Efficient and Effective Residue-to-Binary Converter for Balanced Four Moduli Set

MohammadReza Taheri[1], Zhale Amirjamshidi[2], Mohammad Esmaeildoust[3], Keivan Navi[4], Reza Sabbaghi-nadooshan[2]

[1]Microelectronic Laboratory, Shahid Beheshti University, GC, Tehran, Iran
[2]Electronic Engineering Department, Islamic Azad University, Central Tehran Branch, Tehran, Iran
[3]Faculty of Marine Engineering, Khoramshahr Marine Science and Technology University, Khoramshahr, Iran
[4]Faculty of Electrical and Computer Engineering, Shahid Beheshti University, GC, Tehran, Iran

Abstract- the moduli set $\{2^{n-1}-1, 2^{n+1}-1, 2^n, 2^n-1\}$ has been recently proposed in literature for class of 4*n*-bit dynamic range in residue number system. Due to lack of moduli in the form of 2^k+1, this moduli set enjoys efficient Arithmetical Unit (AU) in its architecture. In this paper, based on mixed radix conversion (MRC), a new residue-to-binary converter architecture in two levels of designs for aforesaid moduli set is proposed. The results show that the proposed reverse converter has hardware cost and conversion delay improvement in comparison with the related state-of-the-art.

Keywords: Mixed radix conversion, Residue arithmetic, Residue number system, Residue-to-binary converter, Digital design

1. Introduction

Carry-free nature of the residue number system (RNS) makes it suitable to be used in arithmetic level in VLSI design to achieve parallelism [1], [2]. In RNS, a weighted number is decomposed into a set of residues. RNS results in high-speed addition, subtraction and multiplication, because arithmetic operations on residues can be performed without carry propagation between residues [3], [4]. The RNS has many applications in digital signal processing (DSP) [5], [6], image processing [7], Cryptography like RSA and Elliptic Curve Cryptography (ECC) algorithm [8-9] and communication systems [10]. Other arithmetic operations like division, sign detection and comparison are the hard operations in RNS.

RNS is based on moduli set which consists of a set of relatively prime integers. The dynamic range of a RNS is defined in terms of the product of the moduli, and it denotes the interval of integers can be uniquely represented in RNS [11].

RNS includes three main parts: binary-to-residue (forward) converter, arithmetic operation and residue-to-binary (reverse) converter. Forward converter transforms a weighted binary number into a residue numbers, with attention to the moduli set. The arithmetic unit contains modular adder, subtractor, and multiplier. The reverse converter converts a residue numbers into a weighted binary number [11]. Careful selection of moduli set determines the efficiency of forward conversion, arithmetic operation and reverse conversion. Reverse converter has more complex architecture and its complexity will grow depending on the number of module. Therefore effective design of reverse converter is needed in order to get the benefit of the RNS [12].

Table I. Comparison of arithmetic operation for different moduli sets for high dynamic range applications

Moduli Set	Design	Critical modulus	Delay
$\{2^n–1, 2^n, 2^n+1, 2^{n+1}–1\}$	[15-17]	2^n+1	$2\log_2 n + 6$
$\{2^n–1, 2^n, 2^n+1, 2^{n+1}+1\}$	[14-15]	$2^{n+1}+1$	$2\log_2(n+1) + 6$
$\{2^n–3, 2^n–1, 2^n+1, 2^n+3\}$	[18]	2^n+3	$2\log_2(n-1) + 7$
$\{2^n, 2^{n+1}–1, 2^n–1, 2^{n-1}–1\}$	[19]	$2^{n+1}–1$	$2\log_2(n+1) + 3$

Many works are reported on balanced four moduli sets such as $\{2^n–1, 2^n, 2^n+1, 2^{n+1}–1\}$ [15-17], $\{2^n–1, 2^n, 2^n+1, 2^{n+1}+1\}$ [14-15], $\{2^n–3, 2^n–1, 2^n+1, 2^n+3\}$ [18] and $\{2^n, 2^{n+1}–1, 2^n–1, 2^{n-1}–1\}$ [19]. Efficiency of arithmetic operations is restricted to critical modulus. Critical modulus in the works reported in [14-19] are shown in Table I. The unit gate delay of the parallel prefix adders 2^k-1, 2^k+1 and 2^k+3 are $2\log_2 n + 3$, $2\log_2 n + 6$ and $2\log_2(n-1) + 7$, respectively [20-22]. Therefore as it's shown in Table I, moduli set $\{2^n, 2^{n+1}–1, 2^n–1, 2^{n-1}–1\}$ [19] provides better arithmetic unit. More efficient reverse converter for the moduli set $\{2^n, 2^{n+1}–1, 2^n–1, 2^{n-1}–1\}$ with less hardware requirement and delay compared to the work reported in [19] and other moduli sets in literature is needed to enjoy efficient reverse converter architecture together with efficient arithmetic operation. Therefore, in this paper, a new design of the reverse converter for the 4-moduli set $\{2^{n-1}-1, 2^{n+1}-1, 2^n, 2^n-1\}$ is presented. The proposed converter has achieved less delay and hardware requirement compared to the state-of-the-art works in literature.

This paper consists of a background about RNS in section 2, design of the proposed RNS to binary converter in section 3, comparison of the performance of the proposed RNS to binary converter with other moduli set in section 4 and finally conclusions of this paper is included in section 5.

2. Background

A residue number system is defined in terms of relatively prime moduli set $\{P_1, P_2, \ldots, P_n\}$ that is gcd $(P_i, P_j) = 1$ for $i \neq j$. An integer number X in the range [0, M-1] where $M = P_1 \times P_2 \ldots \times P_n$ is the dynamic range of the RNS system can be represented as $X = (x_1, x_2, \ldots, x_n)$ where $x_i = X \bmod P_i$, $0 \leq x \leq P_i$ [13].

The algorithms of reverse conversion are principally based on the Chinese remainder theorem (CRT), mixed-radix conversion (MRC), and new Chinese remainder theorems (New CRTs) [11].

By MRC and the moduli set $\{P_1, P_2, \ldots, P_n\}$, the number X can be calculated by using

$$X = v_n \prod_{i=1}^{n-1} P_i + \cdots + v_3 P_2 P_1 + v_2 P_1 + v_1 \tag{1}$$

The coefficients $\{v_1, v_2, \ldots, v_n\}$ can be achieved from

$$v_1 = x_1 \tag{2}$$

$$v_2 = \left| (x_2 - v_1) \left| {P_1}^{-1} \right|_{P_2} \right|_{P_2} \tag{3}$$

$$v_3 = \left| \left((x_3 - v_1) \left| {P_1}^{-1} \right|_{P_3} - v_2 \right) \left| {P_2}^{-1} \right|_{P_3} \right|_{P_3} \tag{4}$$

In general case

$$v_n = \left| \left(\left((x_n - v_1) \left| {P_1}^{-1} \right|_{P_n} - v_2 \right) \left| {P_2}^{-1} \right|_{P_n} - \cdots - v_{n-1} \right) \left| {P_{n-1}}^{-1} \right|_{P_n} \right|_{P_n} \tag{5}$$

Where $\left| {P_i}^{-1} \right|_{P_j}$ is the multiplicative inverse of P_i modulo P_j [11].

Three types of adder are used to realize the hardware architecture of reverse converter, Carry Save Adder (CSA) for operations in modulo 2^n, CSA with end around carry (EAC) for operations in modulo 2^k-1, Carry Propagate Adder (CPA) and Modular Adder (MA). For MA in modulo $2^k - 1$, CPA with end around carry (EAC) is used, which has similar area and double delay in comparison with regular CPA [24].

3. Proposed RNS to binary converter

The proposed RNS to binary converter for the module set $\{2^n-1, 2^{n+1}-1, 2^{n-1}-1, 2^n\}$ mainly follows MRC method, with two level of architecture, for achieving efficient implementation of reverse converter. In the first step of design, number *Y* is calculated from the residues in subset $\{2^n-1\,, 2^{n+1}-1\,, 2^{n-1}-1\}$ by using MRC in a parallel manner. In the second step MRC method is applied to superset $\{(2^{n-1}-1)(2^{n+1}-1)(2^n-1),\ 2^n\}$ and the final result is realized. We describe the proposed reverse converter scheme in two parts, as shown in Fig 1.

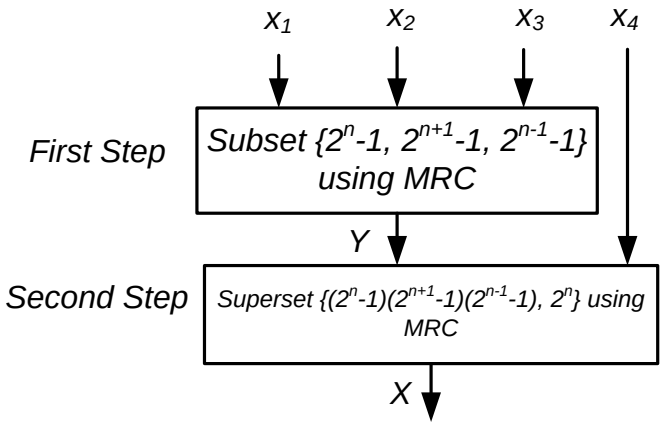

Fig. 1. Proposed Schema for residue-to-binary conversion

3.1.1. First step design

As discussed before, in the first step, reverse converter of subset $\{2^n-1, 2^{n+1}-1\,, 2^{n-1}-1\}$ will be designed. In this step in order to decrease the delay which follows from serial attribute of MRC method, proposed approach in [23] for more parallelism without noteworthy hardware redundancy is utilized. Also to reduce total architecture delay, in the first step all moduli are in the form of(2^k-1) and modulo 2^n will be included in next step. Utilizing modulo 2^n in second step leads to significant improvement in term of delay. Because this modulus has better speed compared to modulus in the forms of 2^k-1. First step design of the reverse converter architecture for subset $\{2^n-1, 2^{n+1}-1\,, 2^{n-1}-1\}$ based on MRC is described in the following. Weighted number *Y* can be calculated as

$$Y = Z_1 + Z_2P_1 + Z_3P_1P_2 \tag{6}$$

Where

$$Z_1 = x_1$$

$$Z_2 = \left|(x_2 - x_1)\left|P_1^{-1}\right|_{P_2}\right|_{P_2}$$

$$Z_3 = \left|\left((x_3 - x_1)\left|P_1^{-1}\right|_{P_3} - Z_2\right)\left|P_2^{-1}\right|_{P_3}\right|_{P_3}$$

And $P_1 = 2^n-1$, $P_2 = 2^{n+1}-1$ and $P_3 = 2^{n-1}-1$.

Theorem 1: Required multiplicative inverses $\left|P_1^{-1}\right|_{P_2}$, $\left|P_1^{-1}\right|_{P_3}$and $\left|P_2^{-1}\right|_{P_3}$ are equal to -2, 1 and $\sum_{i=0}^{\frac{n}{2}-1} 2^{2i}$, respectively.

Proof: we have

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

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

$$|2 - 2^{n+1}|_{2^{n+1}-1} = |1 - (2^{n+1} - 1)|_{2^{n+1}-1} = 1$$

b) $\left|{P_1}^{-1}\right|_{P_3} = 1$

$$\left|(2^n - 1) \times \left|{P_1}^{-1}\right|_{P_3}\right|_{2^{n-1}-1} = 1 \rightarrow |2^n - 1|_{2^{n-1}-1} = |2 \times (2^{n-1} - 1) + 1|_{2^{n-1}-1} = 1$$

c) $\left|{P_2}^{-1}\right|_{P_3} = \sum_{i=0}^{\frac{n}{2}-1} 2^{2i}$

$$\left|(2^{n+1} - 1) \times \left|{P_2}^{-1}\right|_{P_3}\right|_{2^{n-1}-1} = 1 \rightarrow \left|(2^{n+1} - 1) \times \sum_{i=0}^{\frac{n}{2}-1} 2^{2i}\right|_{2^{n-1}-1} =$$

$$\left|(2^{n+1} - 1) \times \frac{1 - 2^n}{-3}\right|_{2^{n-1}-1} = \left|(4 \times (2^{n-1} - 1) + 3) \times \frac{2^n - 1}{3}\right|_{2^{n-1}-1} =$$

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

After realizing multiplicative inverses, Z_2 can be calculated as bellow

$$Z_2 = |(x_2 - x_1) \times (-2)|_{2^{n+1}-1} \tag{7}$$

With bit-level representation of x_2 and x_1 by multiplying -2 based on lemma 1 and 2, Z_2 resulted as:

$$Z_2 = |L_1 - L_2|_{2^{n+1}-1} \tag{8}$$

Where

$$L_1 = x_{1,n-1} \cdots x_{1,0}0$$

$$L_2 = x_{2,n-1} \cdots x_{2,0}x_{2,n}$$

To eliminate the computing of Z_2 in modulo 2^{n+1}-1, following method can be utilized: the result of L_1 and L_2 subtraction can be a positive number smaller than $2^{n+1} - 1$ or be a negative number less greater than $1 - 2^{n+1}$. First case by default has a result in modulo $2^{n+1} - 1$, in second case one summation of subtraction outcome with $2^{n+1} - 1$ resulted in L_1 and L_2 subtraction in modulo $2^{n+1} - 1$. The outgoing carry of the adder utilized for L_1 and L_2 subtraction, disassemble these two cases.

$$Z_2 = \begin{cases} L_1 - L_2, & if \quad L_1 - L_2 \geq 0 \\ L_1 - L_2 + (2^{n+1} - 1), & if \quad L_1 - L_2 < 0 \end{cases} \tag{9}$$

To calculate Z_3, firstly the value of $\left|{P_1}^{-1}\right|_{P_3}$ and $\left|{P_2}^{-1}\right|_{P_3}$ is letting in equation (6) and resulted as

$$Z_3 = \left|\left((x_3 - x_1) \times 1 - Z_2\right) \times (2^0 + 2^2 + \cdots + 2^{n-2})\right|_{2^{n-1}-1} \tag{10}$$

$$Z_3 = |(x_3 - x_1 - Z_2) \times (2^0 + 2^2 + \cdots + 2^{n-2})|_{2^{n-1}-1} \tag{11}$$

If $L_1 > L_2$, we have

$$Z_3 = |(x_3 - x_1 - L_1 + L_2) \times (2^0 + 2^2 + \cdots + 2^{n-2})|_{2^{n-1}-1} \tag{12}$$

For more simplicity $x_3 - x_1 - L_1 + L_2$ is rewritten in bit-level representation and then segregated in numbers with $n-1$ to ease applying its coefficient, $(2^0 + 2^2 + \cdots + 2^{n-2})$.

$$Z_3 = \left| \begin{pmatrix} x_{3,n-2} \cdots x_{3,0} - \underbrace{0 \cdots 0}_{n-2} x_{1,n-1} - x_{1,n-2} \cdots x_{1,0} \\ - \underbrace{0 \cdots 0}_{n-3} L_{1,n} L_{1,n-1} - L_{1,n-2} \cdots L_{1,0} \\ + \underbrace{0 \cdots 0}_{n-3} L_{2,n} L_{2,n-1} + L_{2,n-2} \cdots L_{2,0} \end{pmatrix} \times (2^0 + 2^2 + \cdots + 2^{n-2}) \right|_{2^{n-1}-1} \quad (13)$$

And with utilizing lemma 1 and 2:

$$Z_3 = \left| \begin{pmatrix} x_{3,n-2} \cdots x_{3,0} + \underbrace{1 \cdots 1}_{n-2} \bar{x}_{1,n-1} + \bar{x}_{1,n-2} \cdots \bar{x}_{1,0} \\ + \underbrace{1 \cdots 1}_{n-3} \bar{L}_{1,n} \bar{L}_{1,n-1} + \bar{L}_{1,n-2} \cdots \bar{L}_{1,0} \\ + \underbrace{0 \cdots 0}_{n-3} L_{2,n} L_{2,n-1} + L_{2,n-2} \cdots L_{2,0} \end{pmatrix} \times (2^0 + 2^2 + \cdots + 2^{n-2}) \right|_{2^{n-1}-1} \quad (14)$$

Equation (14) can be simplified as following:

$$Z_3 = \left| \left(Z_{3,1} + Z_{3,2} + Z_{3,3} + Z_{3,4} + Z_{3,5} + Z_{3,6} + Z_{3,7} \right) \times (2^0 + 2^2 + \cdots + 2^{n-2}) \right|_{2^{n-1}-1} \quad (15)$$

where

$$Z_{3,1} = x_{3,n-2} \cdots x_{3,0}$$

$$Z_{3,2} = \underbrace{1 \cdots 1}_{n-2} \bar{x}_{1,n-1}$$

$$Z_{3,3} = \bar{x}_{1,n-2} \cdots \bar{x}_{1,0}$$

$$Z_{3,4} = \underbrace{1 \cdots 1}_{n-3} \bar{L}_{1,n} \bar{L}_{1,n-1}$$

$$Z_{3,5} = \bar{L}_{1,n-2} \cdots \bar{L}_{1,0}$$

$$Z_{3,6} = \underbrace{0 \cdots 0}_{n-3} L_{2,n} L_{2,n-1}$$

$$Z_{3,7} = L_{2,n-2} \cdots L_{2,0}$$

In other case when $L_1 < L_2$, $Z_3 = \left| \left(x_3 - x_1 - L_1 + L_2 - (2^{n+1} - 1) \right) \times (2^0 + 2^2 + \cdots + 2^{n-2}) \right|_{2^{n-1}-1}$ and since $|-(2^{n+1} - 1)|_{2^{n-1}-1} = |-3|_{2^{n-1}-1}$, we have

$$|-(2^{n+1} - 1) \times (2^0 + 2^2 + \cdots + 2^{n-2})|_{2^{n-1}-1} = |-1|_{2^{n-1}-1} = \underbrace{1 \cdots 1}_{n-2} 0 \quad (16)$$

Therefore, Z_3 can be rewritten as

$$Z_3 = \left| \left(Z_{3,1} + Z_{3,2} + Z_{3,3} + Z_{3,4} + Z_{3,5} + Z_{3,6} + Z_{3,7} \right) \times (2^0 + 2^2 + \cdots + 2^{n-2}) + Z_{3,8} \right|_{2^{n-1}-1} \quad (17)$$

Where

$$Z_{3,8} = \underbrace{1 \cdots 1}_{n-2} 0$$

Finally, Z_3 is realized as

$$Z_3 = \begin{cases} |(S_1 + C_1) \times (2^0 + 2^2 + \cdots + 2^{n-2})|_{2^{n-1}-1} \; if \quad L_1 - L_2 \geq 0 \\ |(S_1 + C_1) \times (2^0 + 2^2 + \cdots + 2^{n-2}) + Z_{3,8}|_{2^{n-1}-1} \; if \quad L_1 - L_2 < 0 \end{cases} \tag{18}$$

Hardware implementation of Z_2 and Z_3 are shown in Fig. 2.

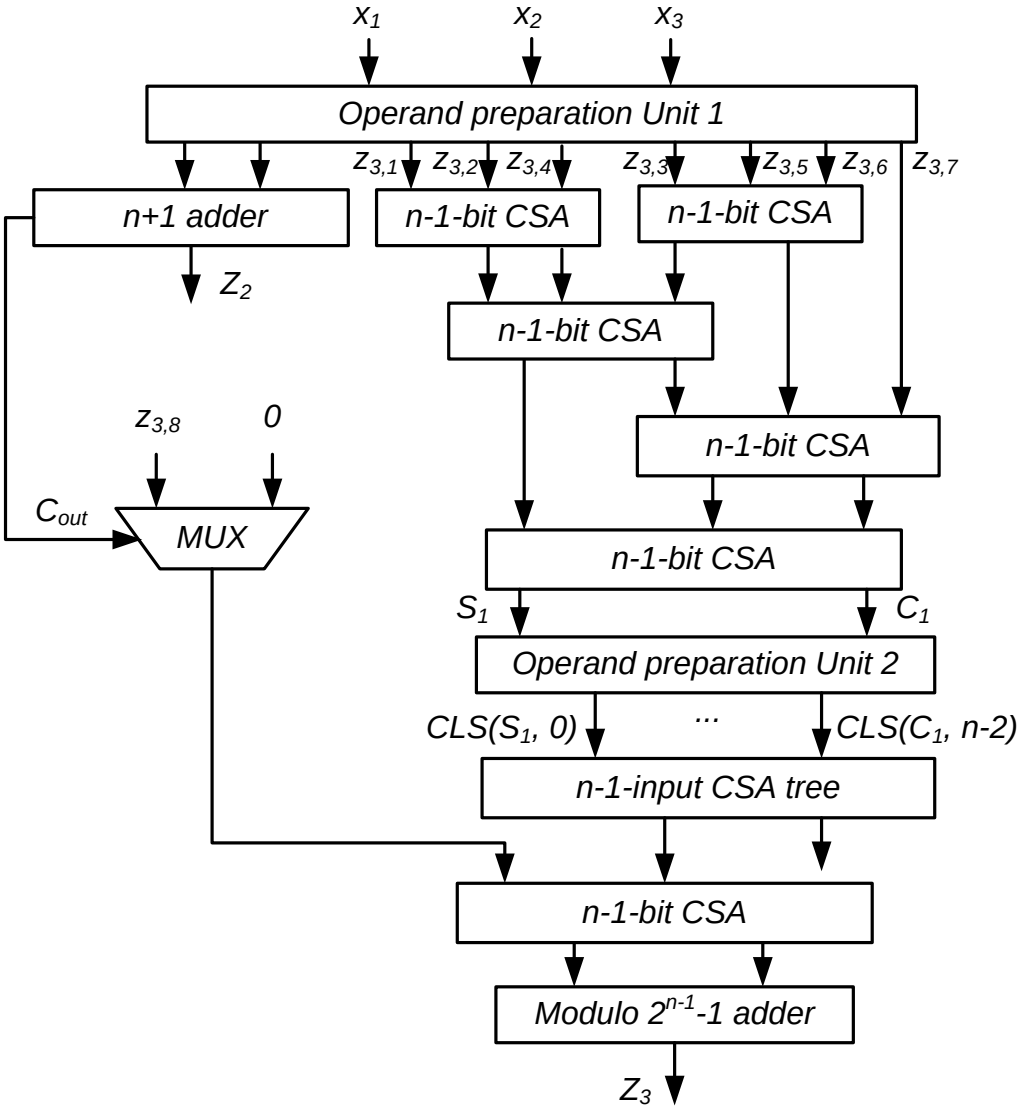

Fig. 2. Hardware schema for first step design

After calculation of Z_2 and Z_3, Y can be achieved from its residues in three moduli $\{2^n - 1\,, 2^{n+1} - 1\,, 2^{n-1} - 1\}$ as

$$Y = Z_1 + Z_2 P_1 + Z_3 P_1 P_2 \tag{19}$$

$$Y = Z_1 + Z_2 \times (2^n - 1) + Z_3 \times (2^n - 1) \times (2^{n+1} - 1) \tag{20}$$

3.1.2. Second step design

After computation of *Y*, the two modulus superset $\{(2^n-1)\times(2^{n+1}-1)\times(2^{n-1}-1), 2^n\}$ is considered for achieving weighted number *X*. The residue of weighted number *X* modulo P_{123} and P_4 is equal to *Y* and x_4, respectively, where $P_{123}=(2^n-1)\times(2^{n+1}-1)\times(2^{n-1}-1)$ and $P_4=2^n$. MRC method for moduli set with two modulus, is utilized to calculate *X* as below:

$$X = v_1 + v_2 P_{123} \tag{21}$$

Where

$$v_1 = Y \tag{22}$$

$$v_2 = \left|(x_4 - Y)\left|{P_{123}}^{-1}\right|_{P_4}\right|_{P_4} \tag{23}$$

Theorem 2: multiplicative inverses of P_{123} in modulo P_4 is equal to $-2^{n-1}-1$.

Proof: Following equation is obtained multiplicative inverses:

$$\begin{aligned}\left|(2^n-1)\times(2^{n+1}-1)\times(2^{n-1}-1)\times\left|{P_{123}}^{-1}\right|_{P_4}\right|_{2^n} &= 1\\ \rightarrow |(2^n-1)\times(2^{n+1}-1)\times(2^{n-1}-1)\times(-2^{n-1}-1)|_{2^n}&\\ = |(-1)\times(-1)\times(-2^{2n-2}+1)|_{2^{n+1}-1} = |(-1)\times(-1)\times 1|_{2^{n+1}-1} &= 1\end{aligned}$$

Thus, $v_2 = |(x_4 - Y)\times(-1)\times(2^{n-1}+1)|_{2^n} = |Y - x_4|_{2^n}$.

With letting the values which computed in first level of design, v_2 can be rewritten as

$$v_2 = |(Z_1 + Z_2\times(2^n-1) + Z_3\times(2^n-1)\times(2^{n+1}-1) + \bar{x}_4 + 1)\times(2^{n-1}+1)|_{2^n} \tag{24}$$

In the operations is in modulo 2^n, only least significant *n* bits of operators are used, therefore v_2 can represented as:

$$v_2 = \left| Z_{1,0}\underbrace{0\cdots 0}_{n-1} + \bar{Z}_{2,0}\underbrace{0\cdots 00}_{n-1} + Z_{3,0}Z_3 + \bar{x}_{4,0}\underbrace{0\cdots 0}_{n-1} + \bar{x}_4 + Z_1 + \bar{Z}_2 + 1 + 1\right|_{2^n} \tag{25}$$

Where

$$k_1 = Xor\langle \bar{x}_{4,0}, \bar{x}_{4,n-1}\rangle \bar{x}_{4,n-2}\cdots\bar{x}_{4,1}\bar{x}_{4,0}$$

$$k_2 = Xor\langle Z_{1,0}, \bar{Z}_{2,0}, Z_{3,0}\rangle Z_3$$

Hardware implementation of v_2 is shown in Fig. 3.

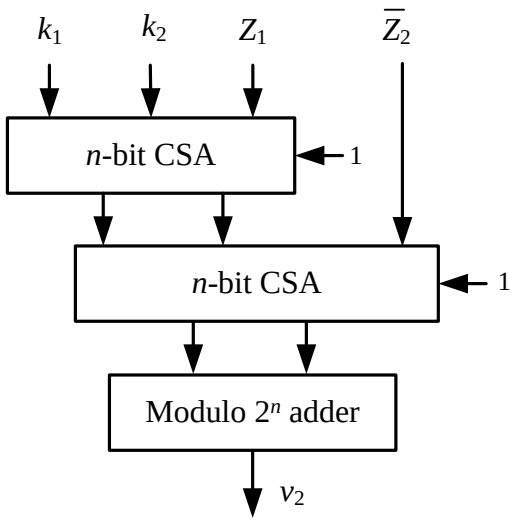

Fig. 3. Calculation of v_2

Calculation the value of weighted number X based on the computed value of v_2 is resulted the equation (26).

$$X = v_1 + v_2 \times (2^n - 1) \times (2^{n+1} - 1) \times (2^{n-1} - 1) \tag{26}$$

Since $v_1 = Y$, we have

$$X = Y + v_2 \times (2^n - 1) \times (2^{n+1} - 1) \times (2^{n-1} - 1) \tag{27}$$

Eq. 27 can be simplified as

$$X = v_2 Z_3 Z_2 Z_1 - v_2 \underbrace{0 \cdots 0}_{2n+1} - v_2 Z_3 Z_2 - v_2 Z_3 v_2 + v_2 \underbrace{0 \cdots 0}_{n+1} + v_2 \underbrace{0 \cdots 0}_{n} + v_2 Z_3 \tag{28}$$

$$X = v_2 Z_3 Z_2 Z_1 + \bar{v}_2 \underbrace{1 \cdots 1}_{2n+1} + \bar{v}_2 \bar{Z}_3 \bar{Z}_2 + \bar{v}_2 \bar{Z}_3 \bar{v}_2 + v_2 \underbrace{0 \cdots 0}_{n+1} + v_2 \underbrace{0 \cdots 0}_{n} + v_2 Z_3 \tag{29}$$

X is the summation of seven values, $\sum_{k=1}^{7} X_k$, where $X_1 = v_2 Z_3 Z_2 Z_1$, $X_2 = \bar{v}_2 \underbrace{1 \cdots 1}_{2n+1}$, $X_3 = \bar{v}_2 \bar{Z}_3 \bar{Z}_2$, $X_4 = \bar{v}_2 \bar{Z}_3 \bar{v}_2$, $X_5 = v_2 \underbrace{0 \cdots 0}_{n+1}$, $X_6 = v_2 \underbrace{0 \cdots 0}_{n}$, $X_7 = v_2 Z_3$. The bit-length of X_k is 4n-bit and enter to carry save adder tree, the outcomes of carry save adder carry go to input of a 4-bit CPA to compute weighted number X. Fig. 4 depicted the architecture of this scenario.

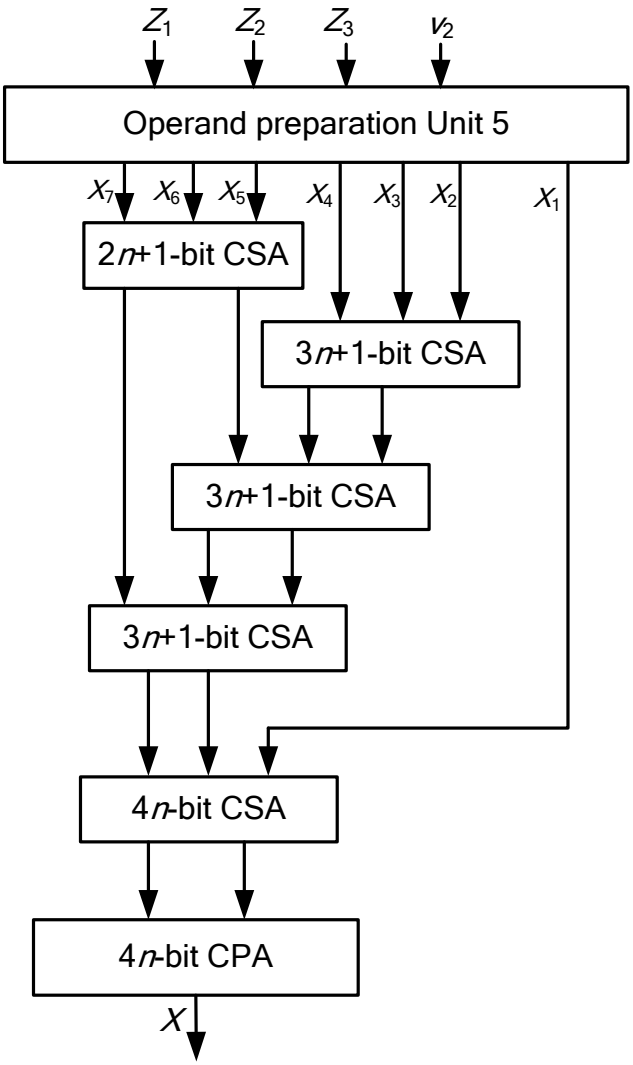

Fig. 4. Hardware implementation for calculation of X

Numerical example: Considering moduli set {63, 127, 31, 64} which is derived from moduli set $\{2^{n-1}-1\,,2^{n+1}-1\,,2^{n}\,,2^{n}-1\}$ when n = 6. The RNS number (33, 7, 5, 63) can be converted to its equivalent in weighted number X as:

First stage:

$x_1=(33)_{10}=(100001)_2$
$x_2=(7)_{10}=(0000111)_2$
$x_3=(5)_{10}=(00101)_2$

With substituting these values in Eq. 6, 8, 15 and 17, we have

$Z_1=(33)_{10}=(100001)_2$
$L_1=1000010_2$
$L_2=0001110_2$
$Z_2=(0110100)_2$

$Z_3 = (11001)_2$

Second stage: by considering Eq. 22, 25 and 29, we have

$v_1 = Y = (107322)_{10} = (11010001100111010)_2$
$k_1 = 0$
$k_2 = (57)_{10} = (111001)_2$
$v_2 = (39)_{10} = (100111)_2$
$X = 107322 + 39 \times 63 \times 127 \times 31 = 9876543$

Thus $X = 9876543$, and verification can be simply done as

$x_1 = |9876543|_{63} = 33$
$x_2 = |9876543|_{127} = 7$
$x_3 = |9876543|_{31} = 5$
$x_4 = |9876543|_{64} = 63$

Table II. Hardware requirement and delay of reverse converters.

Moduli Set	Design	Hardware requirements	Delay
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[15]-1	$(9n + 5 + ((n-4)(n+1)/2))A_{FA} + 2nA_{ex\text{-}Nor} + 2nA_{OR} + (6n+1)A_{INV}$	$(23n+12)/2\ D_{FA}$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[15] -1– Using ROM	$(6n+1)A_{INV} + (8n+4)A_{FA} + 2nA_{ex\text{-}Nor} + 2nA_{OR} + (n+1)2^{n+1}A_{ROM}$	$(9n+6)\ D_{FA}$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$	[15]-2	$(6n+7)A_{INV}+(n^2+12n+12)A_{FA} + 2nA_{EXNOR} + 2nA_{OR}+(4n+8)A_{2:1MUX}$	$(16n+22)\ D_{FA}$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$	[15] -2– Using ROM	$(5n+6)A_{INV} + (9n+10)A_{FA} + 2nA_{EXNOR} + 2nA_{OR} + (2n+2)A_{2:1MUX} + (n+2)2^{n+2}A_{ROM}$	$(11n+14)\ D_{FA}$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$	[14]	$(58n+23 + \log_2(c+1))\ A_{FA} + 36nA_{ROM}$	$(24n+17+ \log_2(c+1))\ D_{FA}$
$\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$	[18]– C1 CE	$(25.5n + 12 + (5n^2/2))A_{FA} + 5nA_{HA} + 3nA_{EXNOR} + 3nA_{OR}$	$(18n+23)\ D_{FA}$
$\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$	[18] – C2 CE	$(20n+17)A_{FA} + (3n-4)A_{HA} + 2^n(5n+2)A_{ROM}$	$(13n+22)\ D_{FA} + 3\ D_{ROM}$
$\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$	[18] – C3 CE	$(23n+11)A_{FA} + (2n-2)A_{HA} + (6n+4)2^nA_{ROM}$	$(16n+14)\ D_{FA} + D_{ROM}$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[17]- Three stage-CE	$(n^2+10n+3)A_{FA}+A_{HA}+ (3n+2)A_{INV} +2A_{2:1MUX}$	$(9n+6+m)\ D_{FA}$*
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[16]	$(28n+15)A_{FA}$	$(14n+8)\ D_{FA}$
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	[19] D1-C-I	$(n^2+16n+6)\ A_{FA}+ (n+2)A_{XNOR} + (3n-5)A_{AND} + (n+2)A_{OR} + (3n-5)A_{XOR} + 4n\ A_{INV}$	$(12n+9+q)\ D_{FA}$*
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	[19] D1-C-II	$(n^2+24n+24)\ A_{FA}+ (2n+3)A_{HA} + 2A_{XNOR} + (2n-5)A_{AND} + 2A_{OR} + (2n-5)A_{XOR}$ $(2n+1)A_{3:1MUX} + 4n\ A_{INV}$	$(8n+11+q)\ D_{FA}$*
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	[19] D1- C-III	$(n^2+22n+22)\ A_{FA}+ (2n+2)A_{HA} + 2A_{XNOR} + (2n-5)A_{AND} + 2A_{OR} + (2n-5)A_{XOR} + 10(2n+1)A_{ROM} + (2n+1)A_{2:1MUX} + 4n\ A_{INV}$	$(8n+11+q)\ D_{FA}$*
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	Proposed	$(n^2+18n+1)A_{FA}+ (4n-3)\ A_{XNOR} + (5n-1)A_{AND} +(4n-3)A_{OR} + (5n+2)A_{XOR}+ (4n-1)\ A_{INV}$	$(7n + 8 + p)\ D_{FA}$* $+ (n-1)\ D_{HA}$

* m, q and p are the number of levels in CSA tree of $(n+2)$, $(n+1)$ and $(n-1)$ inputs, respectively.

4. Comparison

This section presents the comparison of the proposed reverse converter architecture for the moduli set $\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$ with other balanced 4-moduli sets with the same dynamic range class, such as the 4-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$[15-17], $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ [14-15], $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$ [18] and $\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$ [19]. The comparisons are done in terms of delay and area of the reverse converter. Table II shows the comparison between the proposed reverse converter and other state-of-the-art works. In order to achieve fair comparison, assumption for delay and area of the modulus adders and carry save adders are considered such as [24]. As shown in

Table II, the proposed reverse converter for the moduli set $\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$ has achieved higher speed of the reverse converter in terms of delay of one bit full adder (FA) cell compared to $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$[15-17], $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ [14-15], $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$ [18] and $\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$ [19]. It is worth mentioning that, the proposed reverse converter is the fastest adder based reverse converter in class of balanced four moduli set compared to the literature. Less hardware requirement is also needed compared to the one reported in [19].

In order to achieve fair comparison, unit gate delay and area comparison between different adder base reverse converters are included in Table III. In this model FA gates are considered with area of seven gates and delay of four gates. XOR/XNOR gates are considered with two gates area and delay, and each two-input monotonic gates considered with one area and delay [25]. Results in Table III confirm the remarkable improvement in terms of speed of the reverse converter and degraded hardware resources are achieved compared to [14-19].

Table III. Unit gate area and delay of reverse converters.

Moduli Set	Design	Unit gate area	Unit gate delay
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[15]-1	$3.5n^2+72.5n+23$	$46n+24$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$	[15]-2	$7n^2+128n+146$	$64n+88$
$\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$	[18]– C1 CE	$17.5n^2+210.5n+84$	$72n+92$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[17]- Three stage-CE	$7n^2+76n+41$	$36n+24+4m$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	[16]	$196n+105$	$56n+32$
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	[19] D1-C-I	$7n^2+136n+30$	$48n+36+4q$
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	[19] D1-C-II	$7n^2+204n+174$	$32n+44+4q$
$\{2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1\}$	Proposed	$7n^2+170n-12$	$30n + 30 + 4p$

* m, q and p are the delay of CSA tree with $\lceil \log_2(n+2) \rceil$, $\lceil \log_2(n+1) \rceil$ and $\lceil \log_2(n-1) \rceil$ unit gate, respectively.

5. Conclusion

In this paper, an efficient RNS to binary converter for moduli set $\{2^{n-1}-1, 2^{n+1}-1, 2^n, 2^n-1\}$ is presented. This moduli set is free from modulo (2^k+1)-type, providing efficient arithmetic operations in RNS channels. The proposed converter is designed based on MRC in two levels architecture and has achieved higher speed and lower hardware requirements than the other reverse converters for 4-moduli set with the same dynamic range in literature.

6. REFERENCES

[1] M.A. Bayoumi, P. Srinivasan, "Parallel arithmetic: from algebra to architecture," Proc. IEEE Symposium on Circuits and Systems, 1990.

[2] T. Stouratitis and V. Paliouras, "Considering the alternatives in low power design," *IEEE Circuits and Devices*, vol. 7, pp. 23-29, 2001.

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

[4] M. Lu, Arithmetic and Logic in Computer Systems, John Wiley & Sons Publication, 2004.

[5] R. Conway and J. Nelson, "Improved RNS FIR Filter Architectures," IEEE Transactions On Circuits and Systems II, Vol. 51, No. 1, pp. 26- 28, 2004.

[6] R. Chaves, L. Sousa, "RDSP: A RISC DSP based on residue number system," Proceedings of the Euro micro symposium on digital systems design: architectures, methods, and tools, Turkey, pp. 128--135, 2003.

[7] W. Wei and et al., "RNS application for digital image processing," Proceedings of the 4th IEEE international workshop on system-on-chip for real time applications, Canada, pp. 77-80, 2004.

[8] S. Yen, S. Kim, S. Lim and S. Moon, "RSA Speedup with Chinese Remainder Theorem Immune against Hardware Fault Cryptanalysis," IEEE Transactions on Computers, vol. 52, no. 4, pp. 461-472, 2003.

[9] M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, K. Navi, "Efficient RNS Implementation of Elliptic Curve Point Multiplication Over GF(p)", IEEE Transactions on VLSI systems, DOI: 10.1109/TVLSI.2012.2210916.

[10] J. Ramirez, et al., "Fast RNS FPL-Based Communications Receiver Design and Implementation," Proceedings of the 12th Int'l Conf. Field Programmable Logic, pp. 472-481, 2002.

[11] 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.

[12] M. Taheri, E. Khani , M. Esmaeildoust and K. Navi, "Efficient Reverse Converter Design for Five Moduli Set $\{2^n, 2^{2n+1}-1, 2^{n/2}-1, 2^{n/2}+1, 2^n+1\}$" Journal of Computations &Modelling, vol.2, no.1, 93-108, 2012.

[13] W. K. Jenkins and B. J. Leon, "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, 1977.
[14] M. Bhardwaj, T. Srikanthan, and C. T. Clarke, " A reverse converter for the 4-moduli super set $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{+}1\}$, " in IEEE Conf. Computer Arithmetic, Apr. 1999.
[15] P. V. A. Mohan and A. B. Premkumar, "RNS-to-Binary Converters for Two Four-Moduli Set $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{–}1\}$ 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, 2007.
[16] A. P. Vinod and A. B. Premkumar, "A residue to binary converter for the 4-moduli superset $\{2^n\text{–}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{–}1\}$," *Journal of Circuits, Circuits and Computers*, vol. 10, pp. 85-99, 2000.
[17] B. Cao, T. Srikanthan, C.H. Chang, "Efficient reverse converters for the four-moduli sets $\{2^n\text{–}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{–}1\}$ and $\{2^n\text{–}1, 2^n, 2^n\text{+}1, 2^{n-1}\text{–}1\}$,"*IEE Proc. Comput. Digit. Tech.*, vol. 152, pp. 687–96, 2005.
[18] P.V.A. Mohan, "New reverse converters for the moduli set $\{2^n\text{–}3, 2^n\text{–}1, 2^n\text{+}1, 2^n\text{+}3\}$,"*Elsevier Journal of Electronics and Communications (AEU)*, vol. 62, no. 9, pp.643-658, 2008.
[19] M. Esmaeildoust, K. Navi, M. Taheri, A.S. Molahosseini and S. Khodambashi, "Efficient RNS to Binary Converters for the New 4-Moduli Set $\{2^n, 2^{n+1}\text{–}1, 2^n\text{–}1, 2^{n-1}\text{–}1\}$" *IEICE Electron. Exp.*, vol. 9, no. 1, pp. 1-7, 2012.
[20] L. Kalampoukas *et al.*, "High-speed parallel-prefix modulo $2^n\text{-}1$adders," *IEEE Trans. Computers*, vol. 49, no. 7, pp. 673–679, Jul. 2000.
[21] C. Efstathiou, H. T. Vergos, and D. Nikolos, "Fast parallel-prefix modulo $2^n\text{+}1$ adder," *IEEE Trans. Comput.*, vol. 53, no. 9, pp.1211–1216, Sep. 2004.
[22] R. A. Patel, M. Benaissa, N. Powell, and S. Boussakta, "Novel power-delay-area-efficient approach to generic modular addition," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, pp. 1279–1292, 2007.
[23] K. Navi, M. Esmaeildoust and A.S. Molahosseini, A General Reverse Converter Architecture with Low Complexity and High Performance, *IEICE TRANSACTIONS on Information and Systems,* E94-D(2), 264-273, 2011.
[24] A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, S. Timarchi," Efficient Reverse Converter Designs for the New 4-Moduli Sets $2^n\text{-}1, 2^n, 2^n + 1, 2^{2n+1}\text{–}1$ *and* $2^n\text{ -}1, 2^n + 1, 2^{2n}, 2^{2n} + 1$ Based on New CRTs," IEEE Transactions on Circuits and Systems-I, Volume**:** 57 , Issue: 4 , pp. 823 – 835, 2010.
[25] Mohammad Esmaeildoust, Keivan Navi and MohammadReza Taheri, "High speed reverse converter for new five-moduli set $\{2^n, 2^{2n+1}\text{-}1, 2^{n/2}\text{-}1, 2^{n/2}\text{+}1, 2^n\text{+}1\}$", *IEICE Electron. Express*, Vol. 7, No. 3, pp.118-125, 2010.