Residue-to-Binary Converters for the Moduli Set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$

K.A. Gbolagade[1,2], R. Chaves[3], L. Sousa[3], S.D. Cotofana[1]
1. Computer Engineering Laboratory, Delft University of Technology,
The Netherlands. E-mail: {gbolagade,sorin}@ce.et.tudelft.nl
2. University for Development Studies, Navrongo, Ghana,
3. DSP Group, Technical University of Technology, Lisbon.

Abstract*—In this paper, we propose two memoryless converters for the moduli set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$. First, we propose a novel reverse converter, which is purely adder based, using the traditional Chinese Remainder Theorem (CRT). Second, due to the fact that the proposed CRT based structure does not cover the entire dynamic range, a second converter, which covers the entire dynamic range based on Mixed Radix Conversion (MRC), is proposed. The CRT based converter outperforms the MRC based converter both in terms of area and delay. In comparison with related best known state of the art converters, they are all outperformed by the proposed CRT based scheme in terms of both area cost and conversion delay. The theoretical evaluation is supported by the experimental results, which are estimated on a Standard Cell 0.13-μm CMOS technology. These experimental results indicate that, on average, for the same dynamic range, the proposed CRT based converter achieves about 23% delay reduction with more than 3% area reduction, when compared to the existing state of the art MRC based converter. Additionally, the proposed CRT based converter is about 6% faster with about 4% area reduction when compared with the existing CRT based converter.

***Index Terms*—Residue Number System, Reverse Converter, Chinese Remainder Theorem, Mixed Radix Conversion, Memoryless Converter.**

I. Introduction

Residue Number Systems (RNS) have received considerable attention in literature due to their inherent properties such as carry-free operations, parallelism, modularity, and fault tolerance [9]. RNS architectures are typically composed of three main parts namely, a binary-to-residue converter, residue arithmetic units, and a residue-to-binary converter. The residue-to-binary converter is the most complex part of any RNS architecture. Moduli set choice is also an important issue since the complexity and the speed of the resulting conversion structure depend on the chosen moduli set. Several structures have been proposed to perform the reverse conversion for different moduli sets, e.g., $\{2^n, 2^n-1, 2^n+1\}$ [2], $\{2^n, 2^n-1, 2^{n-1}-1\}$ [5], $\{2^{n+1}+1, 2^{n+1}-1, 2^n\}$ [7], $\{2^n, 2^{n+1}-1, 2^n-1\}$ [6], [8], $\{2n+2, 2n+1, 2n\}$ [4]. Efficient memoryless adder-based reverse converter have been proposed for the moduli set $\{2^n, 2^n-1, 2^n+1\}$. This moduli set has the disadvantage [5] that multiplication by powers of 2 with respect to the (2^n+1) modulus is not as simple as left circular rotation in a (2^n-1) modulus. Due to this reason, the moduli set $\{2^n, 2^n-1, 2^{n-1}-1\}$ was proposed in [5]. However, larger dynamic ranges than the one provided by the moduli set $\{2^n, 2^n-1, 2^{n-1}-1\}$ are required. In [6], the moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$ was proposed, by the elimination of the modulus (2^n+1) from the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ proposed in [10]. It avoids the modulo (2^n+1)-type arithmetic, which is more complex and degrades the RNS efficiency. For the cases that the moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$, has insufficient dynamic range, the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$ was proposed in [1] together with a reverse converter based on Mixed Radix Conversion (MRC). Recently, a fast Chinese Remainder Theorem (CRT)-based reverse converter for the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$ was proposed in [3]. The disadvantage of this moduli set is that the binary channel is underutilized.

In this paper, we propose two new reverse converters for the $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$ moduli set by extending the modulus 2^n in the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set. We propose both CRT based (ConverterI) and MRC based (ConverterII) reverse converters. ConverterI is better than ConverterII, the converter in [1], and the one in [3] both in terms of area and delay. However, ConverterII is also useful because it covers the entire dynamic range whereas ConverterI does not. Synthesis results for a 0.13-μm CMOS standard cell-based technology are also obtained and suggest that, on average, the proposed Con-

978-1-4244-3523-4/09/$25.00

verterI improves the conversion performance by about 21%, and also reduced the area cost by about 14%, when compared to proposed ConverterII. Additionally, the synthesis results also indicate that, on average, the proposed ConverterI is 25% and 9% better in terms of Area-Time (AT) product than the reverse converters in [1] and [3], respectively.

The rest of this paper is organized as follows. Section II presents the necessary background. In Section III, the new CRT based algorithm for reverse conversion is proposed. Section IV presents the MRC based conversion technique. Section V presents the hardware realizations of both of the CRT and the MRC based conversion methods while Section VI gives a performance comparison with the similar best known state of the art converters. The paper is concluded in Section VII.

II. Background

RNS is defined in terms of a set of relatively prime moduli set $\{m_i\}_{i=1,k}$, such that $\gcd(m_i,m_j)=1$ for $i\neq j$, where gcd means the greatest common divisor of m_i and m_j, $M=\prod_{i=1}^{k}m_i$, is the dynamic range. When the residues of a decimal number X are represented as $x_i=|X|_{m_i}$, X can be represented in RNS as $X=(x_1,x_2,x_3,...,x_k)$, $0\leq x_i<m_i$. This representation is unique for any integer $X\in[0,M-1]$.

For a moduli set $\{m_i\}_{i=1,k}$, the residues $(x_1,x_2,x_3,...,x_k)$ can be converted into the corresponding decimal number X, according to the traditional CRT, as follows [12]:

$$X=\left|\sum_{i=1}^{k}M_i\left|M_i^{-1}x_i\right|_{m_i}\right|_M, \tag{1}$$

where $M=\prod_{i=1}^{k}m_i$, $M_i=\frac{M}{m_i}$, and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i.

Relatively, performing reverse conversion using MRC is fast since it does not involve the large modulo-M calculations present in CRT. However, the problem with the MRC is that it is naturally a sequential process. Suppose that we have a residue number $(x_1,x_2,x_3,...,x_n)$ for the moduli set $\{m_1,m_2,m_3,...,m_n\}$ and a set of digits $\{a_1,a_2,a_3,...,a_n\}$, which are the Mixed Radix Digits (MRDs). The decimal equivalent X of the residues can be computed as follows [9]:

$$X=a_1+a_2m_1+a_3m_1m_2+...+a_nm_1m_2m_3...m_{k-1} \tag{2}$$

where the MRDs are given as follows:

$$\begin{aligned}
a_1 &= x_1\\
a_2 &= \left|(x_2-a_1)\left|m_1^{-1}\right|_{m_2}\right|_{m_2}\\
a_3 &= \left|\left((x_3-a_1)\left|m_1^{-1}\right|_{m_3}-a_2\right)\left|m_2^{-1}\right|_{m_3}\right|_{m_3}\\
&\ \ .\\
&\ \ .\\
a_k &= \Big|(((x_k-a_1)\left|m_1^{-1}\right|_{m_k}-a_2)\left|m_2^{-1}\right|_{m_k}\\
&\quad -...-a_{k-1})\left|m_{k-1}^{-1}\right|_{m_k}\Big|_{m_k}
\end{aligned} \tag{3}$$

For the MRDs a_i, $0\leq a_i<m_i$, any positive number in the interval $[0,\ \Pi_{i=1}^{k}m_i-1]$ is uniquely represented.

III. CRT Based Conversion Method

Given the RNS number (x_1,x_2,x_3) with respect to the moduli set $\{m_1,m_2,m_3\}$ in the form $\{2^{2n+1}-1,2^{2n},2^n-1\}$, the proposed algorithm computes the decimal equivalent of this RNS number based on a further simplification of the well-known traditional CRT. First, we wish to show that the moduli $2^{2n+1}-1$, 2^{2n}, and 2^n-1 are relatively prime.

Theorem 1: The moduli set $\{2^{2n+1}-1,2^{2n},2^n-1\}$ contains pairwise relatively prime moduli.

Proof: From Euclidean theorem, we have:
$gcd(a,b)=gcd(b,|a|_b)$,
therefore,

$$\begin{aligned}
gcd(2^{2n+1}-1,2^{2n}) &= gcd(2^{2n},\left|2^{2n+1}-1\right|_{2^{2n}})\\
&= gcd(2^{2n},1)=1
\end{aligned}$$

Similarly, it can easily be shown that $gcd(2^{2n+1}-1,2^n-1)=1$ and $gcd(2^{2n},2^n-1)=1$ hold true. Thus, the moduli $2^{2n+1}-1$, 2^{2n}, and 2^n-1 are relatively prime since all the greatest common divisors are equal to 1. ■

Theorem 2: Given the moduli set $\{m_1,m_2,m_3\}$ with $m_1=2^{2n+1}-1, m_2=2^n, m_3=2^n-1$, the following hold true:

$$|(m_1m_2)^{-1}|_{m_3} = 1, \tag{4}$$
$$|(m_1m_3)^{-1}|_{m_2} = 2^n+1, \tag{5}$$
$$|(m_2m_3)^{-1}|_{m_1} = 2^{2n+1}-2^{n+2}-5. \tag{6}$$

Proof: If it can be demonstrated that $|1\times(m_1m_2)|_{m_3}=1$, then 1 is the multiplicative inverse of (m_1m_2) with respect to m_3:

$$\begin{aligned}
R_1 &= \left|2^{2n}(2^{2n+1}-1)\right|_{2^n-1}\\
&= |1\times 1|_{2^n-1}=1,
\end{aligned}$$

thus (4) holds true.

Similarly, (5) and (6) can be easily proved.

■

The following important relations will be utilized in the subsequent theorem: Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^{2n+1}-1, m_2 = 2^{2n}, m_3 = 2^n - 1$, the following holds true:

$$m_1 = 2m_2 - 1, \quad (7)$$

$$m_2 = m_3 m_3 + 2m_3 + 1. \quad (8)$$

Theorem 3: The decimal equivalent of the residues $(x_{i,i=1,3})$ with respect to the moduli set $\{m_{i,i=1,3}\}$ in the form $\{2^{2n+1}-1, 2^{2n}, 2^n - 1\}$, assuming $X \in [0, M-(m_2^2 - 2m_2 + 1))$, can be computed as follows:

$$X = m_2 \left\lfloor \frac{X}{m_2} \right\rfloor + x_2 \quad (9)$$

where

$$\left\lfloor \frac{X}{m_2} \right\rfloor = x_3 - x_2 + m_3 \left| (-2^{n+2} - 4)x_1 + 2^{n+1}x_2 + 2x_2 + 2^{n+1}x_3 + 2x_3 \right|_{m_1} \quad (10)$$

Proof: Since (9) follows the basic integer division definition in RNS, which is always true, we only need to show the correctness of (10). For $k = 3$, (1) becomes:

$$X = \left| \sum_{i=1}^{3} M_i \left| M_i^{-1} x_i \right|_{m_i} \right|_M \quad (11)$$

Substituting (4), (5), and (6) into (11) we obtain the following:

$$X = \left| m_2 m_3 (2^{2n+1} - 2^{n+2} - 5)x_1 + m_1 m_3 (2^n + 1)x_2 + m_1 m_2 x_3 \right|_M,$$

and by substituting (7) and (8) in the above equation, we obtain:

$$X = \left| (m_2 m_3)(2^{2n+1} - 2^{n+2} - 5)x_1 + (2^{n+1} m_2 m_3 x_2 - 2^n m_3 x_2 + 2m_2 m_3 - m_3 x_2) + (2m_3 m_3 m_2 x_3 - m_3 m_3 x_3 + 2m_3 x_3 + x_3) \right|_M. \quad (12)$$

(12) can be further simplified by using the following lemma, presented in [12]:

$$|a m_1|_{m_1 m_2} = m_1 |a|_{m_2} \quad (13)$$

Applying (13) and (8) and given that $\left|2^{2n+1}\right|_{2^{2n+1}-1} = 1$, (12) becomes:

$$\begin{aligned} X &= \left| -2^n m_3 x_2 - m_3 x_2 + m_2 x_3 + m_2 m_3 \left| (-2^{n+2} - 4)x_1 + 2^{n+1}x_2 + 2x_2 + 2m_3 x_3 + 2^2 x_3 \right|_{m_1} \right|_M \\ &= \left| -m_2 x_2 + x_2 + m_2 x_3 + m_2 m_3 \left| (-2^{n+2} - 4)x_1 + 2^{n+1}x_2 + 2x_2 + 2m_3 x_3 + 2^2 x_3 \right|_{m_1} \right|_M \end{aligned} \quad (14)$$

Dividing both sides of (14) by m_2 and taking the floor, we obtain:

$$\left\lfloor \frac{X}{m_2} \right\rfloor = |x_3 - x_2 + m_3 \left| -2^{n+2}x_1 - 2^2 x_1 + 2^{n+1}x_2 + 2x_2 + 2^{n+1}x_3 + 2x_3 \right|_{m_1} \Big|_{m_1 m_3} \quad (15)$$

It can be seen that (15) is the general expression of (10), which is valid for the entire dynamic range, $[0, M-1]$. The next stage of the proof is to demonstrate that the corrective addition required for the calculation of the mod-$m_1 m_3$ can be avoided in most of the cases. We demonstrate that by considering the two extreme cases, i.e., the most positive and most negative value one may get in (15).

- *Most positive value*: in order to get the most positive value in (15), the following must hold true:

$$x_1 = 2^{2n+1} - 2, \quad (16)$$

$$x_2 = 2^{2n} - 1, \quad (17)$$

$$x_3 = 2^n - 2, \quad (18)$$

$$m_1 = 2^{2n+1} - 1, \quad (19)$$

$$R = \left| -2^{n+2}x_1 - 4x_1 + 2^{n+1}x_2 + 2x_2 + 2^{n+1}x_3 + 2x_3 \right|_{m_1}. \quad (20)$$

  - If $x_2 = 0$ and (16) and (18) hold true, then (20) becomes $R = 2^{n+1} + 1$ and (15) reduces to $|2^{2n+1} - 3|_{m_1 m_3}$.
  - Even, if $x_2 = 0$ and $R = 2^{2n+1} - 2$, (15) reduces to $|2^{3n+1} - 2^{2n+1} - 2^n|_{m_1 m_3}$.
  - All the expressions obtained above for (15) are less than $m_1 m_3$, which is equal to $2^{3n+1} - 2^{2n+1} - 2^n + 1$.
  - Therefore, no corrective addition of $m_1 m_3$ is needed in (15). Thus, (10) holds true.
- *Most negative value*: in order to get the most negative value in (15), the following must hold true: $x_3 = 2^n - 2$, $x_2 = 2^{2n} - 1$, and in (20), $R = 2^n$. Substituting these values in (15), we obtain $|-1|_{m_1 m_3}$. The absolute value of -1 is always less than $m_1 m_3$, thus only one corrective addition is needed.

We wish to show that this corrective addition is needed only in very few cases. From (9), the minimum X value that needs a corrective addition of $m_1 m_3$ occurs when $\left\lfloor \frac{X}{m_2} \right\rfloor$ has the lowest value, since $m_2 \left\lfloor \frac{X}{m_2} \right\rfloor$ grows faster than x_2. By using the minimum values in (15), specifically $\left\lfloor \frac{X}{m_2} \right\rfloor = (-(m_2 - 1) + m_1 m_3)$ since $x_2 = m_2 - 1$, the minimum value of X can be computed as:

$$\begin{aligned} X_{min} &= m_2(-m_2 + 1 + m_1 m_3) + (m_2 - 1) \\ &= M - (m_2^2 - 2m_2 + 1) \end{aligned} \quad (21)$$

On the other hand, the maximum value of X that needs a corrective addition is given by:

$$\begin{aligned} X_{max} &= m_2(-1+m_1m_3)+(m_2-1) \\ &= M-m_2+m_2-1=M-1 \end{aligned} \quad (22)$$

Thus, the numbers that need corrective additions lay in the interval $[M-(m_2^2-2m_2+1),M-1]$, which is on the top part of the dynamic range. Generally speaking, if the numbers in the interval $[M-(m_2^2-2m_2+1),M-1]$ require corrective addition, the numbers within the interval $[0,M-(m_2^2-2m_2+1))$ require no corrective addition and thus, (10) holds true. ■

We can further reduce the hardware complexity of the reverse converter by simplifying (10) using the following two properties [7].

Property 1: Modulo (2^s-1) multiplication of a residue number by 2^t, where s and t are positive integers, is equivalent to t bit circular left shifting.

Property 2: Modulo (2^s-1) of a negative number is equivalent to the one's complement of the number, which is obtained by subtracting the number from (2^s-1).

Assuming that (10) is written as:

$$\begin{aligned} \left\lfloor \frac{X}{m_2} \right\rfloor &= x_3-x_2+(2^n-1)A \\ &= x_3-x_2+2^nA-A, \end{aligned} \quad (23)$$

where

$$A = |u_0+u_1+u_2+u_3+u_4|_{2^{2n+1}-1}. \quad (24)$$

For simplicity sake, let us represent (23) as:

$$\left\lfloor \frac{X}{m_2} \right\rfloor = B_1+B_2+B_3, \quad (25)$$

where

$$\begin{aligned} B_1 &= -x_2, \\ B_2 &= 2^nA+x_3, \\ B_3 &= -A. \end{aligned} \quad (26)$$

Let the binary representations of the residues be:

$$\begin{aligned} x_1 &= (x_{1,2n}x_{1,2n-1}...x_{1,1}x_{1,0}), \\ x_2 &= (x_{2,2n-1}x_{2,2n-2}...x_{2,1}x_{2,0}), \\ x_3 &= (x_{3,n-1}x_{3,n-2}...x_{3,1}x_{3,0}). \end{aligned}$$

In (24), u_0, u_1, u_2, u_3, and u_4 are represented as follows:

$$\begin{aligned} u_0 &= \left|-2^{n+2}x_1\right|_{2^{2n+1}-1} \\ &= \left|-2^{n+2}(x_{1,2n}x_{1,2n-1}...x_{1,0})\right|_{2^{2n+1}-1} \\ &= (\underbrace{\overline{x}_{1,n-2}\overline{x}_{1,n-3}...\overline{x}_{1,1},\overline{x}_{1,0}}_{n-1}\underbrace{\overline{x}_{1,2n}\overline{x}_{1,2n-1}...\overline{x}_{1,n-1}}_{n+2}), \\ u_1 &= \left|-2^2x_1\right|_{2^{2n+1}-1} \\ &= (\underbrace{\overline{x}_{1,2n-2}\cdots\overline{x}_{1,0}\overline{x}_{1,2n}\overline{x}_{1,2n-1}}_{2n+1}), \\ u_2 &= \left|2^{n+1}x_2\right|_{2^{2n+1}-1} \\ &= \left|2^{n+1}(x_{2,2n}x_{2,2n-1}...x_{2,0})\right|_{2^{2n+1}-1} \\ &= |2^n(x_{2,2n}x_{2,2n-1}...x_{2,0}0)|_{2^{2n+1}-1} \\ &= \underbrace{x_{2,n-1}x_{2,n-2}\cdots x_{2,0}0}_{n+1}\underbrace{x_{2,2n-1}x_{2,2n-2}\cdots x_{2,n}}_{n}, \\ u_3 &= |2x_2|_{2^{2n+1}-1} \\ &= (\underbrace{x_{2,2n-1}x_{2,2n-2}\cdots x_{2,0}0}_{2n+1}), \\ u_4 &= \left|2^{n+1}x_3+2x_3\right|_{2^{2n+1}-1} \\ &= (\underbrace{x_{3,n-1}x_{3,n-2}\cdots x_{3,0}x_{3,n-1}x_{3,n-2}\cdots x_{3,0}0}_{2n+1}. \end{aligned}$$

Given the binary representation:

$$A = (\underbrace{a_{2n}a_{2n-1}\cdots a_1a_0}_{2n+1}), \quad (27)$$

B_2 can be written as:

$$\begin{aligned} B_2 &= (\underbrace{a_{2n}a_{2n-1}\cdots a_1a_0}_{2n+1}\underbrace{00\cdots0}_{n}) \\ &\quad +(\underbrace{x_{3,n-1}x_{3,n-2}\cdots x_{3,0}}_{n}) \\ &= (\underbrace{a_{2n}a_{2n-1}\cdots a_0x_{3,n-1}x_{3,n-2}\cdots x_{3,0}}_{3n+1}) \end{aligned} \quad (28)$$

In (25), in order to carry out the summation, B_1 and B_3 must be represented with the same number of bits, i.e., $(3n+1)$-bits, as B_2. They can be represented as:

$$\begin{aligned} B_1 &= -x_2 \\ &= -(\underbrace{000\cdots0}_{n+1}\underbrace{x_{2,2n-1}x_{2,2n-2}\cdots x_{2,0}}_{2n}) \\ &= (\underbrace{111\cdots11}_{n+1}\underbrace{\overline{x}_{2,2n-1}\overline{x}_{2,2n-2}\cdots\overline{x}_{2,0}}_{2n}), \end{aligned} \quad (29)$$

$$\begin{aligned} B_3 &= -A \\ &= -(\underbrace{000\cdots0}_{n}\underbrace{a_na_{n-1}\cdots a_0}_{2n+1}) \\ &= (\underbrace{111\cdots11}_{n}\underbrace{\overline{a}_n\overline{a}_{n-1}\cdots\overline{a}_0}_{2n+1}). \end{aligned} \quad (30)$$

IV. MRC Based Conversion Method

Given that the CRT conversion technique presented in the previous section does not cover the entire dynamic range, in this section we present a MRC conversion

approach that is valid for the entire dynamic range for the proposed moduli set.

Theorem 4: Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^n - 1, m_2 = 2^{2n}, m_3 = 2^{2n+1} - 1$, the following holds true:

$$|(m_1)^{-1}|_{m_2} = 2^{2n} - 2^n - 1, \quad (31)$$
$$|(m_2)^{-1}|_{m_3} = 2, \quad (32)$$
$$|(m_1)^{-1}|_{m_3} = 2^{2n+1} - 2^{n+1} - 3. \quad (33)$$

Proof: If it can be shown that $|2^{2n} - 2^n - 1 \times m_1|_{m_2} = 1$, then $2^{2n} - 2^n - 1$ is the multiplicative inverse of m_1 with respect to m_2:

$$\begin{aligned} R_4 &= \left|(2^{2n} - 2^n - 1)(2^n - 1)\right|_{2^{2n}} \\ &= |2^{3n} - 2^{2n} - 2^{2n} + 1|_{2^{2n}} = 1, \end{aligned}$$

thus (31) holds true. Similarly, (32) and (33) can be easily proved. ■

When $k = 3$, (2) becomes

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 \quad (34)$$

Using (31) and (32) in (3), the MRDs a_1, a_2, and a_3 can be represented as

$$a_1 = x_1, \quad (35)$$

$$\begin{aligned} a_2 &= \left|(2^{2n} - 2^n - 1)(x_2 - x_1)\right|_{2^{2n}} \\ &= |2^{2n}x_2 - 2^n x_2 - x_2 - 2^{2n}x_1 + 2^n x_1 + x_1|_{2^{2n}} \\ &= |-2^n x_2 - x_2 + 2^n x_1 + x_1|_{2^{2n}}, \quad (36) \\ a_3 &= \left|((x_3 - x_1)(2^{2n+1} - 2^{n+1} - 3) - a_2)(2)\right|_{2^{2n+1}-1} \\ &= \left|-2^{n+2}x_3 - 4x_3 + 2^{n+2}x_1\right. \\ &\quad \left. + 4x_1 - 2a_2\right|_{2^{2n+1}-1}. \quad (37) \end{aligned}$$

The hardware realization of (34) can be further simplified using the two properties presented in Section III. Let the binary representations of the residues be:

$$\begin{aligned} x_1 &= (x_{1,n-1}x_{1,n-2}...x_{1,1}x_{1,0}), \\ x_2 &= (x_{2,2n-1}x_{2,2n-2}...x_{2,1}x_{2,0}), \\ x_3 &= (x_{3,2n}x_{3,2n-1}...x_{3,1}x_{3,0}). \end{aligned}$$

Given that (36) is represented as

$$a_2 = |v_{1,1} + v_{1,2} + v_{1,3}|_{2^{2n}} \quad (38)$$

where

$$\begin{aligned} v_{1,1} &= |2^n x_1 + x_1|_{2^{2n}} \\ &= (\underbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,0}x_{1,n-1}x_{1,n-2}\cdots x_{1,0}}_{2n}), \\ v_{1,2} &= |-2^n x_2|_{2^{2n}} \\ &= \left|-(\underbrace{x_{2,2n-1}x_{2,2n-2}...x_{2,0}}_{2n}\underbrace{00...0}_{n})\right|_{2^{2n}} \\ &= \underbrace{\bar{x}_{2,n-1}\cdots\bar{x}_{2,0}11\cdots1}_{2n}, \\ v_{1,3} &= |-x_2|_{2^{2n}} \\ &= \underbrace{\bar{x}_{2,2n-1}\bar{x}_{2,2n-2}\cdots\bar{x}_{2,0}}_{2n} \end{aligned}$$

Representing (37) as:

$$a_3 = |v_{2,1} + v_{2,2} + v_{2,3} + v_{2,4}|_{2^{2n+1}-1} \quad (39)$$

where

$$\begin{aligned} v_{2,1} &= \left|-2^{n+2}x_3\right|_{2^{2n+1}-1} \\ &= \left|2^{n+2}(\underbrace{\bar{x}_{3,2n}\bar{x}_{3,2n-1}...\bar{x}_{3,n+2}}_{n-1}\underbrace{\bar{x}_{3,n+1}\bar{x}_{3,n}...\bar{x}_{3,0}}_{n+2})\right|_{2^{2n+1}-1} \\ &= \underbrace{\bar{x}_{3,n+1}\bar{x}_{3,n}...\bar{x}_{3,0}}_{n+2}\underbrace{\bar{x}_{3,2n}\bar{x}_{3,2n-1}...\bar{x}_{3,n+2}}_{n-1}, \\ v_{2,2} &= |-4x_3|_{2^{2n+1}-1} \\ &= \left|2^2\underbrace{\bar{x}_{3,2n}\bar{x}_{3,2n-1}...\bar{x}_{3,0}}_{2n+1}\right|_{2^{2n+1}-1} \\ &= \underbrace{\bar{x}_{3,1}\bar{x}_{3,0}\bar{x}_{3,2n}\bar{x}_{3,2n-1}...\bar{x}_{3,2}}_{2n+1}, \\ v_{2,3} &= \left|2^{n+2}x_1 + 4x_1\right|_{2^{2n+1}-1} \\ &= \left|(\underbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,0}}_{n})\underbrace{00\cdots0}_{n+2}\right. \\ &\quad \left. + \underbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,0}00}_{n+2}\right|_{2^{2n+1}-1} \\ &= (0\underbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,0}x_{1,n-1}x_{1,n-2}\cdots x_{1,0}}_{2n+1}, \\ v_{2,4} &= |-2a_2|_{2^{2n+1}-1} \\ &= \underbrace{\bar{a}_{2,2n-1}\bar{a}_{2,2n-2}\cdots\bar{a}_{2,0}1}_{2n+1} \end{aligned}$$

(34) can be simplified as follows

$$\begin{aligned} X &= a_1 + a_2 m_1 + a_3 m_1 m_2 \\ &= a_1 + 2^n a_2 + 2^{3n} a_3 - 2^{2n} a_3 - a_2. \quad (40) \end{aligned}$$

The hardware realization of (40) can be simplified as follows

$$X = a_4 + a_5 + a_6 \tag{41}$$

where

$$\begin{aligned} a_4 &= a_1 + 2^n a_2 + 2^{3n} a_3 \\ &= \underbrace{(a_{1,n-1}a_{1,n-2}\ldots a_{1,0})}_{n} + \underbrace{(a_{2,2n-1}a_{2,2n-2}\ldots a_{2,0}}_{2n}\underbrace{00\ldots0)}_{n} \\ &\quad + \underbrace{(a_{3,2n}a_{3,2n-1}\ldots a_{3,0}}_{2n+1}\underbrace{00\ldots0)}_{3n} \\ &= \underbrace{a_{3,2n}\ldots a_{3,0}a_{2,2n-1}\ldots a_{2,0}a_{1,n-1}\ldots a_{1,0}}_{5n+1}, \end{aligned} \tag{42}$$

$$\begin{aligned} a_5 &= -2^{2n} a_3 \\ &= -(\underbrace{00\ldots0}_{n}\underbrace{a_{3,2n}\ldots a_{3,0}}_{2n+1}\underbrace{00\ldots0}_{2n}) \\ &= \underbrace{11\ldots1}_{n}\underbrace{\overline{a}_{3,2n}\ldots\overline{a}_{3,0}}_{2n+1}\underbrace{11\ldots1}_{2n} \end{aligned} \tag{43}$$

$$\begin{aligned} a_6 &= -a_2 \\ &= -\underbrace{(a_{2,2n-1}a_{2,2n-2}\ldots a_{3,0})}_{2n} \\ &= -(\underbrace{00\ldots0}_{3n+1}\underbrace{a_{2,2n-1}a_{2,2n-2}\ldots a_{2,0}}_{2n}) \\ &= \underbrace{11\ldots1}_{3n+1}\underbrace{\overline{a}_{2,2n-1}\ldots\overline{a}_{2,0}}_{2n} \end{aligned} \tag{44}$$

V. Descriptions of the Proposed Hardware Structures

In this section, the Hardware structures that result from the two RNS to Binary conversion methods previously described are presented.

A. *CRT Based Approach*

We start by describing the structure that results from the CRT approach, which is based on (24) and (25). As it can be seen from Figure 1, u_0, u_1, u_2, u_3, and u_4 are added by Carry Save Adders (CSAs) with end-around carries (EACs) generating the values s_3 and c_3. These values must be added modulo $2^{2n+1}-1$ in order to obtain A, i.e., with a one's complement adder, namely a Carry Propagate Adder (CPA) with EAC. B_2 is easily obtained by concatenating the operand x_3 with the n-bit left shift of A. This concatenation does not require any additional hardware. The three operands B_1, B_2, and B_3 are added using a CSA with EAC. It should be noted that in order to make B_1 and B_3 $(3n+1)$-bit numbers, 1's are appended to the result of complementations, as given in (29) and (30). Thus, the addition of the most significant $(2n+1)$-bits performed in this CSA can be performed

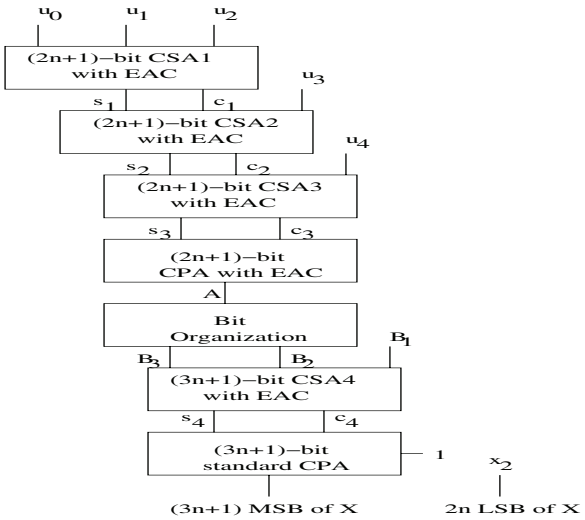

Figure 1. Proposed Reverse Converter I

by Half Adders (HAs). In addition, since these HAs have two inputs equal to 1, the final one's complement adder will always generate an end-around carry. Taking this into consideration the one's complement adder can be reduced to a normal CPA with a constant carry-in equals to 1. The final result, computed from (9) is obtained simply by a shift and a concatenation operation, which do not require any additional hardware.

B. *MRC Based Approach*

The hardware structure that results from the MRC approach is depicted in Figure 2 and it is based on (38), (39), and (41). In (38), the operands are added using a $2n$-bit CSA with EAC, generating s_1 and c_1, which are added by a $2n$-bit CPA with EAC. Similarly, 2-levels of CSAs, followed by a $(2n+1)$-bit one's complement adder, are used to add the 4-operands in (39). In Figure 2, Bit org1 depicts the bit re-organization used to obtain $v_{2,4}$. This is simply the result of complementations of 1-bit right shift of a_2. However, a_4 is simply obtained by concatenating the three operands a_3, a_2, and a_1, which are $(2n+1)$-bit, $2n$-bit, and n-bit, respectively. This concatenation is labeled as Bit org3 and does not require any additional hardware. In order to perform the addition described in (41), which is identical to what we performed on the CRT conversion structure, the two operands a_5 and a_6 must be expanded to $(5n+1)$-bit numbers since a_4 is a $(5n+1)$-bit number. The final adder can be simplified to a standard CPA (CPA3) with a constant carry-in equals to 1, in a similar way to what has been done for the CRT.

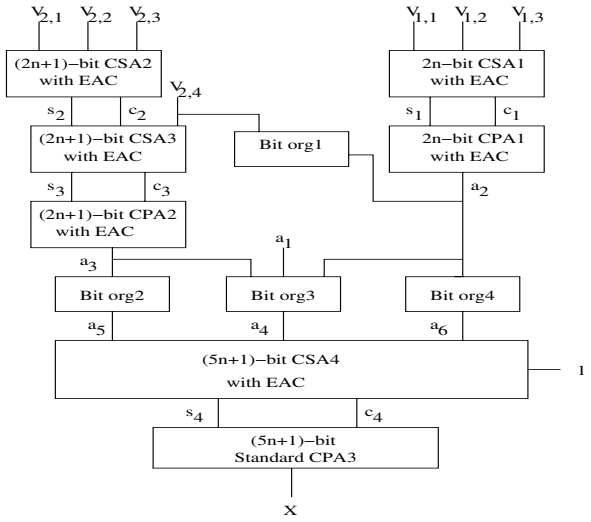

Figure 2. Proposed Reverse Converter II

Table I
AREA AND DELAY COMPARISONS

	FA	HA	Delay
Converter [1]	$2.25D-0.25$	$0.5D+1.5$	$(2.5D+2.5)t_{FA}$
Converter [3]	$2.25D-0.25$	$1.25D+2.75$	$(1.75D+5.25)t_{FA}$
Proposed CI	$2.4D-0.4$	$0.4D+3.6$	$(1.4D+5.6)t_{FA}$
Proposed CII	$3D-1$	$0.8D+2.2$	$(2.6D+4.4)t_{FA}$

VI. PERFORMANCE EVALUATION

In order to evaluate the performance of the proposed reverse converters, we compare them with similar best known state of the art reverse converters in [1] and [3]. In order to properly perform this evaluation, for the same dynamic range, the reverse conversion structures are theoretically analyzed, in terms of gates, and experimentally by implementating them on Application Specific Integrated Circuit (ASIC) on a 0.13μm Standard Cell technology from UMC [11].

We note that in Table I, D stands for the dynamic range. From the theoretical analysis, presented in Table I, it is clear that the proposed CRT technique is clearly more advantageous that the MRC approach, both in terms of area and delay. This suggests that even though the proposed CRT based reverse converter slightly reduces the dynamic range as shown, by (21) and (22), it is preferable to the proposed MRC based reverse converter. The ASIC implementation results presented in Table II, validate this. These results suggest that the CRT based reverse converter is able to improve the conversion computation time by about 20% while reducing the required hardware area by more than 20%, regarding the proposed MRC based reverse converter. For this reason, in the following, comparison with related state of the art reverse converters will only be made with the proposed CRT based reverse converter. When compared with the related art in [1] and in [3], both using the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set, the theoretical analysis presented in Table I suggests that the converter in [3] is slower and with higher area requirements than the one herein proposed. This is expected since the moduli set herein proposed is identical to the moduli set in [3] with a binary channel twice the size. This implies that for the same dynamic range the size of n can be smaller, thus units with smaller length are used. When compared with the converter in [1] the gain in both area and delay is even more significant. When analyzing the ASIC implementation results, depicted in Figures 3 and 4 and Table II, the performance and area advantages of the proposed CRT based reverse converter are confirmed to be significant when compared with the other reverse converters.

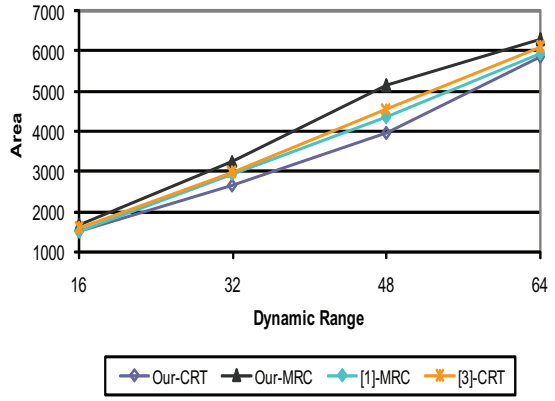

Figure 3. The Converters Area Comparisons

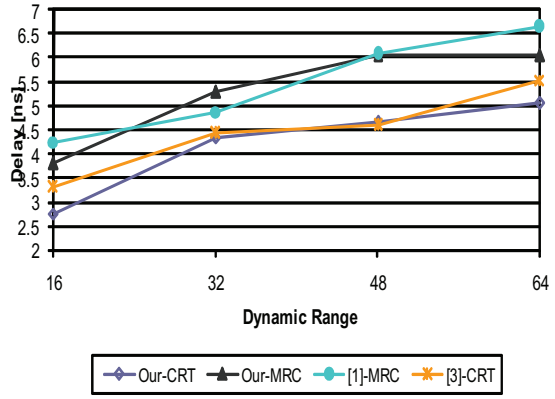

Figure 4. Conversion Delay [ns] Comparisons

Table II
IMPLEMENTATION RESULTS

Dynamic Range	Area-CI	Area-CII	Area [1]	Area [3]	Delay-CI	Delay-CII	Delay [1]	Delay [3]
16	1492	1638	1484	1567	2.76	3.80	4.25	3.30
24	2358	2677	2096	2228	3.78	4.99	5.13	4.13
32	2654	3262	2934	2964	4.35	5.69	5.22	4.66
48	3945	5144	4351	4573	5.26	6.40	6.08	4.43
64	5856	5944	5944	6096	5.06	6.06	6.81	5.54

Experimental results indicate that the proposed CRT based reverse converter is 21% faster and requires 14% less area than the proposed MRC based approach. When compared with the MRC based revere converter proposed in [1], the delay is improved by 23% and the area is reduced by 3%. Comparing the proposed CRT reverse converter with the reverse converter in [3] for the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set, a delay gain of 4% and area reduction of 6% is achieved. Combining these two values, the reverse conversion efficiency is improved by 9%, using the Area/Time (AT) metric.

Comparing the proposed MRC based converter with the one in [1], the results are identical. However it should be noted that both of these converters underperform when compared with the CRT based converters. Therefore, it can be inferred that the CRT is a better approach to design reverse converters for this class of moduli sets.

From these results it can be concluded that the proposed moduli set and respective CRT based reverse converter are able to improve the performance of RNS computation.

VII. CONCLUSIONS

In this paper, two novel reverse converters for the new moduli set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$ are proposed. One of the proposed reverse converters is based on the traditional CRT while the other is based on the MRC. The CRT based reverse converter is better in terms of both area and delay, when compared to the MRC based reverse converter. The MRC based reverse converter is also useful because it covers the entire dynamic range whereas the CRT based converter does not. The proposed reverse converters are memoryless and adder based and can be easily realized in VLSI circuits. We performed both theoretical and experimental evaluation of our proposal. The theoretical analysis shows the advantages of our scheme, which is supported by the experimental results. The synthesis results for a 0.13-μm CMOS standard cell-based technology are obtained and suggest that, for the same dynamic range, the proposed CRT based converter achieves about 23% delay reduction with more than 3% area reduction, when compared to the converter in [1]. Additionally, the proposed CRT based converter is about 6% faster with about 4% area reduction when compared with the existing CRT based converter [3]. Finally, the proposed CRT based converter outperforms all the existing converters in terms of both area and delay.

REFERENCES

[1] K. Navi A.S. Molahosseini and M.K. Rafsanjani. A new residue to binary converter based on mixed-radix conversion. *3rd International Conference On Information and Communication Technologies: From Theory to Applications (ICTTA 2008), pp. 1-6, April*, 2008.

[2] R. Chaves and L. Sousa. Improving residue number system multiplication with more balanced moduli sets and enhanced modular arithmetic structures. *IET Comp. Digital Tech., Vol. 5, No.1, pp.472-480, Sept.*, 2007.

[3] K.A. Gbolagade, R. Chaves, L. Sousa, and S.D. Cotofana. An improved RNS reverse converter for the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set. *Submitted to IEEE Int. Symposium on Circuits and Systems (ISCAS 2010), Paris, France*, 2010.

[4] K.A. Gbolagade and S.D. Cotofana. Residue number system operands to decimal conversion for 3-moduli sets. *Proceedings of 51st IEEE Midwest Symposium on Circuits and Systems, pp.791-794, Knoxville, USA, August*, 2008.

[5] A.A. Hiasat and H.S. Abdel-AtyZohdy. Residue-to-binary arithmetic converter for the moduli set $\{2^k, 2^k-1, 2^{k-1}-1\}$. *IEEE Transactions on Circuits and Systems-II Analog and Digital Signal Processing, Vol.45, No. 2, pp. 204-209, Feb.*, 1998.

[6] P.V.A. Mohan. Rns-to-binary converter for a new three-moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$. *IEEE Trans. on Circuits and Systems-II: Express briefs, Vol. 54, No.9, pp. 775-779, September*, 2007.

[7] A.S. Molahosseini and K. Navi. New arithmetic residue to binary converters. *International Journal of Computer Sciences and Engineering Systems, Vol. 1, No.4, pp. 295-299, October*, 2007.

[8] M. Sheu S. Lin and C. Wang. Efficient vlsi design of residue to binary converter for the moduli set $\{2^n, 2^{n+1}-1, 2^n-1\}$. *IEICE Trans. INF. and SYST., Vol. E91-D, No.7, pp. 2058-2060, July*, 2008.

[9] N. Szabo and R Tanaka. *Residue Arithmetic and its Application to Computer Technology*. MC-Graw-Hill, New York, 1967.

[10] A.P. Vinod and A.B. Premkumar. A memoryless residue to binary converter for the 4-superset $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$. *Journal of Circuits, Syst. and Computers, Vol. 10, pp. 85-99,*, 2000.

[11] Virtual Silicon Technology Inc. *UMC High Density Standard Cells Library - 0.13μm CMOS process*, v2.3 edition, December 1999.

[12] Y. Wang. Residue-to-binary converters based on new chinese remainder theorems. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 47, No.3, pp. 197-205, March*, 2000.