

# ANALYSIS OF AN EFFICIENT MODULAR ADDER DESIGNS BASED ON THERMOMETER & ONE-HOT ENCODING

<sup>1</sup>A SOUMYA, <sup>2</sup>B SHINY SUCHARITA, <sup>3</sup>R ALEKYA

<sup>1,2,3</sup>Assistant professor, ECE Dept.St. Martin's Engineering College, Secunderabad

## ABSTRACT

The key feature of the RNS benefitting modern embedded systems and the Internet-of-Thing (IoT) edge devices is its energy efficiency. Modular addition is the most important and frequent operation applied on the components of RNS, including arithmetic units in the channels as well as forward and reverse converters. The small and medium dynamic range requirements of low-power embedded and edge devices make the usage of the thermometer coding (TC) and one-hot coding (OHC) viable, reducing the power consumption and improving the energy efficiency of modulo addition in comparison to regular binary representations. Based on these techniques, this paper presents two new energy-efficient modular adders, which, due to the carry-free internal computations, are also highly performing. This approach greatly simplifies the modular addition operation by eliminating the carry bit propagation during the arithmetic operation encountered when using the conventional base-2 binary code data format. It also enables practical applications of modular arithmetic for signal processing algorithms in a very efficient way. Circuit for implementing the modular arithmetic units using multiplexers and basic logic gates are also described in this paper.

## INTRODUCTION

Finite Impulse Response Filter (FIR) filter is one of the widely used Digital Signal Processing (DSP) operations which interacts its input through the convolution process. In the convolution, the output is obtained by delaying and after that scaling the input sample values. That is why conventionally, multiply and Accumulate (MAC) based technique was generally used for FIR

realization. In this technique, the throughput is  $N$  i.e. the output is available after  $N$  clock cycles (where  $N = \text{filter order}$ ) [1, 2]. Distributed Arithmetic (DA) in its basic form uses bit-serial technique and avoids direct multiplication [1, 2]. The DA uses Lookup Table (LUT) for the easy access of pre-computed values. Also, compared to MAC approach the DA technique requires the number of clock pulses proportional to bit-width of input data to get the output instead of an order of the filter.

One way to increase the efficiency of modular arithmetic units, i.e., modulo adders, subtractors, and multipliers, is by using the one-hot coding (OHC) [7]. The one-hot residue (OHR) has been considered for designing RNS modular arithmetic circuits based on circular shifting. The OHC circuits based on barrel shifters show a power-delay product (PDP) reduction of up to 85% in comparison to the conventional positional encoding, because they significantly reduce the circuit's activity factor. RNSs based on OHC have also been used on DSP applications due to their high speed. Alternatively, there are other types of coding, such as the thermometer coding (TC) that can be applied to enhance the performance of RNS modular arithmetic. The thermometer is the unary coding, in which the number of 1's corresponds to the magnitude of the displayed number. This means that the Hamming distance between numbers represented in TC has a linear relationship to its difference. This type of coding is a subclass of Gloom coding, used in a variety of applications, including neural networks and data compression. Moreover, the TC together with distributed arithmetic can lead to fast implementations of modular.

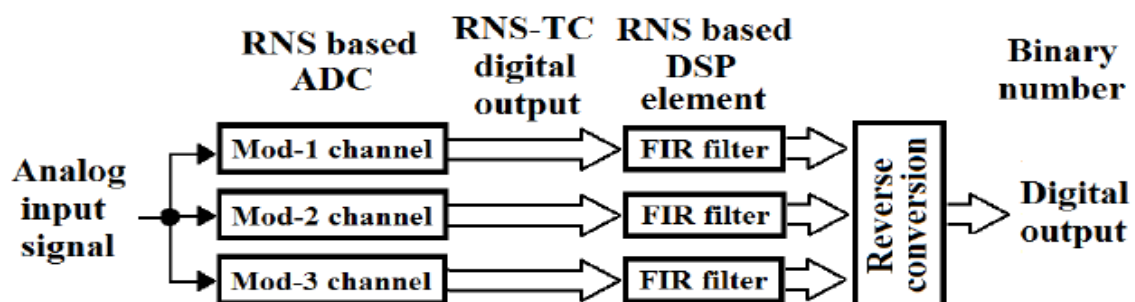


Figure 1: RNS ADC with RNS signal processing

Convention RNS representation of the residues themselves continue to use the base-2 BC format. In

this paper, a base-1 thermometer code (TC) format based RNS method is presented to overcome the

various limitations of the base-2 BC based system as shall be described. The motivation of using the TC based residues introduced here is also due to the availability of a novel RNS based zero-crossing folding ADC [1] which directly converts its input analog signal to digital outputs in the RNS's residues representation that are encoded in TC format as illustrated in Fig. 1. Hence there is no binary to RNS forward conversion involved in generating the RNS based data, which is one of the main practical hindrance for RNS based signal processing in real world applications.

The RNS based ADC hence produces multiple channels of residues each encoded in TC format. Instead of converting these residues to conventional binary representation, it is proposed that the TC modular arithmetic operations presented here are used to process the TC data directly, such as performing the digital filtering using FIR filter as shown in Fig.1, further reducing the overhead of performing the TC to BC encoding process.

It is to emphasize that the TC based method exploits the small dynamic range property of the residues by representing them using the TC format where the number of bits required would not be excessive and hence remains practical. Hence practical implementation would be most feasible by using multiple smaller size moduli, rather than a few medium size moduli.

#### LITERATURE REVIEW

In classification analysis, the dependent variable is frequently influenced not only by ratio scale variables, but also by qualitative (nominal scale) variables. Machine Learning algorithms accept only numerical inputs, hence, it is necessary to encode these categorical variables into numerical values using encoding techniques.

This paper presents a comparative study of seven categorical variable encoding techniques to be used for classification using Artificial Neural Networks on a categorical dataset. The Car Evaluation dataset provided by UCI is used for training. Results show that the data encoded with Sum Coding and Backward Difference Coding technique give highest accuracy as compared to the data pre-processed by rest of the techniques.

This paper presents a new architecture for high-speed Finite Impulse Response (FIR) filter based on Distributed Arithmetic-Residue Number System (DA-RNS) approach. Input is represented in Binary Code (BC) while the output is depicted using Look-up Tables (LUTs) in the proposed method. It is beneficial to use LUTs as it avoids logic complexity due to  $2k$  modulo factor. The Existing approach based on shifters uses Thermometer Code (TC) and One Hot Code (OHC) respectively for input and output representation. Although this technique avoids the complex  $2k$  modulo operation, the number of input TC bits are

increased proportionally to the modulo size which results in degradation of the performance when the actual implementation of the circuit is considered. Residue number systems (RNSs) are efficient alternatives to positional number systems, providing fast and power-efficient computational systems. The key feature of the RNS benefiting modern embedded systems and the Internet-of-Thing (IoT) edge devices is its energy efficiency. Modular addition is the most important and frequent operation applied on the components of RNS, including arithmetic units in the channels as well as forward and reverse converters. The small and medium dynamic range requirements of low-power embedded and edge devices make the usage of the thermometer coding (TC) and one hot coding (OHC) viable, reducing the power consumption and improving the energy efficiency of modulo addition in comparison to regular binary representations. Based on these techniques, this paper presents two new energy-efficient modular adders, which, due to the carry-free internal computations, are also highly performing. The proposed modular adders based on the TC and OHC result in average improvements of 38% and 34.5% for the delay, 27% and 14.5% for the circuit area, 29.5% and 6.3% for energy consumption, and about 54.9% and 44.2% for the area-delay product (ADP), respectively, in comparison with the related state of the art.

This paper presents a novel approach to perform modular arithmetic addition and subtraction using base-1 thermometer code data format for operands corresponding to the residues of the same modulus. Two  $n$ -bit thermometer code operands are first concatenated and logically shifted to produce a normalized  $2n$ -bit thermometer code intermediate sum. Modulo operation is then applied to this  $2n$ -bit intermediate sum to produce an  $n$ -bit datum corresponding to the modular sum of the two input operands. This approach greatly simplifies the modular addition operation by eliminating the carry bit propagation during the arithmetic operation encountered when using the conventional base-2 binary code data format. It also enables practical applications of modular arithmetic for signal processing algorithms in a very efficient way. Circuit for implementing the modular arithmetic units using multiplexers and basic logic gates are also described in this paper.

### 3. CATEGORICAL VARIABLE ENCODING TECHNIQUES

#### 3.1 One Hot Coding

One Hot Coding is the most widely used coding scheme. It compares each level of the categorical variable to a fixed reference level. One hot encoding transforms a single variable with  $n$  observations and  $d$  distinct values, to  $d$  binary variables with  $n$  observations each. Each

observation indicating the presence (1) or absence (0) of the dichotomous binary variable.

### 3.2 Ordinal Coding

In ordinal encoding, an integer is assigned to each category, provided the number of existing categories is known. It does not add any new columns to the data, but implies an order to the variable that may not actually exist. [8]

### 3.3 Sum Coding

Sum coding compares the mean of the dependent variable for a given level to the overall mean of the dependent variable over all the levels. That is, it uses contrasts between each of the first  $k - 1$  levels and level  $k$  in this example, level 1 is compared to all the others, level 2 to all the others, and level 3 to all the others.

### 3.4 Helmert Coding

Helmert Coding compares each level of a categorical variable to the mean of the subsequent levels.

### 3.5 Polynomial Coding

Polynomial coding is a form of trend analysis that looks for linear, quadratic and cubic trends in the categorical variable. This type of coding system should be used only with an ordinal variable in which the levels are equally spaced.

### 3.6 Backward Difference Coding

In this coding system, the mean of the dependent variable for one level of the categorical variable is compared to the mean of the dependent variable for the prior adjacent level.

### 3.7 Binary Coding

In binary coding, first the categories are encoded as ordinal, then those integers are converted into binary code, then the digits from that binary string are split into separate columns.

## PROPOSED MODULAR ADDERS

In this section, new designs for OHR- and TCR-based modular adders are proposed. The proposed hardware structures for modular addition require less circuit area and less delay in comparison to the state of the art.

### A. Thermometer-Based Modular Adder

An important aspect to apply the proposed method to add two modulo  $m$  residues ( $0 \leq A, B < m$ ) represented in TCR is to identify whether  $A + B \geq m$  or not. With this aim, and also for computing the sum, the order of the bits of  $B$  is reversed, which means the rightmost bit becomes the most left bit, and so on. After that, bitwise AND and NOR logic operations are applied to the inputs  $A$  and  $B$ . If any bit of the output of AND gates is 1, then  $A + B$  is equal to or greater than the modulo. More concretely, if exactly one bit of these outputs becomes 1, this means that  $A + B = m$  and the result should be 0. When two bits of the outputs are 1, the result becomes 1, and this process is continued whenever more bits of the output take the value 1. The outputs of the NOR gates are used to

compute the result of the addition when  $A + B < m$ , as it will be observed in Lemma 1. *Lemma 1:* Consider two TCR numbers,  $A$  and  $B$  of  $m - 1$  bits. The condition  $A + B \geq m$  is verified with the bit  $cl$ , and the result of  $A + B \bmod m$  represented with  $m - 1$  bits is computed with the following relations:

$$\text{Sum} = \begin{cases} \text{SUM}_1, & \text{if } cl = 1 (A + B \geq m) \\ \text{SUM}_0, & \text{if } cl = 0 (A + B < m) \end{cases}$$

### RNS and modular arithmetic

RNS has received varying degrees of attention from researchers in the past for implementing computer arithmetic. This number system provides a fundamental methodology for partitioning a large dynamic range system into a number of smaller but independent channels over which computations may be performed in parallel in a carry free manner between the channels. For example, using the moduli set, the decimal number  $X=179$  will be represented using 4,3,8 residue set and the decimal number  $Y=254$  will be represented using 2,6,2 residue set. Arithmetic operation of these two integers can be equally performed in modular arithmetic using

their residues as follows:

$$\begin{aligned} 179 \oplus 254 &\equiv 4,3,8 \oplus 2,6,2 \\ &= 4\oplus 2, 3\oplus 6, 8\oplus 2 \end{aligned}$$

Key advantages in the modular arithmetic are twofold. Firstly is that the dynamic range (DR) of the residue set is

confined to the value of the moduli and hence is normally very much smaller compared to the decimal number that they represent. Secondly, parallel arithmetic operations can be carried out among the pairs of residues independent of each other. Both these mean that the arithmetic circuits required can be of much simpler circuitry and can be much faster when compared to conventional number addition.

### Proposed DA-RNS solution

The problem of modulo  $2^k$  factor is addressed by TCR-OHR (One hot Residue) based DA-RNS solution [2]. In this architecture, for modulo ( $m$ ) TCR based computation,  $(m-1)$  bits are needed for input data representation also OHR use  $m$  bits to encode a particular digit as shown in Fig. 2 [2]. It is evident to state that if modulo size increases then it results in increased input word-length as well as OHR bits. Therefore, the proposed architecture of ref. [2] does not perform well enough for high-speed operation. In our proposed architecture, Binary Code (BC) is used to reduce the input word-length. While, the  $2^k$  modulo factor problem which arises due to the usage of BC at the input of DA-RNS LUT is eliminated by using modulo 'power of two' LUTs. The detailed analysis is given in consequent sections.

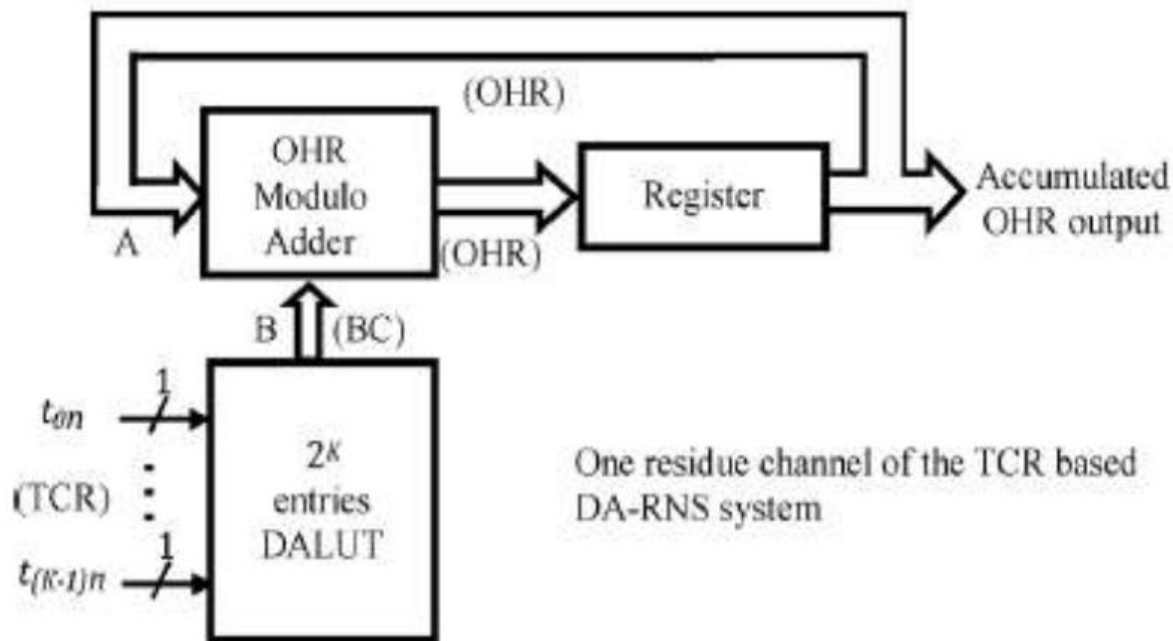


Fig. 2. One of the residue channels of the TCR based DA-RNS system for 1 BAAT operation

In Fig. 3, the architecture for high-speed FIR filter is proposed. Here, the number of  $\emptyset_m$ LUTs are

required is equivalent to the input word-length used.

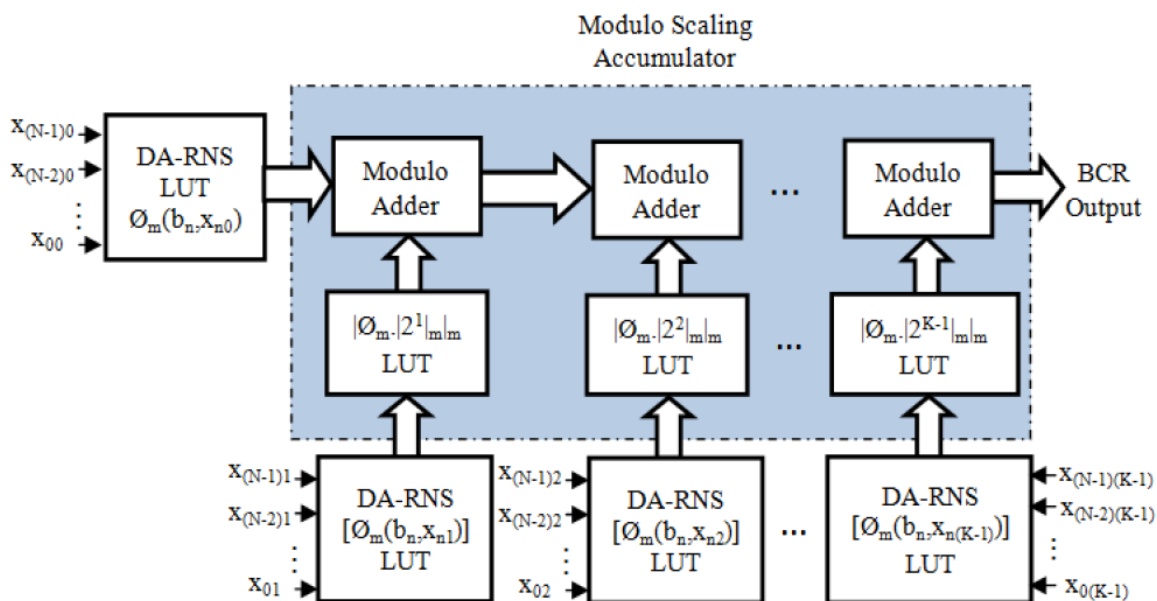


Fig. 3. Proposed high-speed architecture

**CONCLUSION**

In this paper, a new architecture based on DA-RNS approach is presented for high-speed FIR filter. In this architecture, the input and output is presented in binary code due to its shorter word-lengths which results in less hardware. The complexity of  $2^k$  modulo scaling operation involved due to the usage of binary codes at the input is removed by using corresponding ‘power of two’ LUTs. As DA based implementations are always based on LUTs, the inherent advantages of LUTs are maximized by using these at different stages of architecture. A

novel thermometer code based modular arithmetic technique has been described which provide a method to perform modular arithmetic in a highly efficient and high speed manner. Coupled with the new RNS ADC where the output is inherently presented in RNS representation encoded in TC format, the proposed techniques provide a mean to perform signal processing routine using the TC modular arithmetic approach in a very efficient manner.

**REFERENCES**

- [1] A. S. Molahosseini, L. Sousa and C. H. Chang, (Eds.), *Embedded Systems Design with Special Arithmetic and Number Systems*. New York, NY, USA: Springer, 2017.
- [2] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.
- [3] M. Alioto (Ed.), *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. New York, NY, USA: Springer, 2017.
- [4] P. V. A. Mohan, *Residue Number Systems: Theory and Applications*. New York, NY, USA: Springer, 2016.
- [5] M. Anand, M. Saritha, M. Janaki "Performance of efficient CMOS power amplifier for ISM band applications" *International Journal of Innovative Technology and Exploring Engineering*, Vol 9, Issue 2 pp 4579-4584, Dec 2019.
- [6] L. Sousa, S. Antão, and P. Martins, "Combining residue arithmetic to design efficient cryptographic circuits and systems," *IEEE Circuits Syst. Mag.*, vol. 16, no. 4, pp. 6–32, 4th Quart., 2016.
- [7] M. Labafniya and M. Eshghi, "An efficient adder/subtractor circuit for one-hot residue number system," in *Proc. Intl. Conf. Electron. Devices, Syst. Appl. (ICEDSA)*, Apr. 2010, pp. 121–124.
- [8] M. Hosseinzadeh, S. Jafarali Jassbi, and K. Navi, "A novel multiple valued logic OHRNS adder circuit for modulo (RN-1)," in *Proc. 4<sup>th</sup> Int. Conf. Adv. Eng. Comput. Appl. Sci.*, Aug. 2010, pp. 166–170.