An Improved RNS Reverse Converter for the 
\(\{2^{2n+1} - 1, 2^n, 2^n - 1\}\) Moduli Set

K.A. Gbolagade¹, ², R. Chaves³, L. Sousa³, S.D. Cotofana¹

1. Computer Engineering Lab., Delft University of Technology, Delft, The Netherlands,
2. University for Development Studies, Navrongo, Ghana,

Abstract—In this paper, we propose a novel high speed memoryless reverse converter for the moduli set \(\{2^{2n+1} - 1, 2^n, 2^n - 1\}\). First, we simplify the traditional Chinese Remainder Theorem in order to obtain a reverse converter that only requires arithmetic mod-(2^{2n+1} - 1). Second, we further improve the resulting architecture to obtain a purely adder based reverse converter. The proposed converter has a critical path delay of \((7n + 7)\) Full Adders (FA) while the best state of the art converter for this moduli set requires \((10n + 5)\) FA on the critical path. To validate these results, the converters are implemented in a Standard Cell 0.18-\(\mu\)m CMOS technology and the results assert that, on average, the proposed converter achieves about 19% delay reduction at the expense of less than 3% area increase.

I. INTRODUCTION

Residue Number Systems (RNS) have significant advantages over conventional binary number systems. This is due to their inherent features, such as carry free operations, parallelism, modularity, and fault tolerance. RNS have been widely applied in Digital Signal Processing (DSP) applications [5]. However, despite all these advantages, RNS have not found a widespread usage in general purpose processors since sign detection, magnitude comparison, overflow detection, and division are rather difficult to perform. Several solutions for these problems, which rely heavily on RNS to binary conversion, have been proposed [5]. This is one of the major reasons why building efficient RNS to binary converters has become an important research topic.

For a successful RNS utilization, moduli set choice and data conversion are critical, in particular the RNS to binary conversion (reverse conversion). Moduli set choice is an important issue since the complexity and the speed of the resulting conversion algorithm depend on the chosen moduli set. Several structures have been proposed to perform the reverse conversion for different moduli sets, e.g., \(\{2^n, 2^n - 1, 2^n + 1\}\) [1], \(\{2^n, 2^{n+1} - 1, 2^n - 1\}\) [3], [2]. In [3], 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 \(\{2^n - 1, 2^n, 2^n + 1, 2^n - 1\}\) proposed in [6]. The motivation for this is related to the fact that the modulo \((2^n + 1)\)-type arithmetic is more complex and degrades the entire RNS performance, both in terms of area cost and conversion delay. However, the moduli set \(\{2^{n+1} - 1, 2^n, 2^n - 1\}\), which is able to utilize fast modulo operations, is insufficient for applications requiring larger dynamic ranges. Consequently, the moduli set \(\{2^{2n+1} - 1, 2^n, 2^n - 1\}\) was proposed in [4] together with a reverse converter based on Mixed Radix Conversion (MRC).

In this paper, a novel and more efficient reverse converter for the \(\{2^{2n+1} - 1, 2^n, 2^n - 1\}\) moduli set is proposed. First, we simplify the traditional Chinese Remainder Theorem (CRT) and obtain a new converter that only requires mod-(2^{2n+1} - 1) operations. Further simplifications result in a simple and more efficient hardware structure, composed of only Carry Save Adders (CSAs) with end-around carries (EACs) and two Carry Propagate Adders (CPAs).

II. PROPOSED ALGORITHM

The proposed algorithm is described using the following theorems:

**Theorem 1:** 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 holds true:

\[
\begin{align*}
|(m_1 m_2)^{-1}|_{m_3} &= 1, \\
|(m_1 m_3)^{-1}|_{m_2} &= 1, \\
|(m_2 m_3)^{-1}|_{m_1} &= 2^{2n+1} - 2^{n+2} - 3.
\end{align*}
\]

**Proof:** It can be easily shown that

\[
\begin{align*}
|1 \times (m_1 m_2)|_{m_3} &= 1, \\
|1 \times (m_1 m_3)|_{m_2} &= 1, \\
|(2^{2n+1} - 2^{n+2} - 3) \times (m_2 m_3)|_{m_1} &= 1.
\end{align*}
\]
The following important relations are used in the subsequent theorem: 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 holds true:

\[
\begin{align*}
m_1 &= 2m_2m_2 - 1, \\
m_2 &= m_3 + 1.
\end{align*}
\]

**Theorem 2:** The decimal equivalent of the residues \((x_1, x_2, x_3)\) with respect to the moduli set \{m_1, m_2, m_3\} in the form \(\{2^{2n+1} - 1, 2^n, 2^n - 1\}\), assuming \(X \in [0, \prod_{i=1}^3 m_i - m_3^2]\), can be computed as follows:

\[
X = m_2 \left\lfloor \frac{X}{m_2} \right\rfloor + x_2,
\]

\[
\left\lfloor \frac{X}{m_2} \right\rfloor = x_3 - x_2 + m_3 \left\lfloor -2^{n+2} - 2 \right\rfloor x_1 + \]
\[2m_2x_2 + 2m_2x_3 + 2x_3|_{m_1}
\]

From (7), it can be seen easily that (14) is the same as \(\left\lfloor \frac{X}{m_2} \right\rfloor = x_3 - x_2 + m_3 \left\lfloor -2^{n+2} - 2 \right\rfloor x_1 + \]
\[2m_2x_2 + 2m_2x_3 + 2x_3|_{m_1}
\]
Therefore, the numbers within the interval \([0, M - (m_3)^2]\) require no corrective addition and thus, (7) holds true.

We can further reduce the hardware complexity of the reverse converter by simplifying (7) using the following two properties:

**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 \((2^t - 1)\) of a negative number is equivalent to the one’s complement of the number, which is obtained by subtracting the number from \(2^t - 1\).

Suppose that (7) is written as:

\[
\frac{X}{m_2} = x_3 - x_2 + 2^n A - A,
\]

\[
A = |u_0 + u_1 + u_2 + u_3 + u_4|_{2^n+1-1}.
\]

For simplicity sake, let us represent (17) as:

\[
\frac{X}{m_2} = B_1 + B_2 + B_3,
\]

\[
B_1 = -x_2, B_2 = 2^n A + x_3, B_3 = -A.
\]

Let the binary representations of the residues be:

\[
x_1 = (x_1,2n+1,\ldots,x_{1,1})_2,
\]

\[
x_2 = (x_{2,2n-1},x_{2,2n-2},\ldots,x_{2,0})_2,
\]

\[
x_3 = (x_{3,2n-2},x_{3,2n-3},\ldots,x_{3,0})_2.
\]

In (18), \(u_0, u_1, u_2, u_3,\) and \(u_4\) are represented as follows:

\[
u_0 = |-2^{n+2} x_1|_{2^n+1-1} = (x_{1,2n-3},x_{1,2n-2},\ldots,x_{1,0})_{2^{n+1}-1}
\]

\[
u_1 = |-2 x_1|_{2^n+1-1} = (x_{1,2n-1},\ldots,x_{1,0})_{2^n+1},
\]

\[
u_{i,j=2,3} = |2^{n+1} x_j|_{2^n+1-1} = (x_{j,n+2},x_{j,n+1},\ldots,x_{j,0})_{n+1},
\]

\[
u_4 = |2 x_3|_{2^n+1-1} = (0,\ldots,0,x_{3,2n-2},\ldots,x_{3,0})_{n+1}.
\]

Given the binary representation:

\[
A = (a_2 a_2 a_2 \ldots a_1 a_0)
\]

\[
B_2 \text{ can be written as:}
\]

\[
B_2 = (a_2 a_2 a_2 \ldots a_0 a_3 a_3 a_3 \ldots a_0 a_0).
\]

In (19), in order to carry out the summation, \(B_1\) and \(B_3\) must have equal number of bits, i.e., \((3n+1)\)-bits, as \(B_2\). They are represented as:

\[
B_1 = -x_2 = (111\ldots11x_{2,n-1},x_{2,n-2},\ldots,x_{2,0})_{2^n+1},
\]

\[
B_3 = -A = (111\ldots11\bar{a}_n\bar{a}_{n-1},\ldots,\bar{a}_0)_{2^n+1}.
\]

### III. Hardware Realization

The hardware structure of the proposed reverse converter is based on (18) and (19). In Figure 1, \(u_0, u_1, u_2, u_3,\) and \(u_4\) are added by CSAs with end-around carries (EACs) producing the values \(x_3\) 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 CPA with EAC. \(B_2\) is easily obtained by concatenating the operand \(x_3\) with the \(n\)-bit left shift of \(A\). 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 (30) and (31). Thus, the addition of the most significant \((2n+1)\)-bits performed in this CSA can be performed by Half Adders (HA). In addition, since these HA have two inputs equal to 1, the final one’s complement adder will always generate an EAC. Taking this into consideration the one’s complement adder can be reduced to a normal CPA with a constant carry-in of equals to 1. The final result, computed from (6) is obtained simply by a shift and a concatenation operation not requiring any additional hardware.
The performance of the proposed converter is evaluated both theoretically and experimentally by implementing it on an Application Specific Integrated Circuit (ASIC). The results of theoretical analysis are presented in Table I. This table suggests that, in terms of area, the reverse converter in [4] is slightly better than the herein proposed one since our proposal requires \((5n + 4)\times A + 2n\times \text{XNOR} + 2n\times \text{OR}\) gates compared to \((3n - 2)\times A + 2n\times \text{FA}\) faster than the reverse converter in [4].

<table>
<thead>
<tr>
<th>Components</th>
<th>Converter in [4]</th>
<th>Proposed Converter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FA</td>
<td>(9n + 2)</td>
<td>(9n + 2)</td>
</tr>
<tr>
<td>HA</td>
<td>(2)</td>
<td>(5n + 4)</td>
</tr>
<tr>
<td>XNOR</td>
<td>(2n)</td>
<td>(2n)</td>
</tr>
<tr>
<td>OR</td>
<td>(2n)</td>
<td>(2n)</td>
</tr>
<tr>
<td>Delay</td>
<td>((10n + 5)\times \text{FA})</td>
<td>((7n + 7)\times \text{FA})</td>
</tr>
</tbody>
</table>

For the experimental assessment, the converters were described in VHDL and implemented on a 0.18\(\mu\m) Standard Cell technology from UMC [7]. The experimental results, presented in Table II, suggest that, on average, the proposed structure is capable of performing the reverse conversion 19% faster, with an extra area cost of 3%. To compare both conversion structures, the Area-Time (AT) efficiency metric was used. This metric suggests that the proposed reverse converter is 16% more efficient than the one in [4].

V. CONCLUSIONS

In this paper, a novel high speed memoryless residue to binary converter for \(\{2^n+1 - 1, 2^n, 2^n - 1\}\) moduli set is proposed. First, we simplified the traditional CRT to obtain a reverse converter that requires \(\text{mod}-(2^n+1 - 1)\) instead of both \(\text{mod}-(2^m+1 - 1)\) and \(\text{mod}-(2^n - 1)\) required by the reverse converter in [4]. We further simplified the resulting architecture in order to obtain a pure adder-based memoryless converter, which is made up of only CSAs and CPAs. This leads to a structure that is amenable to efficient VLSI chip realization.

The performance of the proposed reverse converter is evaluated both theoretically and experimentally. Experimental results suggest that the proposed structure is, on average, 19% faster, with an additional area cost of 3%. Moreover, the AT metric indicates that the proposed reverse converter is 16% more efficient than the one in [4].

VI. ACKNOWLEDGMENT

The authors wish to acknowledge HIPEAC Network of Excellence.

REFERENCES