Educational Administration: Theory and Practice
2024, 30(5), 2385 - 2400
ISSN: 2148-2403
https://kuey.net/

Research Article

Fast And Area-Efficient Reverse Converters For Five ModuliSet $\{2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1\}$

Danial Alvani[1*], Mohammad Esmaeildoust[2], Amer Kaabi[3]

[1]Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Khorramshahr, Iran.
Email: danial.alvani@kmsu.ac.ir
[2]Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Khorramshahr, Iran.
Email: m_doust@kmsu.ac.ir
[3]Department of Basic Sciences, Abadan Faculty of Petroleum Engineering, Petroleum University of Technology, Abadan, Iran.
Email: kabbi_amer@put.ac.ir

Citation: Danial Alvani, et.al (2024), Fast And Area-Efficient Reverse Converters For Five ModuliSet $\{2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}1, 2^n-1\}$, *Educational Administration: Theory and Practice*, *30*(5), 2385 - 2400
Doi: 10.53555/kuey.v30i5.3289

ARTICLE INFO

ABSTRACT

Introduction: This paper proposes two new reverse converters for balanced and well-formed five-moduli set $\{2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1\}$. The converters are planned in a two-level architecture while appreciating adder base structures without utilizing any ROM, which results in an efficient implementation in VLSI circuits.
Materials and Methods: To design both levels of the proposed reverse converters, Mixed-Radix Conversation (MRC) algorithm is employed.
Results and Discussion: Unit gate delay and area estimation demonstrate the proposed reverse converter (DC1) is faster than other alternatives, the similar five moduli reverse converter, under distinctive dynamic ranges while the second design (DC2) requires less hardware cost.
Conclusion: The synthesis results on Xilinx Virtex-7 FPGA illustrate that, comparing to the latest five moduli set reverse converters, the proposed converter (DC1) has achieved 11%, 12% and 11% improvement in speed for n = 12, 16 and 20, respectively.

Keywords: residue number system (RNS); reverse converter; mixed radix conversion; computer arithmetic.

INTRODUCTION

Characterized by a set of moduli, Residue Number System (RNS) is a non-weighted number system. The most prominent advantage of this system could be the absence of the carry propagation between RNS channels. Accordingly, parallel execution of arithmetic operations, addition, subtraction, and multiplication of smaller numbers, for instance, can be realized in the applications requiring arithmetic operation on large numbers [1]. Unlike addition, subtraction and multiplication operations, dividing, sign detection, and comparing values are difficult to do in RNS. This system is broadly utilized in special-purpose processors to run applications such as public key cryptography algorithm [2-4], RSA [5-8], Elliptic Curve Cryptography (ECC) [9-14], digital signal processing (DSP) [15-19], digital filters [20-23], image processing [24-26], and error correction systems [27-30]. The primary operation in cryptography algorithms such as RSA and ECC, is the modular multiplication on large numbers [12, 31, 32]. Since calculations are performed on residues, applying RNS in these algorithms will result in higher efficiency in terms of fast VLSI implementation and reduced power consumption. RNS comprises three principal parts: binary-to-residue converter (forward converter), which converts a weighted binary number to its equivalent residue number, arithmetic unit including addition, subtraction, and modular multiplication, and finally the residue-to-binary (reverse converter), which converts residues to its equivalent weighted number. The reverse converter is an essential part of the RNS since the speed gain of the RNS arithmetic unit should not be reduced by this part. The complexity of the reverse converter is determined by the selected moduli set as well as the conversion algorithm(s) used in the design of the reverse converter.
3n-bit dynamic range moduli sets has been reported by many works. The most famous RNS moduli set is $\{2^n-1, 2^n, 2^n+1\}$ [33] due to its simple and well-formed balanced moduli. In terms of complexity, arithmetic operation in moduli 2^n+1 are more complex than moduli 2^n and 2^n-1. Therefore, 3n-bit dynamic range RNS moduli sets $\{2^{n-1}-1, 2^n-1, 2^n\}$ [34], $\{2^{\mathrm{n}}-1, 2^n, 2^{\mathrm{n}+1}-1\}$ [35] and $\{2^{2n+1}-1, 2^n, 2^{n-1}\}$ [36] are reported by many

researchers. The latest papers that consider reverse conversion using 3-moduli sets are [37-40]. As applications which require operation on larger numbers grow up, the provided dynamic range by this moduli sets will not be sufficient. Hence, 4*n*-bit dynamic range four-moduli sets {2^n-3, 2^n+1, 2^n-1, 2^n+3} [41] {2^n-1,2^n, 2^n+1, 2^{n+1}-1}, {2^n-1, 2^n, 2^n+1, 2^{n+1}+1} [42], {2^n-1, 2^n, 2^n+1, 2^{2n}+1} [43], {2^n+1, 2^n-1, 2^n, 2^{n+1}+1} [44] and {2^n,2^{n-1}−1,2^n−1,2^{n+1}−1} (*n* Even) [45] have been presented by researchers.

In order to attain a higher degree of parallelism and dynamic range, balanced five moduli set {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1} [46] has been proposed. The moduli set benefits from arithmetic friendly moduli using fast and efficient arithmetic operation together with high dynamic range. The dynamic range and parallelism ensued from this moduli set make it suitable for the applications in which operations on large numbers are required, such as public key cryptography algorithms [5, 9, 12]. Nevertheless, as the number of modules increases, the selected moduli set results in a more complex reverse converter. The studies [47] and [48] present efficient reverse converters for five moduli set {2^n-1, 2^n, 2^n+1, 2^{n+1}+1, 2^{n-1}+1} and {2^n-1, 2^k, 2^n+1, 2^{n-1}-1, 2^{n+1}-1}, respectively, to extend the efficiency of reverse converter. The authors in [48] used the multiplication by constant reported in [49] to increase the efficiency.

In this paper, we introduce two fast and area efficient reverse converters for the five-moduli set {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1} offering a high-speed arithmetic unit because of its balanced moduli in the form 2^k and 2^k±1. To have five moduli set pairwise relatively prime, *n* should be an EVEN positive integer greater than 2. These reverse converters are designed in a two-level structure using Mixed Radix Conversion (MRC) algorithms for both of the levels. High accuracy in choosing the appropriate moduli for these two levels brings about a fast and area efficient reverse converter.

The remains of this paper is organized as follows. Section 2 presents several useful attributes of residue arithmetic are presented. Section 3 illustrates the proposed reverse converter for five-moduli set {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1}. Section 4 explains hardware implementation and performance evaluation with state-of-the-art works in literature. Finally, section 5 concludes the paper.

RELATED BACKGROUND

RNS Background

The RNS is determined in terms of relatively prime moduli set $\{P_1, P_2, \ldots, P_n\}$ that is $\gcd\{(P_i, P_j)\}$ for$\{i \neq j\}$. A weighted number X can be represented as$\{X = (x_1,\ x_2, \ldots, x_n)\}$, where,

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

Such a representation is inimitable for any integer X in the range$[0, M-1]$, where M is the dynamic range of the moduli set$\{P_1, P_2, \ldots, P_n\}$, which is equal to the product of $\{P_i\}$ terms$(M = P_1 \times\ P_2 \times \ldots \times\ P_n)$.

Mixed-Radix Conversion

By MRC, which is calculated sequentially, the weighted number X is obtained from its corresponding residues, i.e.,$(x_1,\ x_2, \ldots, x_n)$, based on the moduli set $\{P_1, P_2, \ldots, P_n\}$ as follows [1]:

$$X = v_1 + v_2 P_1 + v_3 P_2 P_1 + \ldots + v_n \prod_{i=1}^{n-1} P_i \tag{2}$$

Equation (2) for 3-moduli set in MRC can be shown as

$$X = x_1 + v_2 P_1 + v_3 P_1 P_2 \tag{3}$$

The coefficients v_is can be gained from residues by

$$v_1 = x_1 \tag{4}$$

$$v_2 = \left| (x_2 - v_1) \left| P_1^{-1} \right|_{P_2} \right|_{P_2} \tag{5}$$

$$v_3 = \left| ((x_3 - v_1) \left| P_1^{-1} \right|_{P_3} - v_2) \left| P_2^{-1} \right|_{P_3} \right|_{P_3} \tag{6}$$

In general case:

$$v_n = \left| (((v_n - v_1) \left| P_1^{-1} \right|_{P_n} - v_2) \left| P_2^{-1} \right| P_n - \cdots - v_{n-1}) \left| P_{n-1}^{-1} \right|_{P_n} \right|_{P_n} \tag{7}$$

Note that $\left| P_1 \right|_{P_2^{-1}}$ is the multiplicative inverse of P_1 modulo P_2.

Proposed Reverse Converter

As shown in figure 1, two level design using MRC are employed to realize fast and area efficient reverse converter for the moduli set {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1}. Considering moduli set {P_1, P_2, P_3, P_4, P_5} = {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1} and corresponding residues (x_1, x_2, x_3, x_4, x_5), in first and the second levels of the design, converters for the subset {2^n+1, 2^{n-1}-1, 2^n} and {2^n+1×(2^{n-1}-1)×2^n, 2^{n+1}-1, 2^n-1} are designed using MRC. In the following the proposed two level design will be detailed.

Initially at first level, a reverse converter for the subset **{**2^n+1, 2^{n-1}-1, 2^n**}** is designed according to its represented residues (x_1, x_2, x_3) for obtaining RNS number *Y*. Then, at the second level a reverse converter for subset {(2^n+1) (2^{n-1}-1) (2^n), 2^{n+1}-1, 2^n-1} is designed with respect to the result of first level (*Y*) and RNS number x_4 and x_5 to calculate the final weighed number *X*.

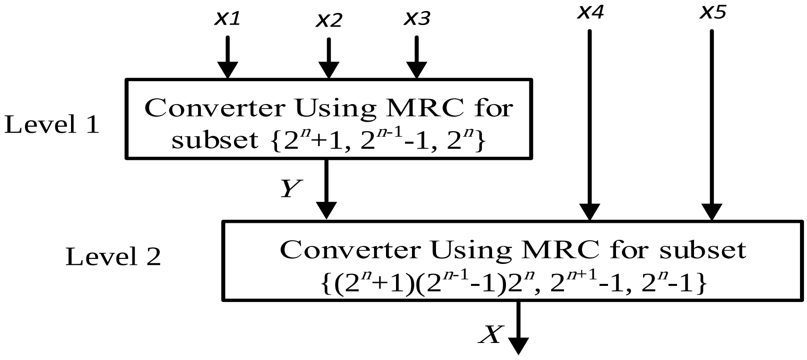

x1
x2
x3
x4
x5
Level 1
Converter Using MRC for subset {2^n+1, 2^{n-1}-1, 2^n}
Y
Level 2
Converter Using MRC for subset {(2^n+1)(2^{n-1}-1)2^n, 2^{n+1}-1, 2^n-1}
X

Figure 1. Two level designs of the reverse converter using MRC

Converter using MRC for subset **{**2^n+1, 2^{n-1}-1, 2^n**}**

Consider the subset {P_1, P_2, P_3} = {2^n+1, 2^{n-1}-1, 2^n} with corresponding weighted number Q = (x_1, x_2, x_3) and using MRC, we have.

(8) $$Y = v_1 + v_2P_1 + v_3P_1P_2$$

Where

(9) $$v_1 = x_1$$

(10) $$v_2 = \left|(x_2 - x_1)\left|_{P_1^{-1}}\right|_{P_2}\right|_{P_2}$$

$$v_3 = \left|((x_3 - v_1)\left|P_1^{-1}\right|_{P_3} - v_2)\left|P_2^{-1}\right|_{P_3}\right|_{P_3} \quad (11)$$

In Eq. (10), $\left|P_1^{-1}\right|_{P_2}$ is the multiplicative inversion of (2^n+1) modulo (2^{n-1}–1), which can be determined as

(12) $$\left|P_1^{-1}\right|_{P_2} = \left|(2^n+1)^{-1}\right|_{2^{n-1}-1} = (1 + 2^2 + 2^4 + ... + 2^{n-2})$$

By substituting Eq. (12) in Eq. (10), we have

(13) $$v_2 = \left|(x_2 - x_1)(1 + 2^2 + \cdots + 2^{n-2})\right|_{2^{n-1}-1}$$

In order to simplified operation in modulo 2^k-1, two lemmas are employed as follows [50].

Lemma 1. The residue of a negative residue number (−*v*) in modulo (2^k − 1) is the one's complement of *v*, where 0 ≤ v < 2^k − 1.

Lemma 2. The multiplication of a positive residue number *v* by 2^P in modulo (2^k − 1) is carried out by *P* bit circular left shift, where *P* is a natural number.

Using lemma 2 in Eq. 13 results

(14) $$v_2 = \left|(L_1 + L_2 + L_3)(1 + 2^2 + ... + 2^{n-2})\right|_{2^{n-1}-1}$$

Where

(15) $$L_1 = x_{2,n-2}...x_{2,0}$$
$$L_2 = \overline{x}_{1,n-2}...\overline{x}_{1,0}$$
$$L_3 = \underbrace{0...0}_{n-3}x_{1,n}x_{1,n-1}$$

In order to calculate v_3 we have:

(16)
$$v_3 = \left| ((x_3 - v_1) \left| P_1^{-1} \right|_{P_3} - v_2) \left| P_2^{-1} \right|_{P_3} \right|_{P_3}$$

In Eq. (16), $\left| P_1^{-1} \right|_{P_3}$ and $\left| P_2^{-1} \right|_{P_3}$ are the multiplicative inverse of (2^n+1) modulo 2^n and (2^{n-1}-1) modulo 2^n, respectively. $\left| P_1^{-1} \right|_{P_3}$ and $\left| P_2^{-1} \right|_{P_3}$ can be determined as

(17)
$$\left| P_1^{-1} \right|_{P_3} = \left| (2^n + 1)^{-1} \right|_{2^n} = 1$$
$$\left| P_2^{-1} \right|_{P_3} = \left| (2^{n-1} - 1)^{-1} \right|_{2^n} = -(2^{n-1} + 1)$$

By substituting Eq. (17) in Eq. (16), we have

(18)
$$v_3 = \left| (x_3 - v_1) \times 1 - v_2)(-(2^{n-1} + 1)) \right|_{2^n}$$

(19)
$$v_3 = \left| (x_3 - v_1 - v_2)(-(2^{n-1} + 1)) \right|_{2^n} = \left| (v_1 + v_2 - x_3)(2^{n-1} + 1) \right|_{2^n}$$

Two cases can be considered for v_2 in Eq. 19

(20)
$$v_2 = \begin{cases} \left| v_{21} + v_{22} \right|_{2^{n-1}-1} & if \ v_{21} + v_{22} < 2^{n-1} \\ \left| v_{21} + v_{22} + 1 \right|_{2^{n-1}-1} & if \ v_{21} + v_{22} \geq 2^{n-1} \end{cases}$$

Using lemma 1 and 2 in Eq. (19) results

(21)
$$v_3 = \left| v_{31} + v_{32} + v_{33} + v_{34} + v_{35} \right|_{2^n}$$

Where

$$v_{31} = v_{1,0} \underbrace{0 \ldots 0}_{n-1}$$
$$v_{32} = \overline{x}_{3,0} \underbrace{00 \ldots 0}_{n-3}10$$
$$v_{33} = v_{1,n-1} \ldots v_{1,0}$$
$$v_{34} = 0 v_{2,n-2} \ldots v_{2,0}$$
$$v_{35} = \overline{x}_{3,n-1} \ldots \overline{x}_{3,0}$$

After calculation of v_2 and v_3, in order to calculate Y, we have

(22)
$$Y = v_1 + (2^n + 1)v_2 + (2^n + 1)(2^{n-1} - 1)v_3$$
$$Y = v_1 + v_2 0 v_2 + 2^{2n-1} v_3 - 2^n v_3 + 2^{n-1} v_3 - v_3$$
$$= v_1 + v_2 0 v_2 - v_3 v_3 + 2^{2n-1} v_3 + 2^{n-1} v_3$$
$$= v_1 + v_3 v_2 v_2 + 2^{n-1} v_3 + \overline{v}_3 \overline{v}_3 + 1$$

Eq.22 can be simplified as

(23)
$$Y = Y_1 + Y_2 + Y_3 + Y_4$$

Where

$$Y_1 = v_1$$
$$Y_2 = v_3 v_2 0 v_2$$
$$Y_3 = v_3 \underbrace{00 \ldots 0}_{n-2}1$$
$$Y_4 = \overline{v}_3 \overline{v}_3$$

In order to achieve faster implementation, Y in Eq. (23) will not calculated in this level and the intermediate results of Y_1, Y_2, Y_3 and Y_4 will sent to next level.

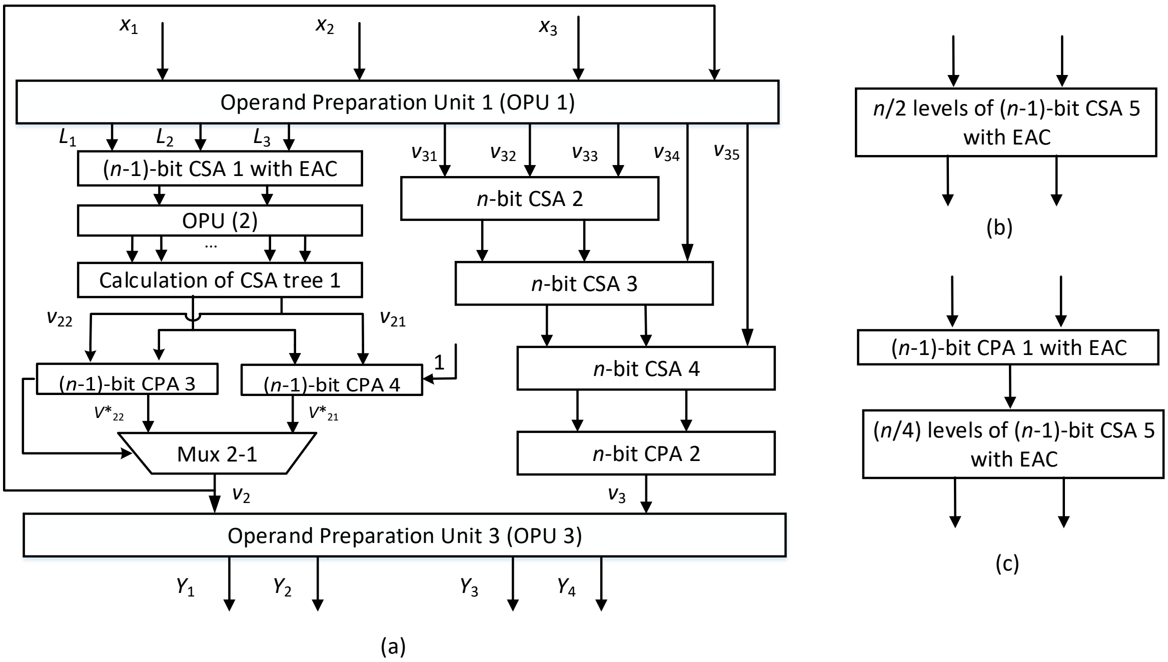


Figure 2. (a) First level design of the reverse converter, (b) calculation of CSA tree 1 (first design-DC1), (c) calculation of CSA tree 1 (second design-DC2)

First level design of the reverse converter including calculation of v_2 and v_3 are shown in figure 2. Hardware Implementation for performing the first level of the proposed reverse converter is specified based on Eqs. (14) and (21). Equation (14) is implemented by a (n-1)-bit carry save adder (CSA) with end around carry (EAC), $n/2$ levels of (n-1)-bit carry save adder (CSA) with (EAC) in DC1, (n/4) levels of (n-1)-bit CSA with EAC in DC2, two (n-1)-bit carry propagate adder (CPA) and a multiplexer. Eq. (21) is implemented by three n-bit CSA and one n-bit carry propagate adder. It should be noted that some full adders (FAs) in these CSAs can be replaced with the couple gates of XOR/AND or XNOR/OR due to some stable values 0 or 1, respectively [51-52].
Table 1 displays the hardware details and delay of each ingredient for the first level design of the proposed reverse converter.

Table 1. Detailed of each component for the first level design of the reverse converter

Component	FA	XOR	AND	XNOR	OR	MUX2-1	Delay
CSA1	2	n-4	n-4	-	-	-	t_{FA}
CSA 5 (DC1)	(n-2)(n-1)	-	-	-	-	-	$n/2\ t_{FA}$
CSA 5 (DC2)	(n/2-2)(n-1)	-	-	-	-	-	$n/4\ t_{FA}$
CPA1 (DC2)	n-1	-	-	-	-	-	(2n-2) t_{FA}
CSA2	1	2	2	-	-	-	t_{FA}
CSA3	n-1	1	1	-	-	-	t_{FA}
CSA4	n	-	-	-	-	-	t_{FA}
CPA2	n	-	-	-	-		$n\ t_{FA}$
CPA3	n-1	-	-	-	-	-	(n-1) t_{FA}
CPA4	n-1	-	-	-	-	-	(n-1) t_{FA}
MUX	-	-	-	-	-	n-1	1t_{FA}

3.2 Converter design using MRC for subset {2^n× (2^{n-1}-1) × (2^n+1), 2^{n+1}-1, 2^n-1}

In order to calculate X from subset {P_{123}, P_4, P_5} = {2^n× (2^{n-1}-1) × (2^n+1), 2^{n+1}-1, 2^n-1} with corresponding residues (Y, x_4, x_5) by using MRC, we have

$$X = v_1 + P_{123}v_2 + P_{123}P_4v_3 \quad (24)$$

$$X = v_1 + (2^n+1)(2^{n-1}-1)2^n v_2 + (2^n+1)(2^{n-1}-1)2^n(2^{n+1}-1)v_3 \quad (25)$$

$$v_1 = Y = Y_1 + Y_2 + Y_3 + Y_4 \quad (26)$$

(27) $$v_2 = \left| (x_4 - (Y_1 + Y_2 + Y_3 + Y_4)) \left| P_{123}^{-1} \right|_{P_4} \right|_{2^{n+1}-1}$$

(28) $$v_3 = \left| ((x_5 - Y\ ) \left| P_{123}^{-1} \right|_{P_5} - v_2) \left| P_4^{-1} \right|_{P_5} \right|_{P_5}$$

The required multiplicative inverses in Eq. 27 and 28 are

$$\left| P_{123}^{-1} \right|_{P_4} = \begin{cases} (2^2 + 2^3 + \sum_{i=1}^{\frac{n-4}{6}} (2^{6i+1} + 2^{6i+2} + 2^{6i+3})) & if \quad n = 6k+4, k = 1, 2, \ldots \\ (1 + \sum_{i=1}^{\frac{n}{6}-1} (2^{6i-1} + 2^{6i} + 2^{6i+1}) + 2^{n-1} + 2^n) & if \quad n = 6k+6 \\ 2^0 + 2^1 + 2^2 + 2^4 + 2^5 + \sum_{i=1}^{\frac{n-8}{6}} (2^{6i+3} + 2^{6i+4} + 2^{6i+5}) & if \quad n = 6k+8 \end{cases}$$

(29) $$\begin{aligned} \left| P_{123}^{-1} \right|_{P_5} &= \left| (2^n + 1)(2^n)(2^{n-1} - 1) \right|_{2^n-1} = -1 \\ \left| P_4^{-1} \right|_{P_5} &= \left| (2^{n+1} - 1)^{-1} \right|_{2^n-1} = 1 \end{aligned}$$

Equation (27) can be rewriten as

(30) $$v_2 = \left| (x_4 + \bar{Y}_1 + Y_{21} + Y_{22} + Y_{23} + Y_{31} + Y_{32} + Y_{41} + Y_{42}) \left| P_{123}^{-1} \right|_{P_4} \right|_{2^{n+1}-1}$$

Where

$$\begin{aligned} Y_{21} &= \bar{v}_{2,0} 1 \bar{v}_{2,n-2} \cdots \bar{v}_{2,0} \\ Y_{22} &= \bar{v}_{3,2} \bar{v}_{3,1} \bar{v}_{3,0} \bar{v}_{2,n-2} \cdots \bar{v}_{2,1} \\ Y_{23} &= 1111 \bar{v}_{3,n-1} \cdots \bar{v}_{3,3} \\ Y_{31} &= \bar{v}_{3,1} \bar{v}_{3,0} \underbrace{11 \cdots 10}_{n-2} \\ Y_{32} &= 111 \bar{v}_{3,n-1} \cdots \bar{v}_{3,2} \\ Y_{41} &= v_{3,0} v_{3,n-1} \cdots v_{3,0} \\ Y_{42} &= 00 v_{3,n-1} \cdots v_{3,1} \end{aligned}$$

In order to simplified Eq. 30, (n+1)-bit CSAs with EAC are employed. As shown in figure 4, after using four levels of (n+1)-bit CSA with EAC, v_{21} and v_{22} will be resulted. Therefore Eq. (30) can be simplified as

(31) $$v_2 = \left| (v_{21} + v_{22}) \times \left| P_{123}^{-1} \right|_{P_4} \right|_{2^{n+1}-1}$$

In case n=6k+4, we have

(32) $$v_2 = \left| (v_{21} + v_{22}) \times (2^2 + 2^3 + 2^7 + 2^8 + 2^9 + 2^{13} + 2^{14} + 2^{15} + \cdots) \right|_{2^{n+1}-1}$$

(33) $$v_2 = \left| \begin{matrix} CLS(v_{21}, 2) + CLS(v_{21}, 3) + CLS(v_{21}, 7) + \cdots + \\ CLS(v_{22}, 2) + CLS(v_{22}, 3) + CLS(v_{22}, 7) + \cdots \end{matrix} \right|_{2^{n+1}-1}$$

In case n=6k+6, we have

$$v_2 = \left| (v_{21} + v_{22}) \times (1 + 2^5 + 2^6 + 2^7 + \cdots + 2^{n-1} + 2^n) \right|_{2^{n+1}-1} \quad (34)$$

$$v_2 = \left| \begin{matrix} CLS(v_{21}, 0) + CLS(v_{21}, 5) + CLS(v_{21}, 6) + \cdots + CLS(v_{21,n-1}) + CLS(v_{21,n}) + \\ CLS(v_{22}, 0) + CLS(v_{22}, 5) + CLS(v_{22}, 6) + \cdots + CLS(v_{22,n-1}) + CLS(v_{22,n}) \end{matrix} \right|_{2^{n+1}-1} \quad (35)$$

In case n = 6k+8 we have

$$v_2 = \left| (v_{21} + v_{22}) \times (2^0 + 2^1 + 2^2 + 2^4 + 2^5 + 2^9 + 2^{10} + 2^{11} + \cdots) \right|_{2^{n+1}-1} \quad (36)$$

$$v_2 = \left| \begin{matrix} CLS(v_{21},0)+CLS(v_{21},1)+CLS(v_{21},2)+CLS(v_{21},4)+\cdots+ \\ CLS(v_{22},0)+CLS(v_{22},1)+CLS(v_{22},2)+CLS(v_{22},4)+\cdots \end{matrix} \right|_{2^{n+1}-1} \qquad (37)$$

In Figure 3, the design of Eq. (31) in three cases are shown.

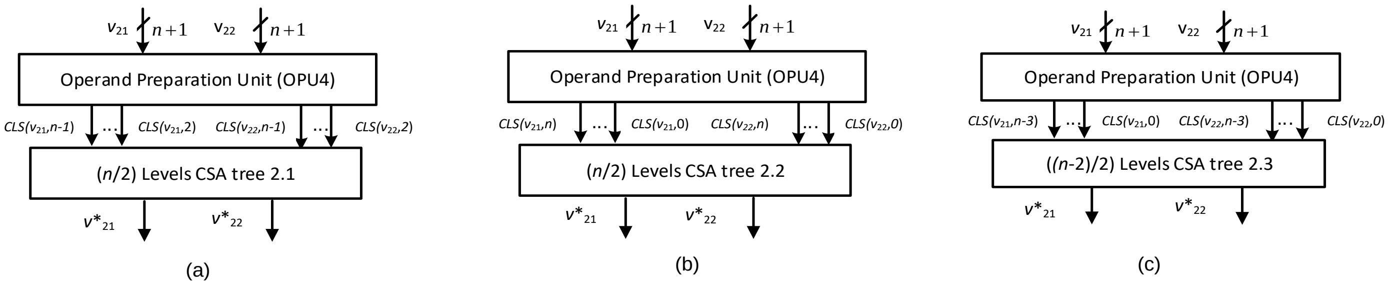


Figure 3: Design Eq. (31): (a)- in case n=6k+4, k=1, 2,..., (b)-in case n=6k+6 ,k=1,2,.., (c)- n=6k+8,k=1,2,...

For v_3, we have

$$v_3 = \left| ((x_5 - Y)\left|P_{123}^{-1}\right|_{P_5} - v_2)\left|P_4^{-1}\right|_{P_5} \right|_{P_5} \qquad (38)$$

By substitution of $\left|P_{123}^{-1}\right|_{P_5} = -1$ and $\left|P_4^{-1}\right|_{P_5} = 1$ in Eq. 38, we have

$$v_3 = \left|((x_5 - Y)(-1) - v_2) \times 1\right|_{2^n-1} \qquad (39)$$

Using lemma 1 and 2 in Eq. 39 results

$$v_3 = \left|Y_1 + Y_2 + Y_3 + Y_4 - x_5 - v_2\right|_{2^n-1} \qquad (40)$$

Two cases can be considered for v_2 in Eq. 40

$$v_2 = \begin{cases} \left|v^*_{21} + v^*_{22}\right|_{2^{n+1}-1} & if\ v^*_{21} + v^*_{22} < 2^{n+1} \\ \left|v^*_{21} + v^*_{22} + 1\right|_{2^{n+1}-1} & if\ v^*_{21} + v^*_{22} \geq 2^{n+1} \end{cases} \qquad (41)$$

By considering Y_is in binary representation as $Y_1 = Y_{1,n} \cdots Y_{1,0}$, $Y_2 = Y_{2,3n-2} \cdots Y_{2,0}$, $Y_3 = Y_{3,2n-2} \cdots Y_{3,0}$, $Y_4 = Y_{4,2n-1} \cdots Y_{4,0}$ and using Lemma 1, Eq. 40 can be rewritten as

$$v_3 = \left|Z_{11} + Z_{12} + Z_{21} + Z_{22} + Z_{23} + Z_{31} + Z_{32} + Z_{41} + Z_{42} + \bar{x}_5 + Z_{51} + Z_{52}\right|_{2^n-1} \qquad (42)$$

Where

$$Z_{11} = Y_{1,n-1}Y_{1,0}$$

$$Z_{12} = \underbrace{00\cdots0}_{n-1}Y_{1,n}$$

$$Z_{21} = Y_{2,n-1} \cdots Y_{2,0}$$

$$Z_{22} = Y_{2,2n-1} \cdots Y_{2,n}$$

$$Z_{23} = 0Y_{2,3n-2} \cdots Y_{2,2n}$$

$$Z_{31} = Y_{3,n-1} \cdots Y_{3,0}$$

$$Z_{32} = 0Y_{3,2n-1} \cdots Y_{3,n}$$

$$Z_{41} = Y_{4,n-1} \cdots Y_{4,0}$$

$$Z_{42} = Y_{4,2n-1} \cdots Y_{4,n}$$

$$Z_{51} = \bar{v}_{2,n-1} \cdots \bar{v}_{2,0}$$

$$Z_{52} = \underset{n-1}{1\cdots1}\bar{v}_{2,n}$$

Finally, *X* can be calculated as

$$X = v_1 + (2^n+1)(2^{n-1}-1)2^n v_2 + (2^n+1)(2^{n-1}-1)2^n(2^{n+1}-1)v_3 \quad (43)$$

$$X = (-2^{2n}-2^n)v_2 + (-2^{3n+1}-2^{2n+1})v_3 \quad (44)$$

Eq. 44 can be simplified as

$$X = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 \quad (45)$$

Where

$$X_1 = v_3 v_2 \underset{3n-1}{Y_2}$$

$$X_2 = v_2 \underset{2n-1}{Y_3}$$

$$X_3 = v_3 \underset{n-1}{0\cdots0}\underset{2n}{Y_4}$$

$$X_4 = v_3 \underset{n-1}{0\cdots0}\underset{n+1}{Y_1}$$

$$X_5 = v_3 \underset{n-2}{0\cdots0}10$$

$$X_6 = \bar{v}_3\bar{v}_2 \underset{2n}{1\cdots1}$$

$$X_7 = \bar{v}_3\bar{v}_2 \underset{n}{1\cdots1}$$

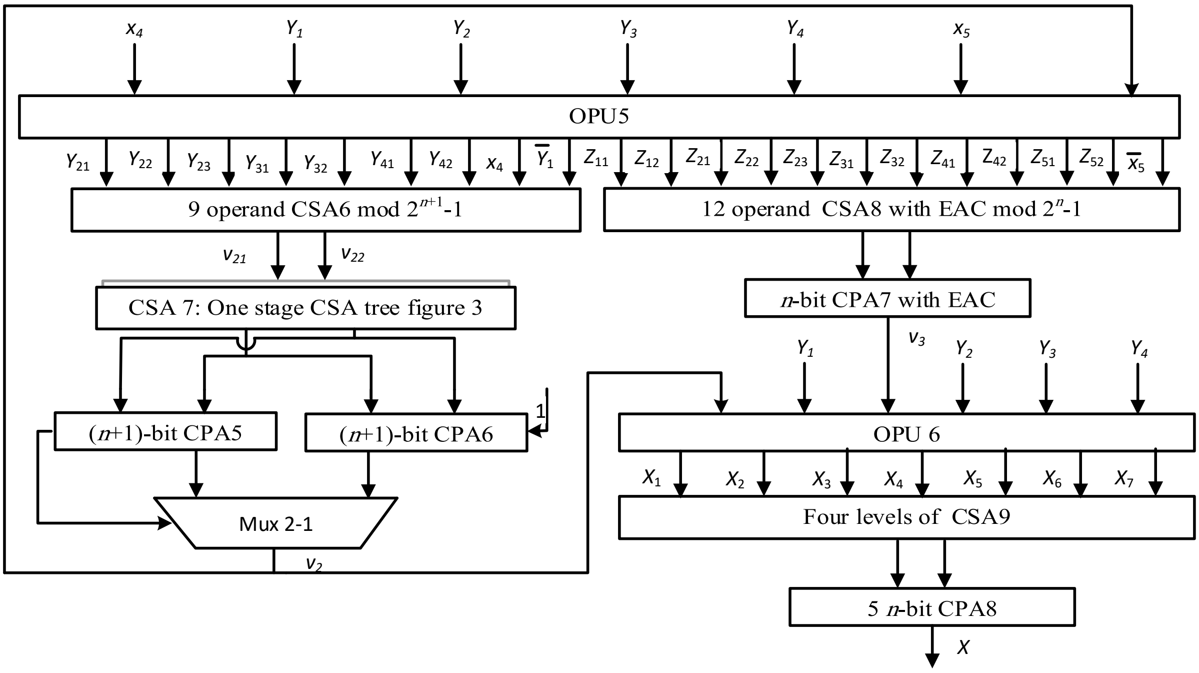


Figure 4. Second level design of the reverse converter

The second level design of the reverse converter including calculation of v_2, v_3 and *X* are shown in figure 4. The hardware cost is specified based on Eqs. (30) and (42). Equation (30) is implemented by 9 operand CSA mod 2^{n+1}-1, one stage CSA tree (figure 3), two (*n*+1)-bit carry propagate adder and a multiplexer. Eq. (42) is implemented by 12 operand CSA with EAC mod 2^n-1, a (*n*)-bit CPA with EAC, after obtaining v_2 and v_3, Equation (43) is implemented by Four levels of carry save adder and (5*n*)-bit carry propagate adder.

Table 2 shows the hardware details and delay of each component for the second level design of proposed reverse converter.

Table 2. Detailed of each component for the second level design of the reverse converter

Component	FA	XOR	AND	XNOR	OR	MUX2-1	Delay
CSA6	7*n*+3	2	2	1	1	-	4*t_{FA}*

CSA7	a:(*n*-2)(*n*+1)
	b:(*n*-2)(*n*+1)
	c: (*n*-4)(*n*+1)
		-	-	-	-	-	a: *n*/2-t_{FA}
							b: *n*/2-t_{FA}
							c: (*n*/2+1)-t_{FA}

CPA5	(*n*+1)	-	-	-	-	-	(*n*+1)-t_{FA}
CPA6	(*n*+1)	-	-	-	-	-	(*n*+1)-t_{FA}
CSA8	9*n*	-	-	-	-	-	5t_{FA}
CPA7	*n*	-	-	-	-	-	2*n*-t_{FA}
CSA9	3*n*	-	-	*n*	*n*	-	4t_{FA}
CPA8	5*n*	-	-	-	-	-	5*n*-t_{FA}
MUX	-	-	-	-	-	*n*+1	1t_{FA}

Numerical Example

Numeral example consider the moduli set $\{2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1\}$ where $n = 10$. Now, with due attention to the moduli set {1025, 511, 1024, 2047, 1023} and the given RNS numbers {12, 8, 15, 22, 3} the corresponding weighted number *X* can be calculated as follows.

First, by putting the values of RNS numbers x_1, x_2, x_3, and *n* in Eqs. (9-19) we have:

$v_1 = x_1 = 12$

$$v_2 = \left|(8-12)\left|1025^{-1}\right|_{511}\right|_{511} = 169$$

Or

$$\left|1025^{-1}\right|_{511} = (1+2^2+2^4+2^6+2^8)$$

$$v_2 = \left|(8-12)(1+2^2+2^4+2^6+2^8)\right|_{2^9-1} = 169$$

Or with using lemma 2:

$$v_2 = \left|(L_1+L_2+L_3)(1+2^2+2^4+2^6+2^8)\right|_{2^9-1}$$

Where

$L_1 = 000001000 = 8$

$L_2 = 111110011 = 499$

$L_3 = 000000000 = 0$

$$v_2 = \left|(499+8+0)\times(341)\right|_{511} = 169$$

$$v_3 = \left|\left((15-12)\left|1025^{-1}\right|_{1024} - 169\right)\left|511^{-1}\right|_{1024}\right|_{1024} = 166$$

$$\left|1025^{-1}\right|_{1024} = 1$$

$$\left|511^{-1}\right|_{1024} = -513$$

Or

$$v_3 = \left|(12+169-15)(513)\right|_{1024} = 166$$

The number *Y* is obtained from Equation. (22)

$Y = 12+(1025\times169)+(1025\times511\times166) = 87119887$

Then, according to equation (26) we have $v_1 = Y = 87119887$ and by putting the values of RNS numbers x_4, x_5 and *n* in Eqs. (27), (29) and (39) we obtained v_2 and v_3.

$$v_2 = \left|(22-87119887)\times 908\right|_{2047} = 1693$$

$$\left|P_{123}^{-1}\right|_{P_4} = \left|((2^{10}+1)(2^{10})(2^9-1))^{-1}\right|_{2^{11}-1} = 2^2+2^3+2^7+2^8+2^9 = 908$$

$$v_3 = \left|((3-87119887)\times(-1)-1693)\times 1\right|_{1023} = 534$$

$$\left|P_{123}^{-1}\right|_{P_5} = \left|((2^{10}+1)(2^{10})(2^9-1))^{-1}\right|_{2^{10}-1} = -1$$

$$\left|P_4^{-1}\right|_{P_5} = \left|(2^{10+1}-1)^{-1}\right|_{2^{10}-1} = 1$$

Finally, the number *X* can be calculated from the Equation. (43)

X= $87119887 + (1025 \times 511 \times 1024 \times 1693) + (1025 \times 511 \times 1024 \times 2047 \times 534) = 587186422889487$

This result is verified as follows:

$$x_1 = \left|587186422889487\right|_{1025} = 12$$

$$x_2 = \left|587186422889487\right|_{511} = 8$$

$$x_3 = \left|587186422889487\right|_{1024} = 15$$

$$x_4 = \left|587186422889487\right|_{2047} = 22$$

$$x_5 = \left|587186422889487\right|_{1023} = 3$$

Performance Evaluation

In this section, the proposed reverse converters have been evaluated and compared with their closest counterparts: $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n\text{-}1}\text{-}1, 2^{n+1}\text{-}1\}$ proposed in [46], $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{+}1, 2^{n\text{-}1}\text{+}1\}$ proposed in [47], and also $\{2^k, 2^n\text{-}1, 2^n\text{+}1, 2^{n+1}\text{-}1, 2^{n\text{-}1}\text{-}1\}$ proposed in [48]. The conversion delay estimation of the proposed converters and other converters in literature are illustrated in Table 3. In table 3, *d*CSA*(a)* denotes the delay of an *a*-operand CSA, *d*CPAm*(a)* [*d*CPAp*(a)*] denotes the delay of an *a*-bit [(*a* + 1*)*-bit] adder mod $2^a - 1$ ($2^a + 1$*)*, whereas *d*ADD*(a)* is the delay of an *a*-bit CPA.

Delay and area of the proposed reverse converter and other works reported in [46-48] are calculated based on full adder and are included in table 4. In order to have fair comparison, delay of modulo 2^k-1 and 2^k+1 adder reported in [42] with 2*k* and 4(*k*+1) delay of full adder and ripple carry adder for CPA are considered.

Reverse converters reported in [48] have one modulo (2^{2n}-1), one modulo (2^{n+1}-1), one modulo ($2^{n\text{-}1}$-1) adder and one 4*n*-bit CPA in its critical path. Considering the delay of modulo 2^k±1 reported in [42], the reverse converter reported in [48] have (12*n*) d_{FA}. It should be noted that delay of CSA tree and multiplication by constant with different inputs will be added to (12*n*) t_{FA} according to version 1 and 2 reported in [48]. Therefore Version 1 and 2 reverse converters reported in [48] have achieved (12n+ O $(\log_{\sqrt{2}}(\frac{n}{2}-1))$ + O $(\log_{\sqrt{2}}\frac{n}{3})$ +18) delay of full adder and (12*n* + O $(\log_{\sqrt{2}}(\frac{n}{2}\text{-}1))$ +O $(\log_{\sqrt{2}}\frac{n}{3})$ +10)t_{FA}, respectively. As discussed in [48], the delay of constant multiplication blocks can be expressed as $O(\log_{\sqrt{2}}\frac{n}{3})$ is very close to the delay of equivalent CSA trees given by *d*CSA*(a)* ≈ *O(*log1.5 *a)* [49].

Table 3. Delay estimation for various five moduli set reverse converter

Converter	Critical path delay estimation of various reverse converters
[46]	$d_{\text{CSA}}(3)+d_{\text{CPAm}}(2n)+d_{\text{CSA}}(4)+d_{\text{CPAm}}(n+1)+d_{\text{CSA}}(\frac{n}{2})+d_{\text{CPAm}}(n+1))+d_{\text{CSA}}(8)+d_{\text{CPAm}}(n\text{-}1)+d_{\text{CSA}}(\frac{n}{3})+d_{\text{CPAm}}(n\text{-}1)+d_{\text{CSA}}(3)+d_{\text{ADD}}(4n)$
[47]	$d_{\text{CSA}}(3)+d_{\text{CPAm}}(2n)+d_{\text{CSA}}(3)+d_{\text{CSAp}}(n+1)+d_{\text{CSA}}(\frac{n}{2})+d_{\text{CPAp}}(n+1)+d_{\text{CSA}}(4)+d_{\text{CPAp}}(n\text{-}1)+d_{\text{CSA}}(\frac{n}{3})+d_{\text{CSAp}}\ (n\text{-}1)+\text{d}_{\text{ADD}}(4n+1)$
[48]- version1	$d_{\text{CSA}}(\left\lceil\frac{k}{2n}\right\rceil+2)+d_{\text{CPAm}}(2n)+d_{\text{CSA}}(\left\lceil\frac{2n+k}{n+1}\right\rceil+1)+\text{O}(\log_{\sqrt{2}}(\frac{n}{2}-1))$ $+d_{\text{CPAm}}(n+1)+d_{\text{CSA}}(\left\lceil\frac{k}{n-1}\right\rceil+7)+\text{O}(\log_{\sqrt{2}}\frac{n}{3})+d_{\text{CPAm}}(n\text{-}1)+d_{\text{CSA}}(3)+d_{\text{ADD}}(4n)$
[48]- version2	$d_{\text{CSA}}(\left\lceil\frac{k}{2n}\right\rceil+2)+\text{d}_{\text{CPAm}}(2n)+d_{\text{CSA}}(3)+\text{O}(\log_{\sqrt{2}}(\frac{n}{2}\text{-}1))+d_{\text{CPAm}}(n+1)+d_{\text{CSA}}(4)+\text{O}$ $(\log_{\sqrt{2}}\frac{n}{3})+d_{\text{CPAm}}(n\text{-}1)+d_{\text{ADD}}(4n)$
Proposed-DC1	$d_{\text{CSA}}(n/2)+d_{\text{CPA}}(n\text{-}1)+d_{\text{CPA}}(n)+d_{\text{CSA}}(10)+d_{\text{CSA}}(n/2+1)+d_{\text{CPA}}(n+1)+d_{\text{CPAm}}(n)+d_{\text{CPA}}(5n)$
Proposed-DC2	$d_{\text{CSA}}(n/4)+d_{\text{CPAm}}(n\text{-}1)+d_{\text{CPA}}(n\text{-}1)+d_{\text{CPA}}(n)+d_{\text{CSA}}(10)+d_{\text{CSA}}(n/2+1)+d_{\text{CPA}}(n+1)+d_{\text{CPAm}}(n)+d_{\text{CPA}}(5n)$

The first design of the proposed reverse converter (DC1) have two *n*-bit CPA, one (*n*+1)-bit CPA, one modulo 2^n-1 adder, and 5*n*-bit CPA in its critical path. Considering the same assumption for modulo 2^k±1 adder [42] and [52], ($10n$+1) d_{FA} in addition to delays of two CSA trees with (n-2) and n inputs will be resulted. The second design structure is the same as DC1 with one extra modulo (2^{n-1}-1) adder in critical path and the number of inputs of the CSA tree 1 is reduced to (n-2)/2. Therefore, the second design has achieved to (12n) t_{FA} in addition to delays of two CSA trees with ((n-2)/2) and n inputs. It can be seen that DC1 converter has achieved a faster conversion compared to converters reported in [48]. DC2 converter has approximately the same delay compared to [48].

Table 4. Hardware costs and delay estimation for various five moduli set reverse converter in terms of delay and area of Full adder

Converter	**Hardware Requirement**	**Conversion Delay in t_{FA}**
[46]	$((5n^2+43n+m^*)/6+16n-1)A_{FA}+$ $(6n+1)A_{NOT}$	$(18n+L^*+7)t_{FA}$
[47]	Inv., $n=6k+1$: $(5n^2+150n+65)/12$ Inv., $n=6k+3$: $(5n^2+146n-3)/12$ Inv., $n=6k+5$: $(5n^2+130n+65)/12$	$d_{CSA}(3)+d_{CPAm}(2n)+d_{CSA}(3)+d_{CSAp}(n+1)+d_{CSA}$ $(\frac{n}{2})$ $+d_{CPAp}(n+1)+d_{CSA}(4)+d_{CPAp}(n-1)+d_{CSA}(\frac{n}{3})$ $+d_{CSAp}(n-1)+d_{ADD}(4n+1)$
[48]- ** version1	3. $(n+1)+n+(n+1).\Omega$ $(2\log_2(\frac{n}{2}-1))$ + $(n-1)+(n-1).\Omega$ $(2\log_2\frac{n}{3})$ $+24n-6$	$(12\mathrm{n}+ O(\log_{\sqrt{2}}(\frac{n}{2}-1)) + O(\log_{\sqrt{2}}\frac{n}{3}) +18)t_{FA}$
[48]- ** version2	$(n+1)+n+(n+1).\Omega(2\log_2(\frac{n}{2}-1))$ $+3.(n-1)+$ $(n-1).\Omega(2\log_2\frac{n}{3})$ $+30n-2$	$(12n + O(\log_{\sqrt{2}}(\frac{n}{2}\text{-}1)) +O(\log_{\sqrt{2}}\frac{n}{3})+10)t_{FA}$
Proposed-DC1	a: $(2n^2+28n+5)A_{FA}+$ $(n+1)A_{XOR}+(n+1)$ $A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+$ $(2n)A_{MUX2\text{-}1}$ b: $(2n^2+28n+5)A_{FA}+$ $(n+1)A_{XOR}+(n+1)$ $A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+$ $(2n)A_{MUX2\text{-}1}$ c:$(2n^2+26n+3)A_{FA}+$ $(n+1)A_{XOR}+(n+1)$ $A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+$ $(2n)A_{MUX2\text{-}1}$	a: $(10n+ d_{CSA}(n/2)+ d_{CSA}(n/2+1) + 13)t_{FA}+ 2t_{MUX}$ b: $(10n+ d_{CSA}(n/2)+ d_{CSA}(n/2+1) + 13)t_{FA}+ 2t_{MUX}$ c: $(10n+ d_{CSA}(n/2)+ d_{CSA}(n/2+1) + 15)t_{FA}+ 2t_{MUX}$
Proposed-DC2	a:$(1.5n^2+29.5n+4)A_{FA}+(n+1)A_{XOR}+$ $(n+1)A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+$ $(2n)A_{MUX2\text{-}1}$ b:$(1.5n^2+29.5n+4)A_{FA}+(n+1)A_{XOR}+$ $(n+1)A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+$ $(2n)A_{MUX2\text{-}1}$ c:$(1.5n^2+27.5n+2)A_{FA}+(n+1)A_{XOR}+$ $(\mathrm{n}+1)A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+$ $(2n)A_{MUX2\text{-}1}$	a: $(12n+ d_{CSA}(n/4)+ d_{CSA}(n/2+1)+ 12)t_{FA}+ 2t_{MUX}$ b: $(12n+ d_{CSA}(n/4)+ d_{CSA}(n/2+1)+ 12)t_{FA}+ 2t_{MUX}$ c: $(12n+ d_{CSA}(n/4)+ d_{CSA}(n/2+1)+ 15)t_{FA}+ 2t_{MUX}$

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

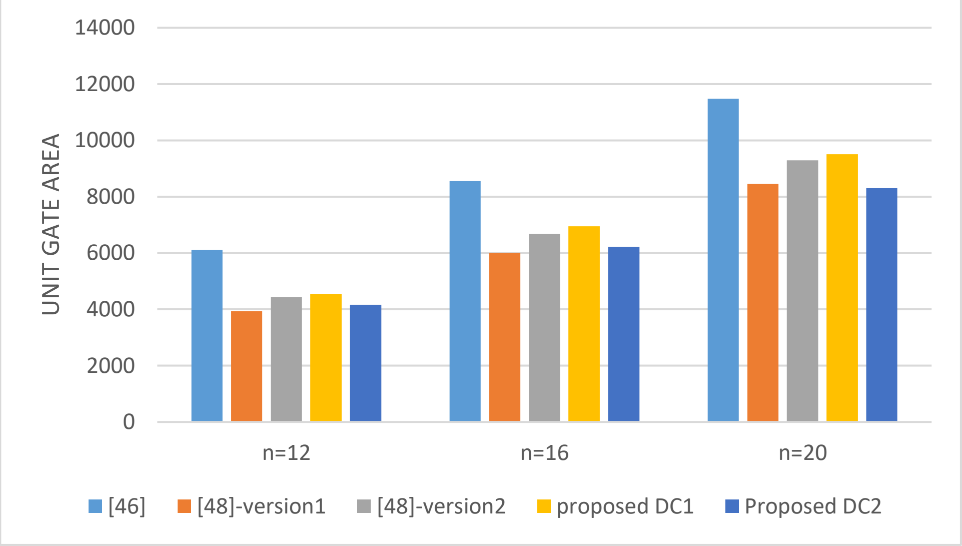
***Big-omega (Ω) impression is used for the lower bound of the area complication of multiplication by the multiplicative inverse [54].*

To have a supreme analogy and concluding area and delay assessment, the unit gate model [53-54] is used. According to this model, each FA, half adder (HA), 2×1 MUX, XOR, XNOR, AND, OR gates considered as 7, 3, 3, 2, 2, 1, 1 gates in area and 4, 2, 2, 2, 2, 1, 1 gates in delay, respectively. Table 5 displays the unit gate area (A) and unit gate delay time (T) for all converters. To provide a more precise analysis, for different values of n the unit gate delay, area and area × time (AT) are computed as shown in figure 5. It can be seen that the proposed converter-DC2 has attain less hardware requirements and the first design (DC1) has attain better efficiency in delay and AT metric compared to the other converters in the literature.

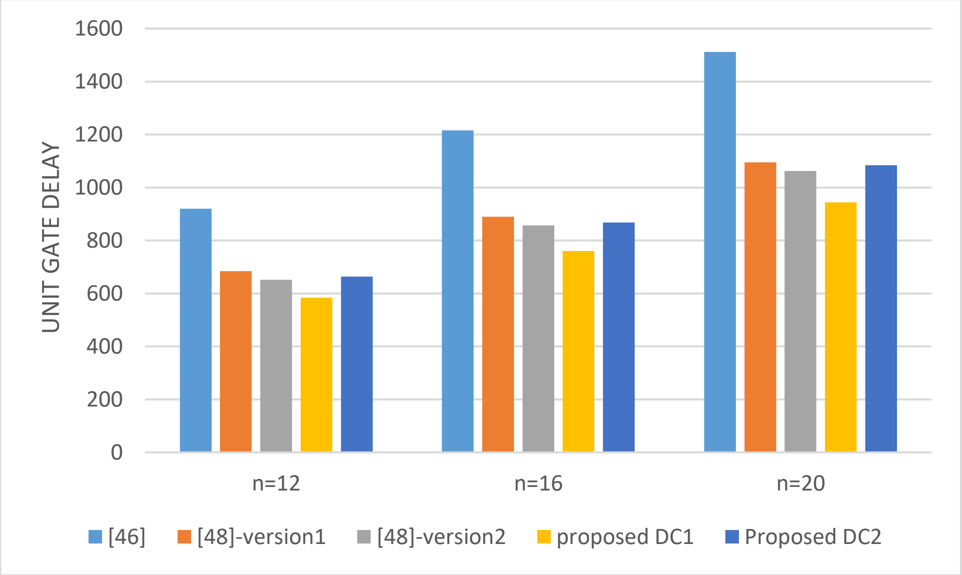
Table 5. Unit gate delay and area comparison for various reverse converter

converter	Unit gate area	Unit gate delay
[46]	$(5n^2+43n+m^*)7/6+118n-6$	$72n+4L^*+28$
[47]	-	-
[48]-version1	$(35/3)n^2+(574/3)n-42$	$(50n+4n/3+68)t_{FA}$
[48]-version2	$(35/3)n^2+(700/3)n-42$	$(50n+4n/3+36)t_{FA}$
Proposed-DC1	a: $14n^2+208n+41$ b: $14n^2+208n+41$ c: $14n^2+194n+27$	a:$(44n+56)t_{FA}+4t_{MUX}$ b: $(44n+56)t_{FA}+4t_{MUX}$ c: $(44n+64)t_{FA}+4t_{MUX}$
Proposed-DC2	a: $10.5n^2+218.5n+34$ b: $10.5n^2+218.5n+34$ c: $10.5n^2+204.5n+20$	a:$(51n+52)t_{FA}+4t_{MUX}$ b: $(51n+52)t_{FA}+4t_{MUX}$ c: $(51n+64)t_{FA}+4t_{MUX}$

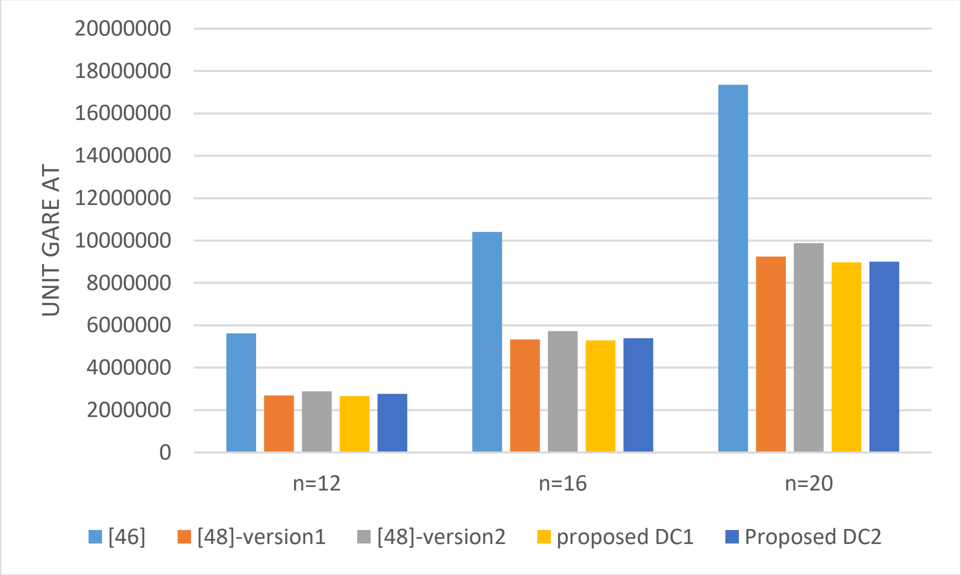
*m=n-4 for n=6k-2, m= 9n-12 for n=6k and m=5n-8 for n=6k+2. L is the number of the levels of a CSA tree with ((n/2) +1) inputs.



(a)



(b)



(c)

Figure 5. Unit gate comparison: a) Unit Gate Area, b) Unit Gate Delay, C) Unit Gate AT

Table 6. FPGA Virtex-7 synthesis results for moduli {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1} reverse converters

Converter	Platform	*n= 12*		*n=16*		*n=20*	
		Delay (ns)	*Area (LUTs)*	*Delay (ns)*	*Area (LUTs)*	*Delay (ns)*	*Area (LUTs)*
[46]	Virtex7	27.901	677	30.542	948	34.147	1372
[48]-version 1	Virtex7	19.489	526	23.894	821	26.721	1116
[48]-version 2	Virtex7	18.515	598	22.903	907	25.546	1240
Proposed- DC1	Virtex7	16.422	625	20.112	917	22.674	1298
Proposed-DC2	Virtex7	19.252	558	23.451	853	25.878	1187

In order to study the effect of different reverse converters for five moduli set {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1} in the FPGA implementation, the converters of [46] and [48] with the same configurations reported in [42] are described with VHDL and also has been synthesized using FPGA, namely, Xilinx Virtex 7 (part xc7vx415t). The Xilinx ISE (version 14.7) tool are used for the synthesis. Table 6 presents experimental results for the FPGAs Virtex 7, from Xilinx. The results show that compared to fastest report in [48]-version2, DC1 reverse converter has achieved 11%, 12% and 11% improvement in speed for n=12, 16 and 20, respectively.

CONCLUSION

This paper proposes two efficient reverse converters for five-moduli set {2^n+1, 2^{n-1}-1, 2^n, 2^{n+1}-1, 2^n-1}. The converters are designed in a two-level structure and offer a high-speed arithmetic unit due to its balanced moduli in the form 2^k and 2^k-1. Furthermore, MRC algorithm is used to design both levels of our novel reverse converters which results in ROM free and adder based structures. According to unit gate delay and area estimation, the first design (DC1) is faster than the similar five moduli reverse converters in literature for the different dynamic range, ensuing a minor penalty in hardware requirement while the second design (DC2) requires less hardware cost. In comparison with state-of-the-art studies, the first design (DC1) has achieved higher performance in delay and AT metric. The proposed converters and recent similar works in literature are described with VHDL and synthesized on Xilinx virtex 7 FPGA. The results show that, comparing to the latest work in literature, the DC1 design has achieved 11%, 12% and 11% improvement in speed for n=12, 16 and 20, respectively.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Conflict of interest

The authors declare that there are no conflict of interests.

REFERENCES

1. Navi K, Molahosseini A, and Esmaeildoust M. How to Teach Residue Number System to Computer Scientists and Engineers, IEEE. 2011. 54:156–163.
2. Bajard JC, Meloni N, and Plantard T. Efficient RNS bases for cryptography. In: Proceedings of IMACS 2005 World Congress, Paris, France. 2005 July.
3. Mohan PVA. Residue Number Systems: Theory and Applications, Cham, Switzerland:Birkhäuser. 2016.
4. Schinianakis D and Stouraitis T. RNS-Based Public-Key Cryptography (RSA and ECC), Cham, Switzerland:Springer. 2017 March; 311-344.
5. Rivest RL, Shamir A, and Adleman LM., A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM. 1978. 21 (2): 120–126.
6. Bajard JC, and Imbert L. A full RNS implementation of RSA. IEEE Trans. on Comput. 2004 June; 53 (6): 769-774.
7. Perin G, Imbert L, Torres L, and Maurine P. Electromagnetic analysis on RSA algorithm based on RNS", Proc. Euromicro Conf. Digital System Design (DSD).2013 Sep;345-352.
8. Ochoa-Jiménez E, Rivera-Zamarripa L, Cruz-Cortés N, and Rodríguez-Henríquez F. Implementation of RSA signatures on GPU and CPU architectures. IEEE Access. 2020. 8: 9928-9941.
9. Miller VS. Advances in Cryptology – CRYPTO '85 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, Ch. Use of Elliptic Curves in Cryptography. 1986:17–426.
10. Schinianakis DM, Fournaris AP, Michail HE, Kakarountas AP, and Stouraitis T. An RNS implementation of an Fp elliptic curve point multiplier", IEEE Trans. Circuits Syst.-I. 2009 June; 56 (6): 1202-1213.
11. Antao S, Bajard JC, and Sousa L. RNS based elliptic curve point multiplication for massive parallel architectures", Comput. J. 2012 May; 55 (5): 629-647.
12. Esmaeildoust M, Schinianakis D, Javashi H, Stouraitis T, and Navi K. Efficient RNS implementation of elliptic curve point multiplication over GF (p). IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 2013 Aug; 21 (8): 1545-1549.
13. Ambrose JA, Pettenghi H, and Sousa L. DARNS: A randomized multi-modulo RNS architecture for double-and-add in ECC to prevent power analysis side channel attacks. Proc. 18th Asia and South Pacific Design Automation Conf. 2013: 620-625.
14. Asif S, Hossain M, Kong Y, and Abdul W. A fully RNS based ECC processor", Integr. VLSI J. 2018. 61:138-149.
15. Ramirez J. RNS-Enabled Digital Signal Processor Design. IEE Electronics Letters.2002. 38 (6): 266-268.
16. Chen J and Hu J. Energy-efficient digital signal processing via voltage-overscaling-based residue number system. IEEE Trans. Very Large Scale Integr. (VLSI) Syst.2013 July; 21 (7): 1322-1332.
17. Albicocco P, Cardarilli GC, Nannarelli A, and Re M. Twenty years of research on RNS for DSP: lessons learned and future perspectives. Proc. 14th Int. Symp. Integrated Circuits (ISIC).2014 Dec; 436-439.
18. Chang CH, Molahosseini AS, Zarandi AAE, and Tay TF. Residue number systems: A new paradigm to datapath optimization for low-power and high-performance digital signal processing applications. IEEE Circuits Syst. Mag. 2015. 15 (4): 26–44.
19. Cardarilli GC, Nannarelli A, and Re M. RNS applications in digital signal processing in Embedded Systems Design with Special Arithmetic and Number Systems, Cham, Switzerland: Springer. 2017: 181-215.
20. Conway R and Nelson J. Improved RNS FIR filter architectures. IEEE Trans. Circuits Syst. II Exp. Briefs. 2004 Jan; 51 (1): 26-28.
21. Chervyakov NI, Lyakhov PA, and Babenko MG. Digital filtering of images in a residue number system using finite-field wavelets", J. Autom. Control Comput. Sci. 2014 May; 48 (3):180-189.
22. Veligosha AV, Linets GI, Kaplun DI, Klionskiy DM, and Bogaevskiy DV. Implementation of non-positional digital filters. Proceedings of the XIX IEEE International Conference on Soft Computing and Measurements (SCM). 2016 May 25-27; 148-150.
23. Cardarilli GC, Nunzio LD, Fazzolari R, Nannarelli A, Petricca M, and Re M. Design space exploration based methodology for residue number system digital filters implementation. IEEE Trans. Emerg. Topics Comput. 2020 May.
24. Wang W, Swamy MNS, and Ahmad MO. RNS application for digital image processing. Proc. 4th IEEE Int. Workshop System-On-Chip for Real Time Appl.2004: 77-80.
25. Younes D and Steffan P. Efficient image processing application using residue number system", Proc. 20th Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES). 2013 Jun; 468-472.
26. Chervyakov N and Lyakhov P. RNS-Based Image Processing in Embedded Systems Design with Special Arithmetic and Number Systems, Cham, Switzerland: Springer. 2017: 217-245.
27. Barsi F and Maestrini P. Error correcting properties of redundant residue number systems", IEEE Trans. Comput. 1973 Mar; 81: 307-315.
28. Goh VT and Siddiqi MU. Multiple error detection and correction based on redundant residue number systems. IEEE Trans. Commun.2008 Mar; 56 (3): 325-330.

29. Chu J and Benaissa M. Error detecting AES using polynomial residue number systems. Microprocessors Microsyst. 2013 Mar; 37 (2): 228-234.
30. Veligosha AV, Kaplun DI, Klionskiy DM, Bogaevskiy DV, Gulvanskiy VV, and Kalmykov AI. Error Correction of Digital Signal Processing Devices using Non-Positional Modular Codes. Automatic Control and Computer Sciences. 2017. 51 (3): 167-173.
31. Gallant RP, Lambert RJ, Vanstone SA. Faster point multiplication on elliptic curves with efficient endomorphisms. In: Kilian, J. (ed.) CRYPTO 2001. LNCS. Springer, Heidelberg. 2001. (2139):190–200.
32. Hankerson D, Menezes A, and Vanstone S. Guide to Elliptic Curve Cryptography, Springer. 2004.
33. Wang Y, Song X, Aboulhamid M, and Shen H. Adder based residue to binary numbers converters for {2^n-1,2^n,2^n+1},” IEEE Trans. Signal Process. 2002 Jul; 50 (7):1772–1779.
34. Wang W, Swamy MNS, Ahmad MO, and Wang Y. A high speed residue-to-binary converter and a scheme of its VLSI Implementation IEEE Trans. Circuits Syst. II, Exp. Briefs. 2000 Dec; 47 (12): 1576–1581.
35. Mohan PVA. RNS-to-binary converter for a new three-moduli set {2^{n+1}-1, 2n, 2n-1}. IEEE Trans. Circuits Syst. II, Exp. Briefs. 2007 Sep; 54 (9):775–779.
36. Gbolagade K, Chaves R, Sousa L, and Cotofana S. An improved RNS reverse converter for the {2^{2n+1}-1, 2^n, 2^n-1} moduli set. Proc. IEEE Int. Symp. Circuits Syst.2010: 2103-2106.
37. Sheu MH, Siao SM, Hwang YT, Sun CC, and Lin YP. New adaptable three moduli set {2^{n+k}, 2n–1, $2n^{-1}$–1} for residue number system – based on finite impulse response implementation. IEICE Electron. Express. 2016. 13 (11):1–9.
38. Hiasat A.An efficient reverse converter for the three-moduli set (2^{n+1} – 1, 2^n, 2^n –1), IEEE Trans. Circuits Syst. II. 2016. 64 (8): 962– 966.
39. Latha MM, Rachh RR, and Mohan PVA. RNS-to-binary converters for a three-moduli set {2^{n-1} – 1, 2^n–1, 2^{n+k}}, IETE J. Edu. 2017. 58 (1): 20– 28.
40. Hiasat A and Sousa L. On the design of RNS inter-modulo processing units for the arithmetic-friendly moduli sets {2^{n+k}, 2^n–1, 2^{n+1}–1}. Comput. J. 2018. 62 (2): 292-300.
41. Mohan PVA. New reverse converters for the moduli set {2^n-3, 2n+1, 2n-1, 2n+3}. J. Electron. Commun. 2008. 62 (9): 643–658.
42. Mohan PVA and Premkumar AB. RNS-to-binary converters for two four-moduli set {2^n-1, 2n, 2n+1, $2n^{+1}$-1} and {2^n-1, 2n, 2n+1, $2n^{+1}$+1},” IEEE Trans. Circuits Syst. I, Reg. Papers. 2007 Jun; 54 (6): 1245–1254.
43. Cao B, Chang C, and Srikanthan T. 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. 2003. 50 (10): 1296–1303.
44. Sousa L, Antao S, and Chaves R. On the design of RNS reverse converters for the four-moduli set {2^n+1, 2^n-1, 2^n, 2^{n+1}+1} ", IEEE Trans. Very Large Scale Integration Syst. 2013 Oct; 21 (10): 1945-1949.
45. Mohan PVA. Reverse converters for the moduli set {2^n, $2n^{-1}$–1, 2n–1, $2n^{+1}$–1} (n Even). Circuits Systems and Signal Processing. 2018. 37: 3605–3634.
46. Cao B, Chang CH, and Srikanthan T. A residue-to-binary converter for a new five-moduli set,” IEEE Trans. Circuits Syst. I, Reg. Papers. 2007 May; 54 (5): 1041–1049.
47. Patronik P, Berezowski K, Biernat J, Piestrak SJ, and Shrivastava A. Design of an RNS reverse converter for a new five-modulus special set. Proc. ACM GLSVLSI.2012 May 3-4; 67-70.
48. Patronik P and Piestrak SJ. Design of reverse converters for a new flexible RNS Five-Moduli set {2^k, 2^n-1, 2^n+1, 2^{n+1}-1, 2^{n-1} -1} (n Even). Circuits Syst. Signal Process. 2017. 36 (11): 4593-4614.
49. Patronik P and Piestrak,SJ. Design of reverse converters for general RNS moduli sets {2^k, 2^n-1, 2^n+1, 2^{n-1}-1} and {2^k, 2^n-1, 2^n+1, 2^{n+1}-1} (n even). IEEE Trans. Circuits Syst. I Reg. Pap. 2014. 61(6): 1687–1700.
50. Hariri A, Navi K, Rastegar R. A new high dynamic range moduli set with efficient reverse converter,” Elsevier Journal of Computers and Mathematics with Applications. 2008. 55 (4):660-668.
51. Piestrak SJ. A high speed realization of a residue to binary converter. IEEE Transactions on Circuits System II analog and Digital Signal Processing. 1995. 42(10): 661–663.
52. Daghlavi MO, Noorimehr MR, and Esmaeildoust M. Efficient two-level reverse converters for the four-moduli set {2^{n-1}, 2^n–1, 2^{n-1}–1, 2^{n+1}–1}. Analog Integrated Circuits and Signal Processing 108. 2021: 75–87.
53. Noorimehr MR, Hosseinzadeh M, and Navi K. Efficient reverse converters for 4-moduli sets 2^{2n-1}-1, 2n, 2n+1, 2n-1 and 2^{2n-1}, 2^{2n-1}-1, 2n+1, 2n-1 based on CRTs algorithms. Circuits Systems and Signal Processing. 2018. 33(10): 3605–3634.
54. Tyagi A. A reduced-area scheme for carry-select adders. IEEE Transactions on Computers. 1993. 42:1163–1170.