Improving the Delay of Residue-to-Binary Converter for a Four-Moduli Set

Amir SABBAGH MOLAHOSSEINI
Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran
sabbagh@iauk.ac.ir

***Abstract*—The residue number system (RNS) is an unconventional number system which can be used to achieve high-performance hardware implementations of special-purpose computation systems such as digital signal processors. The moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ has been recently suggested for RNS to provide large dynamic range with low-complexity, and enhancing the speed of internal RNS arithmetic circuits. But, the residue-to-binary converter of this moduli set relies on high conversion delay. In this paper, a new residue-to-binary converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ using an adder-based implementation of new Chinese remainder theorem-1 (CRT-I) is presented. The proposed converter is considerably faster than the original residue-to-binary converter of the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$; resulting in decreasing the total delay of the RNS system.**

***Index Terms*—Residue Number System (RNS), residue-to-binary converter, digital circuits, computer architecture, high-speed computer arithmetic.**

I. INTRODUCTION

The residue number system (RNS) is a carry-free number system with capability of providing parallel arithmetic. The potentiality of RNS to perform addition, subtraction and multiplication without carry-propagation between residue digits, makes it suitable for reducing the power dissipation in high-performance digital computing systems [1]-[4]. The RNS has a wide range of applications; especially in digital signal processing computations such as FIR filters [5]-[8].

Each RNS system is based on moduli set, and includes binary-to-residue converter, arithmetic unit which includes parallel modulo arithmetic circuits, and residue-to-binary converter [9], [10]. Among all of these, designing residue-to-binary converter is the most complex process which attracts researchers for many decades. The performance of the residue-to-binary converter is drastically depends on the moduli set as well as the selected conversion algorithm. As a result, many special moduli sets and conversion algorithms have been recommended for constructing residue-to-binary converters.

The most prominent moduli set is $\{2^n-1, 2^n, 2^n+1\}$ [11], [12]. Moreover, modulo (2^n+1)-free 3-moduli sets $\{2^{n-1}-1, 2^n-1, 2^n\}$ [13], [14], $\{2^n-1, 2^n, 2^{n+1}-1\}$ [15], [16] have been introduced. Recently, it is shown that non-traditional 3-moduli sets such as $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ and $\{2^{2n}, 2^n-1, 2^{n-1}-1\}$ [17] could significantly reduce the complexity of the residue-to-binary converter. However, due to the increasing demand of some applications for larger parallelism and dynamic range, the Four-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{n+1}\pm1\}$ [18]-[20] and $\{2^n-3, 2^n+1, 2^n-1, 2^n+3\}$ [21], [22] were used as balanced moduli sets for RNS. However, inefficient multiplicative inverses of these sets resulted in complex converter architectures. Due to this, the well-formed moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ [23], $\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$ [24], $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$ [25] have been considered in the recent years. Although these sets do not include balanced moduli, their multiplicative inverses are very simple. Moreover, some five-moduli sets with popular moduli have been also investigated in the recent years [26], [27].

It is expected that the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ plays an important role in designing efficient RNS systems for using in the high-performance DSP and real-time systems with high computation load, due to its particular features. However, the residue-to-binary converter of this moduli set that is presented in [25], has high conversion delay, mainly because of using new Chinese remainder theorem-2 (CRT-II), and its two-level hardware architecture.

In this paper, a fast one-level hardware design for the residue-to-binary converter of the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ through the use of new Chinese remainder theorem-1 (CRT-I) is introduced. This new converter considerably reduces the delay of residue-to-binary conversion, compared to the converter of [25].

In the next, a brief introduction to RNS and CRT-I is presented in Section II. Section III describes the CRT-I-based conversion equations for the four-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$. Hardware implementation of the residue-to-binary conversion equations as well as complexity computation is presented in Section IV, and Section V is conclusion.

II. BACKGROUND

The RNS [1]-[3] is based on a moduli set $\{P_1, P_2, \ldots, P_n\}$ which consists of pair-wise relatively prime numbers. The dynamic range is defined as $M=P_1P_2\ldots P_n$, which is refer to the interval of integer numbers that can be represented in RNS. A weighted number $X<M$ has a unique representation in RNS as $(x_1, x_2, \ldots, x_n)$ where

$$x_i = X \bmod P_i = \left|X\right|_{P_i}, \quad 0 \le x_i < P_i . \tag{1}$$

In the RNS with the 4-moduli set $\{P_1, P_2, P_3, P_4\}$, and based on CRT-I algorithm [25],[28] the RNS number (x_1, x_2, x_3, x_4) can be converted into its corresponding weighted number by

$$X = x_1 + P_1 \left| \begin{matrix} k_1(x_2 - x_1) + k_2P_2(x_3 - x_2) \\ + k_3P_2P_3(x_4 - x_3) \end{matrix} \right|_{P_2P_3P_4} \tag{2}$$

Where

$$\left|k_1 \times P_1\right|_{P_2P_3P_4} = 1 \tag{3}$$

$$\left|k_2 \times P_1 \times P_2\right|_{P_3P_4} = 1 \tag{4}$$

Digital Object Identifier 10.4316/AECE.2011.02006

$$\left|k_3 \times P_1 \times P_2 \times P_3\right|_{P_4} = 1 . \quad (5)$$

The variable k_i's are called multiplicative inverses. The simple form of the multiplicative inverses (powers of two plus/minus one) could greatly reduce the hardware complexity of the residue-to-binary converter. Due to this, the moduli sets which can result in elegant multiplicative inverses are more popular than others.

III. The Proposed Residue-to-Binary Converter

In this section, we use the CRT-I formulas to achieve a fast residue-to-binary converter for the moduli set $\{2^n, 2^{2n+1}-1, 2^n+1, 2^n-1\}$. First, the needed multiplicative inverses are calculated by the following Lemmas.

Lemma 1: The multiplicative inverse of 2^n modulo $(2^{2n+1}-1)\times(2^{2n}-1)$ is $k_1=-2^{3n+1}+2^n+2^{n+1}$.

Proof: By considering (3), we have

$$\begin{aligned}&\left|k_1 \times 2^n\right|_{(2^{2n+1}-1)(2^{2n}-1)}\\ &= \left|(-2^{3n+1}+2^n+2^{n+1})\times 2^n\right|_{(2^{2n+1}-1)(2^{2n}-1)}\\ &= \left|-(2^{4n+1}-2^{2n+1}-2^{2n})\right|_{2^{4n+1}-2^{2n+1}-2^{2n}+1}\\ &= \left|-(-1)\right|_{2^{4n+1}-2^{2n+1}-2^{2n}+1} = 1\end{aligned} . \quad (6)$$

Lemma 2: The multiplicative inverse of $(2^{2n+1}-1)\times 2^n$ modulo $(2^{2n}-1)$ is $k_2=2^n$.

Proof: Substitution the required values in (4) results in

$$\begin{aligned}&\left|k_2 \times 2^n \times (2^{2n+1}-1)\right|_{2^{2n}-1}\\ &= \left|2^{2n} \times (2^{2n+1}-1)\right|_{2^{2n}-1} = \left|1\times 1\right|_{2^{2n}-1} = 1 .\end{aligned} \quad (7)$$

Lemma 3: The multiplicative inverse of $(2^n+1)\times(2^{2n+1}-1)\times 2^n$ modulo (2^n-1) is $k_3=2^{n-1}$.

Proof: Substitution the required values in (4) results in

$$\begin{aligned}&\left|k_3 \times 2^n \times (2^{2n+1}-1)\times(2^n+1)\right|_{2^n-1}\\ &= \left|2^{n-1} \times 2^n \times (2^{2n+1}-1)\times(2^n+1)\right|_{2^n-1}\\ &= \left|2^{n-1}\times 1\times 1\times 2\right|_{2^n-1} = 1\end{aligned} . \quad (8)$$

Now, by considering $\{P_1, P_2, P_3, P_4\}=\{2^n, 2^{2n+1}-1, 2^n+1, 2^n-1\}$, and substituting the values of multiplicative inverses from Lemmas 1-3 to (2), we achieve the conversion equation as

$$X = x_1 + 2^n \left| \begin{array}{l} (-2^{3n+1}+2^n+2^{n+1})(x_2-x_1) \\ +2^n(2^{2n+1}-1)(x_3-x_2) \\ +2^{n-1}(2^{2n+1}-1)(2^n+1)(x_4-x_3) \end{array} \right|_{(2^{2n+1}-1)(2^{2n}-1)} \quad (9)$$

This equation can be simplified by the following arithmetic properties, to achieve a high-performance hardware design.

Property 1: Modulo (2^p-1) multiplication of a residue number by 2^k, where p and k are positive integers, is equivalent to k bit circular left shifting [29].

Property 2: Modulo (2^p-1) of a negative number is accomplished by subtracting this number from (2^p-1), i.e. taking the one's complement of the number [29].

Property 3: $\left|aP_1\right|_{P_1P_2} = P_1 \times \left|a\right|_{P_2}$ [30].

Next, we can consider the following bit-level representations for the RNS number (x_1, x_2, x_3, x_4) corresponding to the moduli set $\{2^n, 2^{2n+1}-1, 2^n+1, 2^n-1\}$:

$$x_1 = (\underbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,1}x_{1,0}}_{n\ bits})_2 \quad (10)$$

$$x_2 = (\underbrace{x_{2,2n}x_{2,2n-1}\cdots x_{2,1}x_{2,0}}_{2n+1\ bits})_2 \quad (11)$$

$$x_3 = (\underbrace{x_{3,n}x_{3,n-1}\cdots x_{3,1}x_{3,0}}_{n+1\ bits})_2 \quad (12)$$

$$x_4 = (\underbrace{x_{4,n-1}x_{4,n-2}\cdots x_{4,1}x_{4,0}}_{n\ bits})_2 . \quad (13)$$

In order to simplify (9), we can rewrite it as below

$$X = x_1 + 2^n \left| \begin{array}{l} (-2^n(2^{2n+1}-1)+2^{n+1})(x_2-x_1) \\ +2^n(2^{2n+1}-1)(x_3-x_2) \\ +2^{n-1}(2^{2n+1}-1)(2^n+1)(x_4-x_3) \end{array} \right|_{(2^{2n+1}-1)(2^{2n}-1)} .(14)$$

Consequently, (14) can be rewritten again as

$$X = x_1 + 2^n \left| \begin{array}{l} 2^{n+1}(x_2-x_1) \\ +(2^{2n+1}-1)(-2^n(x_2-x_1) \\ +2^n(x_3-x_2)+2^{n-1}(2^n+1)(x_4-x_3)) \end{array} \right|_{(2^{2n+1}-1)(2^{2n}-1)} .(15)$$

Note that, x_2 and x_1 are always less than $2^{2n+1}-1$ and 2^n, respectively. So, we have

$$x_2 - x_1 = \begin{cases} \left|x_2-x_1\right|_{(2^{2n+1}-1)} & if \quad x_2-x_1 \ge 0 \\ \left|x_2-x_1\right|_{(2^{2n+1}-1)} - (2^{2n+1}-1) & if \quad x_2-x_1 < 0 \end{cases} . \quad (16)$$

Substituting (16) in (15) results in

$$X = x_1 + 2^n \left| \begin{array}{l} 2^{n+1}\left|x_2-x_1\right|_{2^{2n+1}-1} \\ +(2^{2n+1}-1)(h-2^n(x_2-x_1) \\ +2^n(x_3-x_2)+2^{n-1}(2^n+1)(x_4-x_3)) \end{array} \right|_{(2^{2n+1}-1)(2^{2n}-1)} \quad (17)$$

Where

$$h = \begin{cases} 0 & if \quad x_2-x_1 \ge 0 \\ -2^{n+1} & if \quad x_2-x_1 < 0 \end{cases} . \quad (18)$$

Similar to [13] and [16], we can reduce the size of modulo operation from $(2^{2n+1}-1)\times(2^{2n}-1)$ to $(2^{2n}-1)$ by using Property 3, as follows

$$\begin{aligned} X = x_1 &+ 2^n(2^{2n+1}-1)\left| \begin{array}{l} h-2^n(x_2-x_1) \\ +2^n(x_3-x_2) \\ +2^{n-1}(2^n+1)(x_4-x_3) \end{array} \right|_{2^{2n}-1} \\ &+ 2^n\left|2^{n+1}\left|x_2-x_1\right|_{2^{2n+1}-1}\right|_{(2^{2n+1}-1)(2^{2n}-1)} \end{aligned} . \quad (19)$$

The largest value of $\left|x_2-x_1\right|_{2^{2n+1}-1}$ is $2^{2n+1}-2$, and also it is

clear that

$$2^{n+1}\times(2^{2n+1}-2)<(2^{2n+1}-1)(2^{2n}-1) \tag{20}$$

Therefore, we can rewrite (19) as

$$X = x_1 + 2^n(2^{2n+1}-1)\left|\begin{matrix} h-2^n(x_2-x_1) \\ +2^n(x_3-x_2) \\ +2^{n-1}(2^n+1)(x_4-x_3)\end{matrix}\right|_{2^{2n}-1} + 2^n\times 2^{n+1}\left|x_2-x_1\right|_{2^{2n+1}-1} \tag{21}$$

Now, (21) can be easily simplified based on Properties 1 and 2. Hence, we rewrite (21) as follows

$$X = x_1 + 2^n(2^{2n+1}-1)Y + 2^n\times 2^{n+1}T \tag{22}$$

Where

$$Y = \left|\begin{matrix} h-2^n(x_2-x_1)+2^n(x_3-x_2) \\ +2^{n-1}(2^n+1)(x_4-x_3)\end{matrix}\right|_{2^{2n}-1} \tag{23}$$

$$T = \left|x_2-x_1\right|_{2^{2n+1}-1} = \left|x_2+\bar{x}_1\right|_{2^{2n+1}-1} \tag{24}$$

$$\bar{x}_1 = \underbrace{1\cdots 11}_{n+1\ bits}\ \underbrace{\bar{x}_{1,n-1}\bar{x}_{1,n-2}\cdots\bar{x}_{1,1}\bar{x}_{1,0}}_{n\ bits}. \tag{25}$$

Note that $x_{i,j}$ means the jth bit of x_i. Next, to simplify (23), we evaluate each part of it separately as below

$$R_0 = \left|h\right|_{2^{2n}-1} = \left|-2^{n+1}\right|_{2^{2n}-1} = \left|-(\underbrace{0\cdots 00}_{n-2}1\underbrace{0\cdots 00}_{n+1})\right|_{2^{2n}-1} = \underbrace{1\cdots 11}_{n-2}0\underbrace{1\cdots 11}_{n+1} \tag{26}$$

$$R_1 = \left|2^n x_1\right|_{2^{2n}-1} = \left|2^n(\underbrace{0\cdots 00}_{n}\underbrace{x_{1,n-1}\cdots x_{1,1}x_{1,0}}_{n})\right|_{2^{2n}-1} = \underbrace{x_{1,n-1}\cdots x_{1,1}x_{1,0}}_{n}\underbrace{0\cdots 00}_{n} \tag{27}$$

$$R_2 = \left|-2^{n+1}x_2\right|_{2^{2n}-1} = \left|-2^{n+1}(\underbrace{x_{2,2n}x_{2,2n-1}\cdots x_{2,1}x_{2,0}}_{2n+1})\right|_{2^{2n}-1} = \left|-2^{n+1}(x_{2,2n}\times 2^{2n}+\underbrace{x_{2,2n-1}x_{2,2n-2}\cdots x_{2,1}x_{2,0}}_{2n})\right|_{2^{2n}-1}. \tag{28}$$

It is clear that $\left|2^{2n}\right|_{2^{2n}-1}=1$. So, (28) can be divided into two parts as

$$R_{21} = \left|-2^{n+1}(\underbrace{0\cdots 00}_{2n-1}x_{2,2n})\right|_{2^{2n}-1} = \underbrace{1\cdots 11}_{n-2}\bar{x}_{2,2n}\underbrace{1\cdots 11}_{n+1} \tag{29}$$

$$R_{22} = \left|-2^{n+1}(\underbrace{x_{2,2n-1}x_{2,2n-2}\cdots x_{2,1}x_{2,0}}_{2n})\right|_{2^{2n}-1} = \left|-2^{n+1}(\underbrace{x_{2,2n-1}\cdots x_{2,n}x_{2,n-1}}_{n+1}\underbrace{x_{2,n-2}\cdots x_{2,1}x_{2,0}}_{n-1})\right|_{2^{2n}-1} = \underbrace{\bar{x}_{2,n-2}\cdots\bar{x}_{2,1}\bar{x}_{2,0}}_{n-1}\underbrace{\bar{x}_{2,2n-1}\cdots\bar{x}_{2,n}\bar{x}_{2,n-1}}_{n+1} \tag{30}$$

Also, for coefficients of x_3, we have

$$R_3 = \left|(2^n-2^{2n-1}-2^{n-1})x_3\right|_{2^{2n}-1} = \left|(2^{n-1}-2^{2n-1})x_3\right|_{2^{2n}-1} = \left|(2^{n-1}-2^{2n-1})(\underbrace{x_{3,n}x_{3,n-1}\cdots x_{3,1}x_{3,0}}_{n+1})\right|_{2^{2n}-1} \tag{31}$$

$$R_{31} = \left|2^{n-1}(\underbrace{0\cdots 00}_{n-1}\underbrace{x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1})\right|_{2^{2n}-1} = \underbrace{x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1}\underbrace{0\cdots 00}_{n-1} \tag{32}$$

$$R_{32} = \left|-2^{2n-1}(\underbrace{0\cdots 00}_{n-1}\underbrace{x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1})\right|_{2^{2n}-1} = \bar{x}_{3,0}\underbrace{1\cdots 11}_{n-1}\underbrace{\bar{x}_{3,n}\cdots\bar{x}_{3,2}\bar{x}_{3,1}}_{n}. \tag{33}$$

Also, to substitute the needed modular multiplications by x_4, with circular left shifting, we have

$$R_4 = \left|2^{n-1}(2^n+1)x_4\right|_{2^{2n}-1} = \left|2^{n-1}(2^n+1)(\underbrace{x_{4,n-1}x_{4,n-2}\cdots x_{4,1}x_{4,0}}_{n})\right|_{2^{2n}-1} = \left|2^{n-1}(\underbrace{x_{4,n-1}\cdots x_{4,1}x_{4,0}}_{n}\underbrace{x_{4,n-1}\cdots x_{4,1}x_{4,0}}_{n})\right|_{2^{2n}-1} = x_{4,0}\underbrace{x_{4,n-1}\cdots x_{4,1}x_{4,0}}_{n}\underbrace{x_{4,n-1}\cdots x_{4,2}x_{4,1}}_{n-1}. \tag{34}$$

Before adding these simplified binary vectors, we can combine (29) and (33), since they have some constant bits with value of one. Hence, the following vectors can be used instead of (29) and (33),

$$R'_{21} = \left|\underbrace{1\cdots 11}_{2n}\right|_{2^{2n}-1} = \left|2^{2n}-1\right|_{2^{2n}-1} = 0 \tag{35}$$

$$R'_{32} = \bar{x}_{3,0}\underbrace{1\cdots 11}_{n-3}\bar{x}_{2,2n}1\underbrace{\bar{x}_{3,n}\cdots\bar{x}_{3,2}\bar{x}_{3,1}}_{n} \tag{36}$$

Therefore, (23) can be calculated using this equation

$$Y = \begin{cases}\left|R_1+R_{22}+R_{31}+R'_{32}+R_4\right|_{2^{2n}-1} & if\ \ x_2-x_1\geq 0 \\ \left|R_0+R_1+R_{22}+R_{31}+R'_{32}+R_4\right|_{2^{2n}-1} & if\ \ x_2-x_1<0\end{cases}. \tag{37}$$

Finally, the main conversion equation, i.e. (22), can be rewritten as

$$X = x_1 + 2^n(2^{2n+1}-1)Y + 2^n\times 2^{n+1}T = x_1 + 2^n(2^{n+1}T + 2^{2n+1}Y - Y) = x_1 + 2^n Z \tag{38}$$

Where

$$Z = Z_1 + Z_2 + (Z_3+1) \tag{39}$$

$$Z_1 = 2^{n+1}T = \underbrace{0\cdots 00}_{n-1}\underbrace{T_{2n}\cdots T_1T_0}_{2n+1}\underbrace{0\cdots 00}_{n+1} \tag{40}$$

$$Z_2 = 2^{2n+1}Y = \underbrace{Y_{2n-1}\cdots Y_1Y_0}_{2n}\underbrace{0\cdots 00}_{2n+1} \tag{41}$$

$$Z_3 = \bar{Y} = \underbrace{1\cdots 11}_{2n+1}\underbrace{\bar{Y}_{2n-1}\cdots\bar{Y}_1\bar{Y}_0}_{2n}. \tag{42}$$

Numerical Example: Consider the moduli set {8,127,9,7} which is an special case of the moduli set {2^n, 2^{2n+1}–1, 2^n+1, 2^n–1} for n=3. To convert the RNS number X=(5,31,8,4) into its corresponding weighted binary representation, we have

$$x_1 = 5 = (101)_2$$

$$x_2 = 31 = (0011111)_2$$

$$x_3 = 8 = (1000)_2$$

$$x_4 = 4 = (100)_2 .$$

Based on (26) to (37), Y can be computed as

$$R_1 = (101000)_2 = 40$$

$$R_{22} = (001000)_2 = 8$$

$$R_{31} = (100000)_2 = 32$$

$$R'_{32} = (111011)_2 = 59$$

$$R_4 = (010010)_2 = 18$$

$$Y = \left|40 + 8 + 32 + 59 + 18\right|_{63} = 31 .$$

Note that, R_0 is not considered due to $x_2 - x_1 \geq 0$. Next, with considering (24), T can be obtained using

$$T = \left|31 - 5\right|_{127} = 26 .$$

Finally, based on (22), the final weighted number X can be simply calculated as

$$X = 5 + 8 \times 127 \times 31 + 8 \times 16 \times 26 = 34829 .$$

The result can be verified as below

$$x_1 = \left|34829\right|_8 = 5$$

$$x_2 = \left|34829\right|_{127} = 31$$

$$x_3 = \left|34829\right|_9 = 8$$

$$x_4 = \left|34829\right|_7 = 4 .$$

IV. Hardware Implementation and Complexity Computation

Hardware architecture of the proposed residue-to-binary converter for the moduli set {2n, 22n+1–1, 2n+1, 2n–1} is depicted in Fig. 1. The main conversion equations which should be realized in hardware are (24) and (37)-(39). First, the operand preparation unit 1 (OPU 1) prepares the binary vectors of (25) to (36) using some inverter gates and changing the wiring of the input operands. Next, a (2n+1)-bit carry-propagate adder (CPA) with end-around carry (EAC) [31] is used to realize (24). Also, based on the method of [13] and [16], (37) is also implemented using four carry-save adders (CSAs) with EAC [31], [32] followed by two CPAs with EAC, and a multiplexer (MUX) to obtain the correct value of Y, according to the sign of x2–x1. Hence, the select line of MUX is connected to the carry-out of CPA1.

TABLE I. DETAILS OF THE HARDWARE REQUIREMENTS OF THE PROPOSED REVERSE CONVERTER

Part	NOT	FA	XNOR /OR pairs	XOR /AND pairs	Delay
OPU1	$4n$+2	–	–	–	t_{NOT}
CPA1	–	n	n+1	–	$(4n+2)t_{FA}$
CSA1	–	n	–	n	t_{FA}
CSA2	–	n+1	–	n–1	t_{FA}
CSA3	–	n+2	n–2	–	t_{FA}
CSA4	–	–	1	2n–1	t_{FA}
CPA2	–	2n	–	–	$(4n)t_{FA}$
CPA3	–	2n	–	–	$(4n)t_{FA}$
OPU2	2n	–	–	–	$t_{NOT} + t_{MUX}$
CSA5	–	–	2n	2n+1	t_{FA}
CPA4	–	4n+1	–	–	$(4n+1)t_{FA}$

It should be noted that, some of the full adders (FAs) of the CSAs are reduced to pairs of XNOR/OR or XOR/AND gates, based on the number of constant bits of the operands. Moreover, implementation of (39) relies on a regular CSA followed by a simple CPA. Note that, before adding the output vectors of CSA5, the carry vector (C) of CSA5 should be shifted to left, and then the most significant bit of the shifted carry vector will be ignored. Finally, realization of (38) can be done by a concatenation. Table I presents the details of the hardware requirements of the converter.

Note that, although the total delay of CPA1 is $(4n+2)t_{FA}$ (t_{FA} denotes the delay of one FA), the carry of first round addition of CPA1 will be available after $(2n+1)t_{FA}$, and it is less than the delay of CPA2 or CPA3. Thus, the total delay of the proposed design can be obtained as follows

$$\begin{aligned} \text{Delay} &= t_{NOT} + (4+4n)t_{FA} + t_{MUX} + t_{NOT} + (1+4n+1)t_{FA} \\ &= (8n+6)t_{FA} + 2t_{NOT} + t_{MUX} . \end{aligned} \quad (43)$$

Now, we compare the hardware complexity of proposed design with the original residue-to-binary converter of the moduli set {2n–1, 2n, 2n+1, 22n+1–1} which has been proposed in [25]. Note that comprehensive comparison between the other existing converters for large dynamic range moduli sets have been presented in [25]. Table II shows the total hardware requirements and conversion delays of the converters in terms of FAs and logic gates. It can be seen that the proposed design is considerably faster than [25]; however, our converter relies on more hardware requirements. It should be noted that the delay of residue-to-binary is very important in an RNS system. Because, as indicated in [9], increasing the delay of residue-to-binary converter can counteract the speed gain of RNS arithmetic unit.

TABLE II. HARDWARE COMPLEXITY COMPARISON

Converter	Hardware requirements	Conversion delay
[25]	$(8n+2)A_{FA} + (n-1)A_{XOR} + (n-1)A_{AND} + (4n+1)A_{XNOR} + (4n+1)A_{OR} + (7n+1)A_{NOT} + (n)A_{MUX2\times1}$	$(12n+5)t_{FA} + 3t_{NOT} + t_{MUX}$
Proposed	$(12n+4)A_{FA} + (6n-1)A_{XOR} + (6n-1)A_{AND} + (4n)A_{XNOR} + (4n)A_{OR} + (6n+2)A_{NOT} + (2n)A_{MUX2\times1}$	$(8n+6)t_{FA} + 2t_{NOT} + t_{MUX}$

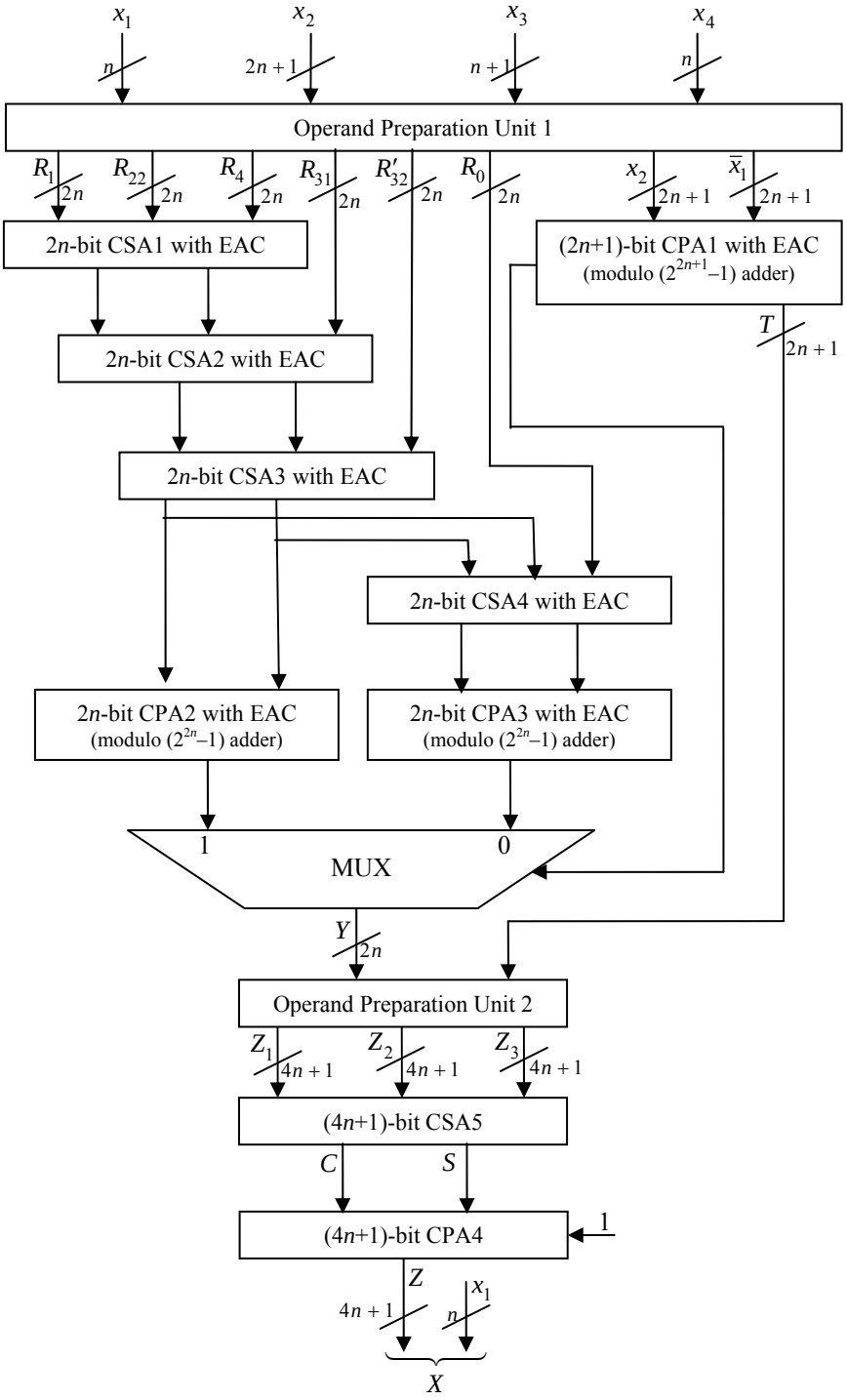

Figure 1. The proposed reverse converter

V. Conclusion

A new high-speed residue-to-binary converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ is designed. The proposed converter has been implemented using full-adders and logic gates, with significantly lower conversion delay, compared to the original residue-to-binary converter of the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$. Therefore, it is expected that this new residue-to-binary converter increases the popularity and applicability of the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ to use in RNS-based computation systems to provide speed enhancement.

References

[1] B. Parhami. Computer Arithmetic: Algorithms and Hardware Design. Oxford University Press, 2000.

[2] A. Omondi and B. Premkumar. Residue Number Systems: Theory and Implementations. Imperial College Press, 2007.

[3] P.V.A. Mohan. Residue Number Systems: Algorithms and Architectures. Kluwer Academic, 2002.

[4] T. Stouratitis and V. Paliouras, "Considering the alternatives in lowpower design," IEEE Circuits and Devices, vol. 7, pp. 23, Jul. 2001. [Online]. Available: http://dx.doi.org/10.1109/101.950050

[5] M.A. Soderstrand and et al. Residue number system arithmetic: modern applications in digital signal processing. IEEE Press, 1986.

[6] R. Conway and J. Nelson, "Improved RNS FIR Filter Architectures," IEEE Trans. Circuits and Systems-II, vol. 51, pp. 26, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1109/TCSII.2003.821524

[7] G.C. Cardarilli, A. Nannarelli and M. Re, "Residue Number System for Low-Power DSP Applications," in Proc. of Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, 2007, pp. 1412-1416. [Online]. Available: http://dx.doi.org/10.1109/ACSSC.2007.4487461

[8] E. Diclaudio, F. Piazza and G. Orlandi, "Fast combinatorial RNS processors for DSP applications," IEEE Trans. Computers, vol. 44, pp. 624, May 1995. [Online]. Available: http://dx.doi.org/10.1109/12.381948

[9] K. Navi, A.S. Molahosseini, M. Esmaeildoust, "How to Teach Residue Number System to Computer Scientists and Engineers," IEEE Trans. Education, vol. 54, pp. 156, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1109/TE.2010.2048329

[10] A.S. Molahosseini, K. Navi, O. Hashemipour, A. Jalali, "An efficient architecture for designing reverse converters based on a general three-moduli set," Elsevier J. Systems Architecture, vol. 54, pp. 929, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.sysarc.2008.03.006

[11] Z. Wang, G.A. Jullien and W.C. Miller, "An Improved Residue-to-Binary Converter," IEEE Trans. Circuits and Systems-I, vol. 47, pp. 1437, Sep. 2000. [Online]. Available: http://dx.doi.org/10.1109/81.883343

[12] Y. Wang, X. Song, M. Aboulhamid and H. Shen, "Adder based residue to binary numbers converters for $(2^n\text{-}1, 2^n, 2^n\text{+}1)$," IEEE Trans. Signal Processing, vol. 50, pp. 1772, Jul. 2002. [Online]. Available: http://dx.doi.org/10.1109/TSP.2002.1011216

[13] W. Wang, M. N. S. Swamy, M. O. Ahmad, and Y.Wang, "A high-speed residue-to-binary converter for three-moduli $\{2^k, 2^k-1, 2^{k-1}-1\}$ RNS and a scheme of its VLSI implementation," IEEE Trans. Circuits and Systems-II, vol. 47, pp. 1576, Dec. 2000. [Online]. Available: http://dx.doi.org/10.1109/82.899659

[14] W. Wang, M. N. S. Swamy, M. O. Ahmad, and Y.Wang, "A Study of the Residue-to-Binary Converters for the Three-Moduli Sets," IEEE Trans. Circuits and Systems-II, vol. 50, pp. 235, Feb. 2003. [Online]. Available: http://dx.doi.org/10.1109/TCSI.2002.808191

[15] P. V. A. Mohan, "RNS-To-Binary Converter for a New Three-Moduli Set $\{2^{n+1}-1, 2^n, 2^n-1\}$," IEEE Trans. Circuits and Systems-II, vol. 54, pp. 775, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1109/TCSII.2007.900844

[16] S.H. Lin, M.H. Sheu, and C.H. Wang, "Efficient VLSI Design of a Residue-to-Binary Converter for the moduli set $(2^n, 2^{n+1}-1, 2^n-1)$," IEICE Trans. Information and Systems, vol. E91-D, pp.2058, Jul. 2008. [Online]. Available: http://dx.doi.org/10.1093/ietisy/e91–d.7.2058

[17] A.S. Molahosseini, C. Dadkhah, K. Navi, M. Eshghi, "Efficient MRC-Based Residue to Binary Converters for the New Moduli Sets $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ and $\{2^{2n}, 2^n-1, 2^{n-1}-1\}$," IEICE Trans. Information and Systems, vol. E92-D, pp. 1628, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1587/transinf.E92.D.1628

[18] M. Bhardwaj , T. Srikanthan, C.T. Clarke, "A reverse converter for the 4-moduli superset $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{+}1\}$," in Proc. of IEEE Symposium on Computer Arithmetic, Adelaide, 1999, pp. 168-175. [Online]. Available: http://dx.doi.org/10.1109/ARITH.1999.762842

[19] A.P. Vinod and A.B. Premkumar, "A residue to binary converter for the 4-moduli superset $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}-1\}$," J. Circuits, Systems and Computers, vol. 10, pp. 85, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0218-1266(00)00004-4

[20] P. V. A. Mohan and A. B. Premkumar, "RNS-to-Binary Converters for Two Four-Moduli Set $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}-1\}$ and $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{+}1\}$," IEEE Trans. Circuits and Systems-I, vol. 54, pp. 1245, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1109/TCSI.2007.895515

[21] M.H. Sheu, S.H. Lin, C. Chen and S.W. Yang, "An efficient VLSI design for a residue to binary converter for general balance moduli $(2^n-3, 2^n\text{+}1, 2^n-1, 2^n\text{+}3)$," IEEE Trans. Circuits and Systems-II, vol. 51, pp. 152, Mar. 2004. [Online]. Available: http://dx.doi.org/10.1109/TCSII.2003.821516

[22] P.V.A. Mohan, "New reverse converters for the moduli set $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$," Elsevier J. Electronics and Communications (AEU), vol. 62, pp. 643, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.aeue.2007.08.008

[23] B. Cao, C. H. Chang and T. Srikanthan, "An Efficient Reverse Converter for the 4-Moduli Set $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{2n}\text{+}1\}$ Based on the New Chinese Remainder Theorem," IEEE Trans. Circuits and Systems-I, vol. 50, pp. 1296, Oct. 2003. [Online]. Available: http://dx.doi.org/10.1109/TCSI.2003.817789

[24] W. Zhang , P. Siy, "An efficient design of residue to binary converter for four moduli set $(2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3)$ based on new CRT II," Elsevier J. Information Sciences, vol. 178, pp. 264, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2007.05.040

[25] A.S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, S. Timarchi, "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," IEEE Trans. Circuits and Systems-I, vol. 57, pp. 823, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1109/TCSI.2009.2026681

[26] B. Cao, C.H. Chang and T. Srikanthan, "A Residue-to-Binary Converter for a New Five-Moduli Set," IEEE Trans. Circuits and Systems-I, vol. 54, pp. 1041, May 2007. [Online]. Available: http://dx.doi.org/10.1109/TCSI.2007.890623

[27] A.S. Molahosseini, C. Dadkhah and K. Navi "A New Five-Moduli Set for Efficient Hardware Implementation of the Reverse Converter," IEICE Electronics Express, vol. 6, pp. 1006, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1587/elex.6.1006

[28] Y. Wang, "Residue-to-Binary Converters Based on New Chinese remainder theorems," IEEE Trans. Circuits and Systems-II, vol. 47, pp. 197, Mar. 2000. [Online]. Available: http://dx.doi.org/10.1109/82.826745

[29] A. A. Hiasat, "VLSI implementation of New Arithmetic Residue to Binary decoders," IEEE Trans. VLSI Systems, vol. 13, pp. 153, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1109/TVLSI.2004.840400

[30] K. H. Rosen. Elementary Number Theory and Its Application. Addison-Wesley, 1988.

[31] S.J. Piestrak, "A high speed realization of a residue to binary converter," IEEE Trans. Circuits and Systems-II, vol. 42, pp. 661, Oct. 1995. [Online]. Available: http://dx.doi.org/10.1109/82.471401

[32] S. J. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," IEEE Trans. Computers, vol. 423, pp. 68, Jan. 1994. [Online]. Available: http://dx.doi.org/10.1109/12.250610