Method for designing two levels RNS reverse converters for large dynamic ranges

Hector Pettenghia, Ricardo Chaves, Roberto de Matos, Leonel Sousa

A R T I C L E   I N F O
Article history:
Received 21 September 2015
Received in revised form 23 February 2016
Accepted 23 February 2016
Available online 2 March 2016

Keywords:
Residue Number Systems
RNS-to-Binary converters
Adder-based converters

A B S T R A C T
In the last years, research on Residue Number Systems (RNS) has targeted larger dynamic ranges in order to further explore their inherent parallelism. In this paper, we start from the traditional 3-moduli set, with an equivalent 3n-bit dynamic range, and propose horizontal and vertical extensions to scale the dynamic range and enhance the parallelism according to the requirements. Two different methods to design general reverse converters for extended moduli sets to the desired dynamic ranges are introduced. Previous converters require complex weight selection of the inputs or complex final conversion steps. In this work the weight selection of the multiplicative terms associated to the inputs is reduced to additions of 2^n-bit length and the final conversion step requires only one comparison. Experimental results suggest that the proposed approaches achieve significant area reductions, up to 61% lower area reductions, in comparison with the state-of-the-art for generic DR purposes. Despite having identical delay metrics as the existing generic state of the art, Area-Delay-Product efficiency metrics improvements up to 2.7 times can be achieved. The obtained results also validate the improved scalability of the proposed approaches, allowing for better results with the increase of n and the DR.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction
Residue arithmetics, based on Residue Number Systems (RNS), have been in use in digital processing systems for many years [1]. RNS is a carry-free arithmetic system with modular characteristics offering the potential for high-speed and parallel computation. Arithmetic operations, such as addition, subtraction, and multiplication, can be carried out more efficiently than in the conventional binary systems [1], independently and concurrently, in several residue channels. The adoption of RNS has provided significant efficiency improvements for different types of Digital Signal Processing (DSP) applications [1], while allowing for an easier scaling for applications with larger dynamic ranges requirements, such as in adaptive filtering and cryptography [2].

The choice of the moduli set is of key importance in order to obtain balanced moduli sets. Moduli sets with a large number of channels can improve the arithmetic computation at the cost of reverse conversion performance.

With efficient reverse converters, capable of supporting large moduli sets, it is possible to compensate this extra cost, especially when several arithmetic operations have to be performed, such as in cryptographic or signal processing systems. In these cases the use of multiple arithmetic moduli channels can lead to better performance metrics.

To support these moduli sets, reverse converters need to be devised. Consequently, reverse conversion structures are usually presented whenever a novel moduli set is proposed. To devise these conversion structures the Chinese Remainder Theorem (CRT) [1], the mixed-radix conversion (MRC) [3] and the New CRT-I [4] algorithms are considered. Each of the moduli sets presented below have an associated conversion structure.

As mentioned before, in applications such as in cryptography [5], very large operands are used for. However, the level of parallelism and the achievable Dynamic Range (DR) provided by the traditional three-moduli set, with a DR of around 3n-bit [6,7], are not enough.
In these cases, horizontal extensions can be used in order to add more moduli to the moduli set. This approach has been considered and proposed in the state-of-the-art, such as the four-moduli sets with a DR of about 4n-bit: \((2^n, 2^n + 1, 2^{n+1} + 1)\) and \((2^n, 2^n + 1, 2^{n+1} + 1)\) [8,9]. Horizontal extensions of five-moduli sets with a DR of about 5n-bit have also been proposed: \((2^n, 2^n + 1, 2^{n+1} + 1, 2^{n+2} + 1, 2^{n+3} + 1)\) [10]. Horizontal extensions of five-moduli sets with a DR of about 4n-bit have also been proposed: \((2^n, 2^n + 1, 2^{n+1} + 1, 2^{n+2} + 1, 2^{n+3} + 1)\) [11], \((2^n, 2^n + 1, 2^{n+1} + 1)\) [12], and \((2^n, 2^n + 1, 2^{n+1} + 1)\) [13] that is composed of co-prime moduli for odd and has been revisited by Hiasat in [14]. The moduli considered in [12] are co-prime numbers for \(n\) even, however, complex multiplicative inverses are required, resulting in expensive reverse conversion structures. In [15] the authors propose a full RNS using the 8 moduli set \((2^n-5, 2^n-3, 1, 2^n-3+1, 2^{n+2}+1, 2^{n+1}+1, 1, 1, 1, 1, 1)\). The proposed moduli set is not regular, presenting channels with \(n\) to \(n-5\) bits with non-co-prime moduli, resulting in a lower DR. As in [12], complex multiplicative inverses are required, resulting in costly and complicated hierarchical reverse converter structures. In addition, vertical extensions of channels have also been proposed in order to increase the DR, such as \(2^n, 2^n + 1\) [7], where \(0 \leq \beta \leq n\) is used to increase the DR up to 4n-bits with a 3-moduli set. This is achieved towards a more balanced moduli set, since the performance difference between the 2\(n\) units and the 2\(n\) k arithmetic units. Therefore the overloading of the 2\(n\) channel up to 2\(n\) can be done without affecting the delay in the arithmetic channels.

Moduli sets with both vertically and horizontally extensions have also been recently proposed \((2^n, 2^n + 1, 2^{n+1} - 1)\) [16], \((2^n, 2^n + 1, 2^{n+1} + 1)\) [17], and \((2^n, 2^n + 1, 2^{n+2} + 1, 2^{n+1} - 1)\) and \((2^n, 2^n + 1, 2^n + 2^{n+1} + 1, 2^{n+1} - 1)\) with \(-\frac{2n-3}{2} \leq \beta \leq 3n\) [18]. The proposals [16,17] provide a DR of \(\approx 6\)-n-bits at a cost of unbalancing the moduli set. In contrast, the proposal [18] provides a more balanced moduli set with a maximum DR of \((8n+1)\)-bit.

In the paper [19] a method based on New CRT-I for designing RNS reverse converter that uses generic hybrid extended moduli sets of the form \((2^n, \beta, 2^n + 1, 2^n + k_1, 2^n + k_2, \ldots, 2^n + k_p)\) is presented, with \(k_i\) being odd values and \(0 \leq \beta \leq n\). However, this method imposes a complex modular weight selection of the multiplicative terms, \(V_{ dip }\), associated to the residue inputs \(R_i\), which is a substantial drawback. Moreover, the modular addition of these weighted inputs requires a large number of comparisons, and consequently a dedicated circuitry is used in the architecture to reduce the complexity of the Final Conversion step (FC). In this work a method is proposed to accommodate the generic moduli set horizontal and vertical extended presented in [19], by reducing the modular values used as the multiplicative terms, \(V_{ dip }\), and requiring only a single conversion in the final conversion operation.

The remaining of this paper is organized as follows. A novel method to design reverse converters to the extended moduli sets is presented in Section 2, and an additional technique that minimizes the number of required levels is presented in Section 3. A performance analysis of a case study is presented in Section 4. The efficiency of the state-of-the-art of reverse converters with large DRs is compared with the one achieved with our proposals, in Section 5. Section 6 concludes this paper with some final remarks.

2. Multi-level hybrid extensions of the three-moduli set \((2^n, 2^n + 1)\)

To simplify the presentation of the method and the description of the architectures, the following notation is adopted [14]: (i) the symbol \(\oplus\) operates the concatenation of the binary representation of two numbers, and (ii) \(R_i\) denotes the residue for \(m_i\).

The dynamic range is equal to the product of the \(N\) of moduli of a defined set \((M = \prod_{i=1}^{n} m_i, m_i = M/m_i, \text{ and } m_i^{-1})\) represents the multiplicative inverse of \(m_i\) with respect to modulus \(m_i\). A value represented in RNS can be converted back to binary \(X\) using the CRT [1]:

\[
X = \sum_{i=1}^{N} m_i \left[ \hat{m}_i^{-1} \right]_{m_i} R_i = \sum_{i=1}^{N} m_i \left[ \hat{m}_i^{-1} \right]_{m_i} R_i - MA(X),
\]

where \(A(X)\) is an integer that depends on the value of \(X\).

As stated above, herein both horizontal and vertical extensions are considered. For the vertical extension the power of two modulus is extended, towards a more balanced moduli set, since the power of two modulus typically allows for more efficient arithmetic operations than the remaining moduli sets for the same word length [18]. This leads to the moduli \((2^n, 2^n + 1, 2^n)\), with \(0 \leq \beta \leq n\), covering DRs up to \((4n)\)-bits [7].

In order to achieve arbitrarily wider moduli sets, horizontal moduli set extensions are herein considered by the addition of conjugate moduli pairs to the above moduli set in the same way as [19]. These conjugate moduli pairs are of the form \(2^n \pm k_j\), \(0 \leq j \leq f\), with \(k_j\) being an odd value in the range \(1 \leq k_j < 2^n - 1\) [20] chosen in such a way that all moduli are co-prime with each other. With this, moduli sets of the form \((2^n, 2^n, 2^n + k_1, 2^n + 2^n)\) are obtained, with a DR around \((1 + \frac{n}{2} + 2 \times (f + 1)) \times n\)-bit, for any integer \(n\).

The values of \(k_j\) can be chosen in order to obtain the highest possible DR, however a cost function can be used to obtain the most balanced moduli sets and minimizing the number of “1’s” in the representation of \(k_j\) in order to derive more efficient architectures, such as the ones presented in the following. It should be noted that the proposed method can also be used to derive reverse converters for moduli sets with non-conjugate moduli pairs. Herein, we only detail moduli sets with conjugate moduli pairs to simplify the explanation.

In order to illustrate the proposed moduli set extensions, we first derive the extension for the moduli set with \(f = 1\) and \(k_0 = 1\), resulting in the moduli set \((2^n, 2^n, 2^n + k_1, 2^n + 1)\). Following, the derivation and discussion of the limitations of extending the moduli set with conjugate moduli pairs for different \(f\) values is also presented.

2.1. Moduli set \((2^{n+1}, 2^n + k_1, 2^n + 1)\)

As presented above, let us consider the value \(0 \leq \beta \leq n\). From now on, the values of the moduli sets are ordered in a decreasing order, excluding the \(2^n+1\) and \(2^n-1\) (placed in the before-last and last positions), resulting in \(m_1 = 2^{n+1}, m_2 = 2^n + k_1, m_3 = 2^n - k_1, m_4 = 2^n + 1, m_5 = 2^n - 1\), whereas \(\hat{m}_1 = 2^n - 2^n(k_1^2 + 1)+k_1^2, \hat{m}_2 = 2^n + 2^n(2^n - k_1)(2^n - k_1) + k_1, \hat{m}_3 = 2^n + 2^n(2^n - 1)(2^n + k_1), m_4 = 2^n + 2^n(2^n - 1)(2^n - k_1), m_5 = 2^n + 2^n(2^n - 1)(2^n - k_1)).

For the proposed extension the following expression \(\hat{m}_i\) is used:

\[
\hat{m}_i = \frac{M}{\prod_{j=1}^{n} m_j} \text{ with } 1 \leq i \leq 5.
\]

(2)

The chosen \(k_1\) needs to satisfy that the resulting moduli set \((2^{n+1}, 2^n + k_1, 2^n + 1)\) is composed of co-prime numbers. For example, \(k_1 = 3\) satisfies this condition for \(n \geq 3\).

The values of the multiplicative inverses are integer numbers, which can be obtained by applying the condition \(\left[ \hat{m}_i^{-1} \right]_{m_i} = 1, 1 \leq i \leq 5\) [14].
It is important to notice that the multiplicative inverse \(\hat{m}_1^{-1}\) satisfies the following equation when \(0 \leq \beta \leq n:\)
\[
|\hat{m}_1^{-1}|_{m_1} = |\hat{m}_1^{-1}|_{m_1} = |(2^{2n} | m_1) | (k_1^2 + 1) + k_1^2 |_{m_1} + 1.
\]
Given that \(|(y,m_1+1)|_{m_1} = 1\), with \(y\) being a positive integer, the multiplicative inverse \(\hat{m}_1^{-1}\) can be expressed as:
\[
|\hat{m}_1^{-1}|_{m_1} = |(y,m_1+1)|_{m_1} = 1 \implies |\hat{m}_1^{-1}|_{m_1} = y(m_1+1)
\]
(4)

Therefore it is possible to reduce the modulo computation from \(M\) to \(\hat{m}_1\) in Eq. (1) as follows:
\[
X = \sum_{i=1}^{5} \frac{V_{i0}R_i}{m_1} + \sum_{i=2}^{5} \frac{V_{i0}R_i}{m_1} | A(X_i) |
\]
(5)
where Eq. (5) can be rewritten as:
\[
X = \sum_{i=1}^{5} \frac{V_{i0}R_i}{m_1} - MA(X) + R_1 = \sum_{i=1}^{5} \frac{V_{i0}R_i}{m_1} - \frac{M}{m_1} R_1 + R_1
\]
(6)
Due to \(X,m_1\) is a shift-left operation of \((n+\beta)\)-bits, the \(X_1\) can be concatenated to \(R_1\) to derive \(X\), where \(R_1\) becomes the less significant \((n+\beta)\) bits of \(X\).

In the second reduction to the modulo \(\hat{m}_2\) computation, the \(V_i\) values are split into two terms, which are the divisible and the indivisible terms of the division \(\frac{V_{i0}}{m_2}\). These divisible and indivisible terms are denoted as \(V_{i0}\) and \(V_{i1}\) respectively, as presented in Eq. (7) and their values can be obtained from the division equation of \(\sum_{i=1}^{5} V_{i0}\) by \(\hat{m}_2\), \(\sum_{i=1}^{5} V_{i1}\) by \(\hat{m}_2\). It is important to note that \(\hat{m}_j\) is divisible by \(\hat{m}_2\) for \(3 \leq j \leq 5\) and consequently \(V_{j0}\) = 0 in these cases:
\[
X_1 = \sum_{i=1}^{5} V_{i0}R_i | \hat{m}_1 \quad X_1 = \sum_{i=1}^{5} V_{i1}R_i | \hat{m}_1 + 2 \frac{V_{i1}R_i}{m_1} | \hat{m}_1
\]
(7)
\[
X_2 = \sum_{i=1}^{5} V_{i2}R_i | \hat{m}_2 \quad \sum_{i=1}^{5} V_{i2}R_i | \hat{m}_2 + 2 \frac{V_{i2}R_i}{m_2} | \hat{m}_2
\]
(8)
At the end:
\[
X = X_1 + R_1, \quad X_1 = \sum_{i=1}^{5} V_{i0}R_i + \sum_{i=1}^{3} \frac{V_{i1}R_i}{m_1} + \sum_{i=1}^{3} \frac{V_{i1}R_i}{m_1}
\]
(9)
It is important to note that \(\phi_{j0} < m_0\) and \(\phi_{j0} < m_2\), and consequently the multiplications of these constants by the corresponding \(R_i\) are non-modular operations. The constraints for \(\phi_{j0}R_i\) and \(\phi_{j0}R_i\) are:
\[
\max(\phi_{j0}R_i) = (m_0 - 1)m_1 - 1 \quad m_0m_1 - 1
\]
(10)
where the range \(0 \leq \beta \leq n\) guarantees Eq. (10).
In order to obtain a non-modular addition of the \(\phi_{j0}R_i\) and \(\phi_{j0}R_i\) terms, \(X = M - 1\) is set as input to provide the maximum residue.
The final conversion steps to derive $X_1$ and $X_2$ require only one comparison, since $\max(X_1 m_3 + \sum_{i=1}^{j-1} \phi_{ij} R_i) < 2 \times \hat{m}_2$ and $\max(X_2 m_3 + \sum_{i=1}^{j-1} \phi_{ij} R_i) < 2 \times \hat{m}_1$. The architecture of the resulting $(2^{2n} \cdot 2^n \pm k_1, 2^n \pm 1)$ converter by using this approach is depicted in Fig. 1(a).

### 2.2. Moduli Set $\{2^{n+\beta}, 2^n \pm k_f, \ldots, 2^n \pm k_2, 2^n \pm k_1, 2^n \pm 1\}$

For a general extension, to which correspond the moduli set $\{2^{n+\beta}, 2^n \pm k_f, \ldots, 2^n \pm k_2, 2^n \pm k_1, 2^n \pm 1\}$, the multiplicative inverses have to satisfy the condition $|\langle \hat{m}_i \rangle | \langle \hat{m}^{-1}_i \rangle | m_i = 1$, for $1 \leq i \leq (3+2 \times f)$. The number of iterative reductions and moduli in the sets are $t = 2 \times f + 1$ and $N = 2 \times f + 3$, respectively.

To obtain the binary value of $X$, Eq. (13) can be extended:

$$X = \sum_{i=1}^{N} V_i R_i$$

where:

$$X_0 = \sum_{i=1}^{m_0} V_i R_i$$

$$X_1 = \sum_{i=1}^{m_1} V_i R_i$$

$$X_2 = \sum_{i=1}^{m_2} V_i R_i$$

$$X_3 = \sum_{i=1}^{m_3} V_i R_i$$

The final conversion steps to derive $X_1$ and $X_2$ require only one comparison, since $\max(X_1 m_3 + \sum_{i=1}^{j-1} \phi_{ij} R_i) < 2 \times \hat{m}_2$ and $\max(X_2 m_3 + \sum_{i=1}^{j-1} \phi_{ij} R_i) < 2 \times \hat{m}_1$. The architecture of the resulting $(2^{2n} \cdot 2^n \pm k_1, 2^n \pm 1)$ converter by using this approach is depicted in Fig. 1(a).

![Fig. 1. Block diagram of the reverse converter $(2^{2n} \cdot 2^n \pm k_1, 2^n \pm 1)$ for (a) Multi-level approach, and (b) two-level approach.](image)
The parameters of Eq. (14) for the \((j+1)\)-reduction, \(1 \leq j \leq t-1\), can be derived as follows:

\[
X_{i+1} = \sum_{i=1}^{N} V_{j} R_{i} \bigg|_{m_{i}} \bigg|_{m_{i+1}} + \sum_{i=1}^{N} V_{j} R_{i} \bigg|_{m_{i}} \bigg|_{m_{i+1}} + \sum_{i=1}^{N} V_{j} R_{i} \bigg|_{m_{i}} \bigg|_{m_{i+1}} + \sum_{i=1}^{N} V_{j} R_{i} \bigg|_{m_{i}} \bigg|_{m_{i+1}},
\]

(15)
in the same way as in Eqs. (7) and (8).

In order to guarantee that to compute \(\sum_{i=1}^{j-1} \phi_{i+1} R_{i}\), a modular addition \(\sum_{i=1}^{j-1} \phi_{i+1} R_{i}\) is not required, the extension of Eq. (12) to the \((j+1)\)-reduction \(1 \leq j \leq t-1\) needs to be satisfied. In this case is also used \(X = M - 1\) as input to reach the maximum residue values in \(R_{i} = m_{i} - 1\), \(1 \leq i \leq N\), as presented in Eq. (11). Therefore:

\[
\sum_{i=1}^{j+1} \phi_{i+1} R_{i} = \sum_{i=1}^{j+1} \phi_{i} R_{i} (m_{i} - 1) < m_{j+1},
\]

(16)
which is satisfied for \(\beta = 0\) and mostly of cases for \(\beta = n\).

### 3. Two-levels hybrid extensions of the three-moduli set \(\mathbb{Z}_n^{2^n + 1}\)

It is possible to simplify the number of iterative reductions to two by using Lemma 1, [21], and Lemma 2:

**Lemma 1.**

\[
|A|_{p} \times k = k \times |A|_{k \times p},
\]

(17)

**Lemma 2.**

\[
|A|_{q} = |A - k \times q| = |A|_{q}; q = k \times p.
\]

(18)

For the moduli set \(\{2^n, 2^n + k_1, 2^n + 1\}\), it is possible to avoid one iterative reduction if Lemma 1 is applied to \(X_2 m_2\) in Eq. (13):

\[
X_2 m_2 = X_2 m_2 + \sum_{i=1}^{3} \phi_{3} R_{i} \bigg|_{m_2} = X_2 m_2 + \sum_{i=1}^{3} \phi_{3} m_2 R_{i} \bigg|_{m_2}.
\]

(19)

Therefore Eq. (13) can be reduced by applying Lemma 2 to:

\[
X = X_{m} m + R_{i};
\]

\[
X = X_{m} m + \sum_{i=1}^{3} \beta_{m} R_{i} \bigg|_{m_2} \bigg|_{m_2},
\]

(20)

with \(\sum_{i=1}^{3} \beta_{m} R_{i} = \sum_{i=1}^{3} \phi_{3} m_2 R_{i} + \sum_{i=1}^{3} \phi_{2} R_{i}\).

It is important to note that the terms \(\beta_{m} R_{i}\) are non-modular multiplications, since \(0 \leq \beta_{m} < m_2 m_3 - 1\):

\[
\begin{align*}
\text{max}(\beta_{m} R_{1}) &= ([m_2 - 1] m_2 + (m_2 - 1) m_1 - 1) m_1 \leq m_2 m_3 m_4 m_5; \\
\text{max}(\beta_{m} R_{2}) &= ([m_2 - 1] m_2 + (m_2 - 1) m_1 - 1) m_1 \leq m_2 m_3 m_5; \\
\text{max}(\beta_{m} R_{3}) &= ([m_2 - 1] m_2 + (m_2 - 1) m_1 - 1) m_1 \leq m_2 m_3 m_5,
\end{align*}
\]

(21)

where the range \(0 \leq \beta \leq n\) guarantees the conditions expressed in Eq. (21).

In order to obtain a non-modular addition \(\sum_{i=1}^{3} \beta_{m} R_{i}\) terms, \(X = M - 1\) is set as input to reach the maximum residue values in \(R_{i} = m_{i} - 1, 1 \leq i \leq 3\), as presented in Eq. (11), therefore:

\[
\beta_{m} (m_1 - 1) + \beta_{m} (m_2 - 1) + \beta_{m} (m_3 - 1) < m_2 m_3 m_4 m_5,
\]

(22)

which is always set for \(\beta = 0\). When the values of \(\beta_{m}\) for \(\beta = n\) are obtained and the condition of Eq. (22) is satisfied we can guarantee a non-modular addition \(\sum_{i=1}^{3} \beta_{m} R_{i}\) as presented in Eq. (20). If the moduli selected does not satisfy Eq. (22), then the solution consists of the reduction of \(\beta\) to \(\beta = 0\).

The architecture of a generic \(\{2^{2n}, 2^n + k_2, 2^n + 1\}\) by using this approximation is depicted in Fig. 1(b). The main difference in comparison with the proposal shown in Fig. 1(a) is the use of only one multiplier and one FC.

For the generic moduli set \(\{2^{2n}, 2^n + k_2, 2^n + 1\}\) with \(t = 2 \times f + 1\) and \(N = 2 \times f + 3\) we apply Lemma 1 2 times and consequently Eq. (20) can be expressed as:

\[
X = X_{m} m + R_{i};
\]

\[
X = X_{m} m + \sum_{i=1}^{N} V_{j} R_{i} \bigg|_{m_{i}} \bigg|_{m_{i+1}}.
\]

(23)

In this case:

\[
\begin{align*}
\sum_{i=1}^{t-1} \beta_{m} R_{i} &= \sum_{i=1}^{t-1} \phi_{2} R_{i} \bigg|_{m_{i}} + \sum_{i=1}^{t-2} \phi_{2} R_{i} \bigg|_{m_{i+1}} \bigg|_{m_{i+2}} \bigg|_{m_{i+3}} + \ldots + \sum_{i=1}^{t-3} \phi_{2} R_{i} \bigg|_{m_{i+3}} \bigg|_{m_{i+4}} \bigg|_{m_{i+3}} \bigg|_{m_{i+5}} + \ldots \bigg|_{m_{i+2}} \bigg|_{m_{i+3}} \bigg|_{m_{i+3}},
\end{align*}
\]

(24)

where the values of \(\phi_{2}\) can be derived from Eq. (15).

In order to guarantee the addition \(\sum_{i=1}^{3} \beta_{m} R_{i}\) can be performed by non-modular adders, the values of \(\beta_{m}\) obtained multiplied by the maximum residue values, \(R_{i} = m_{i} - 1, 1 \leq i \leq N\), need to satisfy:

\[
\sum_{i=1}^{t-1} \beta_{m} R_{i} = \sum_{i=1}^{t-1} \beta_{m} (m_{i} - 1) < m_{i+1},
\]

(25)

which is satisfied for \(\beta = 0\) and mostly of cases for \(\beta = n\).

### 4. Performance estimation: A case study

In order to better illustrate the proposed methodology, a particular case study for the moduli set \(\{2^{2n}, 2^n + 3, 2^n + 1\}\) with \(n = 4\), \(\{m_1, m_2, m_3, m_4, m_5\} = \{256, 19, 13, 17, 15\}\), is herein considered. The following illustrates the resulting parameters for each conversion approaches. Applying CRT [1]:

\[
X = \frac{V_{0}}{3 \times 590 \times 145} R_{1} \bigg|_{16 \times 124 \times 160} + \frac{V_{0}}{3 \times 394 \times 560} R_{2} \bigg|_{16 \times 124 \times 160}
\]

\[
+ \frac{V_{0}}{11 \times 162 \times 880} R_{3} \bigg|_{16 \times 124 \times 160} + \frac{V_{0}}{15 \times 175 \times 680} R_{4} \bigg|_{16 \times 124 \times 160}
\]

\[
+ \frac{V_{0}}{15 \times 409 \times 216} R_{5} \bigg|_{16 \times 124 \times 160}.
\]

(26)

Applying MRC [3]:

\[
X = \frac{V_{0}}{8(4) + 4(1) + V_{1} \bigg|_{15 - V_{2}} + V_{3} \bigg|_{15 - V_{4}}} \bigg|_{15 - V_{5}} + V_{15}
\]

\[
\times \frac{1}{704} \frac{1}{944} + \frac{4(9) \times 17 \times V_{1} \bigg|_{17 - V_{2}} + V_{3} \bigg|_{17 - V_{4}}} \bigg|_{17 - V_{5}} \times \frac{1}{63} \frac{1}{232} + \frac{11(3) \times V_{1} \bigg|_{13 - V_{2}} + V_{3} \bigg|_{13 - V_{4}}} \bigg|_{13} \times \frac{1}{4864}.
\]
Table 1 shows the operations comparison of reverse conversion algorithm approaches for \( (2^{2n}, 2^n \pm 3, 2^n \pm 1), n=4 \).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplications (</td>
<td>V_jR_i</td>
<td>_{\beta_i} )</td>
<td>#</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>#</td>
<td>16 124 160</td>
<td>62 985</td>
<td>255</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>Additions</td>
<td>#</td>
<td>Compressor</td>
<td>(bits)</td>
<td>5:1</td>
<td>6:1; (2 : 1) \times 10</td>
</tr>
<tr>
<td>#</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>#</td>
<td>(6n)</td>
<td>(4n) \times 3;</td>
<td></td>
<td>(2n); (3n); (4n)</td>
<td>(2n); (4n)</td>
</tr>
<tr>
<td>Multiplications</td>
<td>#</td>
<td>Mult. length</td>
<td>(n)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>#</td>
<td>4</td>
<td>( (n+1) \times 2n )</td>
<td></td>
<td>( (2n+1) \times n );</td>
<td>( (2n+1) \times 2n );</td>
</tr>
<tr>
<td>#</td>
<td>( n \times (3n+1) )</td>
<td>( 2n \times n );</td>
<td></td>
<td>( 2n \times (n+1) );</td>
<td>( 2n \times 2n );</td>
</tr>
<tr>
<td>#</td>
<td>( (n+1) \times 4n )</td>
<td>( 3n \times (n+1) ); ( 2n \times 2n );</td>
<td></td>
<td>( 2n \times n );</td>
<td>( 2n \times n );</td>
</tr>
<tr>
<td>Multiplications</td>
<td>#</td>
<td>Comparisons</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>#</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

where the constant values can be extracted from [23].

Applying the methodology described in [19] based on New CRT-I:

\[
X = \left[ \begin{array}{c} \frac{v_{12}}{14 024R_1} |_{62 985} + |_{13 260R_2} |_{62 985} \\
+ |_{43 605R_3} |_{62 985} + |_{59 280R_4} |_{62 985} \\
+ |_{58 786R_5} |_{62 985} \times 256 + R_1. \end{array} \right. \]

(27)

Applying the methodology described in Section 2:

\[
X = \left[ \begin{array}{c} \frac{v_{11}}{56 R_1} |_{255} + \frac{v_{12}}{53 R_2} |_{255} + \frac{v_{13}}{176 R_3} |_{255} \\
+ \frac{v_{14}}{240 R_4} |_{255} + \frac{v_{15}}{238 R_5} |_{255} \times 13 + 10 R_1 \\
+ \frac{v_{16}}{8 R_2 + 7 R_3} \times 19 + \frac{v_{17}}{2 R_1} |_{3315} \\
+ \frac{v_{18}}{7 R_2} \times 256 + R_1. \end{array} \right. \]

(28)

Applying the methodology described in Section 2:

\[
X = \left[ \begin{array}{c} \frac{v_{11}}{56 R_1} |_{255} + \frac{v_{12}}{53 R_2} |_{255} + \frac{v_{13}}{176 R_3} |_{255} \\
+ \frac{v_{14}}{240 R_4} |_{255} + \frac{v_{15}}{238 R_5} |_{255} \times 13 + 10 R_1 \\
+ \frac{v_{16}}{8 R_2 + 7 R_3} \times 19 + \frac{v_{17}}{2 R_1} |_{3315} \\
+ \frac{v_{18}}{7 R_2} \times 256 + R_1. \end{array} \right. \]

(29)

Applying the methodology described in Section 2:

\[
X = \left[ \begin{array}{c} \frac{v_{11}}{56 R_1} |_{255} + \frac{v_{12}}{53 R_2} |_{255} + \frac{v_{13}}{176 R_3} |_{255} \\
+ \frac{v_{14}}{240 R_4} |_{255} + \frac{v_{15}}{238 R_5} |_{255} \times 13 + 10 R_1 \\
+ \frac{v_{16}}{8 R_2 + 7 R_3} \times 19 + \frac{v_{17}}{2 R_1} |_{3315} \\
+ \frac{v_{18}}{7 R_2} \times 256 + R_1. \end{array} \right. \]

(30)

5. Experimental results

In order to assess the performance and cost of the generic reverse conversion approach herein proposed, the proposed and state-of-the-art reverse converters structures were described in a synthesizable VHDL and implemented on a 90 nm Standard Cell ASIC technology from UMC [24], using the Design Vision synthesis tool (version E-2010.12-SP4).

\[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]
To evaluate the scalability of the proposed reverse converters, experimental results for $DR = 6n$, $4 \leq n < 22$, were obtained for the best existing state of the art, namely for the moduli set $(2^{2n}, 2^n \pm 3, 2^n \pm 1)$, presented in [19], and for $(2^n, 2^n \pm 1, 2^n \pm 2^{2n} + 1)$ [17], $(2^n, 2^n \pm 1, 2^n \pm 2^{2n} + 1, 2^n \pm 1 + 1)$ and $(2^n, 2^n \pm 1, 2^n \pm 2^{2n} + 1, 2^n \pm 1 + 1)$ [18]. While additional dedicated reverse conversion structures for particular moduli sets, with $DR = 6n$, exist in the state of the art, these are not herein considered since they present worst results than the reverse conversion structures presented in [17, 18], as shown in [25].

Fig. 2 depicts the obtained results, for the resulting area, delay, Area-Delay-Product (ADP) and dynamic power. As expected, the dedicated conversion structures [17,18] present lower delays and area requirements. However, it is important to notice that they cannot be extended to larger moduli sets than $6n$ and $8n+1$ bits, respectively, which is the main goal of this work. Considering the traditional conversion approaches based on the CRT [1], the obtained area results demonstrate that they are not able to derive efficient ADP values. The MRC [3] solution shows a poor speedup providing high ADP values. When compared with the generic and scalable reverse approaches proposed in [19] ($\beta = 0$ and $\beta = n$), the herein proposed approaches suggest a significant area reduction, in the order of 54% when $\beta = n$. Delay-wise, the proposed multi-level solutions are clearly slower than the architectures presented in [19], being on average 79% slower. On the other hand, in the 2-level approach the delay increase is just of 16% regarding in [19]. This less significant delay increase is achieved given the further parallelism exploited by the 2-level approach. Despite this delay increase, the area reduction is more than enough to compensate the delay increase if the Area-Delay-Product (ADP) efficiency metric is considered. For the ADP, the multi-level approach allows for a 16% efficiency improvement, while the 2-level approach allows for an improvement of 41% with regard to [19] with $\beta = n$ and 71% in comparison with the CRT technique [1]. Between the two proposed approaches, the 2-level solution is clearly faster than the multi-level one, considering the analysed range. While the multi-level approach exhibits smaller area for some particular cases, it is not enough to compensate the delay degradation when ADP is considered. In terms of power, the proposals have similar consumption values. In comparison with the DR scalable architectures [19], for $\beta = 0$ and MRC the proposals have less power consumption.

Given the above results, the following analysis only considers (i) the architecture presented in [19], which has shown to be the only efficient implementation in the state-of-the-art for large DRs and $n$ values, and (ii) the two level approach herein proposed, given the overall better results when compared with the multi-level approach.

In order to better evaluate the scalability of the proposed conversion approaches, a moduli set with a $DR = 10n$-bits is considered. To the best of the authors knowledge, no dedicated conversion structure has been proposed for moduli sets above $8n$ bits.
The scalable architecture presented in [19] for $\beta = n$ has been chosen for this comparison because it has the best ADP results as presented in Fig. 2. Table 2 depicts the obtained conversion metrics for the $DR = 10n$ bits moduli set of the form $(2^{2n}, 2^n \pm k_3, 2^n \pm k_2, 2^n \pm k_1, 2^n \pm 1), 7 \leq n \leq 13$. The chosen for this comparison because it has the best ADP results chosen to de-multiplicative terms associated to the inputs. The 6. Conclusions

In this work a novel method for designing RNS reverse converters for arbitrarily long moduli sets is proposed. Two approaches are proposed that reduce the modular weight selection of the multiplicative terms associated to the inputs. The first approach is based on iterative stages used to reduce the complexity of the final converter step. The second approach minimizes the number of required iterative stages in the conversion to only two levels, with minimal area cost in comparison with the multi-level solution. Experimental results suggest that the proposed approaches allow for significant area reductions in comparison with the state-of-the-art, for generic DR reverse conversion structures. Given the similar delay metrics between the state of the art and the proposed 2-level approach, ADP improvements up to 2.7 times can be achieved, when considering a moduli set with a dynamic range of 10-bits. More importantly, the obtained results suggest that the proposed approaches, in particular the two level approach, are able to efficiently scale with larger moduli sets and $n$.

Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

References


[10] B. Cao, T. Srikanthan, C. Chang. Efficient reverse converters for four-moduli sets $(2^n-1, 2^n+1, 2^n+1, 2^n+1)-1$ and $(2^n-1, 2^n+1, 2^n-1, 2^n+1)-1$, IEEE Proc. Comput. Digit. Tech. 152 (September (5)) (2005) 687–696.


[16] L. Sousa, S. Antão, MBC-based RNS reverse converters for the four-moduli sets $(2^n-1, 2^n+1, 2^n+1)-1$ and $(2^n-1, 2^n+1, 2^n+1)-1$, IEEE Trans. Circuits Syst. II 59 (April (4)) (2012) 244–248.

[17] A. Molahosseini, K. Navi, C. Daddah, O. Kavehei, S. Timarchi, Efficient reverse converter designs for the new 4-moduli sets $(2^n-1, 2^n+1, 2^n+1)-1$ and $(2^n-1, 2^n+1, 2^n+1)-1$ based on new CRTs, IEEE Trans. Circuits Syst. I: Reg. Pap. 57 (April(4)) (2010) 823–835.


