CMAC	11.774	166809	45.1312716	531.375585	18.568
			PMAC	9.335	163853	46.35341706	432.709148	
G1		8	CMAC	10.887	78305	20.6104837	224.386328	28.7
			PMAC	9.096	67629	17.58865566	159.986406	
L3		8	CMAC	11.197	168013	45.19004198	505.992889	27.983
			PMAC	9.318	141948	39.10683089	364.397442	
S1		8	CMAC	11.09	103131	27.63517362	306.474069	30.631
			PMAC	8.845	90996	24.03582156	212.596925	

It is observed that the proposed filter PMAC for the filters G1, L3 and S1 with 16-bit input size and operating size at n=8, has achieved smaller values than the conventional filter CMAC in the aspects of area, delay and power. As the operating bit size n increases the area and power increases whereas delay reduces. It can be seen in table 1 for filters Y1 and S2. For the comparing the performance of PMAC architecture with CMAC architecture, PDP and PDP gain values are considered. The PDP and PDP gain values are reviewed in Table 1. PDP gain values among PMAC and CMAC has small difference in case of higher order filters (S2 and Y1). Conversely, the PDP gain is high in PMAC architecture for lower order filters (G1, L3 and S1). The reason is for higher input bits, no of partial products (PP) are increased and thus the no of CSA stages are increased in PMAC architecture for higher order filters.

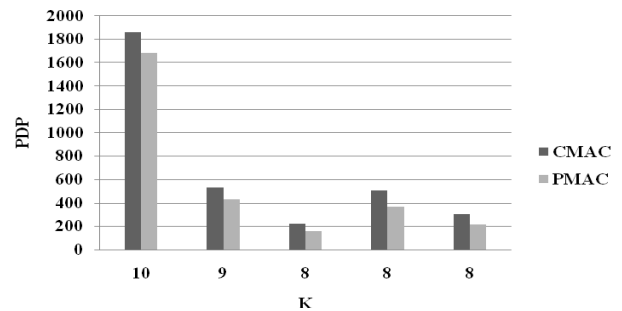


Fig. 6: K vs. PDP

The PDP values are plotted beside K values are shown in fig. 5. For every K value the PDP values are reduced in PMAC when compared with CMAC. In fig. 5, when k=8, PMAC gives improved PDP.

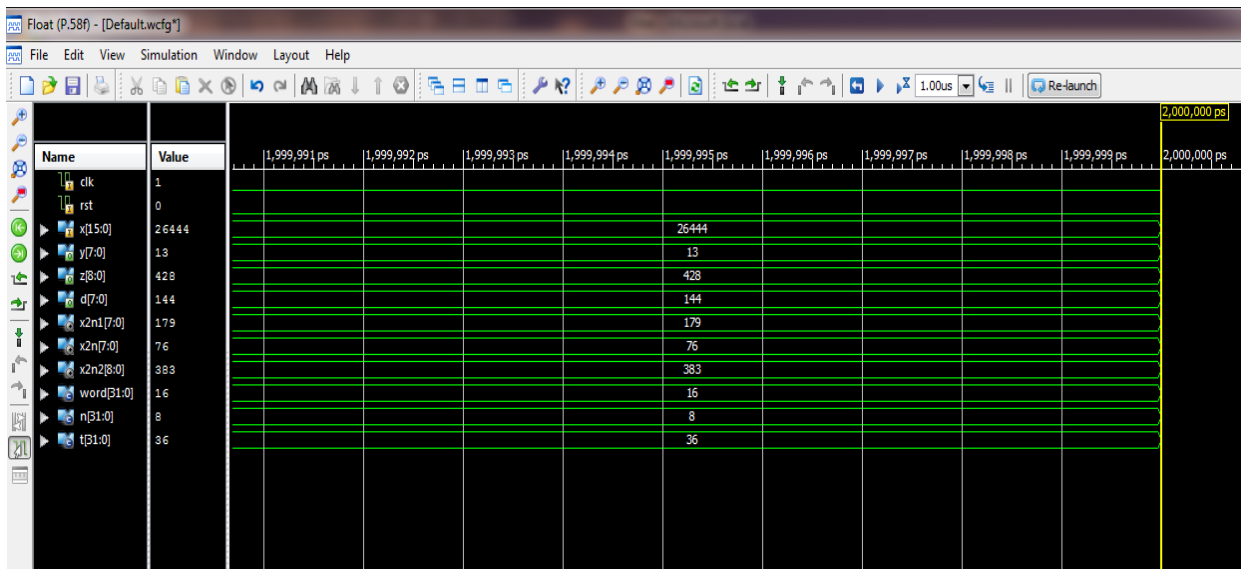


Fig. 7: Simulation Results of PMAC Design

By implementing the PMAC architecture, the simulation results are obtained. These results are shown in fig. 6. In the fig. 6, a binary input of word size 16-bit and clock are given to the PMAC design. The number of filter coefficients of the filter L3 is order of 36. The outputs obtained are modulus of  $\{2^n - 1, 2^n, 2^{n+1} - 1\}$  with n size as 8-bit.

IV. CONCLUSION

In this work, a RNS based multiple-accumulate PMAC is proposed with the  $\{2^n - 1, 2^n, 2^{n+1} - 1\}$  moduli set. PMAC is implemented using 3:2 compressor and a ripple carry adder. With this the computation time has been reduced. The proposed filter PMAC approach for different filter



orders are compared to the conventional CMAC architecture.

## REFERENCES

1. Mohan, Pemmaraju V. Ananda. "RNS-To-Binary Converter for a New Three-Moduli Set  $\{2^{n+1}-1, 2^n, 2^{n-1}\}$ ." IEEE Transactions on Circuits and Systems II: Express Briefs 54, no. 9: 775-779 (2007).
2. Parhi, K. K. (2007). VLSI digital signal processing. India: Wiley.
3. Hiasat, Ahmad A. "New efficient structure for a modular multiplier for RNS." IEEE Transactions on Computers 49, no. 2: 170-174 (2000).
4. Hiasat, Ahmad A., and S. H. Abdel-Aty-Zohdy. "Residue-to-binary arithmetic converter for the moduli set  $(2^{\sup k}, 2^{\sup k-1}, 2^{\sup k-1-1})$ ." IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 45, no. 2 (1998): 204-209
5. Srinivasa Reddy, Kotha, and Subhendu Kumar Sahoo. "An approach for fixed coefficient RNS-based FIR filter." International Journal of Electronics 104, no. 8: 1358-1376 (2017).
6. Wang, Wei, M. N. S. Swamy, M. Omair Ahmad, and Yuke Wang. "A study of the residue-to-binary converters for the three-moduli sets." IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 50, no. 2 (2003): 235-243.
7. Živaljević, Dragana, Negovan Stamenković, and Vidosav Stojanović. "Digital filter implementation based on the RNS with diminished-1 encoded channel." In 2012 35th International Conference on Telecommunications and Signal Processing (TSP), pp. 662-666. IEEE, 2012.
8. Residue Number Systems: Theory and Implementation by A. Omondi (Yonsei University, South Korea) and B. Premkumar (Nanyang Technological University, Singapore)
9. Shi, Dong, and Ya Jun Yu. "Design of linear phase FIR filters with high probability of achieving minimum number of adders." IEEE Transactions on Circuits and Systems I: Regular Papers 58, no. 1 :126-136 (2010).
10. Kotha, Srinivasa Reddy, Sumit Bajaj, and Sahoo Subhendu Kumar. "An lut based rms fir filter implementation for reconfigurable applications." In 18th International Symposium on VLSI Design and Test, pp. 1-6. IEEE, 2014.

## AUTHORS PROFILE



**P Bhargavi** from Vijayawada, Andhra Pradesh, India. She received his B. Tech degree in the year 2017 from PVP Siddhartha, Vijayawada (A.P), India. Currently she is pursuing M. Tech in VLSI & ES from VR Siddhartha engineering college, Vijayawada (A.P), India. Her research interests are digital design implementations.



**K Srinivasa Reddy** He received his Doctorate Degree from BITS-PILANI, Rajasthan. Then he joined as Associate Professor in Vr. Siddhartha Engineering College, Department of Electronics and Communication Engineering, Vijayawada (A.P), India. His area of research interest is VLSI signal processing circuits, arithmetic circuits for DSP and Bio medical applications.



**M Durga Prakash** He received his Doctorate Degree Microelectronics and VLSI at "Innovation Hub for Nano-X Laboratory" in Indian Institute of Technology Hyderabad, India ), in electrical engineering from IIT Hyderabad, India, in 2016. Then he joined as Associate Professor in K L University, Department of Electronics and Communication Engineering, Green Field, Vaddeswaram, Guntur, Andhra Pradesh, India. Currently he is an Associate Professor in VR Siddhartha Engineering College, Vijayawada (A.P), India. His research interests include, among others, Biosensors and miniaturized MEMS devices and VLSI circuits.