16

Study of the Reverse Converters for the Large Dynamic Range Four-Moduli Sets

Amir Sabbagh Molahosseini[1] and Keivan Navi[2]
[1]*Kerman Branch, Islamic Azad University*
[2]*Shahid Beheshti University*
Iran

1. Introduction

The Residue Number System (RNS) is an efficient alternative number system which has been attracted researchers for over three decades. In RNS, arithmetic operations such as addition and multiplication can be performed on residues without carry-propagation between them; resulting in parallel arithmetic and high-speed hardware implementations (Parhami, 2000; Mohan, 2002; Omondi & Premkumar, 2007). Due to this feature, many Digital Signal Processing architectures based on RNS have been introduced in the literature (Soderstrand et al., 1986; Diclaudio et al., 1995; Chaves et al., 2004). In particular, RNS is an efficient method for the implementation of high-speed finite-impulse response (FIR) filters, where dominant operations are addition and multiplication. Implementation issues of RNS-based FIR filters show that performance can be considerably increased, in comparison with traditional two's complement binary number system (Jenkins et al., 1977; Conway et al., 2004; Cardarilli et al., 2007).

As described in (Navi et al., 2011) a typical RNS system is based on a moduli set which is included some pair-wise relatively prime integers. The product of the moduli is defined as the dynamic range, and it denotes the interval of integers which can be distinctively represented in RNS. The main components of an RNS system are a forward converter, parallel arithmetic channels and a reverse converter. The forward converter encodes a weighted binary number into a residue represented number, with regard to the moduli set; where it can be easily realized using modular adders or look-up tables. Each arithmetic channel includes modular adder, subtractor and multiplier for each modulo of set. The reverse converter decodes a residue represented number into its equivalent weighted binary number. The arithmetic channels are working in a completely parallel architecture without any dependency, and this results in a considerable speed enhancement. However; the overhead of forward and reverse converters can counteract this speed gain, if they are not designed efficiently. The forward converters can be designed using efficient methods. In contrast, design of reverse converters have many complexities with many important factors such as conversion algorithm, type and number of moduli.

An efficient moduli set with moduli of the form of powers of two can greatly reduce the complexity of the reverse converter as well as arithmetic channels. Due to this, many different moduli sets have been proposed for RNS which can be categorized based on their

dynamic range. The most well-known 3n-bit dynamic range moduli set is $\{2^n-1, 2^n, 2^n+1\}$ (Gallaher et al., 1997; Bhardwaj et al., 1998; Wang et al., 2000; Wang et al., 2002). The main reasons for the popularity of this set are its well-form and balanced moduli. However, the modulo 2^n+1 has lower performance than the other two moduli. Hence, some efforts have been done to substitute the modulo 2^n+1 with other well-form RNS moduli, and the resulted moduli sets are $\{2^n-1, 2^n, 2^{n-1}-1\}$ (Hiasat & Abdel-Aty-Zohdy, 1998; Wang et al., 2000b), $\{2^n-1, 2^n, 2^{n+1}-1\}$ (Mohan, 2007; Lin et al., 2008).

The dynamic ranges provided by these three moduli sets are not adequate for recent applications which require higher performance. Two approaches have been proposed to solve this problem. First, using three-moduli sets to provide large dynamic range with some specific forms like $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$ where $\alpha<\beta$ (Molahosseini et al., 2008) and $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ (Molahosseini et al., 2009). Second, using four and five moduli sets to increase dynamic range and parallelism in RNS arithmetic unit. The 4n-bit dynamic range four-moduli sets are $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ (Bhardwaj et al., 1999; Mohan & Premkumar, 2007) and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ (Vinod et al., 2000; Mohan & Premkumar, 2007). Although, these four-moduli sets include relatively balanced moduli, their multiplicative inverses are very complicated, and this results in low-performance reverse converters. Furthermore, some recent applications require even more dynamic range than 4n-bit. This demand results in introducing new class of moduli sets which have been called *large dynamic range four-moduli sets*. The first one is the 5n-bit dynamic range moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ that was proposed by (Cao et al., 2003). Next, (Zhang et al., 2008) enhanced the dynamic range to 6n-bit, and introduced the set $\{2^n - 1, 2^n +1, 2^{2n}-2, 2^{2n+1}-3\}$. Moreover, (Molahosseini et al., 2010) proposed the four-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$ in 5n and 6n-bit dynamic range, respectively.

In this chapter, after an introduction about RNS and reverse conversion algorithms, the architecture of the state-of-the-art reverse cnverters which have been designed for the efficient large dynamic range four-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$, $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$ and $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ will be investigated. Furthermore, a recent contribution about modified version of the four-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ that is $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$ will be studied. Finally, we present performance comparison in terms of hardware requirements and conversion delays, between the investigated reverse converters.

2. Background

The fundamental part of RNS (Omondi & Premkumar, 2007) is the moduli set $\{P_1,P_2, \ldots,P_n\}$ where numbers are relatively-prime, i.e. $\gcd(P_i,P_j)=1$ for $i\neq j$. The binary weighted number X can be represented as $X=(x_1,x_2, \ldots ,x_n)$, where

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

This representation is unique for any integer number X in the range $[0,M-1]$, where $M=P_1P_2\ldots P_n$ is the dynamic range of the moduli set $\{P_1,P_2, \ldots,P_n\}$ (Taylor, 1984). Addition (subtraction) and multiplication on RNS numbers can be performed in parallel due to the absence of carry propagation between residues.

The famous algorithms for performing reverse conversion are Chinese remainder theorem (CRT), mixed-radix conversion (MRC) and new Chinese remainder theorems (New CRTs).

In order to design a reverse converter, we have to select appropriate moduli set with considering the required parallelism and dynamic range requirements. Next, the moduli should be substituted in one of mentioned conversion algorithm formulas, and the resulted conversion equations should be simplified using some modulo arithmetic properties to reduce hardware complexity. Finally, hardware implementation of the simplified equations can be done using binary hardware's such as full adders, half adders, logic gates or lock-up tables. In the following, we briefly review the formulas of reverse conversion algorithms for four-moduli RNSs. Hence, consider the moduli set (P_1, P_2, P_3, P_4) with corresponding RNS number (x_1, x_2, x_3, x_4).
By CRT (Parhami, 2000) the weighted number X can be calculated by

$$X = \left| \sum_{i=1}^{4} \left| x_i N_i \right|_{P_i} M_i \right|_M \tag{2}$$

Where

$$M = P_1 P_2 P_3 P_4 \tag{3}$$

$$M_i = M / P_i \tag{4}$$

$$N_i = \left| M_i^{-1} \right|_{P_i} \tag{5}$$

The CRT has capability of parallel implementation; however its final big modulo adder results in inefficient hardware realization if it is considered in direct form.
By MRC (Koc, 1989) the conversion can be done using the following equation:

$$X = v_4 P_3 P_2 P_1 + v_3 P_2 P_1 + v_2 P_1 + v_1 \tag{6}$$

The v_i's coefficients are as follows

$$v_1 = x_1 \tag{7}$$

$$v_2 = \left| (x_2 - v_1) \left| P_1^{-1} \right|_{P_2} \right|_{P_2} \tag{8}$$

$$v_3 = \left| ((x_3 - v_1) \left| P_1^{-1} \right|_{P_3} - v_2) \left| P_2^{-1} \right|_{P_3} \right|_{P_3} \tag{9}$$

$$v_4 = \left| (((x_4 - v_1) \left| P_1^{-1} \right|_{P_4} - v_2) \left| P_2^{-1} \right|_{P_4} - v_3) \left| P_3^{-1} \right|_{P_4} \right|_{P_4} \tag{10}$$

Although MRC implies a sequential process, for two and three-moduli sets it can be lead to simple and efficient reverse conversion equations.
The New CRT-I (Wang, 2000; Molahosseini et al., 2010) uses a more efficient conversion formula

$$X = x_1 + P_1 \left| k_1 (x_2 - x_1) + k_2 P_2 (x_3 - x_2) + k_3 P_2 P_3 (x_4 - x_3) \right|_{P_2 P_3 P_4} \tag{11}$$

Where

$$\left|k_1 \times P_1\right|_{P_2P_3P_4} = 1 \tag{12}$$

$$\left|k_2 \times P_1 \times P_2\right|_{P_3P_4} = 1 \tag{13}$$

$$\left|k_3 \times P_1 \times P_2 \times P_3\right|_{P_4} = 1 \tag{14}$$

Moreover, New CRT-II (Wang, 2000; Molahosseini et al., 2010) provides a tree-like architecture by using the following equations

$$X = Z + P_1P_2\left|k_1(Y - Z)\right|_{P_3P_4} \tag{15}$$

$$Z = x_1 + P_1\left|k_2(x_2 - x_1)\right|_{P_2} \tag{16}$$

$$Y = x_3 + P_3\left|k_3(x_4 - x_3)\right|_{P_4} \tag{17}$$

Where

$$\left|k_1P_1P_2\right|_{P_3P_4} = 1 \tag{18}$$

$$\left|k_2P_1\right|_{P_2} = 1 \tag{19}$$

$$\left|k_3P_3\right|_{P_4} = 1 \tag{20}$$

The New CRTs have potentiality to create higher performance reverse converters than CRT and MRC particularly for some special four-moduli sets. Hence, many research have been done in the recent years to discover efficient four-moduli sets which can be fitted with properties of New CRTs. In the next sections, we investigate the reverse converters that are previously designed for these four-moduli sets.

3. Reverse converter for the moduli set {2^n–1, 2^n, 2^n+1, 2^{2n}+1}

The moduli set {2^n–1, 2^n, 2^n+1, 2^{2n}+1} was introduced by (Cao et al., 2003). They have used New CRT-I to design a fully adder-based reverse converter. In the following, we briefly review the conversion formulas and hardware architecture of the converter of (Cao et al., 2003). First, consider the moduli set {2^n–1, 2^n, 2^n+1, 2^{2n}+1} with corresponding residues (x_1, x_2, x_3, x_4). The residues can be represented in bit-level as below

$$x_1 = (\underbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,1}x_{1,0}}_{n\ bits})_2 \tag{21}$$

$$x_2 = (\underbrace{x_{2,n-1}x_{2,n-2}\cdots x_{2,1}x_{2,0}}_{n\ bits})_2 \tag{22}$$

$$x_3 = (\underbrace{x_{3,n}x_{3,n-1}\cdots x_{3,1}x_{3,0}}_{n+1\,bits})_2 \tag{23}$$

$$x_4 = (\underbrace{x_{4,2n}x_{4,2n-1}\cdots x_{4,1}x_{4,0}}_{2n+1\,bits})_2 \tag{24}$$

With substituting the required multiplicative inverses and values of moduli, i.e. P_1=2^n, P_2=2^n+1, P_3=2^{2n}+1 and P_4=2^n–1 in the New CRT-I formulas (11)-(14), we achieve the following conversion equation:

$$X = x_1 + 2^n \left| \begin{matrix} 2^{3n}(x_3 - x_2) + (2^{3n-2} + 2^{2n-1} - 2^{n-2})(2^n + 1)(x_4 - x_3) \\ +2^{n-2}(2^n + 1)(2^{2n} + 1)(x_1 - x_4) \end{matrix} \right|_{2^{4n}-1} \tag{25}$$

This main conversion equation can be simplified based on the following two well-known modulo (2^n-1) arithmetic properties.

Property 1: The residue of a negative residue number ($-v$) in modulo ($2^n - 1$) is the one's complement of v, where $0 \le v < 2^n - 1$ (Hariri et al. 2008).

Property 2: The multiplication of a residue number v by 2^P in modulo ($2^n - 1$) is carried out by P bit circular left shift, where P is a natural number (Hariri et al. 2008).

Now, (25) can be rewritten as follows

$$X = x_1 + 2^n Z \tag{26}$$

Where

$$Z = \left| v_1 + v_2 + v_{31} + v_{32} + v_{41} + v_{42} \right|_{2^{4n}-1} \tag{27}$$

Next, the binary vectors v_i's which have been simplified based on properties 1 and 2 are as below

$$v_1 = \underbrace{x_{1,1}x_{1,0}}_{2}\underbrace{x_{1,n-1}\cdots x_{1,1}x_{1,0}}_{n}\underbrace{x_{1,n-1}\cdots x_{1,1}x_{1,0}}_{n}\underbrace{x_{1,n-1}\cdots x_{1,1}x_{1,0}}_{n}\underbrace{x_{1,n-1}\cdots x_{1,3}x_{1,2}}_{n-2} \tag{28}$$

$$v_2 = \underbrace{\bar{x}_{2,n-1}\cdots\bar{x}_{2,1}\bar{x}_{2,0}}_{n}\underbrace{1\cdots11}_{3n} \tag{29}$$

$$v_{31} = \underbrace{\bar{x}_{3,1}\bar{x}_{3,0}}_{2}\underbrace{1\cdots11}_{n-1}\underbrace{\bar{x}_{3,n}\cdots\bar{x}_{3,1}\bar{x}_{3,0}}_{n+1}\underbrace{1\cdots11}_{n-1}\underbrace{\bar{x}_{3,n}\cdots\bar{x}_{3,3}\bar{x}_{3,2}}_{n-1} \tag{30}$$

$$v_{32} = 0\underbrace{x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1}\underbrace{0\cdots00}_{n-1}\underbrace{x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1}\underbrace{0\cdots00}_{n-2} \tag{31}$$

$$v_{41} = \underbrace{x_{4,n}\cdots x_{4,1}x_{4,0}}_{n+1}\underbrace{0\cdots00}_{2n-1}\underbrace{x_{4,2n}\cdots x_{4,n+2}x_{4,n+1}}_{n} \tag{32}$$

$$v_{42} = \underbrace{1\cdots11}_{n}\underbrace{\bar{x}_{4,2n}\cdots\bar{x}_{4,1}\bar{x}_{4,0}}_{2n+1}\underbrace{1\cdots11}_{n-1} \tag{33}$$

Therefore, these six operands should be added using a modulo (2^{4n}-1) multi-operand adder which can be realised by four carry-save adders (CSAs) with end-around carry (EAC) followed by a modulo (2^{4n}-1) carry propagate adder (CPA) with EAC (Piestrak, 1994, 1995). The hardware architecture of the resulted converter is shown in Fig. 1.

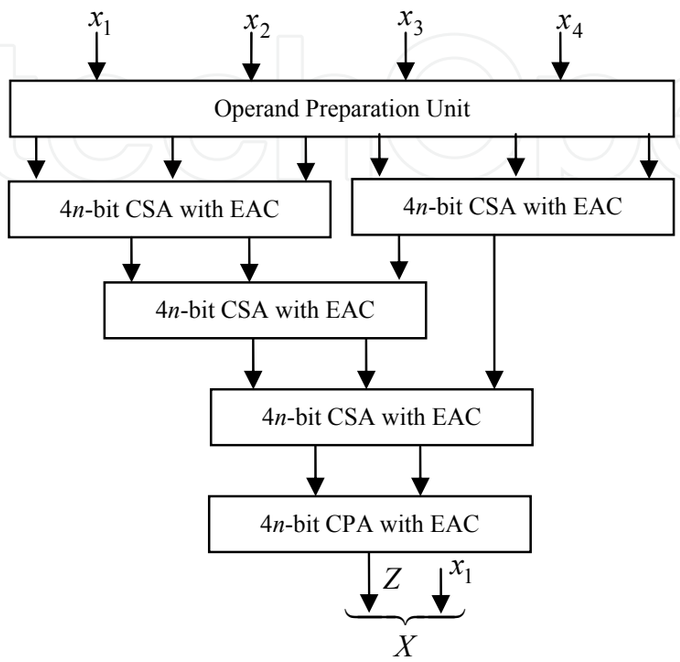


Fig. 1. The converter for moduli set {2^n–1, 2^n, 2^n+1, 2^{2n}+1} (Cao et al., 2003)

4. Reverse converter for the moduli set {2^n–1, 2^n+1, 2^{2n}, 2^{2n}+1}

The moduli set {2^n–1, 2^n+1, 2^{2n}, 2^{2n}+1} has been recently introduced by (Molahosseini et al., 2010) to provide large dyamic range (6n-bit), and high-speed reverse converter. Similar to (Cao et al., 2003), the New CRT-I has used to design converter but with different moduli order, i.e. {2^{2n}, 2^{2n}+1, 2^n+1, 2^n–1}. Therefore, by letting P_1=2^{2n}, P_2=2^{2n}+1, P_3=2^n+1 and P_4=2^n–1, and putting the multiplicative inverses in the New CRT-I formulas (11)-(14), we have the following main conversion equation (Molahosseini et al., 2010).

$$X = x_1 + 2^{2n}\left|\begin{matrix}2^{2n}(x_2 - x_1) + 2^{2n-1}(2^{2n}+1)(x_3 - x_2) \\ +2^{n-2}(2^{2n}+1)(2^n+1)(x_4 - x_3)\end{matrix}\right|_{2^{4n}-1} \tag{34}$$

Simplification of this equation can be done as follows

$$X = x_1 + 2^{2n} Z \tag{35}$$

Where

$$Z = \left|v_1 + v_2 + v_{31} + v_{32} + v_4\right|_{2^{4n}-1} \tag{36}$$

$$v_1 = \underbrace{\bar{x}_{1,2n-1}\cdots\bar{x}_{1,1}\bar{x}_{1,0}}_{2n}\underbrace{\bar{x}_{2,2n}\cdots\bar{x}_{2,2}\bar{x}_{2,1}}_{2n} \tag{37}$$

$$v_2 = \underbrace{x_{2,2n}\cdots x_{2,1}x_{2,0}}_{2n+1}\underbrace{0\cdots00}_{2n-1} \tag{38}$$

$$v_{31} = x_{3,1}x_{3,0}\underbrace{0\cdots00}_{n-1}\underbrace{x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1}\underbrace{0\cdots00}_{n-1}\underbrace{x_{3,n}\cdots x_{3,3}x_{3,2}}_{n-1} \tag{39}$$

$$v_{32} = \bar{x}_{2,0}\underbrace{\bar{x}_{3,n}\cdots\bar{x}_{3,1}\bar{x}_{3,0}}_{n+1}\underbrace{1\cdots11}_{n-1}\underbrace{\bar{x}_{3,n}\cdots\bar{x}_{3,1}\bar{x}_{3,0}}_{n+1}\underbrace{1\cdots11}_{n-2} \tag{40}$$

$$v_4 = x_{4,1}x_{4,0}\underbrace{x_{4,n-1}\cdots x_{4,1}x_{4,0}}_{n}\underbrace{x_{4,n-1}\cdots x_{4,1}x_{4,0}}_{n}\underbrace{x_{4,n-1}\cdots x_{4,1}x_{4,0}}_{n}\underbrace{x_{4,n-1}\cdots x_{4,3}x_{4,2}}_{n-2} \tag{41}$$

Therefore, only five operands should be added using three CSAs with EAC followed by a CPA with EAC (Piestrak, 1994, 1995). Hence, in comparison with (Cao et al., 2003) which needed four CSAs, the (Molahosseini et al., 2010) results in reduction of one 4*n*-bit CSA with EAC; while providing larger dynamic range. The Fig. 2 shows the hardware implementation of this converter.

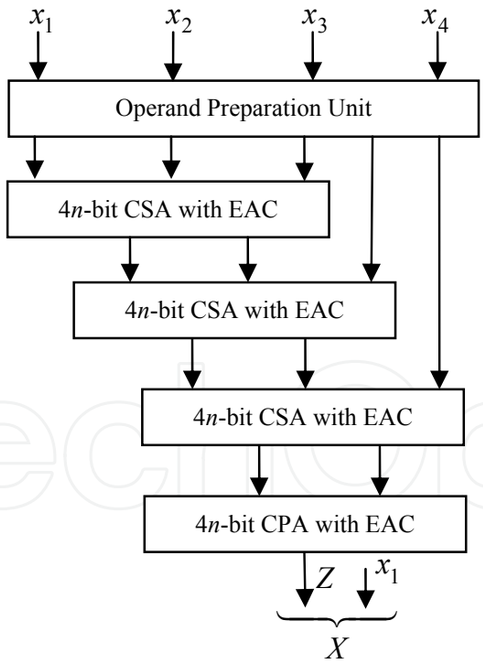


Fig. 2. The converter for moduli set $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$ (Molahosseini et al., 2010)

5. The reverse converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$

The main disadvantage of the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ and $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$ is the modulo $2^{2n}+1$. Because, performance of modulo arithmetic circuits for $2^{2n}+1$ is much

lower than the moduli 2^n–1 and 2^n+1. Hence, (Molahosseini et al., 2010) have been substituted 2^{2n}+1 with well-formed number 2^{2n+1}–1 that results in introducing the large dynamic range four-moduli set {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1}. Besides, they have used New CRT-II to design an efficient reverse converter for this moduli set as described below.
With considering P_1=2^n, P_2=2^{2n+1}–1, P_3=2^n+1, P_4=2^n–1, and the New CRT-II formulas (15)-(17), we have the following conversion equations (Molahosseini et al., 2010)

$$X = Z + 2^n(2^{2n+1}-1)\left|2^n(Y-Z)\right|_{2^{2n}-1} \tag{42}$$

Where

$$Z = x_1 + 2^n\left|2^{n+1}(x_2 - x_1)\right|_{2^{2n+1}-1} \tag{43}$$

$$Y = x_3 + (2^n+1)\left|2^{n-1}(x_4 - x_3)\right|_{2^n-1} \tag{44}$$

Simplified versions of these equations have been computed in (Molahosseini et al., 2010). Here, we briefly review the final simplified equations. First, (43) can be rewritten as

$$Z = x_1 + 2^n H \tag{45}$$

Where

$$H = \left|v_1 + v_2\right|_{2^{2n+1}-1} \tag{46}$$

$$v_1 = \underbrace{x_{2,n-1}\cdots x_{2,1}x_{2,0}}_{n}\underbrace{x_{2,2n}\cdots x_{2,n+1}x_{2,n}}_{n+1} \tag{47}$$

$$v_2 = \underbrace{\bar{x}_{1,n-1}\cdots \bar{x}_{1,1}\bar{x}_{1,0}}_{n}\underbrace{1\cdots 11}_{n+1} \tag{48}$$

Next, for simplifying (44) we have

$$Y = x_3 + (2^n+1)K \tag{49}$$

Where

$$K = \left|v_3 + v_4\right|_{2^n-1} \tag{50}$$

$$v_3 = \underbrace{x_{4,0}x_{4,n-1}\cdots x_{4,2}x_{4,1}}_{n} \tag{51}$$

$$v_4 = \begin{cases} \underbrace{\bar{x}_{3,0}\bar{x}_{3,n-1}\cdots\bar{x}_{3,2}\bar{x}_{3,1}}_{n} & \text{if } x_{3,n}=0 \\ 0\underbrace{1\cdots 11}_{n-1} & \text{if } x_{3,n}=1 \end{cases} \tag{52}$$

Eventually, (42) can be computed as below

$$X = Z + 2^n(2^{2n+1} - 1)T = x_1 + 2^n(P - T) \tag{53}$$

Where

$$P = H + 2^{2n+1}T = \underbrace{T_{2n-1} \cdots T_1 T_0}_{2n} \underbrace{H_{2n} \cdots H_1 H_0}_{2n+1} \tag{54}$$

$$T = \left| v_5 + v_6 + v_7 + v_8 \right|_{2^{2n}-1} \tag{55}$$

$$v_5 = \underbrace{x_{3,n-1} \cdots x_{3,1} x_{3,0}}_{n} \underbrace{0 \cdots 00}_{n-1} x_{3,n} \tag{56}$$

$$v_6 = \underbrace{K_{n-1} \cdots K_1 K_0}_{n} \underbrace{K_{n-1} \cdots K_1 K_0}_{n} \tag{57}$$

$$v_7 = \underbrace{\bar{x}_{1,n-1} \cdots \bar{x}_{1,1} \bar{x}_{1,0}}_{n} \underbrace{1 \cdots 11}_{n-1} \bar{H}_{2n} \tag{58}$$

$$v_8 = \underbrace{\bar{H}_{2n-1} \cdots \bar{H}_1 \bar{H}_0}_{2n} \tag{59}$$

Therefore, two modulo adders needed to realize (46) and (50). Moreover, (55) can be implemented using three CSAs with EAC followed by a CPA with EAC. Note that some of the full adders (FAs) of these CPAs and CSAs are simplified to XOR/AND or XNOR/OR pairs due to the constant bits of the inputs. The final result, i.e. (53) can be obtained by a (4n+1)-bit binaryadder with '1' carry-in. Fig. 3 presents the reverse converter for the moduli set {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1}.

6. Reverse converter for the moduli set {2^n–1, 2^n+1, 2^{2n}, 2^{2n+1}–1}

The moduli set {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1} reduces the total delay of RNS arithmetic unit versus the moduli sets {2^n–1, 2^n, 2^n+1, 2^{2n}+1} and {2^n–1, 2^n+1, 2^{2n}, 2^{2n}+1}. However, still the inter-channel delay of modulo 2^{2n+1}–1 is larger than the other three moduli, i.e. 2^n–1, 2^n and 2^n+1. Due to this, the moduli set {2^n–1, 2^n+1, 2^{2n}, 2^{2n+1}–1} has been recently proposed by (Molahosseini & Navi, 2010). The main advantage of this set is that it provides all of the merits of the moduli set {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1} while providing larger dynamic range (6n-bit). Because, enhancing modulo 2^n to 2^{2n} is not increasing the complexity of the reverse converter.

The converter of (Molahosseini & Navi, 2010) has a two-level architecutre. In other words, they have used a combinatorial conversion algorithm; consisting both CRT and MRC. First, the previous CRT-Based design of reverse converter for the subset {2^{2n}, 2^n–1, 2^n+1} (Hiasat & Sweidan, 2004) is used to achieve the weighted equivalent of the residues (x_1, x_2, x_3) as below

$$Z = x_1 + 2^{2n}Y \tag{60}$$

Where

$$Y = \left| v_1 + v_2 + v_3 + v_4 \right|_{2^{2n}-1} \tag{61}$$

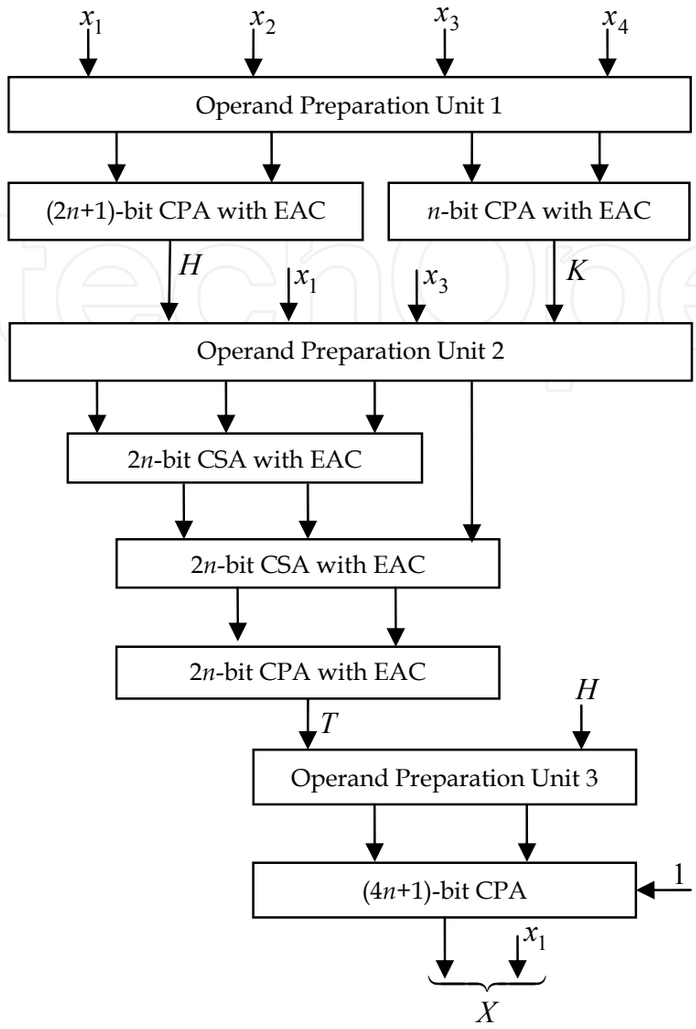


Fig. 3. The converter for moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ (Molahosseini et al., 2010)

$$v_1 = \underbrace{\bar{x}_{1,2n-1} \cdots \bar{x}_{1,1} \bar{x}_{1,0}}_{2n} \tag{62}$$

$$v_2 = \underbrace{x_{2,0} x_{2,n-1} \cdots x_{2,1}}_{n} \underbrace{x_{2,0} x_{2,n-1} \cdots x_{2,1}}_{n} \tag{63}$$

$$v_3 = x_{3,0} \underbrace{\bar{x}_{3,n-1} \cdots \bar{x}_{3,1} \bar{x}_{3,0}}_{n} \underbrace{x_{3,n-1} \cdots x_{3,2} x_{3,1}}_{n-1} \tag{64}$$

$$v_4 = \bar{x}_{3,n} \underbrace{0 \cdots 00}_{n-1} x_{3,n} \underbrace{1 \cdots 11}_{n-1}. \tag{65}$$

Next, two-channel MRC (eqs. (6)-(8)) is used to derive the final result by considering the composite set $\{2^{2n}(2^{2n}-1), 2^{2n+1}-1\}$ and corresponding numbers (Z, x_4). The final conversion equation is as follows

$$X = Z + 2^{2n}(2^{2n}-1)\left|-2^{2n+3}(x_4 - Z)\right|_{2^{2^{n+1}}-1} = x_1 + 2^{2n}(Y + 2^{2n}T - T) \tag{66}$$

Where

$$T = \left|v_5 + v_6 + v_7\right|_{2^{2^{n+1}}-1} \tag{67}$$

$$v_5 = \underbrace{x_{1,2n-2}\cdots x_{1,1}x_{1,0}}_{2n-1}\,0\,x_{1,2n-1} \tag{68}$$

$$v_6 = \underbrace{Y_{2n-1}\cdots Y_1 Y_0}_{2n}\,0 \tag{69}$$

$$v_7 = \underbrace{\overline{x}_{4,2n-2}\cdots\overline{x}_{4,1}\overline{x}_{4,0}}_{2n-1}\,\overline{x}_{4,2n}\overline{x}_{4,2n-1} \tag{70}$$

The hardware implementation of this converter relies on two modulo adders for realization of (61) and (67). In other words, (61) needed two $2n$-bit CSAs with EAC and a $2n$-bit CPA with EAC, and a $(2n+1)$-bit CPA with EAC is used to realize (67). Besides, (66) only requires one $(4n+1)$-bit regular binary adder; the required multiplications all can be done using shift and concatenation. The converter has been depicted in Fig. 4.

7. Complexity comparison

Table 1 presents the total hardware requirements and conversion delays of the reverse converters for the large dynamic range four-moduli sets in terms of logic gates and FAs. Note that A_{FA} and D_{FA} indicate the area and delay of one FA, respectively. It can be seen that the fastest converter is the converter for moduli set $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$. Because, the dynamic range of this set is $6n$-bit while the dynamic range of moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ is $5n$-bit. Therefore, for providing the same dynamic range, the value of n for the first

Moduli set	**Hardware Requirements**	**Conversion Delay**
$\{2^n-1, 2^n, 2^n+1, 2^{2^{n+1}}-1\}$	$(8n+2)A_{FA} + (n-1)A_{XOR} + (n-1)A_{AND}$ $+ (4n+1)A_{XNOR} + (4n+1)A_{OR}$ $+ (7n+1)A_{NOT} + (n)A_{MUX2\times1}$	$(12n+5)D_{FA}+3D_{NOT}+D_{MUX}$
$\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$	$(10n+3)A_{FA} + (n+1)A_{XOR}$ $+ (n+1)A_{AND} + (3n-1)A_{XNOR}$ $+(3n-1)A_{OR} + (7n+3)A_{NOT}$	$(12n+6)D_{FA}+2D_{NOT}$
$\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$	$(11n+6)A_{FA} + (2n-1)A_{XOR}$ $+ (2n-1)A_{AND} + (4n)A_{XNOR}$ $+(4n)A_{OR} + (5n+3)A_{NOT}$	$(8n+3)D_{FA}+D_{NOT}$
$\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$	$(10n+6)A_{FA} + (4n-3)A_{XOR}$ $+ (4n-3)A_{AND} + (2n-3)A_{XNOR}$ $+(2n-3)A_{OR} + (6n+3)A_{NOT}$	$(8n+3)D_{FA}+D_{NOT}$

Table 1. Hardware requirements and conversion delays of the reverse converters for the large dynamic range four-moduli sets

set is smaller than the second set. Furthermore, the reverse converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ relies on less hardware requirements than others. From another side, the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$ results in faster RNS arithmetic units than the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ and $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$.

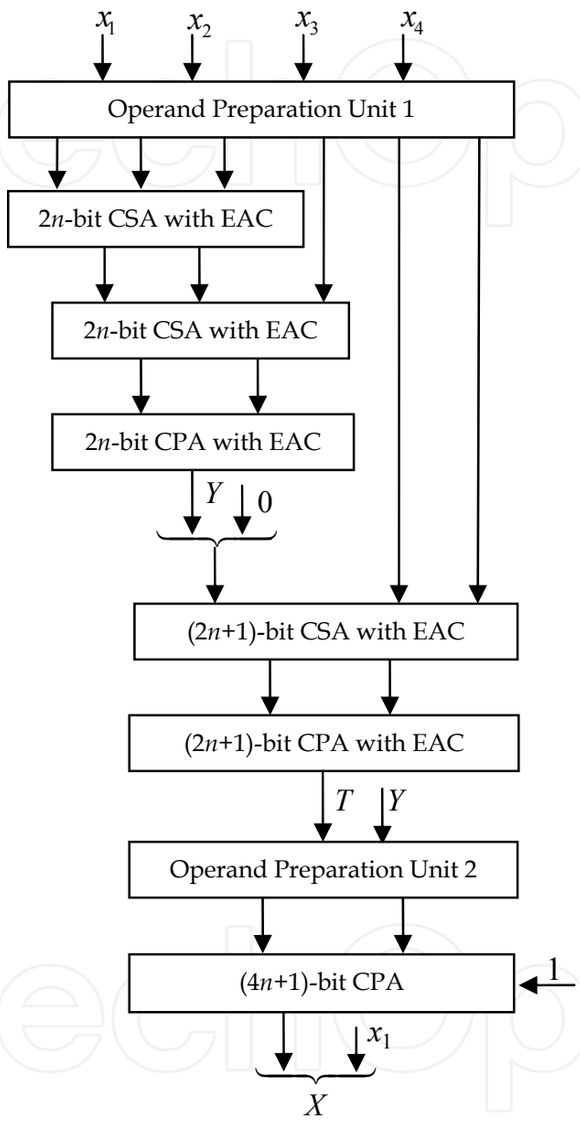


Fig. 4. The converter for moduli set $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$ (Molahosseini & Navi, 2010)

8. Conclusion

The Residue Number System has been recognized as one of the efficient alternative number systems which can be used to high-speed hardware implementation of Digital Signal Processing computation algorithms. However, forward and reverse converters are needed to act as interfaces between RNS and the conventional binary digital systems. The overhead of these converters can frustrate the speed efficiency of RNS, and due to this a lot of research

has been done to design efficient reverse converters. This chapter presents a study on the state-of-the-art reverse converters which have been designed for the recently introduced large dynamic range RNS four-moduli sets. We provide an overview about different reverse conversion algorithms, the recent four-moduli sets, and the reverse converter architectures.

9. References

Bhardwaj, M., Premkumar, A. B. & Srikanthan, T. (1998). Breaking the 2n-Bit Carry Propagation Barrier in Residue to Binary Conversion for the [2^n-1, 2^n, 2^n+1] Modula Set. *IEEE Transactions on Circuits and Systems-I*, Vol. 45, No. 9, pp. 998-1002.

Bhardwaj, M., Srikanthan, T. & Clarke, C.T. (1999). A reverse converter for the 4-moduli superset {2^n-1, 2^n, 2^n+1, 2^{n+1}+1}, *Proceedings of IEEE Symposium on Computer Arithmetic*.

Cao, B., Chang, C.H. & Srikanthan, T. (2003). An Efficient Reverse Converter for the 4-Moduli Set {2^n-1, 2^n, 2^n+1, 2^{2n}+1} Based on the New Chinese Remainder Theorem. *IEEE Transactions on Circuits and Systems-I*, Vol. 50, No. 10, pp. 1296-1303.

Cardarilli, G.C., Nannarelli, A. and Re, M. (2007). Residue Number System for Low-Power DSP Applications, *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, Asilomar, USA.

Chaves, R. & Sousa, L. (2003). RDSP: A RISC DSP based on residue number system, *Proceedings of Euromicro Symposium on digital system design: architectures, methods and tools*.

Conway, R. and Nelson, J. (2004). Improved RNS FIR Filter Architectures. *IEEE Transactions on Circuits and Systems-II*, Vol. 51, No. 1, pp. 26-28.

Diclaudio, E., Piazza F. & Orlandi, G. (1995). Fast combinatorial RNS processors for DSP applications. *IEEE Transactions on Computers*, Vol. 44, pp. 624-331.

Gallaher, D., Petry, F.E. & Srinivasan, P. (1997). The Digit Parallel Method for Fast RNS to Weighted Number System Conversion for Specific Moduli (2^k-1, 2^k, 2^k+1). *IEEE Transactions on Circuits and Systems-II*, Vol. 44, No. 1, pp. 53-57.

Hariri, A., Navi, K. & Rastegar, R. (2008). A new high dynamic range moduli set with efficient reverse converter. *Elsevier Journal of Computers and Mathematics with Applications*, Vol. 55, No. 4, pp. 660-668.

Hiasat A. & Abdel-Aty-Zohdy, H. S. (1998). Residue-to-binary arithmetic converter for the moduli set (2^k, 2^k-1, 2^{k-1}-1). *IEEE Transactions on Circuits and System-II*, Vol. 45, No. 2, pp. 204–208.

Hiasat, A. & Sweidan, A. (2004). Residue-to-binary decoder for an enhanced moduli set. *IEE Proc.-Comput. Digit. Tech.*, Vol. 151, No. 2, pp. 127-130.

Jenkins, W. K. & Leon, B. J. (1977). The use of residue number systems in the design of finite impulse response digital filters. *IEEE Transactions on Circuits and Systems*, Vol. CAS-24, pp. 191–201.

Koc, C.K. (1989). A fast algorithm for mixed-radix conversion in residue arithmetic, *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors*.

Lin, S.H., Sheu, M.H. & Wang, C.H. (2008). Efficient VLSI Design of a Residue-to-Binary Converter for the moduli set (2^n, 2^{n+1}–1, 2^n–1). *IEICE Transactions on Information and Systems*, Vol. E91-D, No. 7, pp.2058-2060.

Mohan, P.V.A. (2002). *Residue Number Systems: Algorithms and Architectures*, Kluwer Academic.

Mohan, P. V. A. (2007). 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*, Vol. 54, No. 9, pp. 775-779.

Mohan P. V. A. & Premkumar, A. B. (2007). RNS-to-Binary Converters for Two Four-Moduli Set $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{–}1\}$ and $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{+}1\}$. *IEEE Transactions on Circuits and Systems-I*, Vol. 54, No. 6, pp. 1245-1254.

Molahosseini, A.S., Dadkhah, C., Navi, K. & Eshghi, M. (2009). Efficient MRC-Based Residue to Binary Converters for the New Moduli Sets $\{2^{2n}, 2^n\text{–}1, 2^{n+1}\text{–}1\}$ and $\{2^{2n}, 2^n\text{–}1, 2^{n-1}\text{–}1\}$. *IEICE Transactions on Information and Systems*, vol. E92-D, pp. 1628-1638.

Molahosseini, A.S. & Navi, K. (2010). A Reverse Converter for the Enhanced Moduli Set $\{2^n\text{–}1, 2^n\text{+}1, 2^{2n}, 2^{2n+1}\text{–}1\}$ Using CRT and MRC, *Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI'10)*, Kefalonia, Greece, Jul. 13-15.

Molahosseini, A.S., Navi, K., Dadkhah, C., Kavehei, O. & Timarchi, S. (2010). Efficient Reverse Converter Designs for the New 4-Moduli Sets $\{2^n\text{–}1, 2^n, 2^n\text{+}1, 2^{2n+1}\text{–}1\}$ and $\{2^n\text{–}1, 2^n\text{+}1, 2^{2n}, 2^{2n}\text{+}1\}$ Based on New CRTs. *IEEE Transactions on Circuits and Systems-I*, vol. 57, no. 4, pp. 823-835.

Molahosseini, A.S., Navi, K., Hashemipour, O. & Jalali, A. (2008). An efficient architecture for designing reverse converters based on a general three-moduli set. *Elsevier Journal of Systems Architecture*, vol. 54, pp. 929-934.

Navi, K., Molahosseini, A.S. & Esmaeildoust, M. (2011). How to Teach Residue Number System to Computer Scientists and Engineers. *IEEE Transactions on Education*, vol. 54, pp. 156-163.

Omondi, A. & Premkumar, B. (2007). *Residue Number Systems: Theory and Implementations*, Imperial College Press, London.

Parhami, B. (2000). *Computer Arithmetic: Algorithms and Hardware Design*, Oxford University Press.

Piestrak, S. J. (1994). Design of residue generators and multioperand modular adders using carry-save adders. *IEEE Transactions on Computers*, Vol. 423, No. 1, pp. 68-77.

Piestrak, S.J. (1995). A high speed realization of a residue to binary converter. *IEEE Transactions on Circuits and Systems-II*, Vol. 42, pp. 661-663.

Soderstrand, M.A. & et al. (1986). *Residue number system arithmetic: modern applications in digital signal processing*, IEEE Press.

Stouratitis, T. & Paliouras, V. (2001). Considering the alternatives in lowpower design. *IEEE Circuits and Devices*, Vol. 7, pp. 23-29.

Taylor, F.J. (1984). Residue arithmetic: a tutorial with examples. *IEEE Computer*, Vol. 17, pp. 50–62.

Vinod, A.P. & Premkumar, A.B. (2000). A residue to binary converter for the 4-moduli superset $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{–}1\}$. *Journal of Circuits, Systems and Computers*, Vol. 10, pp. 85-99.

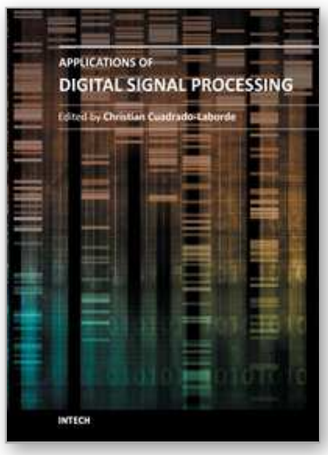
Wang, Y. (2000). Residue-to-Binary Converters Based on New Chinese remainder theorems. *IEEE Transactions on Circuits and Systems-II*, Vol. 47, No. 3, pp. 197-205.

Wang, Z., Jullien, G. A. & Miller, W. C. (2000). An Improved Residue-to-Binary Converter. *IEEE Transactions on Circuits and Systems-I*, Vol. 47, No. 9, pp. 1437-1440.

Wang, Y., Song, X., Aboulhamid, M. & Shen, H. (2002). Adder based residue to binary numbers converters for $(2^n\text{-}1, 2^n, 2^n\text{+}1)$. *IEEE Transactions on Signal Processing*, Vol. 50, No. 7, pp. 1772-1779.

Wang, W., Swamy, M. N. S., Ahmad, M. O. & Wang, Y. (2000b). A high-speed residue-to-binary converter and a scheme of its VLSI implementation. *IEEE Transactions on Circuits and Systems-II*, Vol. 47, No. 12, pp. 1576–1581.

Zhang, W. & Siy, P. (2008). An efficient design of residue to binary converter for four moduli set $(2^n\text{–}1, 2^n\text{+}1, 2^{2n}\text{–}2, 2^{2n+1}\text{–}3)$ based on new CRT II. *Elsevier Journal of Information Sciences*, Vol. 178, No. 1, pp. 264-279.



Applications of Digital Signal Processing

Edited by Dr. Christian Cuadrado-Laborde

ISBN 978-953-307-406-1
Hard cover, 400 pages
Publisher InTech
Published online 23, November, 2011
Published in print edition November, 2011

In this book the reader will find a collection of chapters authored/co-authored by a large number of experts around the world, covering the broad field of digital signal processing. This book intends to provide highlights of the current research in the digital signal processing area, showing the recent advances in this field. This work is mainly destined to researchers in the digital signal processing and related areas but it is also accessible to anyone with a scientific background desiring to have an up-to-date overview of this domain. Each chapter is self-contained and can be read independently of the others. These nineteenth chapters present methodological advances and recent applications of digital signal processing in various domains as communications, filtering, medicine, astronomy, and image processing.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen