Journal of mathematics and computer science 9 (2014), 438-450

Contents list available at JMCS
Journal of Mathematics and Computer Science
Journal Homepage: www.tjmcs.com

High Speed Reverse Converter for the Five Moduli Set $\{2^n, 2^n\text{-}1, 2^n\text{+}1, 2^n\text{-}3, 2^{n-1}\text{-}1\}$

Mohammad Esmaeildoust[1], Amer Kaabi[1]

[1]*Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Iran*

m_doust@kmsu.ac.ir

Article history:
Received December 2013
Accepted February 2014
Available online February 2014

Abstract

The new moduli set $\{2^n, 2^n-1, 2^n+1, 2^n-3, 2^{n-1}-1\}$ is profitable to construct high performance residue number system (RNS) due to well-formed moduli set and high dynamic range. Conversion from residues to binary is a bottleneck in RNS. With growth of number of moduli, this problem has been more critical due to complex multiplicative inverses. In this paper a high speed design of reverse converter for the moduli set $\{2^n, 2^n-1, 2^n+1, 2^n-3, 2^{n-1}-1\}$ is presented. This design is derived by using mix radix conversion (MRC) in three stages. Converter architecture is adder based which is suitable to realize efficient VLSI implementation. Proposed architecture has better delay performance compared to other reverse converters for five moduli sets in literature.

***Keywords**: residue numbers system, reverse converter, mixed radix conversion, digital circuits, VLSI design.*

1. Introduction

Residue number system (RNS) is non-weighted number system which its carry free property results in high speed arithmetic operation such as addition, subtraction, and multiplication. Due to these inherent futures, RNS is suitable to achieve fast and low power architecture in application such Digital Signal Processing (DSP) [1], Image Processing [2], and Cryptography [3] where dominant operation are addition and multiplication.

Choice of moduli is the first step in designing the RNS system. RNS system is described as moduli set which include set of relatively prime integers. The dynamic range is interval of possible number representation in RNS which is equal to product of the moduli. Moduli selection plays an important role in the design of the RNS system [4]. Speed and complexity of arithmetic operations in RNS architecture which consist of binary-to-residue (forward) converter, arithmetic unit, and residue-to-binary (reverse) converter are depends on appropriate moduli selection. The most popular 3n-bit dynamic range moduli set is $\{2^n-1, 2^n, 2^n+1\}$. Forward converters for moduli in the form of 2^n-1, 2^n, and 2^n+1 are simpler than other form of moduli [4-6]. This moduli set enjoys arithmetic friendly moduli and the best report for its reverse converter presented in [7]. Many works have been done to

choose the especial moduli sets such as {2^{n-1}–1, 2^n–1, 2^n} [8] and {2^n–1, 2^n, 2^{n+1}–1} [9]. By increasing the computations bit length in modern application, the dynamic ranges provided by 3n-bit moduli sets are not adequate, so large dynamic moduli set such as {2^n-1, 2^n, 2^n+1, 2^{n+1}-1} [10], {2^n-1, 2^n, 2^{n-1}-1, 2^{n+1}-1} [11], {2^n-3, 2^n-1, 2^n+1, 2^n+3} [12], {2^n-1, 2^n, 2^n+1, 2^{2n} + 1} [13], {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1} [14],{2^n, 2^{2n+1}-1, $2^{n/2}$-1, $2^{n/2}$+1, 2^n+1}[15] and {2^n, $2^{n/2}$-1, $2^{n/2}$+1, 2^n+1, 2^{2n-1}-1} [16] have been introduced. Disadvantage of moduli sets like {2^n-1, 2^n, 2^n+1, 2^{2n} + 1} [13], {2^n, 2^{2n+1}-1, $2^{n/2}$-1, $2^{n/2}$+1, 2^n+1} [15], and {2^n, $2^{n/2}$-1, $2^{n/2}$+1, 2^n+1, 2^{2n-1}-1} [16] is the imbalance modulo channels due to the large bit-width differences between the modulus. Besides, these moduli sets enjoy efficient and simple implementation of the reverse converters. Five moduli set {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1} [14] are designed for even values of n and has balanced moduli with efficient arithmetic operations. High latency of reverse converter is the main disadvantages of this work.
In this paper, a new balanced five moduli set {2^n-1, 2^n+1, 2^n-3, 2^n, 2^{n-1}-1} for even values of n is presented. Also an efficient adder based reverse converter based on mixed radix conversion technique in three stages is proposed. The proposed reverse converter is implemented with faster hardware compared to balanced five moduli set {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1} [14]. This paper is organized as follows; a brief introduction of the RNS with description of mixed radix conversion is presented in section II. An efficient design of reverse converter for balanced moduli set {2^n-1, 2^n+1,2^n-3, 2^n, 2^{n-1}-1} is presented in section III. Details of the delay and area of the proposed reverse converter are evaluated and comparisons are done in section IV and finally section V concludes the paper.

2. Background

A Residue Number System is defined by a set of m integer {P_1, P_2, …, P_m} called moduli set. Moduli set has the property of gcd (P_i, P_j) = 1 for each couple of P_i where$i \neq j, i \in \{0,1,\ldots,m\}$.
A weighted number X can be represented in Residue Number System as $X = (x_1, x_2,\ldots, x_m)$, $0 \leq xi<Pi$ where x_i means the reminder of X in modulo P_i or $\left|X\right|_{P_i}$. Term M=$i=1mPi$ is equal to multiplication of all moduli P_i is named Dynamic Range. If X is chosen in the range [0,M), integer Xwill have a unique representation in RNS.
One of the most important operations in RNS is conversion from residue number representation to binary system. To achieve the binary number from its residues, Mixed Radix Conversion (MRC) and Chinese Reminder Theorem (CRT) can be used. The proposed reverse converter is designed by using four stages of MRC. Therefore in the following, MRC will be presented.

Theorem 1: The number X = (x_1, x_2, …, x_m) in RNS representation can be converted to binary system by using MRC as follows.

$$X = v_m \prod_{i=1}^{m-1} P_i + \ldots + v_3 P_2 P_1 + v_2 P_1 + v_1 \tag{1}$$

Where $v_1 = x_1$ and

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

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

In general

$$v_m = \left| \begin{array}{l} (((x_m - v_1)\left|P_1^{-1}\right|_{P_m} - v_2)\left|P_2^{-1}\right|_{P_m} \\ \quad - \cdots - v_{m-1})\left|P_{m-1}^{-1}\right|_{P_m} \end{array} \right|_{P_m}$$

$\left|P_i^{-1}\right|_{P_j}$ denotes the multiplicative inverse of P_i in modulus P_j. There are two lemmas that we can benefit from them during calculation of v_i.

Lemma 1: In modulo 2^n-1, multiplication of n-bit residue number x by 2^k is equal to k-bit circular left shift residue number x [17].

Lemma2: In modulo 2^n-1, the negative of residue number x is obtained by one's complement of x, where $0 \leq x < 2^n - 1$ [17].

3. Reverse Converter Design

The proposed reverse converter is designed with four stages. Figure 1 shows the different stages of the reverse converter.

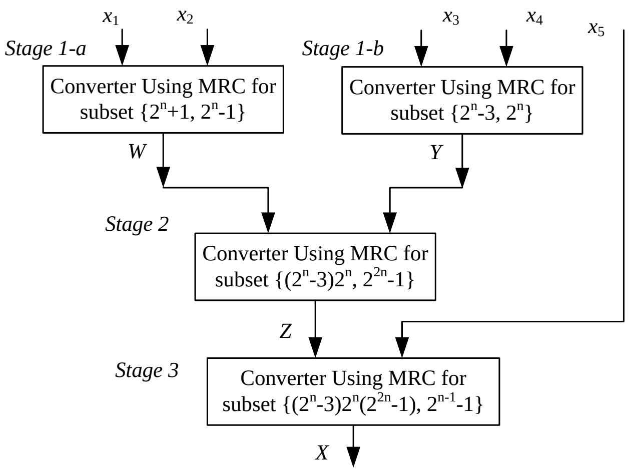

Figure 1. Three stage design of the reverse converter

3.1 Converter design for subset {2^n+1, 2^n-1}

By using MRC for subset {2^n+1, 2^n-1} and considering $x_1 = x_{1,n}\ldots x_{1,0}$ and $x_2 = x_{2,n-1}\, x_{2,n-2}\ldots x_{2,0}$, we have

$$W = v_1 + P_1 v_2 \tag{2}$$

Where

$$v_1 = x_1$$

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

For the required multiplicative inverses we have

$$\left|P_1^{-1}\right|_{P_2} = 2^{n-1}$$

$$\left|\left|P_1^{-1}\right|_{P_2}\left(2^n+1\right)\right|_{2^n-1} = 1 \ \rightarrow\ \left|2^{n-1}\left(2^n+1\right)\right|_{2^n-1} = 1$$

Therefore for v_2 we have

$$v_2 = \left|2^{n-1}\left(x_2 - x_1\right)\right|_{2^n-1} \tag{3}$$

By using Lemma 1 and 2 and considering x_1 and x_2 in binary form, Eq. (3) can be rewritten as

$$v_2 = \left|2^{n-1}\begin{pmatrix} x_{2,n-1}x_{2,n-2}\ldots x_{2,0} - \underbrace{00\ldots0}_{n-1\ bit}x_{1,n} \\ -x_{1,n-1}x_{1,n-2}\ldots x_{1,0} \end{pmatrix}\right|_{2^n-1} \tag{4}$$

$$v_2 = \left|\begin{matrix} \underbrace{x_{2,0}x_{2,n-1}\ldots x_{2,1}}_{v_{23}} + \underbrace{\bar{x}_{1,n}\underbrace{11\ldots1}_{n-1\ bit}}_{v_{22}} \\ +\underbrace{\bar{x}_{1,0}\bar{x}_{1,n-1}\ldots\bar{x}_{1,1}}_{v_{21}} \end{matrix}\right|_{2^n-1} \tag{5}$$

Therefore

$$v_2 = \left|v_{21} + v_{22} + v_{23}\right|_{2^n-1} \tag{6}$$

Where

$$v_{2,1} = \left|x_{2,0}x_{2,n-1}\ldots x_{2,1}\right|_{2^n-1}$$

$$v_{2,2} = \left|\bar{x}_{1,n}\underbrace{11\ldots1}_{n-1\ bit}\right|_{2^n-1}$$

$$v_{2,3} = \left|\bar{x}_{1,0}\bar{x}_{1,n-1}\ldots\bar{x}_{1,1}\right|_{2^n-1}$$

v_2 can be implemented by using one Carry Save Adder (CSA) with End Around Carry (EAC) followed by a Modulo 2^n-1 adder. Therefore to calculate W we have

$$W = x_1 + \left(2^n + 1\right)v_2 \tag{7}$$

Since v_2 has n-bit in binary, therefore it can be concatenate at the end of $v_2\underbrace{00\cdots0}_{n}$, therefore

$$W = x_1 + v_2v_2 \tag{8}$$

Figure 2 shows the hardware implementation of v_2 and W.

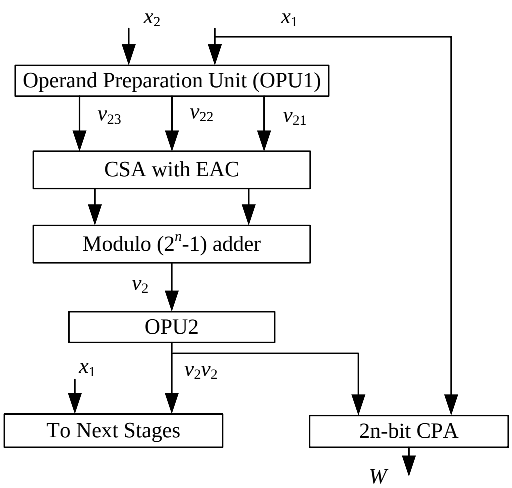

Figure 2. Hardware implementation of v_2 and W

OPU1 provides the required shift and NOT gates according to Eq. (5). After calculation of v_2 the required concatenation in Eq. (8) are done by using OPU2. Note that x_1 and $v_2v_2$are sent to next stages and the calculation of W is parallel with stage three of the design.

3.2 Converter Design for Subset {2^n-3, 2^n}

By using MRC for the moduli set {2^n-3, 2^n} and considering $x_3 = x_{3,n\text{-}1}\ldots x_{3,0}$ and $x_4 = x_{4,n\text{-}1}\ldots x_{4,0}$, it holds that

$$Y = v_3 + P_3 v_4 \tag{9}$$

where

$$v_3 = x_3$$

$$v_4 = \left|(x_4 - v_3)\left|P_3^{-1}\right|_{P_4}\right|_{P_4}$$

For the required multiplicative inverse in Eq. (9), we have

$$\left|\left|P_3^{-1}\right|_{P_4} \times (-3)\right|_{2^n} = 1 \ \rightarrow\ \left|P_3^{-1}\right|_{P_4} = \left|-\frac{1}{3}\right|_{2^n}$$

$$\left|P_3^{-1}\right|_{P_4} = \sum_{i=0}^{i=n/2-1} 2^{2i}$$

Proof.

$$\left|\left|P_3^{-1}\right|_{P_4} \times (-3)\right|_{2^n} = \left|\sum_{i=0}^{i=n/2-1} 2^{2i} \times \left(1-2^2\right)\right|_{2^n}$$

$$= \left|(1+2^2+2^4+\cdots+2^{n-2}) \times \left(1-2^2\right)\right|_{2^n} = 1$$

Therefore

$$v_4 = \left|\begin{matrix}\left(1+2^2+2^4+\cdots+2^{n-2}\right)\times \\ \left(x_{4,n-1}x_{4,n-2}\ldots x_{4,0} - x_{3,n-1}x_{3,n-2}\ldots x_{3,0}\right)\end{matrix}\right|_{2^n} \tag{10}$$

$$v_4 = \left|\begin{matrix}LS\left(x_4,0\right)+\ldots+LS\left(x_4,n-2\right)+ \\ LS\left(\bar{x}_3,0\right)+\ldots+LS\left(\bar{x}_3,n-2\right)\end{matrix}\right|_{2^n} \tag{11}$$

$LS(k, p)$ denotes p-bit left shift of k. After calculation of v_4, we have

$$Y = x_3 + \left(2^n - 3\right)v_4 = v_4 x_3 - 3v_4 \tag{12}$$

$$Y = v_{42} - v_{41} - v_4 \tag{13}$$

where

$$v_{41} = v_4 0$$
$$v_{42} = v_4 x_3$$

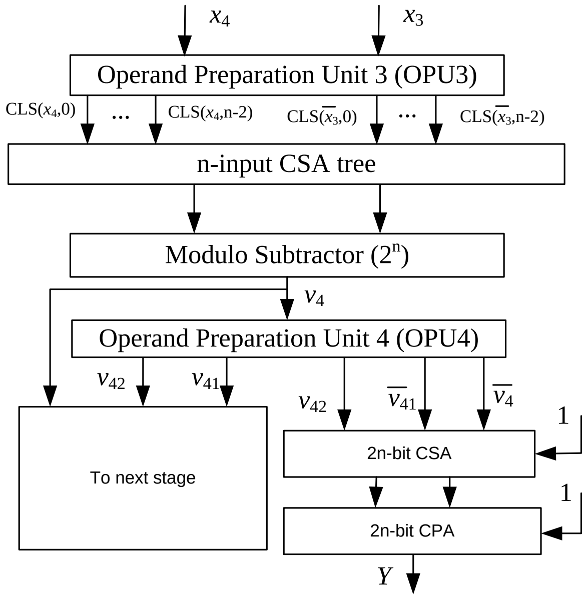

Figure 3. Hardware implementation of Y

Hardware implementation of Y is shown in figure 3. OPU3 provides the required shift and negation in Eq. (11). After that CSA tree followed by modulo 2^n adder calculates v_4. Now, in order to realize$Y$, OPU4 is employed to provide intermediate variables v_{41}, $v_{42}$and their negations. Then $2n$-bit CSA followed by $2n$-bit CPA calculates Y. Note that the variables $v_{41}$and v_{42} before calculation of $Y$are sent to next stage and the operations in next stage are in parallel with calculation of Y.

3.3 Converter Design for Subset {2^{2n}-1, 2^n (2^n-3)}

By using MRC for the superset {2^{2n}-1, 2^n(2^n-3)},we have

$$Z = v_5 + P_{3,4} v_6 \tag{14}$$

where

$$v_5 = Y$$

$$v_6 = \left| (W - Y) \left| P_{3,4}^{-1} \right|_{P_{1,2}} \right|_{P_{1,2}}$$

The multiplicative inverse for Eq.(14) is

$$\left| \left| P_{3,4}^{-1} \right|_{P_{1,2}} \times 2^n \times \left(2^n - 3 \right) \right|_{2^{2n}-1} = 1 \rightarrow$$
$$\left| P_{3,4}^{-1} \right|_{P_{1,2}} = 2^{2n} - 2^{2n-3} - 2^{n-2} - 2^{n-3} - 1$$

Then Eq.(14) can be rewritten as follows

$$v_6 = \left| \begin{array}{c} (W - Y)\times \\ (2^{2n} - 2^{2n-3} - 2^{n-2} - 2^{n-3} - 1) \end{array} \right|_{2^{2n}-1} \tag{15}$$

By replacing *W* from Eq. 8 and *Y* form Eq. 13, results in

$$v_6 = \left| \begin{array}{c} (x_1 + v_2 v_2 - v_4 x_3 + v_4 0 + v_4)\times \\ (-2^{2n-3} - 2^{n-2} - 2^{n-3}) \end{array} \right|_{2^{2n}-1} \tag{16}$$

In binary representation Eq. 16 can be rewritten as

$$v_6 = \left| \begin{array}{l} -x_{1,2}x_{1,1}x_{1,0}\underbrace{00\ldots0}_{n-1}x_{1,n}\ldots x_{1,3} - \\ 0x_{1,n}\ldots x_{1,0}\underbrace{00\ldots0}_{n-2} - 00x_{1,n}\ldots x_{1,0}\underbrace{00\ldots0}_{n-3} \\ -v_{2,2}v_{2,1}v_{2,0}v_{2,n-1}\ldots v_{2,0}v_{2,n-1}\ldots v_{2,3} - \\ v_{2,1}v_{2,0}v_{2,n-1}\ldots v_{2,0}v_{2,n-1}\ldots v_{2,2} \\ -v_{2,2}v_{2,1}v_{2,0}v_{2,n-1}\ldots v_{2,0}v_{2,n-1}\ldots v_{2,3} - \\ v_{4,1}v_{4,0}\underbrace{00\ldots0}_{n}v_{4,n-1}\ldots v_{4,2} \\ -0v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-1} - 00v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-2} - \\ v_{4,2}v_{4,1}v_{4,0}\underbrace{00\ldots0}_{n}v_{4,n-1}\ldots v_{4,3} \\ -00v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-2} - 000v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-3} + \\ x_{3,2}x_{3,1}x_{3,0}v_{4,n-1}\ldots v_{4,0}x_{3,n-1}\ldots x_{3,3} \\ +v_{4,1}v_{4,0}x_{3,n-1}\ldots x_{3,0}v_{4,n-1}\ldots v_{4,2} + \\ v_{4,2}v_{4,1}v_{4,0}x_{3,n-1}\ldots x_{3,0}v_{4,n-1}\ldots v_{4,3} \end{array} \right|_{2^{2n}-1} \tag{17}$$

Using Lemma 1 and 2 results in

$$v_6 = \left| \begin{array}{l} -x_{1,2}x_{1,1}x_{1,0}\underbrace{00\ldots0}_{n-1}x_{1,n}\ldots x_{1,3} - \\ 0x_{1,n}\ldots x_{1,0}\underbrace{00\ldots0}_{n-2} - 00x_{1,n}\ldots x_{1,0}\underbrace{00\ldots0}_{n-3} \\ -v_{2,2}v_{2,1}v_{2,0}v_{2,n-1}\ldots v_{2,0}v_{2,n-1}\ldots v_{2,3} - \\ v_{2,1}v_{2,0}v_{2,n-1}\ldots v_{2,0}v_{2,n-1}\ldots v_{2,2} \\ -v_{2,2}v_{2,1}v_{2,0}v_{2,n-1}\ldots v_{2,0}v_{2,n-1}\ldots v_{2,3} - \\ v_{4,1}v_{4,0}\underbrace{00\ldots0}_{n}v_{4,n-1}\ldots v_{4,2} \\ -0v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-1} - 00v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-2} - \\ v_{4,2}v_{4,1}v_{4,0}\underbrace{00\ldots0}_{n}v_{4,n-1}\ldots v_{4,3} \\ -00v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-2} - 000v_{4,n-1}\ldots v_{4,0}\underbrace{00\ldots0}_{n-3} + \\ x_{3,2}x_{3,1}x_{3,0}v_{4,n-1}\ldots v_{4,0}x_{3,n-1}\ldots x_{3,3} \\ +v_{4,1}v_{4,0}x_{3,n-1}\ldots x_{3,0}v_{4,n-1}\ldots v_{4,2} + \\ v_{4,2}v_{4,1}v_{4,0}x_{3,n-1}\ldots x_{3,0}v_{4,n-1}\ldots v_{4,3} \end{array} \right|_{2^{2n}-1} \tag{18}$$

For simplicity Eq.(18) can be rewritten as

$$v_6 = \left| \begin{array}{l} v_{61} + v_{62} + v_{63} + v_{64} + v_{65} + v_{66} + v_{67} \\ + v_{68} + v_{69} + v_{610} + v_{611} + v_{612} + v_{613} \end{array} \right|_{2^{2n}-1} \quad (19)$$

Hardware implementation of v_6 includes CSA with EAC followed by MA(2^{2n}-1). Figure 4 shows the hardware implementation of v_6. After calculation of v_6, we have

$$Z = Y + 2^n \left(2^n - 3\right) v_6 \quad (20)$$

$$Z = Z_1 + Z_2 + Z_3 \quad (21)$$

where

$$\begin{array}{l} Z_1 = v_6 Y \\ Z_2 = -2^{n+1} v_6 \\ Z_3 = -2^n v_6 \end{array} \quad (22)$$

Figure 4 illustrates the hardware implementation of *Z*. such as described in previous stages, Z_1, Z_2, and Z_3 are sent to next stages in parallel with calculation of *Z*.

3.4 Converter Design for Subset {(2^{2n}-1) 2^n (2^n-3), 2^{n-1}-1}

Using MRCfor superset {(2^{2n}-1)×2^n× (2^n-3),2^{n-1}-1}, results in

$$X = v_7 + P_{1,2,3,4} v_8 \quad (23)$$

where

$$v_7 = Z$$

$$v_8 = \left| \left(x_5 - Z\right) \left| P_{1,2,3,4}^{-1} \right|_{P_5} \right|_{P_5}$$

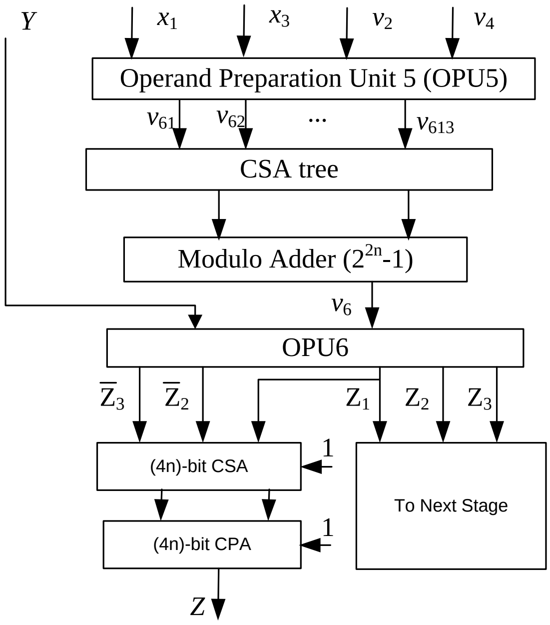

Figure 4. Hardware implementation of *Z*

The multiplicative inverse in Eq. (21) is obtained by

$$\left|\left|P_{1,2,3,4}{}^{-1}\right|_{P_5} \times 2^n \times \left(2^n - 3\right) \times \left(2^{2n} - 1\right)\right|_{2^{n-1}-1} = 1 \;\rightarrow\; \left|P_{1,2,3,4}{}^{-1}\right|_{P_5} = \left|-\frac{1}{6}\right|_{2^{n-1}-1}$$

$$\left|P_{1,2,3,4}{}^{-1}\right|_{P_5} = \left|-\frac{2^{n-2}}{3}\right|_{2^{n-1}-1}$$

$$\left|P_{1,2,3,4}{}^{-1}\right|_{P_5} = \left|-2^{n-2} \times \sum_{i=0}^{n/2-1} 2^{2i}\right|_{2^{n-1}-1}$$

$$= \left|-2^{n-2}\left(1 + 2^2 + 2^4 + \ldots + 2^{n-2}\right)\right|_{2^{n-1}-1}$$

By replacing the multiplicative inverse is Eq. (21), v_8 can be rewritten by Eq. (21).

$$v_8 = \left|\begin{matrix}\left(x_5 - Z\right)\times \\ \left(2^{n-2} + 2^n + 2^{n+2} + \ldots + 2^{2n-4}\right)\end{matrix}\right|_{2^{n-1}-1} \tag{24}$$

By replacing Z from Eq. 22 in Eq. 25 results in

$$v_8 = \left|\begin{matrix}\left(x_5 - (Z_3 + Z_2 + Z_1)\right)\times \\ \left(2^{n-2} + 2^n + 2^{n+2} + \ldots + 2^{2n-4}\right)\end{matrix}\right|_{2^{n-1}-1} \tag{25}$$

$$v_8 = \left|\begin{matrix}\left((x_5 - (Z_3 + Z_2 + Z_1)\right)\times \\ \left(2 + 2^3 + \ldots + 2^{n-3} + 2^{n-2}\right)\end{matrix}\right|_{2^{n-1}-1} \tag{26}$$

By replacing Z_2 and Z_3 form Eq. 22 and simplifying modulo 2^{n-1}-1 results in

$$v_8 = \left|\begin{matrix}\left(Z_1 - x_5\right)\left(2^{n-2} + 2 + 2^3 + \ldots + 2^{n-3}\right) - \\ \left(v_6 0 + v_6\right)\left(1 + 2^2 + 2^4 + \ldots + 2^{n-2}\right)\end{matrix}\right|_{2^{n-1}-1} \tag{27}$$

By adjusting the number of bits for Z_1, $v_6 0$ and $v_6$we have

$$v_8 = \left|\begin{matrix}\left(\begin{matrix}\left(\bar{x}_5 + L_0 + L_1 + L_2 + L_3 + L_4\right)\times \\ \left(2^{n-2} + 2 + 2^3 + \ldots + 2^{n-3}\right)\end{matrix}\right) + \\ \left(\begin{matrix}\left(\bar{L}_5 + \bar{L}_6 + \bar{L}_7 + \bar{L}_8 + \bar{L}_9 + \bar{L}_{10}\right)\times \\ \left(1 + 2^2 + 2^4 + \ldots + 2^{n-2}\right)\end{matrix}\right)\end{matrix}\right|_{2^{n-1}-1} \tag{28}$$

Where

$$L_0 = Y_{n-2} Y_{n-3} \ldots Y_0$$

$$L_1 = Y_{2n-3} Y_{2n-4} \ldots Y_{n-1}$$

$$L_2 = v_{6,n-4} v_{6,n-5} \ldots v_{6,0} Y_{2n-1} Y_{2n-2}$$

$$L_3 = v_{6,2n-5} v_{6,2n-6} \ldots v_{6,n-3}$$

$$L_4 = \underbrace{00\ldots0}_{n-5} v_{6,2n-1} v_{6,2n-2} v_{6,2n-3} v_{6,2n-4}$$

$$L_5 = v_{6,n-3} v_{6,n-4} \ldots v_{6,0} 0$$

$$L_6 = v_{6,2n-4} v_{6,2n-5} \ldots v_{6,n-2}$$

$$L_7 = \underbrace{00\ldots0}_{n-4} v_{6,2n-1} v_{6,2n-2} v_{4,2n-3}$$

$$L_8 = v_{6,n-2} v_{6,n-3} \ldots v_{6,0}$$

$$L_9 = v_{6,2n-3} v_{6,2n-4} \ldots v_{6,n-1}$$

$$L_{10} = \underbrace{00\ldots0}_{n-3} v_{6,2n-1} v_{6,2n-2}$$

After using CAS tree in Eq (28), we have

$$v_8 = \left| \begin{array}{l} (S_0 + C_0)\left(2^{n-2} + 2 + 2^3 + \ldots + 2^{n-5}\right) + \\ (S_1 + C_1)\left(1 + 2^2 + 2^4 + \ldots + 2^{n-4}\right) \end{array} \right|_{2^{n-1}-1} \tag{29}$$

Then in order to get the result of multiplication CLS can be utilized, so we have

$$v_8 = \left| \begin{array}{l} CLS(S_0, n-2) + CLS(S_0, 1) \\ + \ldots + CLS(S_0, n-5) + \\ CLS(C_0, n-2) + CLS(C_0, 1) \\ + \ldots + CLS(C_0, n-5) + S_1 + C_1 + \\ CLS(S_1, 2) + \ldots + CLS(S_1, n-4) + \\ CLS(C_1, 2) + \ldots + CLS(C_1, n-4) \end{array} \right|_{2^{n-1}-1} \tag{30}$$

$$X = Z + 2^n \times \left(2^n - 3\right) \times \left(2^{2n} - 1\right) v_8 \tag{31}$$

$$\begin{array}{l} X = v_8 Z - 2^{3n+1} v_8 - 2^{3n} v_8 \\ -2^{2n} v_8 + 2^{n+1} v_8 + 2^n v_8 \end{array} \tag{32}$$

$$\begin{array}{l} X = v_8 Z - v_8 00 v_8 \underbrace{00\ldots0}_{2n} - 2^{3n} v_8 \\ +2^{n+1} v_8 + 2^n v_8 \end{array} \tag{33}$$

$$X = v_8 Z + \underbrace{11\ldots1}_{n-1}\bar{v}_8 11\bar{v}_8\underbrace{11\ldots1}_{2n} + \underbrace{11\ldots1}_{n}\bar{v}_8\underbrace{11\ldots1}_{3n} + v_8\underbrace{00\ldots0}_{n+1} + v_8\underbrace{00\ldots10}_{n} \tag{34}$$

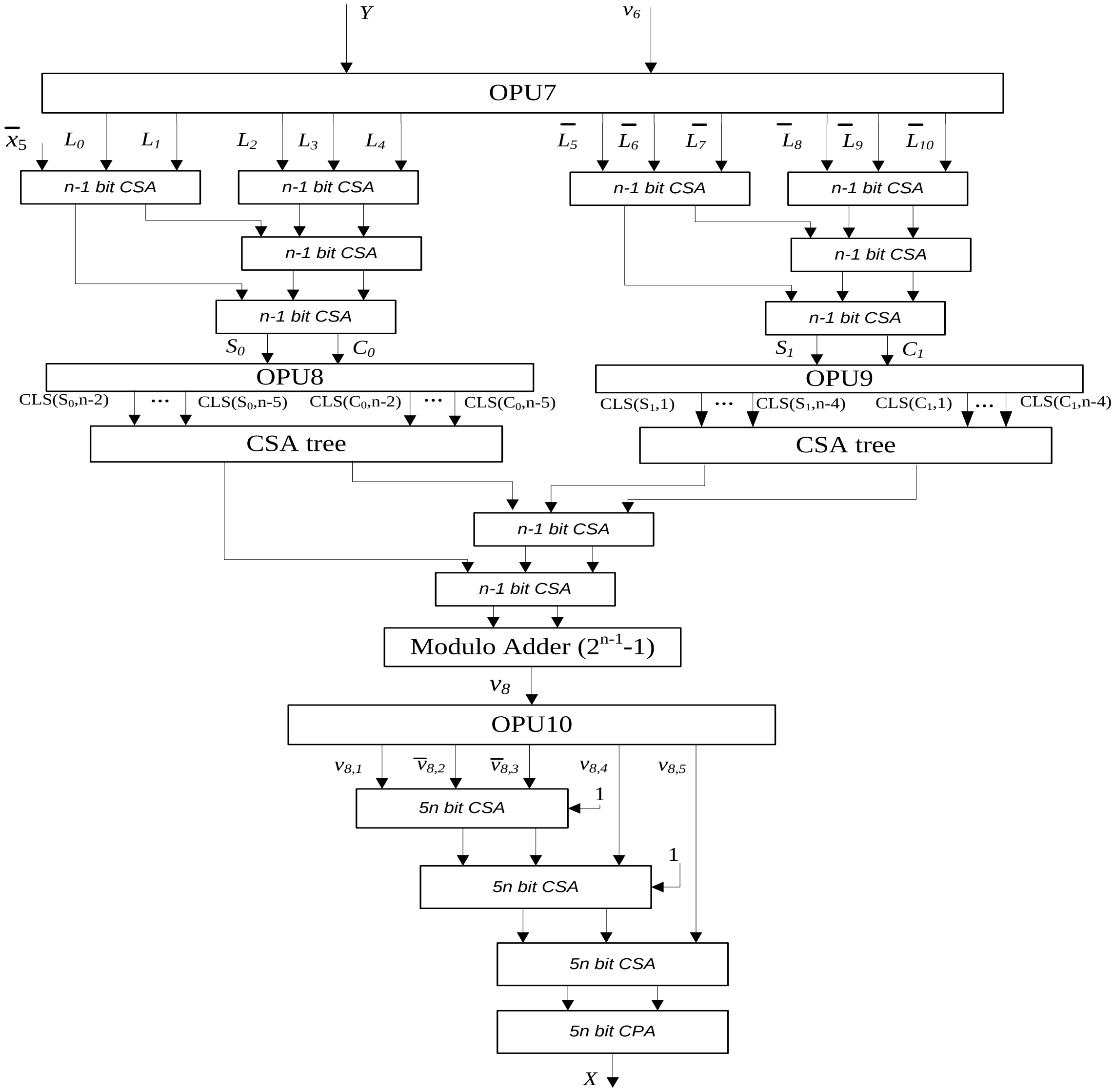

Figure 5. Hardware implementation of X

Figure 5 shows the hardware implementation of X.

3.5 Numerical Example

For n = 6, the moduli set of proposed RNS is {65, 63, 61, 64, 31} with DR equal to 495593280. Let the RNS number of X be 19262232 = (2, 45, 18, 24, 10), weighted number X can be achieved by following step:

1. Binary representation of residue are as bellow:

x_1=0000010
x_2=101101

x_3=010010
x_4=011000
x_5=01010
2. Obtaining W:
v_2=110101

W = 0000010 + 110101110101 = 110101110111

3. Obtaining Y:
v_4=111110
Y=111110010010-1111100-111110=111011011000
4. Obtaining Z:
v_6=001101000110
Z = 001101000110111011011000 -001101000110000000-001101000110000000

Z= 001100011111101001011000=3275352

1. Obtaining X:
v_8 =00001

X = 00001001100011111101001011000 – 0000100000001000000000000 - 00001000000000000000000 + 000010000000+00001000000

X =00001001100011111101001011000 + 11111111101111110111111111111 +*11111*1111011111111111111111111 + 000010000000 + 00001000000 +10
= *000001001001011110101100011000* = *000001001001011110101100011000* = **19262232**

4. Performance evaluation

In this section, performance comparison for new five moduli set $\{2^n, 2^n\text{-}1, 2^n\text{+}1, 2^n\text{-}3, 2^{n-1}\text{-}1\}$ with $\{2^n, 2^n\text{-}1, 2^n\text{+}1, 2^{n+1}\text{-}1, 2^{n-1}\text{-}1\}$ [14] which is the other well-known moduli set in this class is reported. Comparison is done between these moduli set in terms of delay and area. As represented in table 1 the reverse converter for proposed moduli set has a total delay of (13n+ H +12)t_{FA} while total delay of presented reverse converter in [14] is (18n+L +7)t_{FA} where t_{FA} denotes delay of one full adder cell, so delay of proposed reverse converter has about 39% improvement. In a better scenario unit gate model is used for fair comparison in terms of total delay and hardware cost. Based on this model each two monotonic gates and XOR/XNOR gates counts one and two gates in area and delay respectively, one bit full adder cell has seven gates area and four gates delay. As results that are depictured in table 1, proposed design has noticeable improvement in terms of delay in compare with other case.

Table 1. Delay and area comparison of the proposed reverse converter

Converter	Hardware requirements	Unit gate area	Conversion delay	Unit gate delay
[14]	$((5n^2+43n+m^*)/6+16n-1)A_{FA}+(6n+1)A_{NOT}$	$(5n^2+43n+m^*)7/6+118n-6$	$(18n+L^*+7)t_{FA}$	$72n+4L^*+28$
Proposed	$(3n^2+36n+3)A_{FA}+(n-1)A_{HA}+(2n+2)A_{XNOR}+(2n+2)A_{OR}+(11n-11)A_{XOR}+(11n-11)A_{AND}+(9n-1)A_{NOT}$	$(3n^2+36n+3)7+59n-32$	$(13n+H^*+12)t_{FA}$	$52n+4H^*+48$

*m=n-4, 9n-12 and 5n-8 for n=6k-2,6k and 6k +2, respectively, L is the number of the levels of a CSA tree with ((n/2) +1) inputs and H is the number of the levels a CSA with (n-2) inputs.

5. Conclusion

In this paper a new reverse converter architecture for the new balanced moduli set $\{2^n, 2^n-1, 2^n+1, 2^n-3, 2^{n-1}-1\}$ has been proposed. Converter architecture is adder based and does not need any ROM. Conversion technique is based on MRC, in three levels architecture. With this method about 39% speed up together with less hardware requirements are gained compared to the balanced five moduli set in literature.

6. ACKNOWLEDGEMENTS

We would like to thank Khorramshahr University of Marine Science and Technology for supporting this work under research grant contract No. 59.

References

[1] J. Ramirez, U. Meyer-Base, A. Garcia, “Efficient RNS-based Design of Programmable FIR Filters Targeting FPL Technology," *Journal of Circuits, Systems and Computers*, vol. 14, no. 1, (2005) 165-177.

[2] W. Wei, Swamy, M.N.S. ; Ahmad, M.O., “RNS application for digital image processing,” Proceedings of the 4th IEEE international workshop on system-on-chip for real time applications, 2004.

[3] M. Esmaeildoust, D. Schinianakis, H.Javashi, T. Stouraitis, K. Navi, "Efficient RNS Implemenation of Elliptic Curve Point Multiplication Over GF(p)," IEEE Transaction on Very Large Scale Integration (VLSI) Systems, Vol. 21 , No. 8 2013, 1545 - 1549.

[4] K. Navi, A. S. Molahosseini, M. Esmaeildoust, "How to Teach Residue Number System to Computer Scientists and Engineers," IEEE Transactions on Education, Vol. 54, Issue. 1, pp. 156-163, 2011.

[5] I. Koren, *Computer Arithmetic Algorithms*, Prentice-Hall, 1993.

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

[7] Y. Wang, X. Song, M. Aboulhamid and H. Shen, “Adder based residue to binary numbers converters for $(2^n-1, 2^n, 2^n+1)$,” *IEEE Transactions on Signal Processing*, vol. 50, no. 7, (2002) 1772-1779.

[8] W. Wang, M. N. S. Swamy, M. O. Ahmad, and Y.Wang, “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, (2000) 1576–1581.

[9] P. V. A. Mohan, “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, (2007) 775-779.

[10] P. V. A. Mohan and A. B. Premkumar, “RNS-to-Binary Converters for Two Four-Moduli Set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}–1\}$ and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$,” *IEEE Transactions on Circuits and Systems-I*, vol. 54, no. 6, (2007) 1245-1254.

[11] M Esmaeildoust, K Navi, MR Taheri, AS Molahosseini, S Khodambashi, “Efficient RNS to binary converters for the new 4-moduli set {2n, 2n+ 1-1, 2n-1, 2n-1-1}”, IEICE Electronics Express 9 (1), (2002) 1-7.

[12] P.V.A. Mohan, “New reverse converters for the moduli set $\{2^n −3, 2^n −1, 2^n +1, 2^n +3\}$,” *Elsevier Journal of Electronics and Communications (AEU)*, vol. 62, no. 9, (2008) 643-658.

[13] B. Cao, C. H. Chang and T. Srikanthan, “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, (2003) 1296-1303.

[14] B. Cao, C.H. Chang and T. Srikanthan, “A Residue-to-Binary Converter for a New Five-Moduli Set,” *IEEE Transactions on Circuits and Systems-I*, vol. 54, no. 5, (2007) 1041–1049.

[15] Mohammad Esmaeildoust, Keivan Navi and Mohammad Reza Taheri, “High speed reverse converter for new five-moduli set $\{2^n, 2^{2n+1}-1, 2^{n/2}-1, 2^{n/2}+1, 2^n+1\}$”, *IEICE Electron. Express*, Vol. 7, No. 3, (2010) 118-125.

[16] A.S. Molahosseini, C. Dadkhah, K. Navi, “A New Five-Moduli Set for Efficient Hardware Implementation of the Reverse Converter,” *IEICE Electronics Express*, vol. 6, no. 14, (2009) 1006-1012.

[17] N Keivan, M Esmaeildoust, AS Molahosseini, “A General Reverse Converter Architecture with Low Complexity and High Performance”, IEICE TRANSACTIONS on Information and Systems 94 (2), (2011) 264-273.