A New Efficient RNS Reverse Converter for the 4-Moduli Set
\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}

Edem K. Bankas, Kazeem A. Gbolagade

Abstract—In this paper, we propose a new efficient reverse converter for the 4-moduli set \{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\} based on a modified Chinese Remainder Theorem and Mixed Radix Conversion. Additionally, the resulting architecture is further reduced to obtain a reverse converter that utilizes only carry save adders, a multiplexer and carry propagate adders. The proposed converter has an area cost of \((12n + 2)\ FA\)s and \((5n + 1)\ HAs\) with a delay of \((9n + 6)\ t_{FA} + t_{MUX}\). When compared with state of the art, our proposal demonstrates to be faster, at the expense of slightly more hardware resources. Further, the Area-Time square metric was computed which indicated that our proposed scheme outperforms the state of the art reverse converter.

Keywords—Modified Chinese Remainder Theorem, Mixed Radix Conversion, Reverse Converter, Carry Save Adder, Carry Propagate Adder.

I. INTRODUCTION

RESIDUE Number System (RNS) offers very useful applications in addition, subtraction, and multiplication dominated arithmetic operations, for example, digital filtering, fast Fourier Transform (FFT), image processing etc. This is due to the inherent properties of RNS such as parallelism, modularity, fault tolerance, and carry free operations [1], [16]. Additionally, it has been shown that RNS based processors can even reduce power dissipation in very large scale integrated circuits system design [2]. However, RNS has not found a wide spread usage in general purpose computing due to the following difficult RNS arithmetic operations: magnitude comparison, sign detection, overflow detection, moduli selection, reverse and forward conversions.

Moduli selection is one of the greatest RNS challenges. This is because, the speed and complexity of the resulting RNS architecture is dependent on the form and the number of moduli set. It has been well established that powers-of-two moduli sets simplify the required arithmetic operations and generate efficient hardware implementations of the RNS architecture [3].

Many moduli sets with their accompanying reverse converters have been proposed. Among these are \{2^n, 2^n - 1, 2^{2n+1} - 1\} [5], [6], [7], \{2^n, 2^n - 1, 2^{2n-1} - 1\} [8], \{2^{2n+1} - 1, 2^n, 2^n - 1\} [17], and the most popular length-3 moduli set \{2^n, 2^n + 1, 2^n - 1\} [4], [12]. In recent years, due to the increasing demand for some applications that require larger dynamic range and increased parallelism, some length 4 moduli sets such as: \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\} [10], \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\} [9], \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\} [11] and \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\} [14] generally referred to as 4-moduli supersets of \{2^n, 2^n + 1, 2^n - 1\} have been investigated with their respective reverse conversion algorithms. Some of these conversion algorithms use the Chinese Remainder Theorem (CRT), a combination of CRT and Mixed Radix Conversion (MRC) algorithms [15], a combination of New CRT and MRC [14]. In [14], Molahosseini et al. (2010), introduced the two new 4-moduli sets \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\} and \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} + 1\}. Their proposed reverse converter for \{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\} moduli set was obtained from New CRT II with better performance and hardware requirement when compared with other equivalent 5n bit dynamic range state of the art reverse converters. The area, speed and the hardware complexity of the resulting reverse converter for the 4-moduli set \{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\} in [14] can be further reduced and improved.

In this paper, we propose a novel hybridized reverse converter for the moduli set \{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}. First, the proposed algorithm is based on a hybridization of a modified CRT and MRC methods, resulting in a two level design. In the first level, the equivalent weighted number of the residues \(x_1, x_2, x_3\) is obtained by using the algorithm presented in [12] for the popular moduli set \{2^n, 2^n + 1, 2^n - 1\}. Next, the resulting weighted number equivalent from the first level is hybridized with the fourth residue \(x_4\) using MRC with respect to the composite moduli set \{2^{2n} - 2^n, 2^{2n+1} - 1\}.

The resulting architecture is further simplified in order to obtain a reverse converter that utilizes only Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs). Theoretically speaking, the proposed converter outperforms the state of the art reverse converter.

The rest of the paper is structured as follows. Section II provides a brief background information. In Section III, the proposed algorithm is formulated. Section IV describes the hardware implementation of the proposed algorithm and Section V evaluates the performance of the proposed scheme. Finally, the paper is concluded in Section VI.
II. BACKGROUND

Given a moduli set \( \{m_i\}_{i=1}^k \), the residues \((x_1, x_2, \ldots, x_k)\) can be converted into the corresponding decimal number \(X\) in the following ways: First, by the use of the well known CRT, which is given as \([1]\):

\[
X = \sum_{i=1}^{k} m_i \left[ M_i^{-1} \right]_{m_i} x_i \tag{1}
\]

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 \( m_i \).

Second, the MRC, can also be used. Suppose we have a residue number representation \((x_1, x_2, \ldots, x_k)\) with respect to the moduli set \( \{m_i\}_{i=1}^k \) and Mixed Radix Digits (MRDs), \( \{a_i\}_{i=1}^k \), the decimal equivalent of the residues can be computed as follows \([1]\):

\[
X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \ldots + a_n m_1 m_2 \ldots m_{k-1}, \tag{2}
\]

where the MRDs are given as

\[
a_1 = x_1 \tag{3}
\]

\[
a_2 = (x_2 - a_1) \left[ m_1^{-1} \right]_{m_2} \tag{4}
\]

\[
a_3 = ((x_3 - a_1) \left[ m_1^{-1} \right]_{m_2} - a_2) \left[ m_2^{-1} \right]_{m_3} \tag{5}
\]

\[\vdots\]

\[
a_k = \left( ((x_k - a_1) \left[ m_1^{-1} \right]_{m_2} - a_2 \right) \left[ m_2^{-1} \right]_{m_3} - \ldots - a_{k-1} \right) \left[ m_{k-1}^{-1} \right]_{m_k} \tag{6}
\]

Third, the New CRT I proposed in \([13]\) could also be used to convert RNS to its decimal equivalent. Given a 4-moduli set \( \{P_1, P_2, P_3, P_4\} \), the number \(X\) can be converted from its residue representation \((x_1, x_2, x_3, x_4)\) as follows:

\[
X = x_1 + P_1 \left[ k_1 (x_2 - x_1) + k_2 P_2 (x_3 - x_2) + k_3 P_3 P_2 (x_4 - x_3) \right]_{P_2 P_3 P_4} \tag{7}
\]

where

\[
\begin{align*}
|k_1 P_1|_{P_2 P_3} P_4 &= 1 \tag{8} \\
|k_2 P_2|_{P_3} P_4 &= 1 \tag{9} \\
|k_3 P_3 P_2|_{P_4} &= 1 \tag{10}
\end{align*}
\]

Similarly, the New CRT II \([14]\) defined for a 4 moduli set \( \{P_1, P_2, P_3, P_4\} \), the number \(X\) can be computed from its corresponding residues \((x_1, x_2, x_3, x_4)\) by:

\[
X = Z + P_1 P_2 k_1 (Y - Z)_{P_3 P_4} \tag{11}
\]

\[
Z = x_1 + P_1 \left[ k_2 (x_2 - x_1) \right]_{P_2} \tag{12}
\]

\[
Y = x_3 + P_3 \left[ k_3 (x_4 - x_3) \right]_{P_4} \tag{13}
\]

where \(k_1, k_2, \text{ and } k_3\) are the multiplicative inverses.

III. PROPOSED ALGORITHM

Given \( \{2^n, 2^n + 1, 2^{n-1}, 2^{2n+1} - 1\} \) as the 4-moduli set with corresponding residues \((x_1, x_2, x_3, x_4)\), the proposed algorithm which consists of two levels is formulated using the following theorems:

Given the moduli set \( \{m_1, m_2, m_3\} \) with \(m_1 = 2^n, m_2 = 2^n + 1, \text{ and } m_3 = 2^{n-1}\), the decimal equivalent of the residue numbers \((x_1, x_2, x_3)\) is computed as:

\[
A = m_1 \left[ A \right]_{m_1} + x_1 \tag{14}
\]

where

\[
A_{m_1} = [u_1 + u_2 + \frac{1}{m_2} u_3]_{x_1} \quad \text{and} \quad u_1 = (\frac{1}{m_1} - m_2 + 1) x_1, \quad u_2 = (\frac{m_1}{m_2} - 1) x_2 \quad \text{and} \quad u_3 = \frac{m_2}{m_3} x_3
\]

\[
\frac{A}{m_1} \left[ A \right]_{m_1} \text{ can be represented as:}
\]

\[
\alpha = \left[ \frac{A}{m_1} \right] = [u'_1 + u_{13} + u'_2 + u_{23}]_{2^{n-1}} \tag{15}
\]

where

\[
u'_1 = \prod_{k=1}^{n} \pi_{1,n-k+1} \text{ with respect to } x_1, \text{ and } \pi_{1,n} = x_1 \nu_{13} = [x_1]_{2^{n-1}} = \prod_{k=1}^{n} \pi_{1,n-k} \tag{16}
\]

\[
u'_3 = \prod_{k=1}^{n+1} \pi_{3,n-k+1} \text{ with respect to } x_3 \nu'_3 = \prod_{k=1}^{n+1} \pi_{3,n-k} \tag{17}
\]

\[
u'_2 = \prod_{k=1}^{n+1} \pi_{2,n-k+1} \text{ with respect to } x_2 \nu'_2 = \prod_{k=1}^{n+1} \pi_{2,n-k} \tag{18}
\]

\[
u_{23} = [x_2 - x_3]_{2^{n-1}} = \prod_{k=1}^{n+1} \pi_{2,n-k} \tag{19}
\]

\[
u_{23} = [x_2 - x_3]_{2^{n-1}} = \prod_{k=1}^{n+1} \pi_{2,n-k} \tag{20}
\]

\[\textbf{Proof:}\]

This theorem has been proved in \([12]\).
Proof: If it can be demonstrated that
\[
[(2^n(2^{2n} - 1)) \times (2^{2n+1} - 2^{n+2} - 1)]_{2^{2n+1} - 1} = 1,
\]
then \(2^{2n+1} - 2^{n+2} - 1\) is the multiplicative inverse of \((2^n(2^{2n} - 1))\) with respect to \(2^{2n+1} - 1\):
\[
\frac{[(2^n(2^{2n} - 1)) \times (2^{2n+1} - 2^{n+2} - 1)]_{2^{2n+1} - 1}}{2^{2n+1} - 1 - 3 \times 2^n + 2^n} = 1.
\]

Given \(\{2^n, 2^n + 1, 2^n + 1, 2^{2n+1} - 1\}\) as the superset, the decimal equivalent \(X\) of the RNS number \((x_1, x_2, x_3, x_4)\) can be computed as:
\[
X = A + a_4(2^{3n} - 2^n)
\]
where \(a_4 = [(x_4 - A)k]_{2^{2n+1} - 1}\), \(A\) and \(k\) are given by (14) and (21) respectively.

Proof: Given the moduli set \(\{m_1 = 2^n, m_2 = 2^n + 1, m_3 = 2^n - 1, m_4 = 2^{2n+1} - 1\}\) with residues \((x_1, x_2, x_3, x_4)\), using the MRC, its decimal equivalent is computed using (2) for \(\{a_i\}\)\_{\text{val}}. Now, by considering the moduli set \(\{2^n, 2^n + 1, 2^n - 1\}\) with residues \((x_1, x_2, x_3)\), its decimal equivalent is computed by using (14). Therefore, \(A = a_1 + a_2m_1 + a_3m_3m_2\). Next, we consider the composite set \(\{2^{3n} - 2^n, 2^{2n+1} - 1\}\) with residues \((A, x_4)\). For a two moduli set \(\{m_A, m_B\}\), (2) becomes
\[
X = A + a_4m_1m_2m_3 = A + a_4(2^{3n} - 2^n)
\]

In order to reduce the hardware complexity, we use the following properties [11] to simplify (23).

Property 1: The multiplication of a residue number by \(2^k\) in modulo \((2^p - 1)\) is computed by \(k\) bit circular left shifting.

Property 2: A negative number in modulo \((2^p - 1)\) is calculated by subtracting the number in question from \((2^p - 1)\). In binary representation, the ones complement of the number gives the result.

Let the residues \((x_1, x_2, x_3, x_4)\) have binary representation as follows:
\[
x_1 = \underbrace{(x_1, x_1, x_1, \ldots, x_1)}_{n}x_1 \quad (24)
\]
\[
x_2 = \underbrace{(x_2, x_2, x_2, \ldots, x_2)}_{n+1}x_2 \quad (25)
\]
\[
x_3 = \underbrace{(x_3, x_3, x_3, \ldots, x_3)}_{n}x_3 \quad (26)
\]
\[
x_4 = \underbrace{(x_4, x_4, x_4, \ldots, x_4)}_{2n+1}x_4 \quad (27)
\]

Note that:
\[
a_4 = [(x_4 - A)(2^{2n+1} - 2^{n+2} - 1)]_{2^{2n+1} - 1} \quad (28)
\]
\[
a_4 = [(-2^{n+2})]_{2^{2n+1} - 1} = |a_{41} + a_{42}]_{2^{2n+1} - 1} \quad (30)
\]

(30) can then be simplified as follows:
\[
a_{41} = [(-2^{n+2})x_4]_{2^{2n+1} - 1} = 2^{-n-1}x_4 \quad (31)
\]

For \(a_{42}\), it is interesting to note that \(A\) is a \(3n\) bit number and is represented as \(A_{3n-1}A_{3n-2} \ldots A_0\). Therefore, \(a_{42} = 2^{n+2}(2^{2n+1}A_{3n-1}A_{2n+1} + A_{2n}A_{2n-1} \ldots A_0)\) becomes
\[
a_{42}' + a_{42}'' \quad (34)
\]
Thus, in order to obtain \(X\) in (23), \(a_4\) which is a \(2n + 1\) bit number can be represented as \(a_{42}a_{42}' + a_{42}''\). So
\[
X = A + 2^{3n}a_4 - 2^n a_4 \quad (35)
\]

where
\[
2^{3n}a_4 = \underbrace{a_{42}a_{42}a_{42} \ldots a_{42}}_{3n} \quad (36)
\]
and
\[
2^n a_4 = \underbrace{a_{42}a_{42}a_{42} \ldots a_{42}}_{3n} \quad (37)
\]

Note that (37) according to (35) becomes:
\[
\underbrace{1 \ldots 1_{2n+1}}_{n+1} \quad (38)
\]

IV. HARDWARE REALIZATION

The hardware implementation of the proposed reverse converter for the moduli set \(\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}\) is based on (14), (15), (34), and (35). The hardware architecture consists of two levels, the first level is based on a modified CRT and utilizes (14) and (15). Implementation of (15) requires a five operand modulo \(2^{2n} - 1\) adder, where \(2n\) bit numbers \(u_1, u_13, u_3, u_2, u_{23}\) are added with three
levels of $2^n$ bit Carry Save Adder (CSA) with End Around Carry (EAC) followed by a $2^n$ bit Carry Propagate Adder (CPA) with a carry in of 1. It must be noted that, two operands in this level have $n$ and $n-1$ most significant bit input equal to 1. This will result in the final one’s complement adder always generating an end around carry. This phenomenon demonstrates that, the one’s complement adder can be reduced to a normal CPA with a constant carry-in equal to 1. This therefore makes the delay $t_{CPA}(2^n)$. Note also that, the computation of (14) requires no additional hardware since the desired result is obtained by concatenating $n$ bit number $x_1$ with $\alpha$.

In the second level, (34) which consists of three $2n+1$ bit numbers $a_{41}, a_{42}', a_{42}$ and $a_{42}$ are added with two levels of $2n+1$ bit CSA with EAC followed by a $(2n+1)$ bit CPA. The addition of the operands in (34) modulo $(2^{2n+1} - 1)$ can be accelerated with anticipated computation. Thus, we compute $s_4 + c_4$ for both $c_{4i} = 0$ and $c_{4i} = 1$ and the right result is selected with a multiplexer. This process is concluded with the computation of (35) which requires $(5n+1)$ bit subtrahendar implemented by $(5n+1)$ bit regular CPA with a constant carry in of 1. Since $A$ is a $3n$ bit number, the computation of (35) requires no additional hardware as it is easily obtained by concatenating $A$ with the results of $(5n+1)$ bits of sum of (36) and (37). The proposed hardware structure is depicted by Fig. 1.

V. PERFORMANCE ANALYSIS

The performance of the proposed reverse converter is evaluated theoretically in terms of area cost and conversion delay. We compare our proposal with equivalent state of the art reverse converters presented in [14] and [15]. The hardware utilization of our proposal and that of state of the art is computed in terms of Full Adders (FAs) and Half Adders (HAs).

The total delay of our reverse converter is the sum of all the delay of the two levels mentioned above. For the first level, the delay is $(2n + 3)t_{FA}$, while it is $(7n + 3)t_{FA} + t_{MUX}$ in the second level. Therefore, the total delay of the proposed converter is $(9n + 6)t_{FA} + t_{MUX}$ which is faster than that presented in both [14] and [15]. It is clear from Table I that our proposal is faster than the state of the art presented in both [14] and [15] at the expense of slightly more area resources. In order to obtain an adequate comparison, the Area-Time square $(Δτ^2)$ efficiency metric was used. The $Δτ^2$ metric suggests that our proposed scheme is more efficient than the state of the art.

<table>
<thead>
<tr>
<th>Converter</th>
<th>Area Cost in HA($Δ$)</th>
<th>Delay($τ$)</th>
<th>$Δτ^2 \approx$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FA</td>
<td>$8n + 2$</td>
<td>$12n + 5$</td>
<td>$3024n^2$</td>
</tr>
<tr>
<td>HA</td>
<td>$10n + 5$</td>
<td>$13n + 1$</td>
<td>$4963n^2$</td>
</tr>
<tr>
<td>Proposed</td>
<td>$12n + 5$</td>
<td>$13n + 1 + (9n + 6)t_{FA} + t_{MUX}$</td>
<td>$2349n^2$</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

In this paper, we proposed a new efficient reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ based on a modified CRT and MRC. Additionally, we simplify further the resulting architecture in order to obtain a reverse converter that utilizes only CSAs, a multiplexer and CPAs. The proposed converter is purely adder based and memoryless. Theoretically speaking, our proposal outperforms equivalent known state of the art reverse converter in terms delay at the expense of slightly more hardware resources. Further, the Area-Time square metric was computed which indicated that our proposed scheme outperforms the state of the art reverse converter.

REFERENCES


[9] P.V.A. Mohan, A.B. Premkumar. RNS-to-Binary Converters for two Four moduli set \( \{2^n − 1, 2^n, 2^n + 1, 2^n+1\} \) and \( \{2^n − 1, 2^n, 2^n + 1, 2^n+1\} \). IEEE Transactions on Circuits and Systems -I. Regular papers. Vol. 54, No.6, June, 2007.


