Circuits Syst Signal Process (2018) 37:3605–3634
https://doi.org/10.1007/s00034-017-0725-0

Reverse Converters for the Moduli Set $\{2^n, 2^{n-1} - 1, 2^n - 1, 2^{n+1} - 1\}$($n$ Even)

P. V. Ananda Mohan[1]

Received: 15 March 2016 / Revised: 23 November 2017 / Accepted: 24 November 2017 /
Published online: 5 December 2017

Abstract In this paper, two residue number system (RNS) to binary converters for the moduli set $\{2^n, 2^{n-1}-1, 2^n-1, 2^{n+1}-1\}$ for (n even) are presented. One of them uses a two-level conversion, in which, in the first level, two pairs of moduli are considered to obtain two intermediate decoded numbers. A second-level converter obtains the final decoded number corresponding to these two intermediate decoded numbers. Both levels use mixed radix conversion. The second proposed RNS to binary converter uses the conventional MRC of the four-moduli set. The proposed converters are compared with previously reported conversion techniques for this moduli set and converters for other four, five and eight moduli sets for realizing similar dynamic ranges regarding hardware requirement and conversion time. The hardware resource requirement (A), conversion time (T), AT and AT^2 trade-offs are discussed to bring out the relative advantages of various converters. The proposed converters have been shown to need less hardware or less conversion time than the other some of the reported converters for this moduli set. It has been shown by detailed comparison that converters using conjugate moduli and vertical extension generally exhibit better performance (lower hardware /lower conversion time) than those using no vertical extension, while needing differing word lengths of various moduli. These, however, need slightly complex multipliers/adders in the $(2^n + 1)$ channel. Implementation results on FPGA of the proposed converters for few dynamic ranges also have been presented.

Keywords Residue number systems · Reverse converters · Four-moduli sets · CRT · Mixed radix conversion

✉ P. V. Ananda Mohan
anandmohanpv@live.in

[1] Centre For Development of Advanced Computing, Knowledge Park, #1,Old Madras Road, Byappanahalli, Bangalore 560 038, India

1 Introduction

The residue number system (RNS) using three or more moduli of the form $2^a, 2^b - 1, 2^c + 1$ has attracted considerable attention [1–33,35–40] . In view of the relative difficulty of operations using moduli of the form $(2^c + 1)$ as compared to moduli of the form $2^a, 2^b - 1$, designers have considered moduli sets with moduli of the form 2^a and $2^b - 1$ with a and b values as close as possible so that all moduli processors have about the same word length. The modulo addition, modulo subtraction, modulo multiplication are known to be easy for such moduli. A b-bit modulo addition $(u + v) \bmod (2^b - 1)$ where u and v are b-bit numbers can be performed by a b-bit adder adding u and v with end-around-carry (EAC). A b-bit modulo subtraction, i.e., $(u - v) \bmod (2^b - 1)$ can be carried out by addition of u with one's complement of v in a b-bit adder with EAC. The operation $(u \times v) \bmod (2^b - 1)$ can be easily carried out by rewriting the bits of the partial products in the higher bit positions in lower bit positions by using the periodic property of such moduli: $2^{k+b}x_i \bmod (2^b - 1) = 2^k x_i$ and adding the reduced b-bit partial products using a carry-save adder tree followed by a b-bit adder with EAC. Scaling x_i by $2^k \bmod (2^b - 1)$ can be achieved by left circular left of x_i by k bits.

Only two four-moduli sets M1 $\{2^n, 2^{n-1} - 1, 2^n - 1, 2^{n+1} - 1\}$ [12,13,33], M2 $\{2^k, 2^{n-1} - 1, 2^n - 1, 2^{n+1} - 1\}$ [39] with n even, have been described in the literature which use one modulus of the form 2^α and other three moduli of the form $(2^\beta - 1)$. Esmaeildoust et al. [12] have recently presented two designs of RNS to binary converters for the moduli set M1. These were realized using two-moduli RNS $\{M_a, 2^{n-1} - 1\}$, where M_a is the three-moduli set $\{2^n, 2^n - 1, 2^{n+1} - 1\}$ [2]. They have used mixed radix conversion (MRC) for the two-moduli RNS and employing three converters proposed for the three-moduli subset M_a [2]; they have described six RNS to binary converters. Taheri et al. [33] have described a two-stage RNS to binary converter for this moduli set. In the first stage, the reverse conversion for the moduli set $M_b\{2^{n-1} - 1, 2^n - 1, 2^{n+1} - 1\}$ is carried out using MRC. The second stage uses MRC for the composite moduli set $\{M_b, 2^n\}$. However, they avoid the final addition in the MRC-based converter in the first stage in order to reduce the conversion time. The moduli set M1$\{2^n, 2^{n-1} - 1, 2^n - 1, 2^{n+1} - 1\}$ has been suggested by Schinianaikis et al. for use in ECC processors with $n = 64$ [13]. They have used MRC to obtain mixed radix digits. However, in their application, these MRC digits are used to perform base extension to another moduli set using Horner's rule. The last stage needed in MRC-based RNS to binary conversion was not investigated. In [39], using modulus 2^k in place of 2^n in the moduli set M1, the MRC technique has been used to realize a RNS to binary converter. In this paper, we propose one RNS to binary converter based on two-level conversion technique and also extend the mixed radix conversion technique considered in [13] to realize another RNS to binary converter. In Sect. 2, we present the background material needed for this paper. The detailed derivation of the two proposed converters together with the architectures needed for their realization is presented in Sect. 3 in detail. In Sect. 4, the proposed converters are compared regarding hardware requirement and conversion time with earlier described designs of converters for this moduli set in [12,33]. The proposed designs are also compared with converters for other moduli sets using four or more moduli considering their design for similar dynamic range. These are M3$\{2^n, 2^n-1, 2^n+1, 2^{n+1}-1)$ [4,6,15,35] (n even), M4$\{2^n, 2^n-1, 2^n+1, 2^{n-1}-1\}$ [6] (n even), M5$\{2^n, 2^n-1, 2^n+1, 2^{n+1}+1\}$ (n odd) [4,5,29], M6 $\{2^n, 2^n-1, 2^n+1, 2^{n-1}+1\}$ (n odd) [19], M7$\{2^n, 2^n-1, 2^n+1, 2^{n+1}-1, 2^{n-1}-1\}$ [8], M8$\{2^n, 2^n-1, 2^n+1, 2^{2n+1}-1\}$ [17,30], M9$\{2^{2n}, 2^n-1, 2^n+1, 2^{2n+1}-1\}$ [30], M10$\{2^n, 2^n-1, 2^n+1, 2^{2n}+1\}$ [7], M11$\{2^{2n}, 2^n-1, 2^n+1, 2^{2n}+1\}$ [17] and M12$\{2^{n-5}-1, 2^{n-3}-1, 2^{n-3}+1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$ [26]. Implementation results of the two proposed converters on FPGA for few n values are also presented. In Sect. 5, we conclude the paper.

2 Background Material

A residue number system uses l moduli $\{m_1, m_2, m_3, m_4, \ldots, m_l\}$ and has a dynamic range $D = m_1 \times m_2 \times m_3 \times \cdots \times m_l$, thus being capable of representing uniquely all the numbers X between 0 and D-1 using the residues $\{x_1, x_2, x_3, \ldots, x_l\}$ where X mod $m_i = x_i$ for i= 1, 2,..., l. The process of obtaining the binary number X corresponding to given residues $\{x_1, x_2, x_3, \ldots, x_l\}$ is known as reverse conversion and can be carried out by several techniques [1,18,28,32] and most popular among these are based on using Chinese remainder theorem (CRT) [1] and mixed radix conversion (MRC) [1]. In the classical MRC technique, (l-1) sequential steps are required to compute the various mixed radix digits d_i for $i = 1, 2, \ldots, l-1$ defined as

$$X = x_1 + d_1\, m_1 + d_2\, m_1\, m_2 + \cdots + d_{l-1}\, m_1\, m_2\, m_3 \ldots m_{l-1} \tag{1}$$

In the jth step, the mixed radix digit d_{j-1} determined in the $(j-1)$th step is subtracted from the residues of other moduli m_k where $k = j+1, j+2, \ldots, l$ and multiplied with the multiplicative inverses $\left(\frac{1}{m_j}\right)_{m_k}$. Note that $w = \left(\frac{1}{y}\right)_z$ is the multiplicative inverse of y with respect to modulus z defined such that $w \times y = 1$ mod z.

MRC can be carried out in $\lceil \log_2 l \rceil$ steps also considering several pairs of moduli in each step for an l-moduli RNS. In the first step (considering l = even, i.e., even number of moduli without loss of generality), the pairs can be chosen as $\{m_1, m_2\}, \{m_3, m_4\}, \ldots, \{m_{l-1}, m_l\}$. Mixed radix conversion can be used for each pair $\{m_g, m_{g+1}\}$ to obtain the intermediate decoded numbers $X_{g,g+1}$ as

$$X_{g,g+1} = x_g + m_g \left((x_{g+1} - x_g)_{m_{g+1}} \left(\frac{1}{m_g}\right)_{m_{g+1}} \right)_{m_{g+1}} \tag{2}$$

Each of the next steps considers pairs of residues corresponding to the composite moduli (e.g., in second step composite moduli are of the form $(m_g \times m_{g+1})$) and computes the decoded numbers. Evidently, the size of the operands (composite moduli as well residues corresponding to these moduli) increases progressively from first step to last step. We consider the application of the classical MRC as well as two-level MRC techniques in this paper.

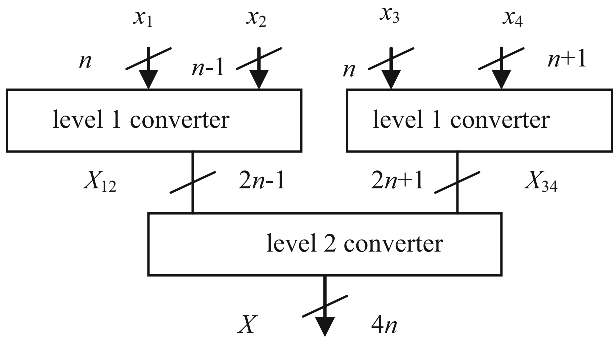

Fig. 1 Two-level mixed radix conversion

3 Proposed RNS to Binary Converters for the Four-Moduli Set

We first denote the moduli as $m_1 = 2^n, m_2 = 2^{n-1} - 1, m_3 = 2^n - 1, m_4 = 2^{n+1} - 1$ with n even and the corresponding residues as x_1, x_2, x_3 and x_4, respectively. It can be easily verified that the moduli are mutually prime. This moduli set has a dynamic range $M = m_1 \times m_2 \times m_3 \times m_4 = (2^{4n} + 7 \times (2^{2n-1} - 2^{3n-1}) - 2^n)$. We describe two RNS to binary converters in the next two subsections.

3.1 Two-Level MRC-Based Converter

We consider the two pairs of moduli $\{2^n, 2^{n-1} - 1\}$ and $\{2^n - 1, 2^{n+1} - 1\}$ in the first level and use MRC on the pairs of respective residues to obtain the two intermediate numbers X_{12} and X_{34}. In the second level, we use MRC for the residues (X_{12}, X_{34}) in the composite two-moduli set $\{M_{12}, M_{34}\}$ where $M_{12} = m_1 \times m_2 = 2^n(2^{n-1} - 1)$ and $M_{34} = m_3 \times m_4 = (2^n - 1)(2^{n+1} - 1)$ to obtain X(Fig. 1). Note that other pairings of moduli in the first level are also possible. The resulting composite moduli sets are $\{M_{13}, M_{24}\}$ and $\{M_{14}, M_{23}\}$. One of the multiplicative inverse needed in the first level for the pairing $\{M_{13}, M_{24}\}$ is not simple power of 2 thus making modulo multiplication with the multiplicative inverse expensive in hardware and time. The multiplicative inverses needed in the first level are simple for the set $\{M_{14}, M_{23}\}$ which are 2 and -2 (see "Appendix"). In the second level, MRC needs multiplicative inverses which have more number of bits which are 1 in the case of pairing $\{M_{13}, M_{24}\}$, thus making the modulo multiplication operation with the multiplicative inverse complex. The multiplicative inverses needed in the second level, while using the pairing $\{M_{14}, M_{23}\}$ has only $(n/2)$ +1 bits which are "1," whereas in case of using $\{M_{12}, M_{34}\}$, the partial products are $n/2$ (see "Appendix").

3.1.1 Computation of X_{12}

It can be seen that X_{12} can be obtained from x_1 and x_2 following (2) as

$$X_{12} = x_1 + m_1 \left(ab\right)_{m_2} = x_1 + 2^n \left(ab\right)_{2^{n-1}-1}, \tag{3}$$

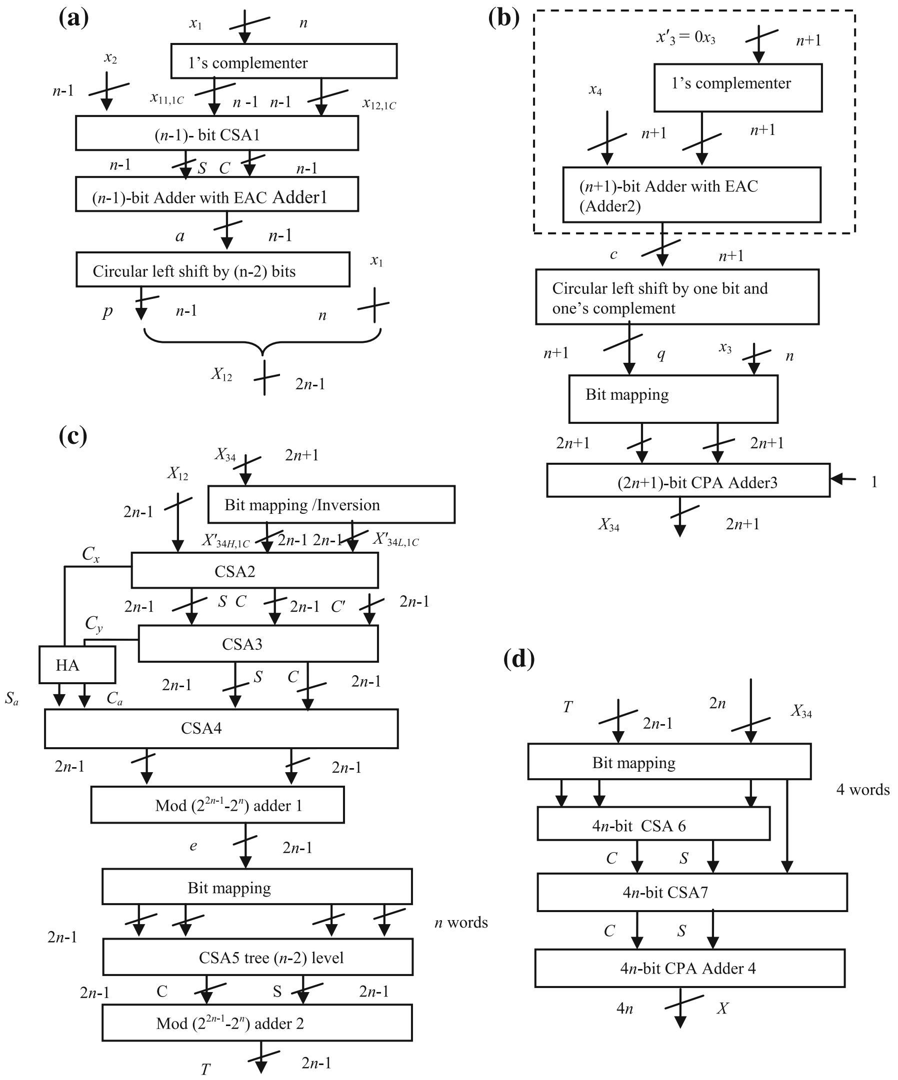

Fig. 2 **a** Block computing X_{12}, **b** block computing X_{34}, **c** block for computing T and **d** block for obtaining final decoded word X

where $a = (x_2 - x_1)_{2^{n-1}-1}$ and $b = \left(\frac{1}{2^n}\right)_{2^{n-1}-1} = 2^{n-2}$. Since the word lengths of x_1 and x_2 are different, x_1 needs to be written as $x_1 = x_{12}2^{n-1} + x_{11}$ where $x_{11} = x_{1,n-2}x_{1,n-3}\ldots x_{1,1}x_{1,0}$ and $x_{12} = 00\ldots 0x_{1,n-1}$ are $(n-1)$-bit words. One's complements $x_{11,1C}$ and $x_{12,1C}$ of these two words x_{11} and x_{12} need to be added to x_2 mod $(2^{n-1} - 1)$ for computing a. A $(n - 1)$-bit carry-save adder (CSA) stage (CSA1) followed by a $(n - 1)$-bit carry-propagate adder (CPA) (Adder 1 in Fig. 2a) both with end-around carry (EAC) will be required. Since $(n - 2)$ bits of the word $x_{12,1C}$ are

"1," the hardware requirements of the CSA stage are 1 full-adder (FA) and $(n-2)$ EXNOR/OR pairs of gates. Thus, totally, this stage needs n full-adders and $(n-2)$ EXNOR/OR pairs. The computation time is $(2n-1)D_{FA}$ where D_{FA} is the delay of a full-adder since the EAC adder exhibits twice the delay of a $(n-1)$-bit adder. Next, the multiplication mod $(2^{n-1}-1)$ with the multiplicative inverse $b=2^{n-2}$ can be realized by circular left shift of a by $(n-2)$ bits. The $(n-1)$-bit result $p=(ab)_{2^{n-1}-1}$ can be concatenated with x_1 as n LSBs to yield a $(2n-1)$-bit word X_{12} following (3). The architecture for computing X_{12} is presented in Fig. 2a.

3.1.2 Computation of X_{34}

We next consider evaluation of X_{34}. Using MRC for the moduli set $\{m_3, m_4\}$, we have

$$X_{34} = x_3 + m_3\,(cd)_{m_4} = x_3 + (2^n-1)\,(cd)_{2^{n+1}-1}, \tag{4}$$

where $c=(x_4-x_3)_{2^{n+1}-1}$. It can be easily verified that the multiplicative inverse of m_3 with respect to m_4 is

$$d = \left(\frac{1}{m_3}\right)_{m_4} = \left(\frac{1}{2^n-1}\right)_{2^{n+1}-1} = (-2)_{2^{n+1}-1}\,. \tag{5}$$

The computation of $c=(x_4-x_3)_{2^{n+1}-1}$ needs a $(n+1)$-bit CPA (Adder 2 in Fig. 2b) with EAC adding x_4 and one's complement of x_3' where x_3' is obtained by prepending x_3 with one zero bit as most significant bit (MSB). The multiplication using the multiplicative inverse d mod $(2^{n+1}-1)$ for obtaining $q=(cd)_{2^{n+1}-1}$ can be realized by one-bit circular left shift of c and taking one's complement. Next computation of X_{34} from (4) can be carried out by adding the two $(2n+1)$-bit words $q2^n+x_3$ and two's complement of q:

$$\begin{array}{l} q_n q_{n-1}\ldots q_2 q_1 q_0 x_{3,n-1}\ldots x_{3,1} x_{3,0} \\ 11\ldots 11 q_n' q_{n-1}'\ldots q_1' q_0' \\ \hphantom{11\ldots 11 q_n' q_{n-1}'\ldots q_1' q_0' q_0 q_0 q_0 q_0 q_0 q_0 q_0}1 \end{array}$$

(Note that $q2^n+x_3$ is obtained by concatenating q as $(n+1)$ bit MSB word with x_3). Note also that the primes indicate inverted bits. (Since the result is always positive, the adder can be $(2n+1)$-bit; sign bit of the two's complement word need not be considered). The addition of these words needs a $(2n+2)$-bit CPA (Adder 3 in Fig. 2b) which can be simplified taking into account the fact that n bits in one of the words are 1. The hardware requirements are n full-adders and $(n+1)$ pairs of exclusive-NOR (XNOR) and OR gates. Thus, the total hardware requirements for computing X_{34} is $(2n+1)$ full-adders and $(n+1)$ EXNOR/OR pairs of gates. The computation time is $(4n+3)D_{FA}$.

3.1.3 Computation of X

The computation of the final result X uses mixed radix conversion corresponding to the residues (X_{12}, X_{34}) for the moduli set $\{M_{12}, M_{34}\}$:

$$X = X_{34} + M_{34}\,(ef)_{M_{12}}, \tag{6}$$

where $e = (X_{12} - X_{34})_{M_{12}}$ and

$$f = \left(\frac{1}{M_{34}}\right)_{M_{12}} = \frac{2^{2n-1} - 2^{n+1} + 3}{3} = 2^{n+1}\left(\frac{2^{n-2} - 1}{3}\right) + 1. \tag{7}$$

Note that $\left(\frac{2^{n-2}-1}{3}\right) = \frac{2^{n-2}-1}{2^2-1} = 2^{n-4} + 2^{n-6} + \cdots + 4 + 1$ for $n \geq 6$ and for $n = 4$, $\left(\frac{2^{n-2}-1}{3}\right) = 1$. Hence, $\left(\frac{2^{n-2}-1}{3}\right)$ has $(n-2)/2$ bits which are "1." Thus, f has $n/2$ bits which are "1" leading to $n/2$ partial products which are left-shifted versions of e. As an illustration, for $n = 4, 6, 8, 10$, we have $f = \left(\frac{1}{M_{34}}\right)_{M_{12}} = 33, 641, 10753, 174081$ having 2, 3, 4 and 5 bits which are "1." The proof for (7) is as follows:

We need to find f such that $(f \times M_{34}) = 1 \bmod M_{12}$ or

$$f \times (2^{2n+1} - 3 \times 2^n + 1) = 1 \bmod (2^{2n-1} - 2^n) \tag{8a}$$

which is same as

$$f \times (2^n + 1) = 1 \bmod (2^{2n-1} - 2^n) \tag{8b}$$

after mod $(2^{2n-1} - 2^n)$ reduction. Multiplying (8b) both sides by $(2^n - 4)$, we have

$$f \times (-2^n - 4) = (2^n - 4) \bmod (2^{2n-1} - 2^n). \tag{8c}$$

Adding (8b) and (8c), we have

$$f \times (-3) = (2^n - 3) \bmod (2^{2n-1} - 2^n) \tag{8d}$$

or

$$f = \left(\frac{3 - 2^n}{3}\right) \bmod (2^{2n-1} - 2^n) = \left(\frac{2^{2n-1} - 2^{n+1} + 3}{3}\right). \tag{8e}$$

We consider next computation of e. The word lengths of X_{12} and X_{34} are $2n - 1$ and $(2n+1)$ bits, respectively. Since mod M_{12} where $M_{12} = (2^{2n-1} - 2^n)$ operation is required, we consider the $(2n+1)$-bit word X_{34} as $X_{34H}2^{2n-1} + X_{34L}$, where X_{34L} is a $(2n-1)$-bit word $x_{34,2n-2}x_{34,2n-3}\ldots x_{34,0}$ and X_{34H} is a 2-bit word $x_{34,2n}x_{34,2n-1}$. We next note that $2^{2n-1} \bmod (2^{2n-1} - 2^n) = 2^n$. Thus, we have

$$(X_{34}) \bmod (2^{2n-1} - 2^n) = (X_{34H} 2^{2n-1} + X_{34L}) \bmod (2^{2n-1} - 2^n)$$
$$= (X_{34H} 2^n + X_{34L}) \bmod (2^{2n-1} - 2^n). \quad (9)$$

Next, the $(n+2)$-bit word $X_{34H}2^n$ is prepended with $(n-3)$ number of "0" bits to make it a $(2n-1)$ bit word X'_{34H}. Thus, e can be computed as

$$e = (X_{12} - X_{34})_{M_{12}}$$
$$= \left(X_{12} + X_{34L,1C} + X'_{34H,1C} + C'\right)_{M_{12}}, \quad (10)$$

where 1C indicates one's complement. Note that a correction term $C' = 2^{2n-1} - 3 \times 2^n + 2$ needs to be added to take care of one's complementing operation of X_{34L} and X_{34H} and mod $(2^{2n-1} - 2^n)$ reduction in (10):

$$X_{34L,1C} + X'_{34H,1C} + 2^{2n-1} - 3 \times 2^n + 2 = (2^{2n-1} - 1 - X_{34L})$$
$$+ (2^{2n-1} - 1 - 2^n X_{34H}) + 2^{2n-1} - 3 \times 2^n + 2$$
$$= (-X_{34}) \bmod M_{12}. \quad (11)$$

Note that $X_{12} + X_{34L,1C} + X'_{34H,1C} + C'$ in (10) can exceed M_{12} when X_{12} is maximum (= $2^{2n-1} - 2^n - 1$) and $X_{34} = 0$ and can be at most $(2^{2n-1} - 2^n - 1) + 2(2^{2n-1} - 1) + 2^{2n-1} - 3 \times 2^n + 2 = 2^{2n+1} - 2^{n+2} - 1$. The terms $X_{12}, X_{34L,1C}, X'_{34H,1C}$ can be added using a CSA stage (CSA2) needing 2 FA and $(2n-3)$ pairs of XNOR/AND gates since $(2n-3)$ bits are "1" in $X_{34H,1C}$ (Fig. 2c). Next, $C' = 2^{2n-1} - 3 \times 2^n + 2 = 2^n(2^{n-1} - 3) + 2$ can be added with the SUM and CARRY output vectors of this CSA2 in another level of CSA (CSA3) comprising nHA (where HA stands for half-adder) and $(n-1)$ XNOR/OR pairs since C' contains n bits which are zeros and $(n-1)$ bits which are "1." These two CSA levels generate two carry bits C_x and C_y which are added using a HA to obtain carry C_a and sum $S_a$bits of weights 2^{2n} and 2^{2n-1}. These can be reduced mod $(2^{2n-1} - 2^n)$ using the relation $(C_a 2^{2n} + S_a 2^{2n-1}) \bmod (2^{2n-1} - 2^n) = C_a 2^{n+1} + S_a 2^n$.

This implies adding C_a and S_a at the $(n+1)$th and nth bit positions in another level of CSA (CSA4) needing $(n-3)$HA and 2FA. This is followed by a modulo $(2^{2n-1} - 2^n)$ Adder 1 which needs two cascaded $(2n-1)$-bit CPAs. Thus, the CSAs in Fig. 2c need 4FA+$(2n-3)$HA+$(3n$-4)XNOR/OR pairs of gates and the two CPAs need $(4n$-2)FA. The total time needed for computing e is $(4n+1)D_{FA}$.

The computation of $T = (ef)_{M_{12}}$needed in (6) requires mod $(2^{2n-1} - 2^n)$ reduction in the $n/2$ partial products (PPs) T_i which are left-shifted versions of e.Some of these PPs $PP_i = e f_i 2^i$ where f_i is the ith bit of $f$which is "1," extend beyond $(2n-1)$ bits up to $(4n$-4) bits. Hence, denoting $PP_i = PP_{i2}\, 2^{3n-2} + PP_{i1}\, 2^{2n-1} + PP_{i0}$, we have

$$(PP_i) \bmod (2^{2n-1} - 2^n) = (PP_{i2}\, 2^{3n-2} + PP_{i1}\, 2^{2n-1} + PP_{i0}) \bmod (2^{2n-1} - 2^n)$$
$$= (PP_{i2}\, 2^n + PP_{i1}\, 2^n + PP_{i0}) \bmod (2^{2n-1} - 2^n) \quad (12)$$

since $(PP_{i1}\, 2^{2n-1}) \bmod (2^{2n-1} - 2^n) = PP_{i1}\, 2^n$ and $(PP_{i2}\, 2^{3n-2}) \bmod (2^{2n-1} - 2^n) = (PP_{i2}\, 2^n) \bmod (2^{2n-1} - 2^n)$. Note that for some partial products with length less than $3n$-2 bits, $PP_{i2} = 0$. Hence, the bits in the bit positions $(3n - 3)$ to $(2n - 1)$ shall be mapped as bit positions $2n$-2 to n bit positions as additional words. Note that for partial products having more than $(3n-2)$ bits, the bits in the bit positions $4n$-4 to $3n$-2 will also be mapped between $2n$-2 to n bit positions. Since the partial products are of different lengths, some of the repositioned bits can be accommodated in vacant bit positions. Thus, the number of $(2n-1)$-bit words to be added is only n.

An example will illustrate this computation. Consider multiplication of $e \times 10753$ mod 32512 for the moduli set {256,127,255,511} where e is $e_{14}e_{13}e_{12}e_{11}e_{10}e_9e_8e_7e_6e_5$ $e_4e_3e_2e_1e_0$. (Note that $M_{12} = 32{,}512$, $M_{34} = 130, 305$ and $f = 10{,}753$ for $n = 8$). The modulo $(2^{2n-1} - 2^n)$ reduced partial products are as illustrated in Table 1. Note that only partial products corresponding to "1" in the multiplier 10,753 need to be considered. The partial product corresponding to the multiplier 2^9 can be seen to be folded two times in the second, third and fourth rows shown in bold in Table 1.

It can be seen that only bits beyond nth bit position need to be added using a carry-save adder tree. The carry bits can be placed in the vacant bit in the n-bit carry vector due to the mod $(2^{2n-1} - 2^n)$ operation. The computation of T needs $(n-2)(n-1)$ full-adders for the CSA tree (CSA5) and two $(2n-1)$ bit CPAs cascaded for realizing addition mod M_{12}(Mod $(2^{2n-1} - 2^n)$ Adder 2) thus needing $(4n$-2) full-adders.

The computation of (6) next can be carried out as follows:

$$X = X_{34} + M_{34}\,T = X_{34} + T\,(2^{2n+1} - 2^{n+1} - 2^n + 1). \tag{13}$$

Denoting T as the $(2n-1)$-bit word $t_{2n-2}t_{2n-3}\ldots t_1t_0$, the bit matrix corresponding to (13) is presented in Table 2. In order to take care of the two's complementing the two words $T \times 2^n$, $T \times 2^{n+1}$, a carry-in of 1 can be added to both the CSA stages. Note that the four $4n$-bit words need to be added using two-level CSA (CSA6 and CSA7) followed by a $4n$-bit CPA (Adder 4) to obtain the final decoded word X as shown in Fig. 2d. The CSA needs $(3n$-3) full-adders and $(2n + 2)$ EXNOR/OR pairs since many bits are "1," and the CPA needs $4n$ full-adders and the computation time is $(4n+2)D_{FA}$. The conversion time is the sum of computation times of X_{34}, T and X since the computation time is more for X_{34} than X_{12}.

3.2 Mixed Radix Conversion-Based Converter

We consider next RNS to binary converter using mixed radix conversion for the same moduli set. The MRC technique and its implementation architecture are presented in Fig. 3a, b. This needs three steps. In the first step, three operations are performed in parallel each needing one modulo subtraction followed by one modulo multiplication with a multiplicative inverse following (1). In the second step, two such operations are performed in parallel, and in third step one such operation is needed. The final stage uses the mixed radix digits d_1, d_2 and d_3 obtained in these three steps to compute the most significant $3n$ bits of X denoted as X_H as

$$X_H = d_1 + d_2(2^{n-1} - 1) + d_3(2^n - 1)(2^{n-1} - 1). \tag{14}$$

Table 1 Modulo $(2^{2n-1} - 2^n)$ reduced partial products for computing $(e \times 10753)$ mod (32512)

e_{14}	e_{13}	e_{12}	e_{11}	e_{10}	e_9	e_8	e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0
$\boldsymbol{e}_5$	$\boldsymbol{e}_4$	$\boldsymbol{e}_3$	$\boldsymbol{e}_2$	$\boldsymbol{e}_1$	$\boldsymbol{e}_0$	$\boldsymbol{e}_{13}$								
$\boldsymbol{e}_{12}$	$\boldsymbol{e}_{11}$	$\boldsymbol{e}_{10}$	$\boldsymbol{e}_9$	$\boldsymbol{e}_8$	$\boldsymbol{e}_7$	$\boldsymbol{e}_6$								
e_3	e_2	e_1	e_0	e_{13}	e_{12}	e_{11}								
e_{10}	e_9	e_8	e_7	e_6	e_5	e_4								
e_8	e_7	e_6	e_5	e_4	e_3	e_2								
e_1	e_{14}	e_{13}	e_{12}	e_{11}	e_{10}	e_9								
	e_0		e_{14}		$\boldsymbol{e}_{14}$									

Table 2 Bit matrix for computing X following (13)

t_{2n-2}	t_{2n-3}	$\dots$	t_{n-1}	t_{n-2}	t_{n-3}	$\dots$	$\dots$	t_1	t_0	$x_{34,2n}$	$x_{34,2n-1}$	$x_{34,2n-2}$	$\dots$	$x_{34,n+1}$	$x_{34,n}$	$x_{34,n-1}$	$\dots$	$x_{34,1}$	$x_{34,0}$
0	0	$\dots$	0	0	0	$\dots$	$\dots$	0	0	0	0	t_{2n-2}	$\dots$	t_{n+1}	t_n	t_{n-1}	$\dots$	t_1	t_0
1	1	$\dots$	1	1	$\overline{t_{2n-2}}$	$\dots$	$\dots$	$\overline{t_{n+2}}$	$\overline{t_{n+1}}$	$\overline{t_n}$	$\overline{t_{n-1}}$	$\overline{t_{n-2}}$	$\dots$	$\overline{t_1}$	$\overline{t_0}$	1	$\dots$	1	1
1	1	$\dots$	1	$\overline{t_{2n-2}}$	$\overline{t_{2n-3}}$	$\dots$		$\overline{t_{n+1}}$	$\overline{t_n}$	$\overline{t_{n-1}}$	$\overline{t_{n-2}}$	$\dots$	$\dots$	$\overline{t_0}$	1	1	$\dots$	1	1

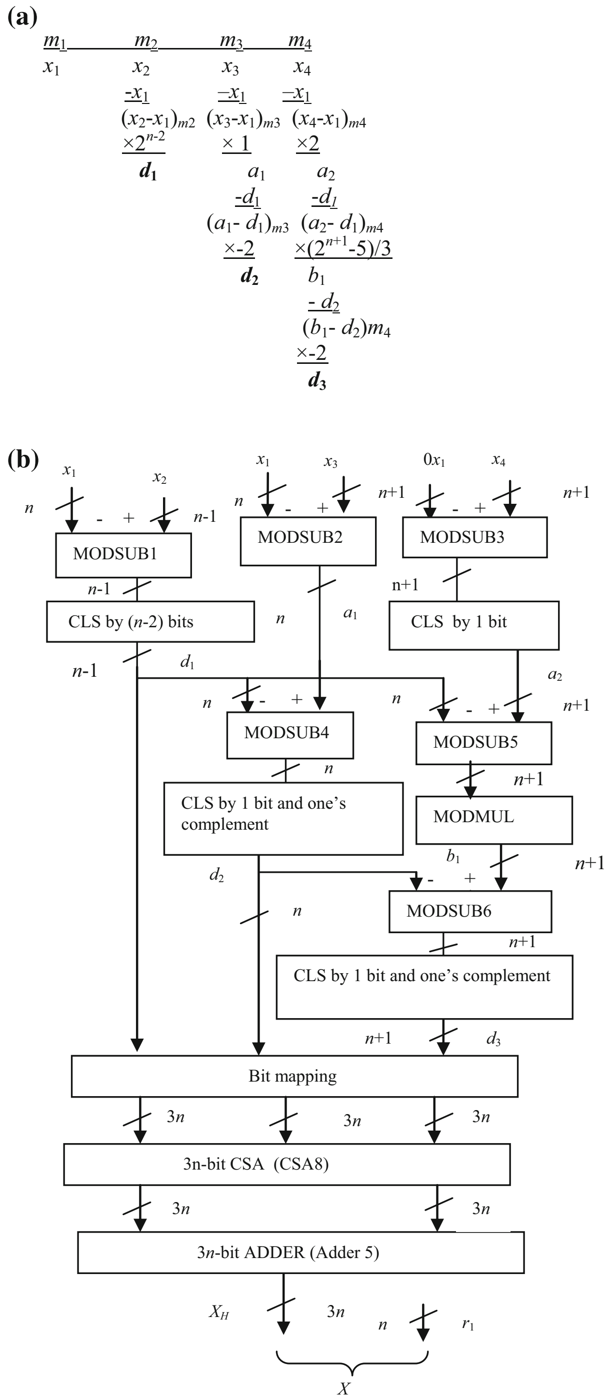

Fig. 3 Converter based on mixed radix conversion **a** algorithm and **b** architecture

The multiplicative inverses needed in the first step are

$$\left(\frac{1}{m_1}\right)_{m_2} = \left(\frac{1}{2^n}\right)_{2^{n-1}-1}$$
$$= 2^{n-2}, \left(\frac{1}{m_1}\right)_{m_3} = \left(\frac{1}{2^n}\right)_{2^n-1} = 1, \left(\frac{1}{m_1}\right)_{m_4} = \left(\frac{1}{2^n}\right)_{2^{n+1}-1} = 2. \quad (15a)$$

The multiplicative inverses needed in the second step are

$$\left(\frac{1}{m_2}\right)_{m_3} = \left(\frac{1}{2^{n-1}-1}\right)_{2^n-1} = -2, \left(\frac{1}{m_2}\right)_{m_4} = \left(\frac{1}{2^{n-1}-1}\right)_{2^{n+1}-1} = \frac{2^{n+1}-5}{3} = \gamma. \quad (15b)$$

The last step needs the multiplicative inverse

$$\left(\frac{1}{m_3}\right)_{m_4} = \left(\frac{1}{2^n-1}\right)_{2^{n+1}-1} = -2. \quad (15c)$$

Thus, all the multiplicative inverses are simple except $\gamma = \left(\frac{1}{m_2}\right)_{m_4} = \frac{2^{n+1}-5}{3}$. Note that γ can be derived as follows: We need to find γ such that

$$\gamma \times m_2 = 1 \bmod m_4$$

or

$$\gamma \times (2^{n-1}-1) = 1 \bmod (2^{n+1}-1) \quad (16a)$$

Multiplying both sides of (16a) with 4, we have

$$\gamma \times (2^{n+1}-4) = 4 \bmod (2^{n+1}-1) \quad (16b)$$

or

$$\gamma \times (-3) = 4 \bmod (2^{n+1}-1). \quad (16c)$$

Thus, we have

$$\gamma = \left(\frac{-4}{3}\right) \bmod (2^{n+1}-1) = \left(\frac{2^{n+1}-5}{3}\right) \bmod (2^{n+1}-1). \quad (16d)$$

Note that $\left((x_2 - x_1)_{2^{n-1}-1}\, 2^{n-2}\right)_{2^{n-1}-1}$ is realized by the MODSUB1 block followed by left circular shift of $(n-2)$ bits (Fig. 3b) to obtain d_1. This block is same as that shown in Fig. 2a. The computation of $(x_3 - x_1)_{2^n-1}$ needs addition of x_3 with one's complement of $x_1$in a n-bit CPA with EAC (MODSUB2 in Fig. 3b) to yield directly a_1 since multiplication by the multiplicative inverse 1 does need any additional computation. Next, $(x_4 - x_1)_{2^{n+1}-1}$needs addition of, one's complement of, x_1 prepended with one zero bit, with x_4 in a $(n + 1)$-bit CPA with EAC (MODSUB3). The multiplication of this result with the multiplicative inverse 2 mod $(2^{n+1} - 1)$ is realized by left circular shift (CLS) of the output of MODSUB3 by one bit to obtain a_2.

The computation of $(a_1 - d_1)_{2^n-1}$ can be carried out by adding a_1 with one's complement of d_1 prepended with a zero bit, in a n-bit CPA with EAC (MODSUB4). The multiplication with multiplicative inverse $-2\text{mod}(2^n - 1)$ is realized by one-bit left circular shift (CLS) followed by one's complementing (inversion of all bits) for obtaining d_2. Similarly, $(a_2 - d_1)_{2^{n+1}-1}$ needs addition of a_2 with one's complement of d_1 prepended with two zero bits, in a $(n + 1)$-bit CPA with EAC (MODSUB5). Note that the output of MODSUB5 block needs to be multiplied with γ (see (15b)) and reduced mod $(2^{n+1} - 1)$ in the block MODMUL.

The multiplicative inverse γ can be expressed as

$$\gamma = \left(\frac{1}{2^{n-1}-1}\right)_{2^{n+1}-1} = \frac{2^{n+1}-5}{3}$$
$$= \frac{2^{n+1}-5}{2^2-1} = 2^{n-1} + 2^{n-3} + \cdots + 2^3 + 1 \text{ for } n \quad \geq 4. \tag{17}$$

It can be seen for $n = 4, 6, 8, 10$ to be $9, 41, 169, 681$. The multiplicative inverse γ will have $n/2$ bits which are 1 thus needing only $n/2$ number of partial products. The multiplication with 2^m mod $(2^{n+1}-1)$ can be accomplished by left circular shift (CLS) of m bits. Thus, the number of partial products obtained are n/2 each having $(n + 1)$ bits. The addition of these with EAC needs a CSA tree of $(n/2)$-2 levels followed by a $(n + 1)$-bit CPA with EAC. The hardware and time requirements for multiplication of $(a_2 - d_1)_{m_4}$with γ in the MODMUL block in Fig. 3b are $(n + 1)\left(\frac{n}{2} - 1\right) FA$ and $(5n/2)\,\Delta_{FA}$.

The computation of $(b_1 - d_2)_{2^{n+1}-1}$ can be realized using the block shown in dotted lines in Fig. 2b (with b_1 in place of x_4 and d_2 in place of x_3 using MODSUB6). The multiplication with the multiplicative inverse -2 mod $(2^{n+1} - 1)$ is carried out by left circular shift (CLS) by one bit followed by inversion of all the bits (one's complementing) to obtain d_3.

Denoting d_1, d_2 and d_3 in (14) of bit lengths $(n-1)$, n and $(n+1)$ bits, respectively, as $d_{1(n-2)}d_{1(n-3)}\ldots d_{11}d_{10}$, $d_{2(n-1)}d_{2(n-2)}\ldots d_{21}d_{20}$ and $d_{3n}d_{3(n-1)}\ldots d_{31}d_{30}$, the various words that need to be added for the computation of X_H following (14) for n = 8 are presented in the bit matrix in Table 3. Note that all the 1 bits due to one's complementing and addition of three "1"s for two's complementing of three terms have been combined as a single word $h = 2^{3n} - 3 \times 2^{2n} + 2^{n-1} + 1$. The bit mapping block (Fig. 3b) obtains the various $3n$-bit words from d_1, d_2 and d_3 which are added in a CSA (CSA8) needing $(4n + 3)FA + (2n - 1)$ HA followed by a CPA (Adder 5)

Table 3 Bit matrix for computing (17)

d_{38}	d_{37}	…	…	d_{32}	d_{31}	d_{30}	d_{27}	d_{26}	d_{25}	d_{24}	d_{23}	d_{22}	d_{21}	d_{20}	d_{16}	d_{15}	d_{14}	d_{13}	d_{12}	d_{11}	d_{10}
h_{23}	h_{22}	…	…	h_{17}	h_{16}	$\overline{d_{38}}$	$\overline{d_{37}}$	$\overline{d_{36}}$	$\overline{d_{35}}$	$\overline{d_{34}}$	$\overline{d_{33}}$	$\overline{d_{32}}$	d_{38}	d_{37}	d_{36}	d_{35}	d_{34}	d_{33}	d_{32}	d_{31}	d_{30}
					$\overline{d_{38}}$	$\overline{d_{37}}$	$\overline{d_{36}}$	$\overline{d_{35}}$	$\overline{d_{34}}$	$\overline{d_{33}}$	$\overline{d_{32}}$	$\overline{d_{31}}$	$\overline{d_{30}}$	$\overline{d_{27}}$	$\overline{d_{26}}$	$\overline{d_{25}}$	$\overline{d_{24}}$	$\overline{d_{23}}$	$\overline{d_{22}}$	$\overline{d_{21}}$	$\overline{d_{20}}$
						h_{15}	h_{14}	h_{13}	t_{12}	h_{11}	h_{10}	h_9	$\overline{d_{31}}$	$\overline{d_{30}}$	h_6	h_5	h_4	h_3	h_2	h_1	h_0
													h_8	h_7							

needing $3n$FA. The computation time is $(3n+3)\Delta_{FA}$. Next, X_H is appended with x_1 as LSBs to yield the final decoded word.

4 Hardware and Conversion Time Evaluation of Proposed Converters

The hardware requirement and conversion time of the proposed converters for moduli set M1 (n even) using two-level MRC (Converter 1) and MRC (Converter 2) are presented as entries D1 and D2 in Table 4. The hardware requirements of the three converters described in [12] for Design I are presented as entries D3–D5 in Table 4 for comparison. Note that the design D5 uses a ROM, whereas the others use combinational logic. Three other converters described in [12] Design II need (n^2-15n)/2 less number of full-adders than those given in Table 4 in entries D3–D5 and their conversion time is also less by (2n-2+$p-q$) D_{FA}. The two-stage converter D6 of Taheri et al. [33] uses first-stage converter of three-moduli set M_b $\{2^n-1, 2^{n+1}-1, 2^{n-1}-1\}$ using MRC and uses a second stage using MRC of the composite moduli set $\{M_b, 2^n\}$. The design D7 for moduli set M2 [39] using modulus 2^k in place of 2^n in the moduli set M1 is based on MRC.

We also consider the five converters D8–D12 for the moduli set M3$\{2^n, 2^n-1, 2^n+1, 2^{n+1}-1)$. The two-stage designs D8 [4] and D9 [15] are based on MRC of the composite moduli set $\{M_c, 2^{n+1}-1\}$ where M_c is the popular three-moduli set $\{2^n-1, 2^n+1, 2^n\}$ in the second stage and use the three-moduli RNS to binary converter of Piestrak and Dhurkadas [10,23] in the first stage. The design D10 [35] has employed CRT. The designs D11 and D12 [6] also use MRC of the composite moduli set $\{M_c, 2^{n+1}-1\}$, but use the three-moduli RNS to binary converter Converter 1 due to Wang et al. [36]. The designs D13 and D14 [6] for the moduli set M4$\{2^n, 2^n-1, 2^n+1, 2^{n-1}-1\}$ [6] (n even) also use Wang et al. Converter 1 [36] in the front end followed by MRC of the composite moduli set $\{M_c, 2^{n-1}-1\}$. The converter D15 for moduli set M5 $\{2^n, 2^n-1, 2^n+1, 2^{n+1}+1\}$ (n odd) [4] uses MRC for the composite moduli set $\{M_c, 2^{n+1}+1\}$ and uses the reverse converter of Piestrak [23] and Dhurkadas [10] in the front end for three-moduli RNS to binary converter. The design D16 [29] for moduli set M5 uses a two-level converter similar to the architecture in Fig. 1 and use MRC in both the levels. The design D17 [5] for moduli set M5 is based on CRT. The converter D18 for the moduli set M6$\{2^n, 2^n-1, 2^n+1, 2^{n-1}+1\}$ (n odd) [19] uses MRC on the composite moduli set $\{M_c, 2^{n-1}+1\}$ and has used the three-moduli RNS to binary converter of Wang, Jullien and Miller [37] in the first stage. The converters D19 and D20 for moduli set M6 use two-level reverse converter following Fig. 1. The two-stage converter D21 for the moduli set M7 $\{2^n, 2^n-1, 2^n+1, 2^{n+1}-1, 2^{n-1}-1\}$ [8] uses MRC on the composite moduli set $\{$M3$, 2^{n-1}-1\}$ and uses the four-moduli reverse converter of [6] for the moduli set M3 in the first stage. The converters D22 [30] and D23 [30] for moduli sets M8 $\{2^n, 2^n-1, 2^n+1, 2^{2n+1}-1\}$ and M9 $\{2^{2n}, 2^n-1, 2^n+1, 2^{2n+1}-1\}$, respectively, also use same architecture as in Fig. 1 considering two pairs of moduli in first stage. The design D24 [7] for moduli set M10$\{2^n, 2^n-1, 2^n+1, 2^{2n}+1\}$ is based on New CRT-I [38]. The converter D25 [17] for moduli set M8$\{2^n, 2^n-1, 2^n+1, 2^{2n+1}-1\}$ is based on the architecture of Fig. 1, whereas the converter D26 [17] for moduli set

Table 4 Hardware and conversion time evaluation of various reverse converters

	Converter	Moduli set	Hardware requirement	Conversion time in D_{FA}
D1	Proposed Converter I MRC(2-2)	M1 n even $\{2^n, 2^{n-1}-1, 2^n-1, 2^{n+1}-1\}$	$(n^2+15n+1)A_{FA}+(2n-3)A_{HA}+(7n-5)A_{XOR/XNOR}$	$(17n+2)$
D2	Proposed Converter II MRC		$(12.5n+(n^2/2)+7)A_{FA}+(3n-4)A_{HA}$	$(11.5n+7)$
D3 [12]	Design 1 Converter I MRC (3-1)		$(n^2+16n+6)A_{FA}+(n+2)(A_{OR}+A_{XNOR})+(3n-5)(A_{AND}+A_{XOR})$	$12n+9+q^{**}$
D4 [12]	Design 1 Converter 2 MRC (3-1)		$(n^2+24n+24)A_{FA}+(2n+3)A_{HA}+2(A_{OR}+A_{XNOR})+(2n-5)(A_{AND}+A_{OR})+(2n+1)A_{3:1MUX}$	$8n+11+q^{**}$
D5 [12]	Design I Converter 3 MRC (3-1)		$(n^2+22n+22)A_{FA}+(2n+2)A_{HA}+(2n-5)(A_{AND}+A_{EXOR})+2(A_{XNOR}+A_{OR})+(2n+1)A_{2:1MUX}+10(2n+1)A_{ROM}$	$8n+11+q^{**}$
D6 [33]	MRC(3-1)		$(n^2+21n-11)A_{FA}+(3n+1)(A_{XNOR}+A_{OR})+(6n+9)A_{AND}+(3n+9)A_{XOR}+nA_{OR3}$	$8n+1$
D7 [39]	MRC	M2 n even $\{2^k, 2^{n-1}-1, 2^n-1, 2^{n+1}-1\}$	$(9.5n+23\times n^2/2+5+n\alpha)A_{FA}$‡	$11.5n+3+\alpha$

Table 4 continued

	Converter	Moduli set	Hardware requirement	Conversion time in D_{FA}
D8 [4]	Converter I MRC(3-1)	M3 n even $\{2^n, 2^n-1, 2^n+1, 2^{n+1}-1)$	$(9n+5+((n-4)(n+1)/2))A_{FA} + 2nA_{EXNOR} + 2nA_{OR}$	$[(23n+12)/2]$
D9 [15]	MRC (3-1)		$(10n+6+(n-4)(n+1)/2)A_{FA} + (6n+2)A_{XNOR} + (6n+2)A_{OR} + (n+3)A_{MUX2:1} + (2n+1)A_{MUX3:1}$	$((15n+22)/2)$
D10 [35]	CRT		$(37n+14)A_{FA}$	$(14n+8)$
D11 [6]	four-stage CE MRC(3-1)		$(n^2/2+7n/2+7n+4)A_{FA} + A_{HA} + 2A_{2:1MUX}$	$(11n+l_1+8)^*$
D12 [6]	three-stage CE MRC(3-1)		$(n^2+10n+3)A_{FA} + A_{HA} + 2A_{2:1MUX}$	$(9n+m_1+6)^*$
D13 [6]	four-stage CE MRC(3-1)	M4 n even $\{2^n, 2^n-1, 2^n+1, 2^{n-1}-1\}$	$(n^2/2+3n/2+7n-3)A_{FA} + A_{HA} + 2A_{2:1MUX} + (2n-8)A_{XNOR} + 6A_{AND} + (2n-8)A_{OR} + 6A_{XOR}$	$(11n+l_2-1)\dagger$
D14 [6]	three-stage CE MRC(3-1)		$(n^2+7n-2)A_{FA} + A_{HA} + 2A_{2:1MUX} + (2n-8)A_{XNOR} + 6A_{AND} + (2n-8)A_{OR} + 6A_{XOR}$	$(9n+m_2+1)\dagger$
D15 [4]	Converter II MRC	M5 n odd $\{2^n, 2^n-1, 2^n+1, 2^{n+1}+1\}$	$(n^2+12n+12)A_{FA} + 2nA_{EXNOR} + 2nA_{OR} + (4n+8)A_{2:1MUX}$	$(16n+22)$

Table 4 continued

	Converter	Moduli set	Hardware requirement	Conversion time in D_{FA}
D16 [29]	MRC (2-2)		$(2n^2 + 11n + 3)A_{FA}$	$11.5n + 2log_2 n + 2.5$
D17 [5]	CRT		$(58n + 23 + log_2(c + 1))A_{FA} + 36nA_{ROM}$	$(24n + 17 + log_2(c + 1))$#
D18 [19]	Version I Converter MRC(3-1)	M6 n odd $\{2^n, 2^n - 1, 2^n + 1, 2^{n-1} + 1\}$	$(13n + (n^2/2) - 6.5)A_{FA} + nA_{OR} + A_{2:1MUX}$	$10n + 5 + \lfloor \log_{1.5}((n-3)/2) \rfloor$
D19 [19]	Version II Converter MRC (2-2)		$(13n + (n^2/2))A_{FA} + 2A_{AND} + 3A_{XOR}$	$8n + 5 + \lfloor \log_{1.5}((n-5)/2) \rfloor$
D20 [19]	Version III Converter MRC (2-2)		$(2n^2 + 10n)A_{FA} + 2A_{AND} + 3A_{XOR}$	$8n + 2 + \lfloor \log_{1.5}(n-1) \rfloor$
D21 [8]	MRC (4-1)	M7 n even $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$	$((5n^2 + 43n + m)/6 + 16n - 1)A_{FA}$	$18n + l_2 + 7$††
D22 [30]	MRC (2-2)	M8 $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$	$13n + 2$	$8n + 1$
D23 [30]	MRC (2-2)	M9 $\{2^{2n}, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$	$16n + 1$	$8n + 2$
D24 [7]	New CRT-I	M10 $\{2^n, 2^n - 1, 2^n + 1, 2^{2n} + 1\}$	$(11n + 6)A_{FA} + (2n - 1)A_{XOR} + (2n - 1)A_{AND} + 4nA_{XNOR} + 4nA_{AND}$	$(8n + 3)$
D25 [17]	MRC (2-2)	M8 $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$	$(8n + 2)A_{FA} + (n - 1)A_{XOR} + (n - 1)A_{AND} + (4n + 1)A_{XNOR} + (4n + 1)A_{OR} + nA_{MUX}$	$(12n + 5)$

Table 4 continued

	Converter	Moduli set	Hardware requirement	Conversion time in D_{FA}
D26 [17]	New CRT-I	M11 $\{2^{2n}, 2^n-1, 2^n+1, 2^{2n}+1\}$	$(10n+6)A_{FA}+(4n-3)A_{XOR}+(4n-3)A_{AND}+(2n-3)A_{XNOR}+(2n-3)A_{OR}$	$(8n+3)$
D27 [26]	Three- level MRC,MRC,CRT	M12$n=2k, k=4,5,6,\ldots$	$(66n^2-87n-15)A_{FA}$	$46n-42+2log_2(2n-6)+2log_2(4n-12)+4log_2n+2log_2(4n-1)$

**q number of levels in CSA tree of $(n+1)$ inputs

*l_1 is the number of levels in a $((n/2)+1)$-input CSA and m_1 the number of levels in a $(n+2)$-input CSA

†Note thatl_2 is the number of levels in a $(n/2)$-input CSA and m_2 the number of levels in a n-input CSA

††$m=n$-4, 9n-12,and 5n-8 for n = 6k-2,6k and 6k+2, l_2 as defined above

#$\log_2$(c+1) indicates the overflow bits of the final adder before modulo reduction

‡ $\alpha = ceil\ (k/n)+ceil\ (k/(n-1))+ceil\ (k/(n+1))$ and ceil (x) indicates nearest integer greater than x

M11$\{2^{2n}, 2^n - 1, 2^n + 1, 2^{2n} + 1\}$ uses New CRT-I [38]. The converter D27 [26] for moduli set M12 $\{2^{n-5}-1, 2^{n-3}-1, 2^{n-3}+1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$ considers four pairs of moduli in the first level and two pairs of composite moduli in second level and one pair of composite moduli in the third level. It uses MRC for each two-moduli reverse converter in first and second levels and uses CRT in the third level. The hardware requirements in terms of gates as well as conversion time for all the 27 converters are presented in Table 4. In Table 4, we denote the converters following the architecture of Fig. 1 as MRC(2-2) and converters using converters for three-moduli set M_a or M_b or M_c in first stage and MRC in second stage as MRC (3-1) converter. Note that the three-moduli converters could have been designed by any technique—MRC, CRT or New CRT-I. The other converters presented in Table 4 have been described as using New CRT-I or MRC (4-1) or MRC or three-stage (MRC, MRC, CRT) types. The corresponding hardware requirements and conversion times using the unit-gate model [34] are presented in Table 5. We have considered the gates needed for full-adder, half-adder, 2:1 MUX, EXOR, AND, OR gates as 7, 4, 3, 2, 1 and 1 and the delays as 4, 2, 2,2,1,1 unit delays Δ_g, respectively. Note that we have not considered the equivalent gates of the ROM for the converters D5 and D17, but only number of ROM bits needed are given in brackets.

If we consider converters D8, D9, D11–D15, D18 using M_c, D9 needs lowest AT^2. Among the converters D24 and D26 using New CRT-I, converter D26 has lowest AT^2. For the five-moduli converter D21, we note that AT^2 is lower than D1 and D15 for all dynamic ranges, whereas AT^2 is lower than that of converters D3 for DR 16, 48 and 64 bits and lower than that of converter D10 and D16 for DR of 48 bits. Among the converters D7, D22–D26, converter D26 exhibits lowest AT^2 for DR 16, 32 and 64 bits and D23 has lowest for 48-bit DR.

All the moduli sets M1–M12 have different dynamic ranges (DR) varying from about $4n$-1 bits to ($8n$-15) bits for the chosen n(and k values). The moduli sets M5 and M6 are possible for only odd n, and the moduli sets M1-M4 and M7 are possible for even $n$only. The other moduli sets are possible for even or odd n values. The gate requirement and conversion times estimated following Table 5 for realizing the four dynamic ranges 16 bits, 32 bits, 48 bits and 64 bits are presented together with the needed n and k values in Table 6. The plots showing area in unit gates, conversion time, AT (area$\times$ Time) and A$\times$T^2(area $\times$ Time^2) for four dynamic ranges 16 bits, 32 bits, 48 bits and 64 bits are presented in Fig. 4a–d for appropriate choice of n (and k wherever applicable).

The converter designs D1, D16, D19, D20, D22, D23 and D25 use MRC (2-2) two-stage reverse converters following Fig. 1. The converter designs D2–D9, D11–D15, D18–D20 are also two-stage converters, but in the first stage they employ a three-moduli reverse converter for the moduli sets M_a or M_b or M_c. The two-stage reverse converter for five-moduli set M7 converter D21 uses MRC in second stage, and the converter for four-moduli set M3 is used in first stage. All these use mixed radix conversion in the second stage. The converter D2 uses conventional MRC needing three successive steps for finding mixed radix digits and finally computes the decoded word. The converters D24 and D26 uses New CRT-I, whereas converters D10 and D17 use CRT.

Table 5 Hardware and conversion time evaluation of various reverse converters in terms of unit gates

	Converter	Moduli set	Hardware requirement	Conversion time in D_{FA}
D1	Proposed Converter I MRC(2-2)	M1 n even	$7n^2 + 125n - 12$	$68n + 8$
D2	Proposed Converter II MRC	M1 n even	$3.5n^2 + 96.5n + 37$	$46n + 28$
D3	Design 1 Converter I MRC (3-1)	M1 n even	$7n^2 + 124n + 33$	$48n + 36 + 4q^{**}$
D4	Design 1 Converter 2 MRC (3-1)	M1 n even	$7n^2 + 192n + 162$	$32n + 44 + 4q^{**}$
D5	Design I Converter 3 MRC (3-1)	M1 n even	$7n^2 + 172n + 154 + 10(2n + 1)A_{ROM}$	$32n + 44 + 4q^{**}$
D6	MRC(3-1)	M1 n even	$7n^2$+170n-47	$30n$+4
D7	MRC	M2 n even	$66.5n + 3.5n^2 + 35 + 7n\alpha$	$46n + 12 + 4\alpha$
D8	Converter I MRC(3-1)	M3 n even	$58.5n + 3.5n^2 + 21$	$46n + 24$
D9	MRC (3-1)	M3 n even	$3.5n^2 + 92.5n + 49$	$30n + 44$
D10	CRT	M3 n even	$259n + 98$	$56n + 32$
D11	Four-stage CE MRC(3-1)	M3 n even	$3.5\ n^2 + 73.5n + 37$	$44n + 4l_1 + 32^*$
D12	Three-stage CE MRC(3-1)	M3 n even	$7n^2 + 70n + 30$	$36n + 4m_1 + 24^*$
D13	Four-stage CE MRC(3-1)	M4 n even	$3.5n^2 + 65.5n - 18$	$44n + 4l_2 - 4\dagger$
D14	Three-stage CE MRC(3-1)	M4 n even	$7n^2 + 55n - 11$	$36n + 4m_2 + 4\dagger$
D15	Converter II MRC	M5 n odd	$7n^2 + 102n + 108$	$64n + 88$
D16	MRC (2-2)	M5 n odd	$14n^2$+$77n + 21$	$46n + 8\log_2 n + 10$
D17	CRT	M5 n odd	$406n$+161+7$\log_2(c$+1)) +$36nA_{ROM}$	$96n$+68 + 4$\log_2(c$+1))τ_{FA}
D18	Version I Converter MRC(3-1)	M6 n odd	$92n + 3.5n^2 - 42.5$	$40n + 20 + 4 \times \lfloor \log_{1.5}((n-3)/2) \rfloor$
D19	Version II Converter MRC (2-2)	M6 n odd	$91n + 3.5n^2 + 8$	$32n + 20 + 4 \times \lfloor \log_{1.5}((n-5)/2) \rfloor$

Table 5 continued

	Converter	Moduli set	Hardware requirement	Conversion time in D_{FA}
D20	Version III Converter MRC (2-2)	M6 n odd	$14n^2 + 70n + 8$	$32n + 8 + 4\lfloor \log_{1.5}(n-1) \rfloor$
D21	MRC (4-1)	M7 n even	$(35/6)n^2 + 162n + (7/6)m - 7$	$72n + 4l + 28$††
D22	MRC (2-2)	M8	$91n+14$	$32n+4$
D23	MRC (2-2)	M9	$112n+7$	$32n+8$
D24	New CRT-I	M10	$95n+39$	$32n+12$
D25	MRC (2-2)	M8	$74n+14$	$48n+20$
D26	New CRT-I	M11	$88n+24$	$32n+12$
D27	Three- level MRC,MRC,CRT	M12 $n = 2k, k = 4, 5, 6, \ldots$	$462n^2 - 609n - 105$	$184n - 168 + 8\log_2(2n-6) + 8\log_2(4n-12) + 16\log_2 n + 8\log_2(4n-1)$

See foot note of Table 4 for details about c, l, m, q and α

Table 6 Gate requirement and conversion time of various reverse converters for various DR

	Moduli set	DR 16 bit			DR 32bit			DR 48bit			DR 64bit		
		n	Area	Delay	*n*	Area	Delay	*n*	Area	Delay	*n*	Area	Delay
D1	M1	6	990	416	10	1938	688	14	3110	960	18	4506	1232
D2	M1	6	742	304	10	1362	488	14	2074	672	18	2908	856
D3	M1	6	1029	344	10	1973	552	14	3141	760	18	4533	968
D4	M1	6	1562	256	10	2770	400	14	4202	544	18	5858	688
D5	M1	6	1438 (130)	256	10	2574 (210)	400	14	3884 (290)	544	18	5452 (370)	688
D6	M1	6	1225	184	10	2353	304	14	3705	424	18	5281	544
D7	M2	4 k=5	441	216	8 k=9	1071	400	12 k=13	1757	584	16 k=18	2667	768
D8	M3	4	311	208	8	713	392	12	1227	576	16	1853	760
D9	M3	4	475	164	8	1013	284	12	1663	404	16	2425	524
D10	M3	4	1134	256	8	2170	480	12	3206	704	16	4242	928
D11	M3	4	387	208	8	849	392	12	1423	576	16	2109	760
D12	M3	4	422	176	8	1038	336	12	1878	496	16	2942	656
D13	M4	6	501	288	10	987	452	14	1585	636	18	2295	820
D14	M4	6	571	244	10	1239	396	14	2131	556	18	3247	716
D15	M5	5	793	408	9	1593	664	13	2617	920	17	3865	1176
D16	M5	5	756	264	9	1848	456	13	3388	640	17	5376	832

Table 6 continued

	Moduli set	DR 16 bit			DR 32bit			DR 48bit			DR 64bit		
		n	Area	Delay	n	Area	Delay	n	Area	Delay	n	Area	Delay
D17	M5	5	2191 (180)	548	9	3815 (324)	932	13	5439 (468)	1316	17	7063 (612)	1700
D18	M6	5	505	224	9	1069	388	13	1745	552	17	2533	712
D19	M6	5	550	184	9	1110	312	13	1782	444	17	2566	580
D20	M6	5	708	176	9	1772	308	13	3284	444	17	5244	568
D21	M7	4	771	320	8	1743	616	10	2299	764	14	3551	1060
D22	M8	4	378	132	7	651	228	10	924	324	13	1197	420
D23	M9	3	343	104	6	679	200	8	903	264	11	1239	360
D24	M10	4	419	140	7	704	236	10	989	332	13	1274	428
D25	M8	4	310	212	7	532	356	10	754	500	13	976	644
D26	M11	3	288	108	6	552	204	9	816	300	11	992	364
D27	M12	–	–	–	–	–	–	9	31836	1672	11	49098	2040

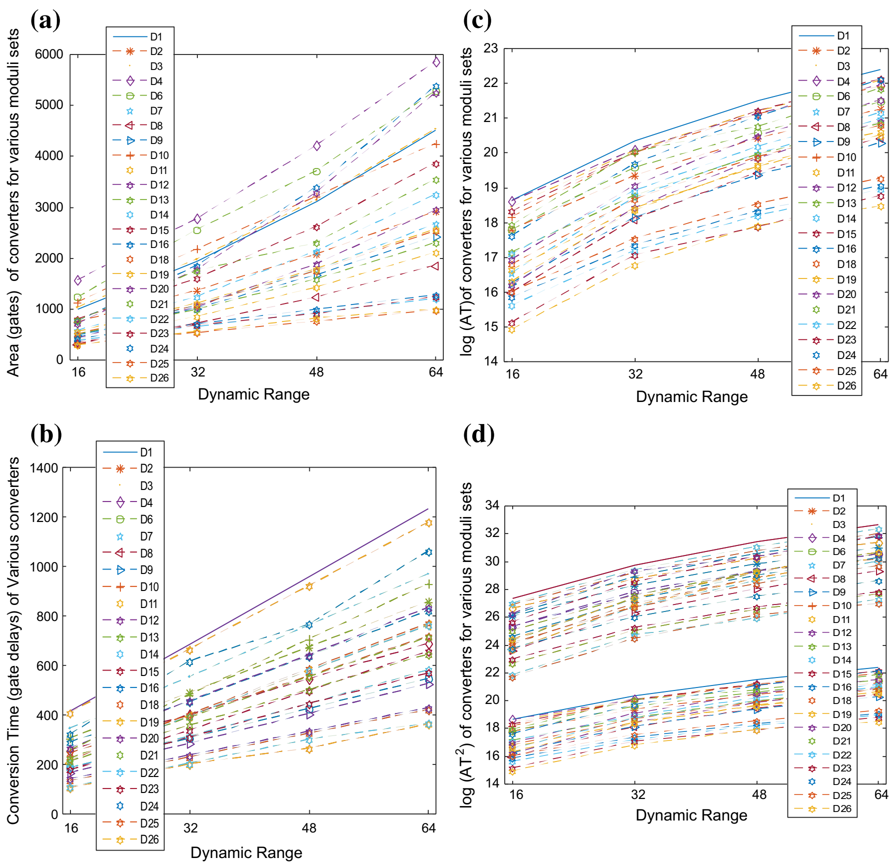

Fig. 4 **a** Area (unit gates) A and **b** conversion time T, **c** $A \times T$ and **d** $(A \times T^2)$ of various reverse converters for dynamic ranges 16, 32, 48 and 64 bits

The following conclusions can be arrived at from Table 6 for the four DRs considered. Among all the reverse converters considered, D26 exhibits lowest area for 16-bit DR and D25 needs lowest resources for other dynamic ranges. Regarding conversion time, D23 needs least among all designs. For 64-bit DR, D26 is comparable with D23 regarding conversion time.

Among all the converters for the moduli set M1, converter D2 needs lowest hardware resources, whereas D6 needs the least conversion time. The converter D1 needs low area than D3, D4, D5 and D6 for all dynamic ranges. The converter D2 needs lower conversion time than converter D3. The converter D7 using modulus 2^k needs lower area than all converters for the moduli set M1 and more conversion time than converter D6 for all DRs and more conversion time than D4 and D5 for DRs of 48 bits and 64 bits.

Among the MRC (2-2) converters following Fig. 1 (D1, D16, D19, D20, D22, D23, D25), the design D25 needs lowest area, whereas D23 needs least conversion time. Among all the four-moduli reverse converters using converter for moduli set M_c in the

Table 7 FPGA implementation results of converters D1 and D2

Design	n values	Area (slices)	Delay (ns)
D1	n= 4	38	13.773
	n= 8	86	17.383
	n= 16	195	16.347
D2	n= 4	48	13.403
	n= 8	63	16.319
	n= 16	124	14.530

front end (D8, D9, D11–D15, D18), the converter D8 needs lowest area and converter D9 has least conversion time. Among the two converters using New CRT-I, converter D26 needs less area and conversion time than that of converter D24.

The five-moduli reverse converter D21 needs lower area than converters D1, D3–D6, D10, D17 for all dynamic ranges and less area than converters D16 and D20 for dynamic ranges 32, 48 and 64 bits. It also needs less area than converters D15 for DR of 16, 48 and 64 bits. The converter D21 needs less conversion time than that of converters D1, D17 for all dynamic ranges and less than that of converter D3 for 16 bits, and converters D15 for DR of 16, 32 and 64 bits.

We also consider the moduli sets for comparison which have widely differing word lengths (n to $(2n + 1)$ bits) for the moduli. These are D7, D22–D26. Among these, converter D25 needs lowest area, whereas converter D23 has lowest conversion time.

We next consider comparison using Table 6 and plots in Fig. 4c regarding AT. AT is lowest for converter D26 considering all converters. Among all the converters D1–D6 for moduli set M1, D6 needs lowest AT for 16-, 32- and 48-bit DR and for 64-bit DR D2 needs lowest AT. Considering D7 also, the converter D7 needs lowest AT. Among MRC (2-2) converters D1, D16, D19, D20, D22, D23 and D25, the converter D23 has lowest AT. If we consider converters D8, D9, D11–D15, D18 using M_c, for DR 16 bits and 32 bits, D8 needs lowest AT, whereas for 48-bit and 64-bit DR, D9 needs lowest AT. Among the converters D24 and D26 using New CRT-I, converter D26 has lowest AT. For the five-moduli converter D21, we note that AT is lower than D1, D3 and D4 for all dynamic ranges, whereas AT is lower than that of converters D10 and D15 for DRs of 16, 48 and 64 bits and lower than that of converter D16 for DRs of 48 and 64 bits. Among the converters D7, D22–D26, converter D26 exhibits lowest AT for DRs of 16, 32 and 64 bits and D23 has lowest AT for 48-bit DR.

We next consider comparison of various converters regarding AT^2. AT is lowest for converter D26 considering all converters. Among all the converters D1–D6 for moduli set M1, D6 needs lowest AT^2 and considering D7 also, the converter D7 needs lowest AT^2 for DRs of 16, 32 and 48 bits. Among MRC (2-2) converters D1, D16, D19, D20, D22, D23 and D25, the converter D23 has lowest AT^2.

4.1 FPGA Implementation Results

The converters D1 and D2 have been realized for n = 4, 8 and 16 on Xilinx Device Virtex6—xc6vhx380t, Package: ff1923of Speed Grade 3 using Verilog HDL. These roughly correspond to dynamic ranges of 16, 32 and 64 bits, respectively. The post-placement and routing results of area requirement in slices and the conversion time are presented in Table 7. It can be seen that the converter D2 needs less area than converter D1 for n = 8 and n = 16, whereas the conversion times comprise of only combinational logic path delay.

5 Conclusion

In this paper, novel RNS to binary converters for the moduli set $\{2^n, 2^n - 1, 2^{n+1} - 1, 2^{n-1} - 1\}$ for n even using two-level MRC-based converter are presented. The proposed converters have been compared with converters proposed earlier for this moduli set as well as other moduli sets using four, five and eight moduli described in the literature for similar dynamic range. The proposed converters have been found to be having advantage ether in hardware requirement or conversion time over some of the previous reported converters. The advantage of this moduli set is the use of moduli which are convenient for fast arithmetic operations such as modulo addition, subtraction and multiplication and binary to RNS conversion.

The comparison among all the converters considered has shown that four-moduli RNS converters using vertical extension, i.e., one modulus 2^x with x larger than the word length of conjugate moduli $\{2^n - 1, 2^n + 1\}$ may lead to lower conversion time and/or lower hardware resource requirement (converters D22–D26). These can have the fourth modulus of word length n to $(2n + 1)$ bits. However, it must be noted that the arithmetic components for the moduli channels like modulo adders, modulo multipliers in channels of bigger word length using moduli $(2^{2n+1} - 1)$ or $(2^{2n} + 1)$ may limit the overall performance of RNS arithmetic operations, whereas operations in the 2^x moduli channel can be area efficient and time efficient. The computation in case of moduli of type $(2^\alpha - 1)$ and 2^x can be advantageous in situations like FIR filters where repeated MAC operations need to be carried out and in cryptography applications [13].

Appendix

In this Appendix, we consider the two different pairings of moduli in the two-level MRC. In the case of the pairing of Moduli M_{13}, M_{24}, the various multiplicative inverses are as follows:

In the first level we have $\left(\frac{1}{2^n}\right)_{2^n-1} = 1$ and $\left(\frac{1}{2^{n-1}-1}\right)_{2^{n+1}-1} = \frac{2^{n+1}-5}{3} = 2^{n-1} + 2^{n-3} + \cdots + 2^3 + 1$ and in the second level, we have $\left(\frac{1}{(2^{n-1}-1)(2^{n+1}-1)}\right)_{2^n(2^n-1)} = 2^{2n-1} - 7 \times 2^{n-1} + 1 = 2^{n-1}(2^n - 7) + 1$. This has been derived using extended Euclid algorithm. Thus, in the first level, multiplication with one of the multiplicative inverses takes more time than in the case of choice of M_{12},M_{34}. In the second level, the multiplicative inverse has $(n - 1)$ number of bits which are "1" and hence $(n - 1)$ partial products are needed to be added. The modulo M_{13} reduction can follow a similar method as described in the case of mod M_{12} reduction.

In the case of the pairing of Moduli M_{14}, M_{23}, the various multiplicative inverses are as follows:

In the first level, we have $\left(\frac{1}{2^n}\right)_{2^{n+1}-1} = 2$ and $\left(\frac{1}{2^{n-1}-1}\right)_{2^n-1} = -2$. In the second level, we have $\left(\frac{1}{(2^{n-1}-1)(2^n-1)}\right)_{2^n(2^{n+1}-1)} = \frac{2^{2n}+19\times 2^{n-1}+3}{3} = 2^{n-1}(2^{n-1}+2^{n-3}+\cdots+2^4+1)+1$. This has been obtained using extended Euclid algorithm.

The multiplicative inverses in the first level are simple, whereas that in the second level has $(n/2)+1$ bits which are "1," thus leading to $(n/2)+1$ partial products which need to be reduced mod M_{14} following a similar method as described in the case of mod M_{12} reduction.

References

1. P.V. Ananda Mohan, *Residue Number Systems: Algorithms and Architectures* (Kluwer Academic Publishers, New York, 2002)
2. P.V. Ananda Mohan, RNS to binary converter for the new three moduli set $2^{n+1}-1, 2^n, 2^n-1$. IEEE Trans. Circuits Syst. II Express Brief **54**(9), 775–779 (2007)
3. P.V. Ananda Mohan, New reverse converters for the moduli set $2^n-3, 2^n+1, 2^n-1, 2^n+3$. AEU **62**, 643–658 (2008)
4. P.V. Ananda Mohan, A.B. Premkumar, RNS to Binary converters for two four moduli sets $2^n-1, 2^n, 2^n+1, 2^{n+1}-1$ and $2^n-1, 2^n, 2^n+1, 2^{n+1}+1$. IEEE Trans. Circuits Syst. I **54**, 1245–1254 (2007)
5. M. Bhardwaj, T. Srikanthan , C.T. Clarke, A reverse converter for the 4 moduli super set $2^n-1, 2^n, 2^n+1, 2^{n+1}+1$, in *IEEE Conference on Computer Arithmetic* (1999), pp. 168–175
6. B. Cao, T. Srikanthan, C.H. Chang, Efficient reverse converters for the four-moduli sets $2^n-1, 2^n, 2^n+1, 2^{n+1}-1$ and $2^n-1, 2^n, 2^n+1, 2^{n-1}-1$. IEE Proc. Comput. Digit. Tech. **152**, 687–696 (2005)
7. B. Cao, C.H. Chang, 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 Trans. Circuits Syst. I **50**, 1296–1303 (2003)
8. B. Cao, C.H. Chang, T. Srikanthan, A residue to binary converter for a new five-moduli set. IEEE Trans. Circuits Syst. I **54**, 1041–1049 (2007)
9. G. Chalivendra, V. Hanumaiah, S. Vrudhula, A new balanced 4-moduli set $2^k, 2^n-1, 2^n+1, 2^{n+1}-1$ and its reverse converter design for efficient reverse converter implementation, in *Proceedings of ACM GSVLSI*, (Lausanne, Switzerland, 2011), pp. 139–144
10. A. Dhurkadas, Comments on A high-speed realization of a residue to binary number system converter. IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process. **45**(3), 446–447 (1998)
11. L.S. Didier, P.Y. Rivaille, A generalization of a fast RNS conversion for a new 4-modulus base. IEEE Trans. Circuits Syst. II Express Br. **56**, 46–50 (2009)
12. M. Esmaeildoust, K. Navi, M. Taheri, A.S. Molahosseini, S. Khodambashi, Efficient RNS to binary converters for the new 4-moduli set $2^n, 2^{n+1}-1, 2^n-1, 2^{n-1}-1$. IEICE Electron. Express **9**(1), 1–7 (2012)
13. M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, K. Navi, Efficient RNS implementation of elliptic curve point multiplication over GF(p). IEEE Trans. VLSI Syst. **21**, 1545–1549 (2013)
14. A.A. Hiasat, VLSI implementation of new arithmetic residue to binary decoders. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **13**, 153–158 (2005)
15. M. Hosseinzadeh, A. Molahosseini, K. Navi, An improved reverse converter for the moduli set $2^n+1, 2^n-1, 2^n, 2^{n+1}-1$. IEICE Electron. Express **5**, 672–677 (2008)
16. G. Jaberipur, H. Ahmadifar, A ROM-less reverse converter for moduli set $2^q\pm 1, 2^q\pm 3$. IET Comput. Digit. Tech. **8**, 11–22 (2014)
17. A.S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, S. Timarchi, Efficient reverse converter designs for the new 4-moduli sets $2^n-1, 2^n, 2^n+1, 2^{2n+1}-1$ and $2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1$ based on new CRTs. IEEE Trans. Circuits Syst. I **57**, 823–835 (2010)

18. A. Omondi, B. Premkumar, *Residue Number Systems: Theory and Implementation* (Imperial College Press, London, 2007)
19. P. Patronik, S.J. Piestrak, Design of reverse converters for the new RNS moduli set $2^n + 1, 2^n, 2^n - 1, 2^{n-1} + 1$ (n odd). IEEE Trans. Circuits Syst. I **61**, 3436–3449 (2014)
20. P. Patronik, S.J. Piestrak, 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 **61**, 1687–1700 (2014)
21. H. Pettenghi, R. Chaves, L. Sousa, Method to design general RNS converters for extended moduli sets. IEEE Trans. Circuits Syst. II **60**, 877–881 (2013)
22. H. Pettenghi, R. Chaves, L. Sousa, RNS reverse converters for moduli sets with dynamic ranges up to ($8n$+1) bits. IEEE Trans. Circuits Syst. **60**, 1487–1500 (2013)
23. S.J. Piestrak, A high-speed realization of residue to binary system converter. IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process. **42**(10), 661–663 (1995)
24. A. Skavantzos, T. Stouraitis, Grouped-moduli residue number systems for fast signal processing, in *Proceedings of IEEE ISCAS* (1999), pp. 478–483
25. A. Skavantzos, M. Abdallah, Implementation issues of the two-level residue number system with pairs of conjugate moduli. IEEE Trans. Signal Process. **47**, 826–838 (1999)
26. A. Skavantzos, M. Abdallah, T. Stouraitis, D. Schinianakis, Design of a balanced 8-modulus RNS, in *Proceedings of IEEE ISCAS* (2009), pp. 61–64
27. M.H. Sheu, S.H. Lin, C. Chen, S.W. Yang, An efficient VLSI design for a residue to binary converter for general balance moduli ($2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3$). IEEE Trans. Circuits Syst. Express Br. **51**, 52–55 (2004)
28. M.A. Soderstrand, W.K. Jenkins, G.A. Jullien, F.J. Taylor, *Residue Number System Arithmetic: Modern Applications in Signal Processing* (IEEE Press, New York, 1986)
29. L. Sousa, S. Antao, R. Chaves, 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. VLSI Syst. **21**, 1945–1949 (2013)
30. L. Sousa, S. Antao, MRC based RNS reverse converters for the four-moduli sets $2^n + 1, 2^n - 1, 2^n, 2^{2n+1} - 1$ and $2^n + 1, 2^n - 1, 2^{2n}, 2^{2n+1} - 1$. IEEE Trans. Circuits Syst. II **59**, 244–248 (2012)
31. N. Stamenkovic, B. Jovanovic, Reverse converter design for the 4-moduli set $2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1$ based on the mixed-radix conversion. Facta Univ. (NIS) Ser. Electron. Energ. **24**, 91–105 (2011)
32. N.S. Szabo, R.I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology* (Mc-Graw Hill, New York, 1967)
33. M.R. Taheri, N. Shafiee, M. Esmaeildoust, Z. Amirjamshidi, R. Sabbaghi-nadooshan, K. Navi, A high speed residue-to-binary converter for balanced 4-moduli set. J. Comput. Secur. **2**, 43–54 (2015)
34. A. Tyagi, A reduced-area scheme for carry-select adders. IEEE Trans. Comput. **42**, 1163–1170 (1993)
35. A.P. Vinod, A.B. Premkumar, A residue to binary converter for the 4-moduli superset $2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1$. JCSC **10**, 85–99 (2000)
36. Y. Wang, X. Song, M. Aboulhamid, H. Shen, Adder-based residue to binary number converters for $2^{n+}1 - 1, 2^n, 2^n - 1$. IEEE Trans. Signal Process. **50**(7), 1772–1779 (2002)
37. Z. Wang, G.A. Jullien, W.C. Miller, An improved residue to binary converter. IEEE Trans. Circuits Syst. I **47**, 1437–1440 (2000)
38. Y. Wang, Residue to binary converters based on new Chinese remainder theorems. IEEE Trans. Circuits Syst. II **47**, 197–205 (2000)
39. M. Wesolowski, P. Patronik, K. Berezowski, J. Biernat, Design of a novel flexible 4-moduli RNS and reverse converter, in *Proceedings of Signals and Systems Conference (IS SC 2012), IET Irish, IET* (2012), pp 1–6
40. W. Zhang, P. Siy, An efficient design of residue to binary converter for the moduli set $2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3$ based on new CRT II. Elsevier J. Inf. Sci. **178**, 264–279 (2008)