Reverse Converters for RNSs with Diminished-one Encoded Channels

Evangelos Vassalos[#1], Dimitris Bakalis[#2], Haridimos T. Vergos[*3]

[#] *Electronics Laboratory, Department of Physics, University of Patras, Patras, Greece*

[1] vassalos@upatras.gr

[2] bakalis@physics.upatras.gr

[*] *Computer Engineering and Informatics Department, University of Patras, Patras, Greece*

[3] vergos@ceid.upatras.gr

Abstract— **The diminished-one encoding is often considered when representing the operands in the modulo 2^k+1 channels of a Residue Number System (RNS) since it can offer increased arithmetic processing speed. However, limited research is available on the design of residue-to-binary (reverse) converters for RNSs that use the diminished-one encoding in one or more channels. In this paper we introduce a simple methodology for designing such converters which can be applied to reverse converters based on the Chinese Remainder Theorem (CRT) or the New CRT-I method. Efficient converters for three moduli sets, covering different dynamic ranges, are also analytically presented. The proposed converters are shown to be area, delay and power efficient for several moduli sets.**

Keywords: Residue number system, diminished-one encoding, reverse converter, Chinese remainder theorem, New CRT-I

I. Introduction

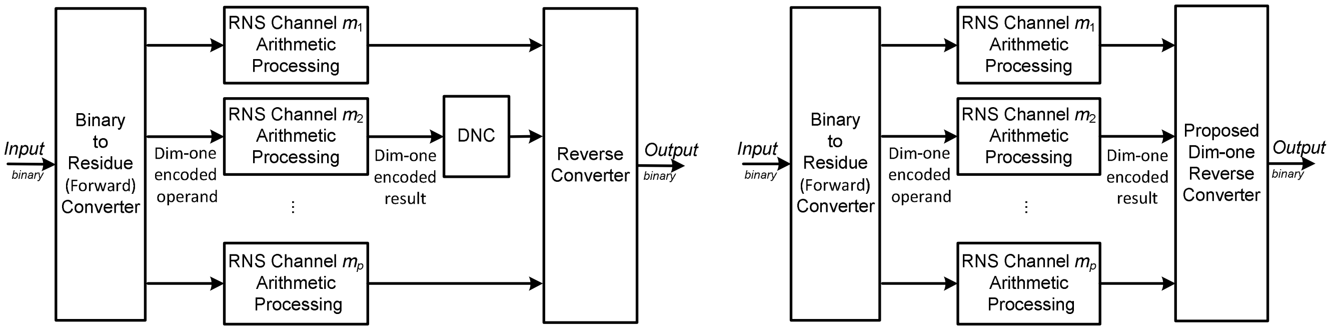
The Residue Number System (RNS) is commonly adopted for speeding up computations in digital signal processing, cryptography and telecommunication applications [18], [25]. An RNS is defined by a set of moduli $\{m_1, m_2, ..., m_p\}$ that are pair-wise relatively prime integers. An operand $X \in [0, M)$, with $M = m_1 \times m_2 \times ... \times m_p$, is uniquely represented in this RNS by the set $\{x_1, x_2, \ldots, x_p\}$ of residues, where $x_i = |X|_{m_i}$ is the residue of X when divided by m_i.

An RNS-based system consists of three main blocks. At first, all operands are converted to their corresponding sets of residues with binary-to-residue (forward) converters, according to the specified moduli set. Then, the arithmetic processing is performed in parallel in each channel following the corresponding modulo arithmetic. Finally, the RNS representation of the results is converted back to binary with residue-to-binary (reverse) converters.

RNSs with moduli of the 2^k, 2^k-1 and 2^k+1 forms have received significant attention since arithmetic circuits for the 2^k±1 moduli are almost as efficient as the binary ones while several efficient forward and reverse converters have been proposed for them. To this end, in the following we assume RNSs with moduli sets that consist of one modulus of the 2^k form while all the remaining moduli are of the 2^k±1 forms, which is the most common case. In such RNSs, a modulo 2^k+1 channel has to deal with operands one bit wider than the corresponding modulo 2^k-1 or 2^k channels, leading to a performance bottleneck. To avoid this, the diminished-one encoding was introduced in [16]. In this representation each modulo 2^k+1 operand is encoded by k+1 bits; k of them encode a value that is decreased by one compared to the value of the normal (weighted) encoding while an extra bit indicates a zero operand or result. Arithmetic units perform their calculations on the k bits of each operand, while zeros are treated in a special way. As a result, the architectures for diminished-one modulo 2^k+1 addition, multiplication and squaring that have been presented [2], [8-10], [14], [27], [30] are more delay and/or area efficient than those for the normal encoding.

Forward conversion for the diminished-one encoding is equally and in some cases even more efficient than the conversion for the normal encoding [7], [29]. On the other hand, the reverse converters that have been reported in the open literature, excluding the converter reported in [7] for the $\{2^n\text{-}1, 2^{n+k}, 2^n\text{+}1\}$ moduli set, assume a normal encoding for the residues of the 2^k+1 form. That is, it is silently assumed that a two-step approach is required for the reverse conversion in an RNS–based system that uses a diminished-one encoded channel: a diminished-to-normal converter (DNC) has to be used before the final reverse conversion as shown in Figure 1(a). The diminished-to-normal converter can be based on a controlled binary incrementer [10] but its logarithmic delay may cancel all the speedup achieved in the arithmetic processing. An initial attempt to deal with this problem was reported in [28] where modulo 2^k+1 adders, subtractors and multipliers with diminished-one encoded inputs and normally encoded outputs have been presented. However, a more elegant solution that is based on a simple and generic methodology is presented in this work.

In the following we present a methodology for designing efficient reverse converters, for any RNS that uses one or more channels of the 2^k+1 form with their operands encoded in diminished-one, by embedding the diminished-to-normal conversion within the reverse conversion (see Figure 1(b)). The proposed methodology can be applied to the design of reverse converters that are based on the Chinese Remainder Theorem (CRT) [18], [25] or the New CRT-I [31], that are the most frequently adopted theorems for that purpose. The Mixed Radix Conversion (MRC) [18], [25] is also


Input
binary
Binary to Residue (Forward) Converter
Dim-one encoded operand
RNS Channel m_1 Arithmetic Processing
RNS Channel m_2 Arithmetic Processing
RNS Channel m_p Arithmetic Processing
Dim-one encoded result
DNC
Reverse Converter
Output
binary
Input
binary
Binary to Residue (Forward) Converter
Dim-one encoded operand
RNS Channel m_1 Arithmetic Processing
RNS Channel m_2 Arithmetic Processing
RNS Channel m_p Arithmetic Processing
Dim-one encoded result
Proposed Dim-one Reverse Converter
Output
binary

(a) (b)

Fig. 1. (a) RNS system with normal reverse converter driven by diminished-to-normal converter(s) , (b) RNS system adopting the proposed diminished-one residue to binary converter

considered for certain design cases. The resulting converters are in all cases more efficient compared to the equivalent solution that utilizes normally-encoded reverse converters driven by diminished-to-normal converter(s), in terms of area, delay and power dissipation. In most examined cases, they are also equally or more efficient than the converters that assume normal encoding of the operands.

The rest of the paper is organized as follows. In Section II we formally derive the proposed converters and present analytical examples of them for three different moduli sets, while Section III presents evaluation and comparisons. Conclusions are drawn in the last section.

II. Proposed Methodology for Reverse Converters With Diminished-One Encoded Inputs

Consider an RNS with the p-moduli set $\{m_1, m_2, \ldots, m_p\}$ and a number $X \in [0, M)$, where $M = m_1 \times m_2 \times \ldots \times m_p$. X is uniquely represented in this RNS as $\{x_1, x_2, \ldots, x_p\}$, where $x_1 = |X|_{m_1}, x_2 = |X|_{m_2}, \ldots, x_p = |X|_{m_p}$.

Moduli sets usually contain a modulus with an even value which is a power of two. Without loss of generality we assume that this modulus is m_1. Assume that a modulus of the form 2^k+1 exists among the remaining moduli $\{m_2, \ldots, m_p\}$. Let this modulus be m_i and let its residue be x_i, $1 < i \leq p$. If x_i is normally encoded, then it consists of (k+1) bits. If it is encoded in diminished-one, then x_i is represented by (x_{iz}, x_{id}), where x_{iz} is the zero indication bit and x_{id} denotes the k-bit diminished-one number part. For this encoding it holds that

$$(x_{iz}, x_{id}) = \begin{cases} (1,0) & ,\text{when } x_i = 0 \\ (0, x_i - 1) & ,\text{when } x_i \neq 0 \end{cases} \tag{1}$$

or equivalently, x_i=x_{id} when x_{iz}=1 and x_i=x_{id}+1 when x_{iz}=0 [10].

An efficient reverse converter assuming a diminished-one encoding on x_i can be derived from the corresponding reverse converter that assumes that x_i is normally encoded. We show in the following and analytically prove that all that is required is the replacement in the latter converter of the residue x_i with, $x_i' + 1$ where $x_i' = 2^k x_{iz} + x_{id}$. Note that x_i' is also a (k+1)-bit vector while its increment by one can be achieved by considering a constant term, whose bits in most moduli set cases can be merged with the rest constants required. Although it is obvious that adding 1 is required when x_{iz}=0, the main difficulty stems from having to deal with zero operands as well, that is, when x_{iz}=1 the increment by 1 should be cancelled. In the proposed methodology we show that this can be efficiently achieved without severe performance degradation, by considering x_{iz} as a bit with a weight equal to 2^k along with the bits of x_{id}.

The proposed methodology can be adopted in any CRT-based or New CRT-I based reverse converter irrespectively of the number of the moduli that are of the 2^k+1 form.

A. CRT-based Converters

Given a pair-wise relatively prime moduli set $\{m_1, m_2, \ldots, m_p\}$, the CRT [18, 25] states that a number X can be derived from its residues $\{x_1, x_2, \ldots, x_p\}$ by the following equation:

$$X = \left| \sum_{j=1}^{p} \hat{m}_j \left| \hat{m}_j^{-1} \right|_{m_j} x_j \right|_M = \sum_{j=1}^{p} \hat{m}_j \left| \hat{m}_j^{-1} \right|_{m_j} x_j - A(X)M \qquad \text{or}$$

equivalently: $$X + A(X)M = \sum_{j=1}^{p} \hat{m}_j \left| \hat{m}_j^{-1} \right|_{m_j} x_j \tag{2}$$

where $\hat{m}_j = M / m_j$, $\left| \hat{m}_j^{-1} \hat{m}_j \right|_{m_j} = 1$, $1 \leq j \leq p$ and $A(X)$ is a non-negative integer which is a function of X (or equivalently x_is) .

Assuming that m_1 is equal to a power of two, it is common practice to divide both parts of Equation (2) with m_1 and take the result modulo $m_2 \ldots m_p$. We then have that:

$$\left| \lfloor X/m_1 \rfloor \right|_{m_2 \ldots m_p} = \left| \tilde{m}_1 \left| \hat{m}_1^{-1} \right|_{m_1} x_1 + \ldots + \tilde{m}_i \left| \hat{m}_i^{-1} \right|_{m_i} x_i + \ldots \right.$$
$$\left. + \tilde{m}_p \left| \hat{m}_p^{-1} \right|_{m_p} x_p \right|_{m_2 \ldots m_p}$$

where $\tilde{m}_j = \hat{m}_j / m_1$. X can then be computed by $X = x_1 + m_1 \left| \lfloor X/m_1 \rfloor \right|_{m_2 \ldots m_p}$, that is, by concatenating the bits that are derived by the modulo $m_2 \ldots m_p$ addition with the bits of x_1 [25].

Assume that among the set $\{m_2, \ldots, m_p\}$ there exists one modulus, m_i, $1 < i \leq p$, which is of the 2^k+1 form and its corresponding operand x_i is encoded in diminished-one. Then, according to the proposed methodology, x_i should be replaced by $x_i'+1$, where $x_i' = 2^k x_{iz} + x_{id}$. Thus, two steps are required: i) the (*k*+1)-bit vector x_i' should be used instead of the vector x_i and ii) since x_i is multiplied by $\tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i}$, a constant correction term equal to $CT = \left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} \right|_{m_2 \ldots m_p}$ should be also added in order to accommodate for the required increment of x_i' by 1. A proof of the validity of the proposed methodology follows.

Proof:

Suppose that x_i is replaced by x_i' and *CT* is also taken into account. We can then distinguish the following two cases:

a) if $x_i \neq 0$, then $x_{iz}=0$, $x_i' = x_{id}$ and

$$\left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} x_i' + CT \right|_{m_2 \ldots m_p} = \left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} (x_{id}+1) \right|_{m_2 \ldots m_p}$$
$$= \left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} x_i \right|_{m_2 \ldots m_p}$$

b) if $x_i = 0$ then $x_{iz}=1$, $x_i' = 2^k$ and

$$\left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} x_i' + CT \right|_{m_2 \ldots m_p}$$
$$= \left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} (2^k+1) \right|_{m_2 \ldots m_p} = \left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} m_i \right|_{m_2 \ldots m_p}$$
$$= \left| m_2 \ldots m_p \left|\hat{m}_i^{-1}\right|_{m_i} \right|_{m_2 \ldots m_p} = 0 = \left| \tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i} x_i \right|_{m_2 \ldots m_p}$$

Hence, in all cases the correct result is derived.

■

If more than one moduli of the 2^k+1 form exist in the same moduli set (eg. as in the RNS considered in [15]) and all or some of their corresponding residues are encoded in diminished-one, then the proposed methodology still holds. The correction term that is then required is equal to the sum taken modulo $m_2 \ldots m_p$ of the corresponding $\tilde{m}_i \left|\hat{m}_i^{-1}\right|_{m_i}$ terms of the moduli that adopt the diminished-one encoding.

Example 1: Consider the 4-moduli set $\{2^{p+k}, 2^p+1, 2^p-1, 2^{2p}+1\}$ which has a dynamic range approximately equal to (5*p*+*k*) bits and two channels of the 2^k+1 form. This moduli set was considered in [15]. In order to compute *X* from its residues $x_1 = |X|_{2^{p+k}}$, $x_2 = |X|_{2^p+1}$, $x_3 = |X|_{2^p-1}$ and $x_4 = |X|_{2^{2p}+1}$, where x_2 and x_4 are normally encoded, the CRT is utilized in [15] resulting in the relation given below:

$$X = x_1 + 2^{p+k} \left| Ax_1 + (B_1+B_2)x_2 + Cx_3 + (D_1+D_2)x_4 \right|_{2^{4p}-1}$$

with

$$A = -2^{3p-k}$$
$$B_1 = -(2^{4p-k-2} + 2^{2p-k-2})$$
$$B_2 = (2^{3p-k-2} + 2^{p-k-2})$$
$$C = (2^{4p-k-2} + 2^{3p-k-2} + 2^{2p-k-2} + 2^{p-k-2})$$
$$D_1 = 2^{3p-k-1}, \ D_2 = -2^{p-k-1}$$

Let $x_{1,p+k-1} \ldots x_{1,0}$, $x_{2,p} \ldots x_{2,0}$, $x_{3,p-1} \ldots x_{3,0}$ and $x_{4,2p} \ldots x_{4,0}$ denote the bits of the residues x_1, x_2, x_3 and x_4, respectively. Recall that for a single bit z it holds that $\left|z2^{k+t}\right|_{2^k-1} = \left|2^t z\right|_{2^k-1}, 0 \le t \le k-1$ and let $\bar{z}$ denote the complement of z. According to [15], the following six 4*p*-bit vectors can be derived:

$$V_1 = |Ax_1|_{2^{4p}-1} = \underbrace{\bar{x}_{1,p+k-1} \ldots \bar{x}_{1,k}}_{p} \underbrace{\bar{x}_{1,k-1} \ldots \bar{x}_{1,0}}_{k} \underbrace{1 \ldots 1}_{3p-k}$$

$$V_2 = |B_1 x_2|_{2^{4p}-1} = \underbrace{\bar{x}_{2,k+1} \ldots \bar{x}_{2,0}}_{k+2} \underbrace{1 \ldots 1}_{p-1} \underbrace{\bar{x}_{2,p} \ldots \bar{x}_{2,0}}_{p+1} \underbrace{1 \ldots 1}_{p-1} \underbrace{\bar{x}_{2,p} \ldots \bar{x}_{2,k+2}}_{p-k-1}$$

$$V_3 = |B_2 x_2|_{2^{4p}-1} = \underbrace{0 \ldots 0}_{k+1} \underbrace{x_{2,p} \ldots x_{2,0}}_{p+1} \underbrace{0 \ldots 0}_{p-1} \underbrace{x_{2,p} \ldots x_{2,1}}_{p} \underbrace{x_{2,0} \, 0 \ldots 0}_{p-k-1}$$

$$V_4 = |Cx_3|_{2^{4p}-1} = \underbrace{x_{3,k+1} \ldots x_{3,0}}_{k+2} \underbrace{x_{3,p-1} \ldots x_{3,0}}_{p} \underbrace{x_{3,p-1} \ldots x_{3,0}}_{p}$$
$$\underbrace{x_{3,p-1} \ldots x_{3,1}}_{p-1} \underbrace{x_{3,0} x_{3,p-1} \ldots x_{3,k+2}}_{p-k-1}$$

$$V_5 = |D_1 x_4|_{2^{4p}-1} = \underbrace{x_{4,p+k} \ldots x_{4,0}}_{p+k+1} \underbrace{0 \ldots 0}_{2p-1} \underbrace{x_{4,2p} \ldots x_{4,p+k+1}}_{p-k}$$

$$V_6 = |D_2 x_4|_{2^{4p}-1} = \underbrace{1 \ldots 1}_{p+k} \underbrace{\bar{x}_{4,2p} \ldots \bar{x}_{4,0}}_{2p+1} \underbrace{1 \ldots 1}_{p-k-1}$$

However, if the (3*p*–*k*) least significant bits of vectors V_1 and V_6 are swapped, then the vectors V_1' and V_6' are formed, where $V_1' = \underbrace{\bar{x}_{1,p+k-1} \ldots \bar{x}_{1,k}}_{p} \underbrace{\bar{x}_{1,k-1} \ldots \bar{x}_{1,0}}_{k} \underbrace{\bar{x}_{4,2p} \ldots \bar{x}_{4,0}}_{2p+1} \underbrace{1 \ldots 1}_{p-k-1}$ and $V_6' = \underbrace{1 \ldots 1}_{4p}$. Since it holds that $|V_6'|_{2^{4p}-1} = |1 \ldots 1|_{2^{4p}-1} = 0$, V_6' can be ignored and hence the reverse converter can be realized with one less Carry Save Adder (CSA) with End-Around-Carry (EAC) than reported to [15] and a final fast modulo 2^{4p}-1 adder.

If both x_2 and x_4 are encoded in diminished-one, then according to the proposed methodology, x_2 and x_4 are replaced by the vectors $x_2' = 2^p x_{2z} + x_{2d}$ and $x_4' = 2^{2p} x_{4z} + x_{4d}$, respectively, and the constant correction term, *CT*, that should be taken into consideration is:

$$CT = \left| (B_1+B_2) + (D_1+D_2) \right|_{2^{4p}-1}$$
$$= \left| (-2^{4p-k-2} + 2^{3p-k-2} - 2^{2p-k-2} + 2^{p-k-2}) \right.$$
$$\left. + (2^{3p-k-1} - 2^{p-k-1}) \right|_{2^{4p}-1}$$

We show in the following that the resulting diminished-one reverse converter does not introduce any area and delay overheads compared to the simplified normally-encoded one previously presented. Let $x_{2,p} \ldots x_{2,0}$ and $x_{4,2p} \ldots x_{4,0}$ also denote the bits of x_2' and x_4', respectively. By summing all constant bits of vectors V_1' and V_2 along with *CT* modulo 2^{4p}-1, we derive a single overall correction term denoted as CT_{ov}.

Specifically, the 1s from vectors V_1' and V_2 can form the values $T_{V_1'}$ and T_{V_2} respectively, with:

$$T_{V_1'} = 2^{p-k-1} - 1 \ , \ T_{V_2} = (2^{p-1}-1)2^{3p-k-1} + (2^{p-1}-1)2^{p-k-1} \ .$$

Hence CT_{ov} is equal to:

$$\begin{aligned} CT_{ov} &= \left| CT + T_{V_1'} + T_{V_2} \right|_{2^{4p}-1} \\ &= \left| 2^{3p-k-2} - 2^{p-k-1} + 2^{p-k-2} - 1 \right|_{2^{4p}-1} \\ &= \left| (2^{2p-1}-1)2^{p-k-1} + (2^{p-k-2}-1) \right|_{2^{4p}-1} \\ &= \underbrace{0...0}_{p+k+2}\ \underbrace{1...1}_{2p-1}\ 0\ \underbrace{1...1}_{p-k-2} \end{aligned}$$

Furthermore, vector V_2 is now written as:

$$V_2' = \underbrace{\overline{x}_{2,k+1}...\overline{x}_{2,0}}_{k+2}\underbrace{0...0}_{p-1}\underbrace{\overline{x}_{2,p}...\overline{x}_{2,0}}_{p+1}\underbrace{0...0}_{p-1}\underbrace{\overline{x}_{2,p}...\overline{x}_{2,k+2}}_{p-k-1} \ .$$

CT_{ov} can be merged with the non-constant bits of vector V_1', V_2' and V_5 forming the vectors:

$$V_1'' = \underbrace{\overline{x}_{1,p+k-1}...\overline{x}_{1,k}}_{p}\ \underbrace{\overline{x}_{1,k-1}...\overline{x}_{1,0}}_{k}\ \underbrace{\overline{x}_{4,2p}...\overline{x}_{4,0}}_{2p+1}\ 0\ \underbrace{1...1}_{p-k-2} \ ,$$

$$V_2'' = \underbrace{\overline{x}_{2,k+1}...\overline{x}_{2,0}}_{k+2}\underbrace{0...0}_{p-1}\underbrace{\overline{x}_{2,p}...\overline{x}_{2,0}}_{p+1}\underbrace{1...1}_{p-1}\underbrace{\overline{x}_{2,p}...\overline{x}_{2,k+2}}_{p-k-1} \quad \text{and}$$

$$V_5' = \underbrace{x_{4,p+k}...x_{4,0}}_{p+k+1}\,0\underbrace{1...1}_{p}\underbrace{0...0}_{p-2}\,\underbrace{x_{4,2p}...x_{4,p+k+1}}_{p-k} \ .$$

Thus, the final vectors to be added are V_1'', V_2'', V_3, V_4 and V_5'. Fig. 2 presents a block diagram of the proposed converter. Hence, the proposed converter with diminished-one encoded channels is as efficient as the one that follows normal encoding.

■

Example 2: Consider the well-known 3-moduli set $\{m_1=2^n, m_2=2^n-1, m_3=2^n+1\}$ which offers a dynamic range approximately equal to $3n$ bits. For computing X from its residues $x_1 = |X|_{2^n}$, $x_2 = |X|_{2^n-1}$ and $x_3 = |X|_{2^n+1}$, the CRT can be utilized resulting in the following Equation [1]:

$$X = x_1 + 2^n \left| -2^n x_1 + (2^{2n-1}+2^{n-1})x_2 + (-2^{2n-1}+2^{n-1})x_3 \right|_{2^{2n}-1} \quad (3)$$

Thus X can be reconstructed by concatenating the $2n$ bits derived by the modulo 2^{2n}-1 summation with the n bits of x_1.

A very efficient residue-to-binary converter for this moduli set, assuming that x_3 is normally encoded, is the one presented in [33]. It utilizes the CRT and Equation (3) and its implementation requires the modulo 2^{2n}-1 summation of only three vectors. If x_3 is encoded in diminished-one then, according to the proposed methodology, the $(n+1)$-bit vector $x_3' = 2^n x_{3z} + x_{3d}$ must be used instead of x_3 and an additional $2n$-bit vector ($CT = \left| -2^{2n-1} + 2^{n-1} \right|_{2^{2n}-1}$) is required in order to increase it by 1.

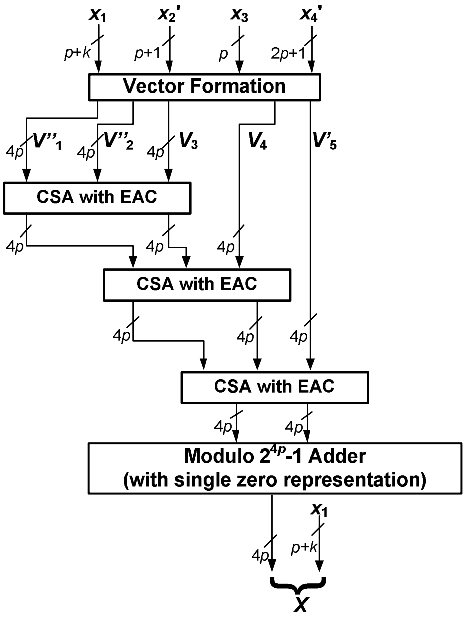


Fig. 2. Proposed reverse converter for the $\{2^p-1, 2^{p+k}, 2^p+1, 2^{2p}+1\}$ RNS

However, CT can be efficiently merged with the remaining vectors. Let $x_{3,n-1}\ldots x_{3,0}$, $x_{2,n-1}\ldots x_{2,0}$ and $x_{1,n-1}\ldots x_{1,0}$ denote the bits of the n-bit vectors x_{3d}, x_2 and x_1, respectively. Assume at first that $x_3 \neq 0$. Then $x_{3z}=0$, $x_{3d}=x_3-1$ and $(x_3'+1) = x_3$. We can easily derive the $2n$-bit vectors for the terms of Equation (3):

$$\left| -2^n x_1 \right|_{2^{2n}-1} = \left| (2^{2n}-1) - 2^n x_1 \right|_{2^{2n}-1} = \overline{x}_{1,n-1}\ldots\overline{x}_{1,0}1\ldots1 \quad (4)$$

$$\left| (2^{2n-1}+2^{n-1})x_2 \right|_{2^{2n}-1} = x_{2,0}x_{2,n-1}\ldots x_{2,0}x_{2,n-1}\ldots x_{2,1} \quad (5)$$

$$\begin{aligned} &\left| (-2^{2n-1}+2^{n-1})(x_3'+1) \right|_{2^{2n}-1} \\ &= \left| (-2^{2n-1}+2^{n-1})(x_{3d}+1) \right|_{2^{2n}-1} = \left| (-2^{2n-1}+2^{n-1})x_{3d} + CT \right|_{2^{2n}-1} \\ &= \left| (\overline{x}_{3,0}x_{3,n-1}\ldots x_{3,0}\overline{x}_{3,n-1}\ldots\overline{x}_{3,1}) + (0\underbrace{1\ldots1}_{n}\underbrace{0\ldots0}_{n-1}) + CT \right|_{2^{2n}-1} \quad (6) \\ &= \left| (\overline{x}_{3,0}x_{3,n-1}\ldots x_{3,0}\overline{x}_{3,n-1}\ldots\overline{x}_{3,1}) - CT + CT \right|_{2^{2n}-1} \\ &= \overline{x}_{3,0}x_{3,n-1}\ldots x_{3,0}\overline{x}_{3,n-1}\ldots\overline{x}_{3,1} \end{aligned}$$

Assume now that $x_3=0$ ($x_{3z}=1$, $x_{3d}=0$). Then the increment in $(x_3'+1)$ should be cancelled since it holds that $x_{3d}=x_3$ instead of $x_{3d}=x_3-1$. According to the proposed methodology, this can be achieved by using the value 2^n as x_3'. Equivalently we may choose to subtract 1 from x_3' and therefore add $-CT = (2^{2n-1} - 2^{n-1}) = 0\underbrace{1\ldots1}_{n}\underbrace{0\ldots0}_{n-1} = 0\underbrace{x_{3z}\ldots x_{3z}}_{n}\underbrace{0\ldots0}_{n-1}$ to the value of the $2n$-bits wide vector of Equation (6). Since x_{3z} and $x_{3,i}$, $0 \leq i < n$, cannot be simultaneously at 1, we can simply replace the vector of Equation (6) with:

$$\overline{x}_{3,0}\hat{x}_{3,n-1}\ldots\hat{x}_{3,0}\overline{x}_{3,n-1}\ldots\overline{x}_{3,1} \quad (7)$$

where $\hat{x}_{3,i} = x_{3,i} \vee x_{3z}$, $0 \le i < n$, and ∨ denotes a logic OR.

Hence, in all cases, the modulo 2^{2n}-1 summation of Equation (3) can be performed by adding the $2n$-bits vectors of Equations (4), (5) and (7), without any extra cost related to the introduction of the *CT*. Compared to the converter of [33], the proposed converter requires the same delay but has smaller area.

■

B. New CRT-I based Converters

The New CRT-I [31] was introduced as an alternative method that reduces the size of the modulo operation needed by the CRT. New CRT-I states that, given the number $\{x_1, x_2, \ldots, x_p\}$ in the p-moduli set $\{m_1, m_2, \ldots, m_p\}$ RNS, its equivalent binary number X can be derived as [32]:

$$X = x_1 + m_1 \left| C_1 x_1 + \ldots + C_i x_i + \ldots + C_p x_p \right|_{m_2 \ldots m_p} \quad (8)$$

where

$C_1 = -k_1$,

$C_2 = (k_1 - k_2 m_2)$,

$C_i = (k_{i-1} m_2 \ldots m_{i-1} - k_i m_2 \ldots m_i)$, $2<i<p$,

$C_p = k_{p-1} m_2 \ldots m_{p-1}$

and

$$\left| k_1 m_1 \right|_{m_2 \ldots m_p} = 1,\ \left| k_2 m_1 m_2 \right|_{m_3 \ldots m_p} = 1, \ldots,\ \left| k_{p-1} m_1 m_2 \ldots m_{p-1} \right|_{m_p} = 1.$$

The methodology that was proposed for CRT-based converters can also be applied in the case of converters that are based on the New CRT-I. Assume that the 2^n modulus of the RNS is m_1 and that among $\{m_2, \ldots, m_p\}$ exists a modulus of the 2^k+1 form. Suppose that this is m_i, with $1<i\le p$. If x_i is encoded in diminished-one, then, similarly to the CRT case, it should be replaced in Equation (8) by $x_i' + 1$, where $x_i' = 2^k x_{iz} + x_{id}$. Thus, i) the vector x_i' must be used instead of the vector x_i and ii) since x_i in Equation (8) is multiplied by C_i, a constant correction term equal to $CT = \left| C_i \right|_{m_2 \ldots m_p}$ must be also added. A proof of the validity of the proposed methodology follows.

Proof:

If x_i is replaced by x_i' and *CT* is also taken into account then:

Case 1: if $x_i \neq 0$, then $x_i' = x_{id}$ and

$$\left| C_i x_i' + CT \right|_{m_2 \ldots m_p} = \left| C_i (x_{id} + 1) \right|_{m_2 \ldots m_p} = \left| C_i x_i \right|_{m_2 \ldots m_p} \text{ and}$$

Case 2: if $x_i = 0$ then $x_i' = 2^k$ and

$$\left| C_i x_i' + CT \right|_{m_2 \ldots m_p} = \left| C_i (2^k + 1) \right|_{m_2 \ldots m_p} = \left| C_i m_i \right|_{m_2 \ldots m_p}$$

We thus have to prove that $\left| C_i m_i \right|_{m_2 \ldots m_p} = \left| C_i x_i \right|_{m_2 \ldots m_p} = 0$.

We consider all different cases for the value of i:

(i) $2<i<p$

From the definitions of the k_i terms it holds that:

$$\left| k_{i-1} m_1 \ldots m_{i-1} \right|_{m_i \ldots m_p} = 1 \Rightarrow k_{i-1}\, m_1 \ldots m_{i-1} = 1 + a(m_i \ldots m_p)$$

$$\left| k_i m_1 \ldots m_i \right|_{m_{i+1} \ldots m_p} = 1 \Rightarrow k_i\, m_1 \ldots m_i = 1 + b(m_{i+1} \ldots m_p)$$

where a, b are constant non-negative integers. Subtracting the second equation from the first we get:

$$k_{i-1}\, m_1 \ldots m_{i-1} - k_i\, m_1 \ldots m_i = a(m_i \ldots m_p) - b(m_{i+1} \ldots m_p)$$

$$\Rightarrow m_1 \ldots m_{i-1} (k_{i-1} - k_i m_i) = (a m_i - b) m_{i+1} \ldots m_p$$

and taking modulo $m_{i+1} \ldots m_p$ we have: $\left| m_1 \ldots m_{i-1} (k_{i-1} - k_i m_i) \right|_{m_{i+1} \ldots m_p} = 0$.

It is known that if $ab = \left| ac \right|_m$ and $gcd(a,m)$=1, where gcd(a,m) denotes the greatest common divisor of a and m, then $b = \left| c \right|_m$ [26, pg. 18]. Considering that: $\gcd(m_1 \ldots m_{i-1}, m_{i+1} \ldots m_p) = 1$, we get:

$$\left| m_1 \ldots m_{i-1} (k_{i-1} - k_i m_i) \right|_{m_{i+1} \ldots m_p} = 0$$

$$\Rightarrow \left| m_1 \ldots m_{i-1} (k_{i-1} - k_i m_i) \right|_{m_{i+1} \ldots m_p} = 0 \cdot (m_1 \ldots m_{i-1}) \quad (9)$$

$$\Rightarrow \left| k_{i-1} - k_i m_i \right|_{m_{i+1} \ldots m_p} = 0$$

Since $\left| ab \right|_{ac} = a \left| b \right|_c$ [25, pg. 30], we finally prove that:

$$\left| C_i m_i \right|_{m_2 \ldots m_p} = \left| k_{i-1} m_2 \ldots m_{i-1} m_i - k_i m_2 \ldots m_i m_i \right|_{m_2 \ldots m_p}$$

$$= m_2 \ldots m_i \left| k_{i-1} - k_i m_i \right|_{m_{i+1} \ldots m_p} = 0$$

(ii) i=2

According to Equation (9), $\left| k_1 - k_2 m_2 \right|_{m_3 \ldots m_p} = 0$. Hence:

$$m_2 \left| k_1 - k_2 m_2 \right|_{m_3 \ldots m_p} = 0 \Rightarrow \left| C_2 m_2 \right|_{m_2 \ldots m_p}$$

$$= \left| (k_1 - k_2 m_2) m_2 \right|_{m_2 \ldots m_p} = 0$$

(iii) i=p

If is obvious that

$$\left| C_p m_p \right|_{m_2 \ldots m_p} = \left| k_{p-1} m_2 \ldots m_{p-1} m_p \right|_{m_2 \ldots m_p} = 0.$$

We conclude that in all cases the correct result is derived.

■

If more than one moduli of the 2^k+1 form are present in the moduli set (e.g. the RNS sets considered in [3], [23]) and all or some of their corresponding residues are diminished-one encoded, then the required correction term is equal to the modulo $m_2 \ldots m_p$ sum of the corresponding C_i terms of the moduli that adopt the diminished-one representation.

Example 3: Consider the $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{2n}\text{+}1\}$ RNS which has a dynamic range approximately equal to $5n$ bits. X can be computed from its residues, $x_1 = \left| X \right|_{2^n - 1}$, $x_2 = \left| X \right|_{2^n}$, $x_3 = \left| X \right|_{2^n + 1}$ and $x_4 = \left| X \right|_{2^{2n} + 1}$, where x_3 and x_4 are normally encoded, according to the New CRT-I as follows [3]:

$$X = x_2 + 2^n \left| C_1 x_1 + C_2 x_2 + C_3 x_3 + C_4 x_4 \right|_{2^{4n} - 1}$$

where:

$C_1 = 2^{4n-2} + 2^{3n-2} + 2^{2n-2} + 2^{n-2}$

$C_2 = -2^{3n}$

$C_3 = -2^{4n-2} + 2^{3n-2} - 2^{2n-2} + 2^{n-2}$

$C_4 = 2^{3n-1} - 2^{n-1}$

Let $x_{1,n-1}\ldots x_{1,0}$, $x_{2,n-1}\ldots x_{2,0}$, $x_{3,n}\ldots x_{3,0}$ and $x_{4,2n}\ldots x_{4,0}$ denote the bits of the residues x_1, x_2, x_3 and x_4, respectively. According to [3], the following six $4n$-bit vectors are derived:

$$V_1 = x_{1,1}x_{1,0}\underbrace{x_{1,n-1}\ldots x_{1,0}}_{n}\underbrace{x_{1,n-1}\ldots x_{1,0}}_{n}\underbrace{x_{1,n-1}\ldots x_{1,0}}_{n}\underbrace{x_{1,n-1}\ldots x_{1,2}}_{n-2}$$

$$V_2 = \underbrace{\overline{x}_{2,n-1}\ldots\overline{x}_{2,0}}_{n}\underbrace{1\ldots1}_{3n}$$

$$V_3 = \overline{x}_{3,1}\overline{x}_{3,0}\underbrace{1\ldots1}_{n-1}\underbrace{\overline{x}_{3,n}\ldots\overline{x}_{3,0}}_{n+1}\underbrace{1\ldots1}_{n-1}\underbrace{\overline{x}_{3,n}\ldots\overline{x}_{3,2}}_{n-1}$$

$$V_4 = 0\underbrace{x_{3,n}\ldots x_{3,0}}_{n+1}\underbrace{0\ldots0}_{n-1}\underbrace{x_{3,n}\ldots x_{3,0}}_{n+1}\underbrace{0\ldots0}_{n-2}$$

$$V_5 = \underbrace{x_{4,n}\ldots x_{4,0}}_{n+1}\underbrace{0\ldots0}_{2n-1}\underbrace{x_{4,2n}\ldots x_{4,n+1}}_{n}$$

$$V_6 = \underbrace{1\ldots1}_{n}\underbrace{\overline{x}_{4,2n}\ldots\overline{x}_{4,0}}_{2n+1}\underbrace{1\ldots1}_{n-1}$$

However, the n most significant bits of vectors V_2 and V_6 can be swapped forming the vectors V_2' and V_6', where $V_2' = \underbrace{1\ldots1}_{4n}$ and $V_6' = \underbrace{\overline{x}_{2,n-1}\ldots\overline{x}_{2,0}}_{n}\underbrace{\overline{x}_{4,2n}\ldots\overline{x}_{4,0}}_{2n+1}\underbrace{1\ldots1}_{n-1}$ and since it holds that $|V_2'|_{2^{4n}-1} = |1\ldots1|_{2^{4n}-1} = 0$, V_2' can be ignored. As a result, only three $4n$-bit EAC CSAs and a final modulo 2^{4n}-1 adder are required for the reverse converter.

Assume now that both x_3 and x_4 are encoded in diminished-one. Then, according to the proposed methodology, x_3 and x_4 are replaced by $x_3' = 2^n x_{3z} + x_{3d}$ and $x_4' = 2^{2n} x_{4z} + x_{4d}$ respectively, and the following correction term, CT, is taken into account:

$$CT = |C_3 + C_4|_{2^{4n}-1} = \Big|(-2^{4n-2} + 2^{3n-2} - 2^{2n-2} + 2^{n-2}) + (2^{3n-1} - 2^{n-1})\Big|_{2^{4n}-1}$$

Let $x_{3,n}\ldots x_{3,0}$ and $x_{4,2n}\ldots x_{4,0}$ denote now the bits of the residues x_3' and x_4', respectively. The constant bits of vectors V_3 and V_6' form the values $T_{V_3} = (2^{n-1}-1)2^{3n-1} + (2^{n-1}-1)2^{n-1}$ and $T_{V_6'} = (2^{n-1}-1)$ respectively. By summing T_{V_3} and $T_{V_6'}$ along with CT, modulo 2^{4n}-1, a single overall correction term, CT_{ov}, can be formed with:

$$CT_{ov} = |CT + T_{V_3} + T_{V_6'}|_{2^{4n}-1} = |2^{3n-2} - 2^{n-1} + 2^{n-2} - 1|_{2^{4n}-1}$$
$$= \underbrace{0\ldots0}_{n+1}\underbrace{01\ldots1}_{2n-1}1\underbrace{01\ldots1}_{n-1}$$

We can merge the bits of CT_{ov} with the non-constant bits of V_3, V_5 and V_6' so as to form the vectors:

$$V_3' = \overline{x}_{3,1}\overline{x}_{3,0}\underbrace{0\ldots0}_{n-1}\underbrace{\overline{x}_{3,n}\ldots\overline{x}_{3,0}}_{n+1}\underbrace{0\ldots1}_{n-1}\underbrace{\overline{x}_{3,n}\ldots\overline{x}_{3,2}}_{n-1},$$

$$V_5' = \underbrace{x_{4,n}\ldots x_{4,0}}_{n+1}\underbrace{01\ldots1}_{2n-1}\underbrace{x_{4,2n}\ldots x_{4,n+1}}_{n} \text{ and}$$

$$V_6'' = \underbrace{\overline{x}_{2,n-1}\ldots\overline{x}_{2,0}}_{n}\underbrace{\overline{x}_{4,2n}\ldots\overline{x}_{4,0}}_{2n+1}\underbrace{01\ldots1}_{n-1}$$

Thus, the final vectors to be added in a modulo 2^{4n}-1 fashion are V_1, V_3', V_4, V_5' and V_6''. We conclude that even in the case where x_3 and x_4 are encoded in diminished-one, the proposed converter has zero area and delay overheads compared to the simplified one previously presented for channels that follow the normal representation.

■

III. Evaluation and Comparison

In this section we evaluate the converters that are derived by following the methodology introduced in the previous section for several moduli sets. We assume that all channels of the 2^k+1 form follow the diminished-one encoding in all cases. The proposed converters are based on the corresponding converters for the normally encoded residues and are designed by replacing these residue(s) with the corresponding diminished-one encoded residue(s) and by considering a constant correction as well. Hence, in the worst case, a simplified CSA with EAC will be required to accommodate this constant correction, resulting in a maximum delay overhead equal to that of the delay of a half-adder (HA) or a simplified full-adder (SFA) and a maximum area overhead equal to that of the area of t HAs or SFAs, with t denoting the width of the constant correction term. However, merging the constant correction with the constant bits that are already present in the converters cancels in most cases all area and delay overheads (see Examples 1-3).

Reference [7] has presented reverse converters for the {2^n-1, 2^{n+k}, 2^n+1} (k>0) moduli set for a diminished-one encoded 2^n+1 channel. However, [7] does not present any details on how zero values are handled and considers only a specific moduli set.

Since most reported reverse converters assume normally encoded modulo 2^k+1 operands at their inputs, we compare the proposed diminished-one converters against those that assume a normal encoding in these residues. We do not include the diminished-to-normal converter in the latter which in every case adds considerable area and delay. Table I presents comparison results for several CRT-based and New CRT-I based converters for various moduli sets and reports the area and delay savings. A_{HA} (T_{HA}) denotes the area (delay) of a HA or an SFA, while A_{FA} denotes the area of a full-adder (FA). Table I reveals that the proposed converter for the well-known {2^n-1, 2^n, 2^n+1} moduli set is more efficient compared to the converter of [33] that assumes a normal encoding for the modulo 2^n+1 operands since it has the same delay while is smaller since n FAs are replaced by HAs/SFAs. Besides that, in several other moduli set cases, the delay overhead of the proposed converters is zero and/or the area overhead is very small, that is, a few HAs of SFAs, unlike the logarithmic to n overhead that a diminished-to-normal converter would introduce. This is due to merging the correction term bits of the proposed methodology with the rest constant bits in each converter.

TABLE I
COMPARISON BETWEEN REVERSE CONVERTERS WITH DIMINISHED-ONE ENCODED AND NORMALLY ENCODED INPUTS

Method	*Moduli set*	*Area savings*	*Delay savings*
CRT	$\{2^n\text{-}1,2^n,2^n+1\}$ [33]	$n(A_{FA}–A_{HA})$	0
	$\{2^n\text{-}1,2^{2n},2^n+1\}$ [13]	0	0
	$\{2^{n\text{-}1},2^n\text{-}1,2^n+1\}$ [12]	$–(n–1)(A_{FA}–A_{HA})$	0
	$\{2^n,2^{2n}\text{-}1,2^{2n}+1\}$ [11][19]	$–1A_{HA}$	$–T_{HA}$
	$\{2^{p+k},2^p+1,2^p\text{-}1,2^{2p}+1\}$ [15]	0	0
New CRT-I	$\{2^\alpha,2^\beta\text{-}1,2^\beta+1\}$ $(\alpha<\beta)$ [24]	$–1A_{HA}$	$–T_{HA}$
	$\{2^{2n+1},2^n\text{-}1,2^n+1\}$ [17]	$–1A_{HA}$	$–T_{HA}$
	$\{2^n\text{-}1,2^n+1,2^{2n},2^{2n}+1\}$ [23]	$–2A_{HA}$	0
	$\{2^n\text{-}1,2^n,2^n+1, 2^{2n}+1\}$[3]	0	0
First stage: CRT or New CRT-I Last stage: MRC	$\{2^n\text{-}1,2^n,2^n+1,2^{n\pm1}\text{-}1\}$ [5]	$n(A_{FA}–A_{HA})$	0
	$\{2^n\text{-}1,2^n,2^n+1,2^{n+1}\text{-}1\}$ [20]	$n(A_{FA}–A_{HA})$	0
	$\{2^n\text{-}1,2^n,2^n+1,2^{n+1}\text{-}1,2^{n\text{-}1}\text{-}1\}$ [4]	$n(A_{FA}–A_{HA})$	0
	$\{2^k,2^n\text{-}1,2^n+1,2^{n+1}\text{-}1\}$ [6]	0	0
	$\{2^n,2^{n/2}\text{-}1,2^{n/2}+1,2^n+1,2^{2n\text{-}1}\text{-}1\}$ [21]	0	0
	$\{2^n\text{-}1,2^{2n},2^n+1,2^{2n+1}\text{-}1\}$ [22]	0	0

The MRC is another method that is frequently combined with the CRT or the New CRT-I methods in the design of reverse converters. Table I also presents comparison results for several such converters. The reverse converters proposed in [5], [20] moduli sets are realized in two stages, utilizing the CRT in the first stage for the moduli subset $\{2^n\text{-}1, 2^n, 2^n+1\}$ and the MRC in the second stage for the last modulo which is of the $2^k\text{-}1$ form. It is evident that if the modulo 2^n+1 channel is encoded in diminished-one, then the converter of the Example 2 can be utilized and no other modification is required. The same holds for the reverse converter of the moduli set presented in [4] since in the first stage the CRT-based reverse converter for the $\{2^n\text{-}1, 2^n, 2^n+1\}$ set is used and then the MRC technique is applied twice for the remaining moduli. A similar approach can be used in the converters of [6], [21] and [22] leading to diminished-one reverse converters with the same area and delay requirements as the normally-encoded ones.

We have to note that the actual savings offered by the proposed converters would be significantly higher since all converters with normally-encoded operands at their inputs have to be accompanied by a diminished-to-normal converter (DNC) for every modulus channel that adopts the diminished-one encoding. The latter was not considered in Table I.

To attain more realistic results, reverse converters for the moduli sets of Examples 1-3 were described in HDL for values of n equal to 4, 8, 16 and 32 bits with both normally-encoded operands (hereafter denoted as normal reverse converters, NRCs) and diminished-one encoded operands (hereafter denoted as proposed) of the 2^k+1 form. For the sake of completeness we also examine the corresponding circuits where the diminished-one encoded operands drive DNC(s) followed by the normally encoded reverse converter (denoted as DNC+NRC). After simulating the resulting descriptions, the converters were synthesized and mapped to a 90 nm power-characterized CMOS implementation technology. The Synopsys Design Compiler tool in the topographical mode was used for synthesis and mapping. Each converter was recursively optimized for speed using a bottom-up approach. A final area recovery step was then applied. For obtaining average power dissipation data, we assumed an operating frequency of 400MHz and equiprobable inputs.

Tables II, III and IV present the attained area, delay and power dissipation results for the converters of Examples 1-3 respectively. It is evident that the proposed methodology results in reverse converters with up to 10% area and 16% power savings without any loss in delay compared to the corresponding NRCs. Furthermore, the results indicate that if the diminished-one encoding is adopted in the modulo 2^k+1

TABLE II
CMOS VLSI RESULTS OF REVERSE CONVERTERS FOR THE $\{2^{n+k}, 2^n+1, 2^n\text{-}1, 2^{2n}+1\}$ MODULI SET WITH $k=n/2$.

	NRC [15]			**DNC+NRC [15]**			**Proposed**			**Savings over NRC [15]**			**Savings over DNC+NRC [15]**		
n	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (%)***	***Delay (%)***	***Power (%)***	***Area (%)***	***Delay (%)***	***Power (%)***
4	6538	1.37	0.91	7300	1.71	0.95	6550	1.369	0.92	-0.2	0.0	-0.7	10.3	19.9	3.8
8	14339	1.53	1.88	16347	2.20	1.97	14385	1.530	1.89	-0.3	0.2	-0.5	12.0	30.5	4.0
16	31710	1.68	3.90	36140	2.72	4.11	31721	1.676	3.91	0.0	0.2	-0.3	12.2	38.4	4.8
32	69231	1.99	8.46	79227	3.06	9.00	69222	1.978	8.49	0.0	0.4	-0.4	12.6	35.4	5.7

TABLE III
CMOS VLSI RESULTS OF REVERSE CONVERTERS FOR THE $\{2^n\text{-}1, 2^n, 2^n+1\}$ MODULI SET.

	NRC [33]			**DNC+NRC [33]**			**Proposed**			**Savings over NRC [33]**			**Savings over DNC+NRC [33]**		
n	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (%)***	***Delay (%)***	***Power (%)***	***Area (%)***	***Delay (%)***	***Power (%)***
4	2397	0.86	0.35	2615	1.22	0.40	2256	0.86	0.31	5.9	0.4	10.4	13.7	29.5	22.7
8	5480	1.05	0.77	6119	1.58	0.87	5089	1.03	0.69	7.1	2.6	10.6	16.8	34.9	21.2
16	12699	1.23	1.73	13905	2.09	1.94	11624	1.24	1.52	8.5	-0.7	12.1	16.4	40.8	21.6
32	28142	1.42	3.74	31515	2.53	4.05	25366	1.43	3.15	9.9	-0.2	15.8	19.5	43.5	22.2

TABLE IV
CMOS VLSI RESULTS OF REVERSE CONVERTERS FOR THE $\{2^n\text{-}1, 2^n, 2^n+1, 2^{2n}+1\}$ MODULI SET.

	NRC [3]			**DNC+NRC [3]**			**Proposed**			**Savings over NRC [3]**			**Savings over DNC+NRC [3]**		
n	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (μm^2)***	***Delay (ns)***	***Power (mW)***	***Area (%)***	***Delay (%)***	***Power (%)***	***Area (%)***	***Delay (%)***	***Power (%)***
4	6413	1.38	0.90	7214	1.69	0.95	6386	1.38	0.896	0.4	-0.3	-0.1	11.5	18.3	5.8
8	14266	1.57	1.82	16105	2.22	1.94	14310	1.55	1.819	-0.3	1.1	-0.2	11.2	30.3	6.0
16	31780	1.70	3.90	36037	2.79	4.25	31654	1.70	3.920	0.4	-0.1	-0.6	12.2	39.1	7.7
32	69143	1.92	8.39	78714	3.07	8.98	69256	1.89	8.405	-0.2	1.3	-0.2	12.0	38.4	6.4

channels, then the proposed reverse converters offer significant savings in all terms (area, delay and power dissipation) compared to the only equivalent solution which is to use NRCs driven by DNCs for the diminished-one encoded operands. In this case, the comparison reveals savings in area, delay and power dissipation up to 20%, 44% and 23% respectively. The delay savings are justified by the logarithmic delay introduced by the controlled binary incrementer used for the diminished-to-normal conversion, which was designed as a simplified parallel-prefix-based structure, in order to be as fast as possible.

IV. Conclusions

An efficient methodology for designing reverse converters that assume diminished-one encoded modulo 2^k+1 operands has been proposed. The methodology can be applied to converters that use the CRT or the New CRT-I methods. Evaluation and experimental results of the reverse converters for several moduli sets have shown that they are as efficient as those that assume the normal encoding for the modulo 2^k+1 operands in most cases, while in few cases a very small area and delay overhead may be required. If however the cost of the diminished-to-normal converters is also taken into account, then the proposed converters are much more efficient in all cases. Hence, adopting the diminished-one encoding for the modulo 2^k+1 residues in an RNS now becomes even more preferable, since the costly diminished-to-normal conversion is no longer required before the residue-to-binary conversion.

Acknowledgment

This work was supported by the Caratheodory Programme of the University of Patras (D.178).

References

[1] S. Andraos, and H. Ahmed "A new efficient memoryless residue to binary converter," *IEEE Trans. Circuits Syst.*, vol 35, pp. 1441-1444, 1988.

[2] D. Bakalis, H. T. Vergos, and A. Spyrou, "Efficient Modulo 2^n±1 Squarers," *Integration, the VLSI journal*, vol. 44, no. 3, pp. 163-174, 2011.

[3] B. Cao, C.-H. Chang, and T. Srikanthan "An efficient reverse converter for the 4-moduli Set $\{2^n$-1, 2^n, 2^n+1, 2^{2n}+1$\}$ based on the new Chinese remainder theorem," *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 1296-1303, 2003.

[4] B. Cao, C.-H. Chang, and T. Srikanthan "A residue-to-binary converter for a new five-moduli set," *IEEE Trans. Circuits Syst. I*, vol. 54, pp. 1041-1049, 2007.

[5] B. Cao, T. Srikanthan, and C.-H. Chang, "Efficient reverse converters for 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.*, vol. 152, pp. 687-696, 2005.

[6] G. Chalivendra, V. Hanumaiah, and 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 FIR filter implementation," in *Proc. Great Lakes Symp. on VLSI*, Utah, 2011, pp. 139-144.

[7] R. Chaves, and L. Sousa, "Improving residue number system multiplication with more balanced moduli sets and enhanced modular arithmetic structures," *IET Comput. Digit. Tech.*, vol. 1, pp. 472-480, 2007.

[8] J.W. Chen, and R.H. Yao, "Efficient modulo 2^n+1 multipliers for diminished-1 representation," *IET Circuits Dev. Syst.*, vol. 4, pp. 291-300, 2010.

[9] C. Efstathiou et al., "Efficient diminished-1 modulo 2^n+1 multipliers," *IEEE Trans. Comput.*, vol. 54, pp. 491-496, 2005.

[10] C. Efstathiou, H.T. Vergos, and D. Nikolos, "Handling zero in diminished-one modulo 2^n+1 adders," *Int. J. Electronics*, vol. 90, pp. 133–144, 2003.

[11] A. Hariri, K. Navi, and R. Rastegar, "A new high dynamic range moduli set with efficient reverse converter," *Comput. Math. Appl.*, vol. 55, pp. 660-668, 2008.

[12] A. Hiasat and A. Sweidan, "Residue number system to binary converter for the moduli set $\{2^{n-1}$, 2^n-1, 2^n+1$\}$," *J. Systems Architecture*, vol. 49, pp. 53-58, 2003.

[13] A. Hiasat and A. Sweidan, "Residue-to-binary decoder for an enhanced moduli set," *IEE Proc. Comput. Digit. Tech.*, vol. 151, pp. 127-130, 2004.

[14] G. Jaberipur, and S. Nejati, "Balanced minimal latency RNS addition for moduli set $\{2^n$-1, 2^n, 2^n+1),” in *Proc. 18th Int. Conf. Syst. Signals and Image Proc.*, 2011, pp. 1-7.

[15] Y. Kuo et al., "Efficient VLSI design of a reverse RNS converter for new flexible 4-moduli set (2^{p+k}, 2^p+1, 2^p-1, 2^{2p}+1)," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2009, pp. 437–440.

[16] L. Leibowitz, "A Simplified binary arithmetic for the Fermat number transform," *IEEE Trans. Acoustics Speech Signal Process.*, vol. 24, pp. 356-359, 1976.

[17] S. Lin et al., "Efficient VLSI design for RNS reverse converter based on new moduli set (2^n-1, 2^n+1, 2^{2n+1})," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, 2006, pp. 2020-2023.

[18] P.V.A. Mohan, *Residue Number Systems: Algorithms and Architectures, The Springer International Series in Engineering and Computer Science, Vol. 677,* Ed. Norwell, MA: Kluwer Academic Publishers, 2002.

[19] P.V.A. Mohan, "Reverse Converters for a New Moduli Set $\{2^{2n}$-1, 2^n, 2^{2n}+1$\}$," *Circuits Syst. Signal Process.*, vol. 26, pp. 215-227, 2007.

[20] P.V.A. Mohan, and 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*, vol. 54, pp. 1245-1254, 2007.

[21] A. Molahosseini, C. Dadkhah, and K. Navi, "A new five-moduli set for efficient hardware implementation of the reverse converter," *IEICE Electronics Express*, vol. 6, pp. 1006-1012, 2009.

[22] A. Molahosseini, and K. Navi, "A reverse converter for the enhanced moduli set $\{2^n$-1, 2^n+1, 2^{2n}, 2^{2n+1}-1$\}$ *using CRT and MRC*," in *Proc. IEEE Comput. Soc. Annu. Symp. on VLSI*, 2010, pp. 456-457.

[23] A. Molahosseini et al., "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*, vol. 57, pp. 823-835, 2010.

[24] A. Molahosseini et al, "An efficient architecture for designing reverse converters based on a general three-moduli set," *J. Syst. Architecture*, vol. 54, pp. 929-934, 2008.

[25] A. Omondi, and B. Premkumar, *Residue Number Systems: Theory and Implementation,* Ed. London: Imperial College Press, 2007.

[26] T.R. Rao, *Error Coding for Arithmetic Processors,* Ed. Academic Press, 1974.

[27] L. Sousa, and R. Chaves, "A universal architecture for designing efficient modulo 2^n+1 multipliers," *IEEE Trans. Circuits Syst. I*, vol. 52, pp. 1166-1178, 2005.

[28] E. Vassalos, D. Bakalis, and H.T. Vergos, "Modulo 2^n+1 arithmetic units with embedded diminished-to-normal conversion," in *Proc. Euromicro Conf. Digit. Syst. Design*, 2011, pp. 468-475.

[29] H.T. Vergos, and D. Bakalis, "On implementing efficient modulo 2^n+1 arithmetic components," *J. Circuits Syst. Comput.*, vol. 19, pp. 911-930, 2010.

[30] H.T. Vergos, C. Efstathiou, and D. Nikolos, "Diminished-one modulo 2^n+1 adder design," *IEEE Trans. Comput.*, vol. 51, pp. 1389-1399, 2002.

[31] Y. Wang, "Residue-to-binary converters based on new Chinese remainder theorems," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 197-205, 2000.

[32] Y. Wang et al., "Adder based residue to binary number converters for (2^n-1, 2^n, 2^n+1)," *IEEE Trans. Signal Process.*, vol. 50, pp. 1772-1779, 2002.

[33] Z. Wang, G.A. Jullien, and W.C. Miller, "An improved residue-to-binary converter," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 1437-1440, 2000.

[34] D.G. Baily, Design for embedded image processing on FPGAs, 1st ed.John Wiley & Sons: Asia, 2011