

A Robust Redundant Residue Representation in Residue Number System with Moduli Set

$$\{r^n - 2, r^n - 1, r^n\}$$

Hossein Khademolhosseini and Mehdi Hosseinzadeh

Abstract—The residue number system (RNS), due to its properties, is used in applications in which high performance computation is needed. The carry free nature, which makes the arithmetic, carry bounded as well as the paralleling facility is the reason of its capability of high speed rendering. Since carry is not propagated between the moduli in this system, the performance is only restricted by the speed of the operations in each modulus. In this paper a novel method of number representation by use of redundancy is suggested in which $\{r^n - 2, r^n - 1, r^n\}$ is the reference moduli set where $r = 2k + 1$ and $k = 1, 2, 3, \dots$. This method achieves fast computations and conversions and makes the circuits of them much simpler.

Keywords—Binary to RNS converter, Carry save adder, Computer arithmetic, Residue number system.

I. INTRODUCTION

THE residue number system (RNS) is a system for number representation. During the past decades, this system has received unprecedented attention. In this system, residues of the original number with respect to a moduli set are represented instead of the original number itself. Thus the number will be split in-to some smaller numbers which are independent and operations can be accomplished on them separately and concurrently which makes the computations simpler and much faster.

RNS can support carry limited and high speed arithmetic and error detection as well as error correction applications [1]. Because of all of these unique features, its usage in variety of fields is growing rapidly. Some applications of RNS are digital signal processing (DSP), digital image processing, digital filters, fast Fourier transform (FFT) computations as well as digital communications [2]–[10].

The moduli set should be selected such that its members are pairwise relatively prime [12]. If the moduli set $\{m_1, m_2, \dots, m_j\}$ is selected, number X within the dynamic range M is represented by its corresponding residues as a j -tuple set (x_1, x_2, \dots, x_j) such that:

Hossein Khademolhosseini and Mehdi Hosseinzadeh are with the Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. (e-mails: h.khademolhosseini@srbiau.ac.ir; hosseinzadeh@srbiau.ac.ir).

$$M = m_1 \times m_2 \times \dots \times m_j = \prod_{i=1}^j m_i \quad (1)$$

$$X = m_i q_i + x_i \quad 0 \leq x_i \leq m_i \quad i = 1, 2, \dots, j \quad (2)$$

In (2) q_i is a non-negative integer [11], [12].

Introducing several moduli sets such as $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$, $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$, $\{2^{n+1} - 1, 2^n, 2^n - 1\}$, $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$, ... have been brought about by recent works in RNS [13]–[15]. In [16] we used the moduli set $\{3^n - 2, 3^n - 1, 3^n\}$, however $\{r^n - 2, r^n - 1, r^n\}$ where $r = 2k + 1$ and $k = 1, 2, 3, \dots$ is used as our reference moduli set in this paper which makes the method more general compared to $\{3^n - 2, 3^n - 1, 3^n\}$.

The rest of the paper is organized as follows. Section 2 provides some explanations about the redundant residue number system (RRNS). In section 3 the proposed representation method is introduced. Section 4 shows the methodology to find the RRNS equivalent of a number in moduli $\{r^n - 2, r^n - 1, r^n\}$ and the related circuits. Section 5 demonstrates the adders in this moduli set. In section 6 comparison of our proposed method with another method is provided. Finally, section 7 offers the conclusions.

II. REDUNDANCY IN COMPUTER ARITHMETIC AND REDUNDANT RNS

Three benefits are provided while redundancy is applied to computer arithmetic. The first benefit is to improve reliability, the second one is to increase speed of computation and the last one is to provide structural flexibility [17]. To implement the first one, hardware redundancy and/or redundant arithmetic codes are used. For the second and the third cases several methods have been introduced. One is utilization of faster elements like carry save adders (CSA). Redundancy in representation is another robust method which has been used a lot [17].

In RNS three types of redundancy is considered [11]:

- 1) Extra residues (moduli) which is appropriate when the goal is production of a desired dynamic range.
- 2) Redundant representation of conventional residues which is appropriate when faster arithmetic is needed.

3) Redundant residues with a range exceeding the m values in $[a, m+a)$ for some a (in mod- m residues).

Researchers have focused on the first type and the others have got less consideration. In this paper we use a mixture of the second and the third types.

III. REDUNDANT REPRESENTATION OF RESIDUES IN THE PROPOSED APPROACH

In this paper we use the high radix moduli set $\{r^n - 2, r^n - 1, r^n\}$ where $r = 2k + 1$ and $k = 1, 2, 3, \dots$ which has been extensively investigated in [18]. In order to represent numbers in the conventional system, the multiple-valued logic (MVL) has been employed. In comparison with the binary logic, MVL is capable of representing larger numbers with fewer positions.

Although n digits are adequate to represent residues in moduli r^n , $r^n - 1$ and $r^n - 2$, we use $2n$ digits in our method, which means that we benefit from n -bit redundancy. CSA is our basic element to implement the circuits. We define the partial *Sum* and the partial *Carry* of the last CSA in circuits as the final outputs which show the residue, so the produced residue has $2n$ digits instead of n digits this way (n -digit *Sum* and n -digit *Carry*). In this method to represent a residue, a class of equivalence exists which contains all the possible representations of the residue with n -digit *Sum* and n -digit *Carry* with a predefined restriction. (*Sum* and *Carry* are the outputs of CSA.) The *Sum* digits belong to $[0, r - 1]$ however the *Carry*'s interval is restricted to $[0, 6]$ except the least significant digit of *Carry* which could be up to 12.

IV. CONVERTERS FOR MODULI SET $\{r^n - 2, r^n - 1, r^n\}$ WITH REDUNDANCY

$3n$ digits are needed to represent numbers with the moduli set $\{r^n - 2, r^n - 1, r^n\}$ in the conventional system, because all possible integers in the dynamic range must be covered uniquely. It is proved as follows.

Theorem 1: $3n$ digits cover the dynamic range of the moduli set $\{r^n - 2, r^n - 1, r^n\}$.

Proof: We prove that the following inequality is correct so the theorem will be proved consequently.

$$\begin{aligned} (r^n)(r^n - 1)(r^n - 2) < r^{3n} &\Rightarrow r^{3n} - 3r^{2n} + 2r^n < r^{3n} \\ &\Rightarrow 1 - 3r^{-n} + 2r^{-2n} < 1 \\ &\Rightarrow -3r^{-n} + 2r^{-2n} < 0 \\ &\Rightarrow 2r^{-2n} < 3r^{-n} \Rightarrow 3 > 2r^{-n} \end{aligned} \quad (3)$$

In the last inequality, the case where the right side is the largest occurs when $r = 3$ and $n = 1$.

So we have

$$3 > 2 \times \left(\frac{1}{3}\right)^1 \Rightarrow 3 > \frac{2}{3} \quad (4)$$

The last inequality is always correct so the theorem is proved.

Thus number X is represented as follows:

$$X = \underbrace{a_{3n-1} \dots a_{2n+1}}_A a_{2n} \underbrace{a_{2n-1} \dots a_{n+1}}_B a_n \underbrace{a_{n-1} \dots a_1 a_0}_C \quad (5)$$

According to what was discussed formerly, A , B and C are n -digit. Residues of X with respect to the moduli will be obtained as follows.

A. Residue in Modulus r^n

It is apparent that the residue in this modulus is the number which is shown by the right most digits of X (C).

$$[X] \bmod r^n = a_{n-1} \dots a_1 a_0 = C \quad (6)$$

We should only convert the residue to the adopted redundant representation in our method. This will be achieved by resetting all the n digits of *Carry* to zero. Thus the representation will be as the following:

$$R_1 = \underbrace{00 \dots 0}_{n\text{-digit Carry}} \underbrace{a_{n-1} \dots a_1 a_0}_{n\text{-digit Sum}} \quad (7)$$

B. Residue in Modulus $r^n - 1$

This residue will be obtained as follows:

$$\begin{aligned} R_2 &= [a_{3n-1} \dots a_{2n+1} a_{2n} a_{2n-1} \dots a_{n+1} a_n a_{n-1} \dots a_1 a_0] \bmod r^n - 1 \\ &= [(a_{3n-1} \dots a_{2n+1} a_{2n} a_{2n-1} \dots a_{n+1} a_n) \times (r^n - 1 + 1) \\ &\quad + (a_{n-1} \dots a_1 a_0)] \bmod r^n - 1 \\ &= [(a_{3n-1} \dots a_{2n+1} a_{2n}) \times (r^n - 1 + 1) \\ &\quad + (a_{2n-1} \dots a_{n+1} a_n) + (a_{n-1} \dots a_1 a_0)] \bmod r^n - 1 \\ &= [(a_{3n-1} \dots a_{2n+1} a_{2n}) + (a_{2n-1} \dots a_{n+1} a_n) \\ &\quad + (a_{n-1} \dots a_1 a_0)] \bmod r^n - 1 = (A + B + C) \bmod r^n - 1 \end{aligned} \quad (8)$$

Equation (8) shows that firstly we should calculate $A + B + C$ and then compute the residue of the obtained sum modulo $r^n - 1$.

To verify the correct design, different possible values of A , B and C have been analyzed. The dynamic range of the moduli set is $r^{3n} - 3r^{2n} + 2r^n$ so the representable numbers belong to $[0, r^{3n} - 3r^{2n} + 2r^n - 1]$. Hence the largest representable number is $r^{3n} - 3r^{2n} + 2r^n - 1$. This number is computed as the following:

$$r^{3n} = \left(\underbrace{1\ 00\dots 0}_{3n \text{ digits}} \right)_r \quad (9)$$

$$3r^{2n} = \left(\underbrace{3\ 00\dots 0}_{2n \text{ digits}} \right)_r \quad (10)$$

$$2r^n = \left(\underbrace{2\ 00\dots 0}_n \right)_r \quad (11)$$

Therefore we have

$$\begin{aligned} r^{3n} - 3r^{2n} + 2r^n - 1 &= \left(\underbrace{100\dots 0}_{3n+1 \text{ digits}} - \underbrace{00\dots 0300\dots 0}_{3n+1 \text{ digits}} + \underbrace{00\dots 0200\dots 0}_{3n+1 \text{ digits}} - 1 \right)_r \quad (12) \\ &= \left(\underbrace{r-1r-1\dots r-1r-1}_{n\text{-digit } A} \underbrace{3\ 00\dots 01}_{n\text{-digit } B} \underbrace{r-1r-1\dots r-1}_{n\text{-digit } C} - 1 \right)_r \end{aligned}$$

In (12), summation of A , B and C , outputs the carry-out 1. Now consider the following case within the dynamic range which dictates how the circuits can be designed:

$$\left(\underbrace{r-1r-1\dots r-1r-4}_{n\text{-digit } A} \underbrace{r-1r-1\dots r-1}_{n\text{-digit } B} \underbrace{r-1r-1\dots r-1}_{n\text{-digit } C} \right)_r \quad (13)$$

Where $r = 2k + 1$ and $k = 2, 3, \dots$
 And for $r = 3$:

$$\left(\underbrace{22\dots 212}_{n \text{ digits}} \underbrace{22\dots 2}_{n \text{ digits}} \underbrace{22\dots 2}_{n \text{ digits}} \right) \quad (14)$$

In (14), when calculating $A+B+C$, carry-out is 2. Since in this modulus each unit of carry-out (*Carry*) has the value of 1, to take this value into account carry save adder with end around carry (CSA with EAC) is utilized. The popular full adder (FA) cell in binary is 3-input and it is 4-input in ternary therefore the CSA in the former is 3-to-2 and in the latter is 4-to-2 [19]. The CSA in radix r is described in theorem 2.

Theorem 2: CSA in radix r may have up to $(r + 1)$ inputs.

Proof: The full adder basic cell in radix r has two single-digit outputs (*Carry* and *Sum*) and the maximum value of each output is $(r - 1)$. Hence the maximum representable number in the output of a cell in radix r is $[(r - 1) + r(r - 1)]$. Assume that all inputs are $(r - 1)$. The sum of these inputs must be represented in the output so the maximum number of inputs (l) is determined by the largest representable number in the output.

Therefore we have

$$\begin{aligned} l \times (r - 1) &= (r - 1) + r(r - 1) = (r - 1)(r + 1) \\ \Rightarrow l &= (r + 1) \end{aligned} \quad (15)$$

It is proved that CSA in radix r may have up to $(r + 1)$ inputs, as well.

Each cell of the CSAs which are used hereafter may take $(r + 1)$ digits as inputs and returns two digits (outputs) as *Sum* and *Carry*. If less than $(r + 1)$ inputs are needed, we will apply all zeros to the inputs which are not connected, however these unused inputs are not illustrated in the future figures. Using a CSA with EAC in implementation, the outputs show the correct result in the all cases. Fig.1 shows the converter modulo $r^n - 1$.

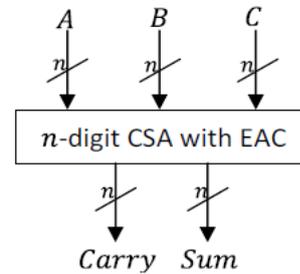


Fig.1 Converter for modulus $r^n - 1$

C. Residue in Modulus $r^n - 2$

To obtain the residue in this modulus the following procedure is used:

$$\begin{aligned} R_3 &= [a_{3n-1} \dots a_{2n+1} a_{2n} a_{2n-1} \dots a_{n+1} a_n a_{n-1} \dots a_1 a_0] \bmod r^n - 2 \\ &= [(a_{3n-1} \dots a_{2n+1} a_{2n} a_{2n-1} \dots a_{n+1} a_n) \times (r^n - 2 + 2) \\ &\quad + (a_{n-1} \dots a_1 a_0)] \bmod r^n - 2 = [2 \times (a_{3n-1} \dots a_{2n+1} a_{2n}) \\ &\quad \times (r^n - 2 + 2) + 2 \times (a_{2n-1} \dots a_{n+1} a_n) \\ &\quad + (a_{n-1} \dots a_1 a_0)] \bmod r^n - 2 \\ &= [4 \times (a_{3n-1} \dots a_{2n+1} a_{2n}) + 2 \times (a_{2n-1} \dots a_{n+1} a_n) \\ &\quad + (a_{n-1} \dots a_1 a_0)] \bmod r^n - 2 = (4A + 2B + C) \bmod r^n - 2 \end{aligned} \quad (16)$$

While radix r CSAs are used, the maximum generated carry-out in calculating $4A + 2B + C$, is 6. In this modulus each unit of carry-out has the value of 2 therefore carry-out should be added to the resultant *Sum* and *Carry* at the next levels. In this modulus we should design the converter for five cases separately. In radix $r = 3$ and $r = 5$ the maximum number of inputs are 4 and 6, respectively. The converter modulo $3^n - 2$ where $n = 2, 3$ is demonstrated in Fig. 2 and this converter will be simplified for $n \geq 4$ as Fig. 3. Since for $n \geq 4$ the third level CSA does not generate carry-out, this simplifications will be applied and carry-out of the second level will be placed in the least significant position of the final *Carry*. As it is clear,

for $n = 1$, the residue is always 0 in this modulus. Since the number of inputs is limited, two parallel CSAs are employed at the first level.

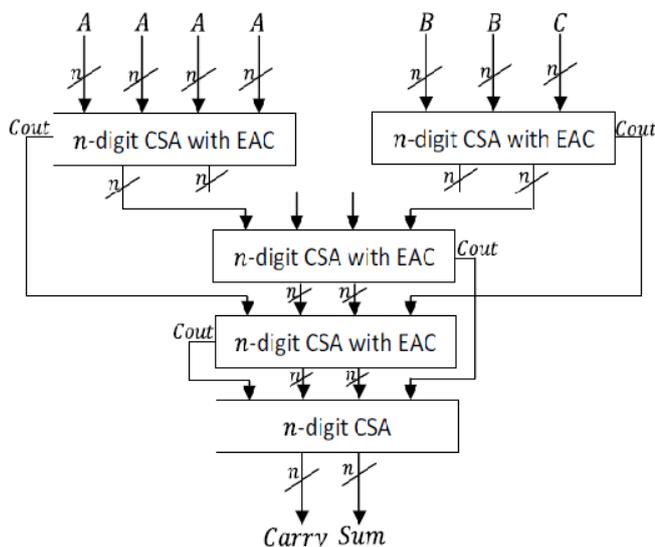


Fig. 2 Converter for modulus $3^n - 2$ where $n = 2, 3$

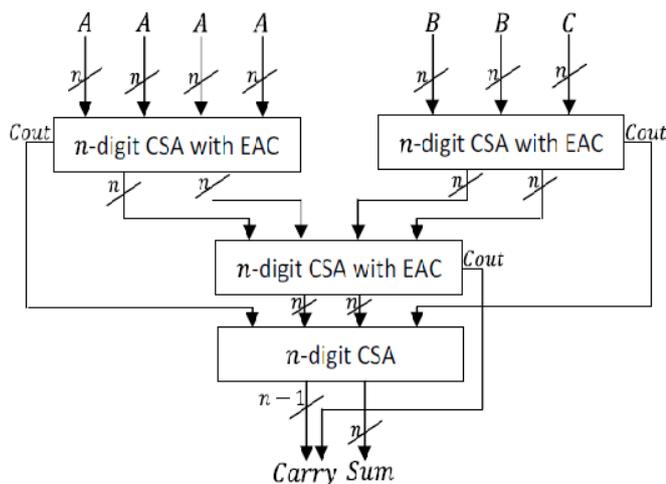


Fig. 3 Converter for modulus $3^n - 2$ where $n \geq 4$

Fig. 4 illustrates the converter in modulus $5^n - 2$. Since each digit of the last Carry is up to 2 (obtained by analysis), only one radix r full adder (FA) is utilized to add the carry-out. (*Cout* and the least significant digit are the inputs.) In this modulus the *Sum* of the FA, will be placed in the least significant position of the final Carry. As the previous converter this conversion is done in 3 levels however the hardware needed is reduced. It can clearly be perceived that converters are getting simpler. This fact will be more apparent in the future figures which show the converters in higher radices.

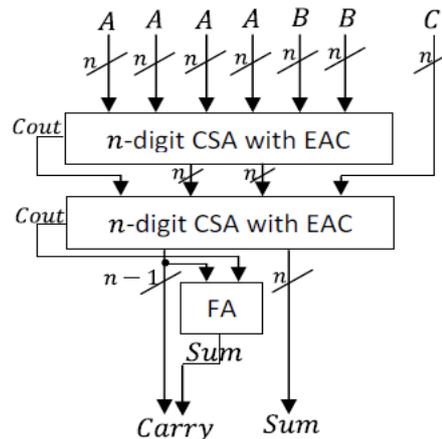


Fig. 4 Converter for modulus $5^n - 2$

For $r = 7, 9, 11$ the covertor modulo $r^n - 2$ is as Fig. 5.

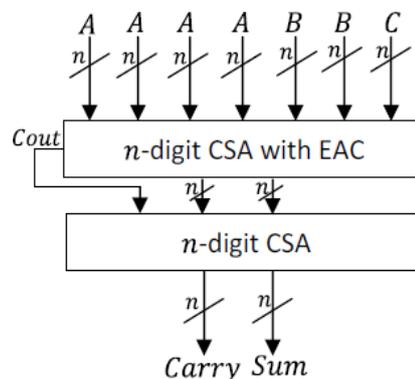


Fig. 5 Converter for modulus $r^n - 2$ where $r = 7, 9, 11$

For $r \geq 13$ where $r = 2k + 1$ the converter in modulus $r^n - 2$ is realized as Fig. 6

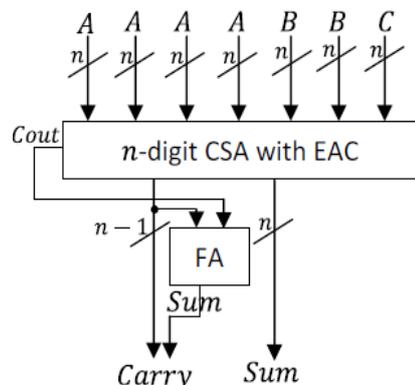


Fig. 6 Converter for modulus $r^n - 2$ where $r \geq 13$

V. RNS ADDERS WITH MODULI SET $\{r^n - 2, r^n - 1, r^n\}$

When number X is converted to the RRNS form as result of

above mentioned equations, arithmetic operations may be carried out. Recalling that the produced residue has a *Sum* as well as an accompanying *Carry* and that in RNS each modulus has a separate adder, the adders modulo r^n , $r^n - 1$ and $r^n - 2$ will be designed as follows. Assume that the numbers to be added are *D* and *E* and the resultant number is *F*.

A. Modulo r^n Adder

According to what explained previously, in our method, *Carries* of the residues modulo r^n is all zeros. Hence we do not use *Carries* to perform add operation and we only add the *Sums*. Since *Carry* digits are not allowed to be more than 6 (The least significant digit is not allowed to be more than 12.), the adder is designed to be as Fig.7.

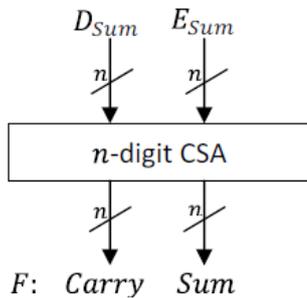


Fig. 7 Modulo r^n adder

B. Modulo $r^n - 1$ Adder

In a similar manner like the previous adder, the adder in this modulus is realized as Fig. 8. In this adder both *Sums* and *Carries* are used as the inputs of the CSA.

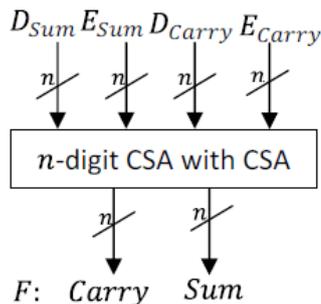


Fig. 8 Modulo $r^n - 1$ adder

C. Modulo $r^n - 2$ Adder

The maximum carry-out of the first CSA in the modulo $3^n - 2$ adder is 2, however it is 1 for all other cases. Accordingly Fig. 9 and Fig. 10 show the adders in modulus $3^n - 2$ and other moduli of the type $r^n - 2$, respectively. *Cout* and the least significant digit of *Carry* are the inputs of the FA in Fig. 10.

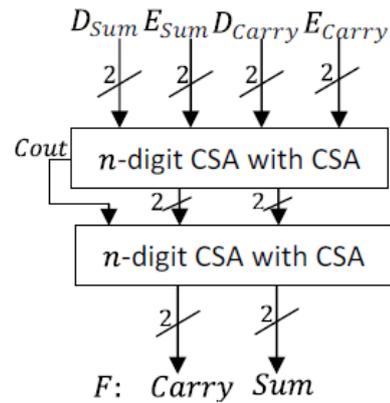


Fig. 9 Modulo $3^n - 2$ adder

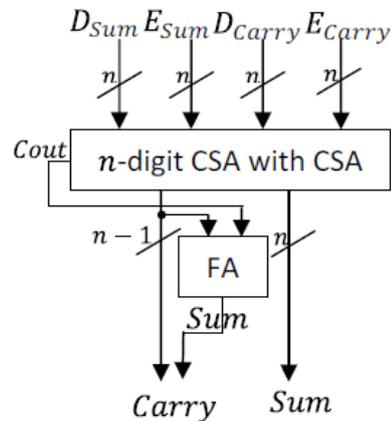


Fig. 10 Modulo $r^n - 1$ adder where $r \neq 3$ and $n \neq 2$

VI. COMPARISONS

Since the moduli set $\{r^n - 2, r^n - 1, r^n\}$ has been introduced and fully described in [18], we compare our proposed design with [18]. These comparisons are presented in the following tables. Table I shows the comparison of delay in conversion and Table II shows the comparison of delay in addition. The tables demonstrate improvements in terms of speed in circuits. The circuits of the forward converters and the adders are able to perform n -independently. (n is the exponent of r in r^n modulus.) Hence a large reduction in the delays of the circuits occurs which is more apparent in large radices. Moreover, in comparison with the circuits presented in [18], for large radices the corresponding circuits in our method are much simpler.

TABLE I
 COMPARISON OF DELAY IN CONVERSION

Maximum delay in the proposed method	Maximum delay in [18]	Modulus
$4t_{FA3}$	$(n+1)t_{FA3} + t_{MUX (3 \times 1)}$	$r^n - 2$ ($r = 3$)
$3t_{FA5}$	$(n+1)t_{FA5} + t_{MUX (3 \times 1)}$	$r^n - 2$ ($r = 5$)
$2t_{FAr}$	$(n+1)t_{FAr} + t_{MUX (3 \times 1)}$	$r^n - 2$ ($r \geq 7$)
t_{FAr}	$nt_{FAr} + t_{MUX (3 \times 1)}$	$r^n - 1$
0	0	r^n

TABLE II
 COMPARISON OF DELAY IN ADDITION

Maximum delay in the proposed method	Maximum delay in [18]	Modulus
$2t_{FA3}$	$nt_{FA3} + t_{MUX (2 \times 1)}$	$r^n - 2$ ($r = 3$)
$2t_{FA5}$	$nt_{FA5} + t_{MUX (2 \times 1)}$	$r^n - 2$ ($r = 5$)
$2t_{FAr}$	$nt_{FAr} + t_{MUX (2 \times 1)}$	$r^n - 2$ ($r \geq 7$)
t_{FAr}	$nt_{FAr} + t_{MUX (2 \times 1)}$	$r^n - 1$
t_{FAr}	nt_{FAr}	r^n

VII. CONCLUSION

In this paper we used a high radix moduli set. In order to represent the residues, a novel redundant method is employed. The use of the redundant method leads to efficient designs in terms of delays of the related circuits. At the expense of using n redundant digits, a large reduction in the delays of the conversion and computational circuits in residue arithmetic will be achieved. The delays of the presented circuits using the general moduli set $\{r^n - 2, r^n - 1, r^n\}$ does not depend on n , the exponent of the moduli. Hence the larger moduli which generate larger residues will have short delays in their circuits and furthermore, for $r \geq 7$ the delays are constant multiples of the delays of the basic full adder cells and this evidently shows a development in the design of RNS circuits.

REFERENCES

- [1] Barsi, F.; Maestrini, P.; , "Error Detection and Correction by Product Codes in Residue Number Systems," *Computers, IEEE Transactions on* , vol.C-23, no.9, pp. 915- 924, Sept. 1974.
- [2] Di Claudio, E.D.; Piazza, F.; Orlandi, G.; , "Fast combinatorial RNS processors for DSP applications," *Computers, IEEE Transactions on* , vol.44, no.5, pp.624-633, May 1995.
- [3] Banerjee, S.; Sinha, A.; , "A reconfigurable Digital Signal Processor using residue number system," *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on* , vol., no., pp.405-408, 10-13 May 2010.
- [4] Wei Wang; Swamy, M.N.S.; Ahmad, M.O.; , "RNS application for digital image processing," *System-on-Chip for Real-Time Applications, 2004.Proceedings. 4th IEEE International Workshop on* , vol., no., pp. 77- 80, 19-21 July 2004.
- [5] Taleshmekaeli, D.K.; Mousavi, A.; , "The use of Residue Number System for improving the Digital Image Processing," *Signal Processing (ICSP), 2010 IEEE 10th International Conference on* , vol., no., pp.775-780, 24-28 Oct. 2010.
- [6] Mirshekari, Ali; Mosleh, Mohammad; , "Hardware implementation of a fast FIR filter with residue number system," *Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on* , vol.2, no., pp.312-315, 30-31 May 2010.
- [7] Shahana, T.K.; James, R.K.; Jose, B.R.; Jacob, K.P.; Sasi, S.; , "Performance analysis of FIR digital filter design: RNS versus traditional," *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on* , vol., no., pp.1-5, 17-19 Oct. 2007.
- [8] Taylor, F. J.; Papadourakis, G.; Skavantzios, A.; Stouraitis, A.; , "A Radix-4 FFT Using Complex RNS Arithmetic," *Computers, IEEE Transactions on* , vol.C-34, no.6, pp.573-576, June 1985.
- [9] Ben-Dau Tseng; Jullien, G.A.; Miller, W.C.; , "Implementation of FFT Structures Using the Residue Number System," *Computers, IEEE Transactions on* , vol.C-28, no.11, pp.831-845, Nov. 1979.
- [10] Panella, M.; Martinelli, G.; , "RNS quasi-chaotic generator for self-correcting secure communication," *Electronics Letters* , vol.37, no.5, pp.325-327, 1 Mar 2001.
- [11] Parhami, B.; , "RNS representations with redundant residues," *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on* , vol.2, no., pp.1651-1655 vol.2, 2001.
- [12] Garner, Harvey L.; , "The Residue Number System," *Electronic Computers, IRE Transactions on* , vol.EC-8, no.2, pp.140-147, June 1959.
- [13] Molahosseini, A.S.; Navi, K.; Dadkhah, C.; Kavehei, O.; Timarchi, S.; , "Efficient Reverse Converter Designs for the New 4-Moduli Sets $\{2^n-1, 2^n, 2^{n+1}, 2^{2n+1}-1\}$ and $\{2^n-1, 2^{n+1}, 2^{2n}, 2^{2n+1}\}$ Based on New CRTs," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.57, no.4, pp.823- 835, April 2010.
- [14] Mohan, P.V.A.; , "RNS-To-Binary Converter for a New Three-Moduli Set $\{2^{n+1}-1, 2^n, 2^n-1\}$," *Circuits and Systems II: Express Briefs, IEEE Transactions on* , vol.54, no.9, pp.775-779, Sept. 2007.
- [15] Cao, B.; Chang, C.-H.; Srikanthan, T.; , "A Residue-to-Binary Converter for a New Five-Moduli Set," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.54, no.5, pp.1041-1049, May 2007.
- [16] Khademolhosseini, H.; Roohi, A.; , "A new redundant method on representing numbers with moduli set $\{3^n, 3^{n-1}, 3^{n-2}\}$," *Computer, Communication and Electrical Technology (ICCET), 2011 International Conference on* , vol., no., pp.163-166, 18-19 March 2011.
- [17] Atkins, D.E.; , "Introduction to the Role of Redundancy in Computer Arithmetic," *Computer* , vol.8, no.6, pp.74-77, June 1975.
- [18] Hosseinzadeh, M.; Navi, K.; Gorgin, S.; , "A New Moduli Set for Residue Number System: $\{r^n-2, r^n-1, r^n\}$," *Electrical Engineering, 2007. ICEE '07. International Conference on* , vol., no., pp.1-6, 11-12 April 2007.
- [19] Abdallah, M.; Skavantzios, A.; , "On MultiModuli residue number systems with moduli of forms r^a, r^b-1, r^c+1 ," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.52, no.7, pp. 1253- 1266, July 2005.