An Power Efficient RNS Backward Converter for Novel Moduli Set \( \{2^{2n+1}, 2^{2n} - 1, 2^{2n} + 1\} \)

I.B.K.Raju¹, P. Rajesh Kumar², Ch. Divya³

¹²Padmasri Dr. B.V. Raju Institute of Technology, CVD, ECE Department, Narsapur, Medak Dt, TELANGANA
³P.V.P Siddhartha Institute of Technology, Vijayawada, AP

Abstract

In this paper we propose new 3-moduli set \( \{2^{2n+1}, 2^{2n} - 1, 2^{2n} + 1\} \) for a large dynamic range of 6n-bits. Adder based RNS backward converter for this 3-moduli set is proposed based on New Chinese Remainder Theorem-I to achieve high performance. Hardware architecture of the proposed converter employs only Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs). We implemented the proposed backward converter for different widths using standard cell 180nm CMOS technology libraries and result analysis indicated that, the proposed converter achieves about 20% power reduction over the other equivalent state of the art backward converters.

Keywords-Residue Number system (RNS), Chinese Remainder Theorem (CRT), New Chinese Remainder Theorem-I(New CRT-I),backward converter.

1. Introduction

Residue number system (RNS) is an efficient unconventional number system which has attracted researches for over five decades. RNS is based on the discovery of association with residue of a number first discovered by a Chinese Scholar named Sun Tzu dated in third century and mentioned RNS in his book Suan Ching in the form of a mathematical riddle and solved by a method called Tai Yen [1]. The theory of residue numbers was again adduced by Carl Friedrich Gauss in 19th century[2].

Residue Number System (RNS) is a non-weighted number system which has significant advantages over weighted binary number system. The implicit characteristics of RNS such as absence of carry propagation in additions and multiplications, fault tolerance, parallelism and modularity make it as an alternative to the weighted number system for efficient hardware implementation of DSP computation algorithms [1]. RNS can be used efficiently to operate on large numbers namely for applications such as communications engineering [3 ], computer security (cryptography) [4], signal processing [5], image processing systems especially RNS image coding which can offer high speed VLSI implementation of secure image processing algorithms [6], computer arithmetic[14] and also redundant RNS is extensively used for design of the error detection and correction codes [7]. The effectiveness of RNS is limited by number of issues such as complexity in performing fast modulo operations and fast conversion as magnitude comparison, sign detection, overflow detection etc and also due to overhead in the forward and backward conversions. Of all these issues backward conversion is a major overhead which limits wider use of RNS [8]. The conversion from residue to binary i.e. backward conversion is one of the critical impediments to the adoption of RNS System. Backward converter is an critical part of the RNS system because the conversion delay should not counteract the speed gain of RNS arithmetic unit [9] The first step of designing a backward converter is the moduli set selection. The proper selection of a moduli set has an important role in the design of the RNS system because the speed of RNS arithmetic unit and as well as the complexity of residue to binary converter depend on the form and the number of the moduli. Another important concern for reverse converter design is the selection of an appropriate conversion algorithm. The algorithms of reverse conversion are mainly based on the Chinese Remainder Theorem (CRT) Mixed-Radix Conversion (MRC) and the New Chinese Remainder Theorems (New CRTs)[12].

Several interesting backward converters have been proposed for moduli sets

\[ \{2^n - 1, 2^n, 2^n + 1\} \quad [11], \quad \{2^n - 1, 2^n, 2^n + 1, 2^{2n + 1} - 1\} \quad [11], \quad \{2^n - 1, 2^n, 2^n + 1, 2^{2n + 1} + 1\} \quad [10], \quad \{2^n - 1, 2^n, 2^n + 1\}, \{2^{2n + 1} - 1, 2^{2n + 1} + 1\} \quad [15] \]

in order to achieve efficient RNS processors.

This paper introduces new novel 3-moduli set \( \{2^{2n+1}, 2^{2n} - 1, 2^{2n} + 1\} \) with 6n-bits dynamic range. The backward converter for this moduli set is implemented based on New CRT-I theorem. When compared to the proposed converter in [10], the proposed converter has shown 20% better performance in aspects of power. The proposed converter outperforms converter [10] in terms of power. Additionally the proposed converter requires less power when compared to similar
6n dynamic range similar state of art backward converters.

This paper backward converter design for the moduli set is proposed and analyzed with the other backward converter designs. The rest of paper is structured as follows Section 2 provides brief background on New CRT-I. The proposed backward converter design for the moduli set \( \{2^{2n+1}, 2^n - 1, 2^{2n} + 1\} \) along with hardware implementation is presented in Section 3, performance of the backward converter designs are analyzed and evaluated in Section 4 and it is followed by conclusion and references in Section 5.

2. BACK GROUND

Residue Number System (RNS): An RNS can be defined in terms of a relatively-prime moduli set \( \{M_1, M_2, \ldots, M_n\}\) where \( \gcd(M_i, M_j) = 1 \) for \( i \neq j \) and \( \gcd(a, b) \) denotes the greatest common divisor of \( a \) and \( b \). A weighted number \( X \) can be represented as \( X = (x_1 x_2 \ldots x_n) \) where \( x_i = X \mod M_i = [X]_{M_i}, \ 0 \leq x_i < M_i \) (1)

Such a representation is unique for any integer \( X \) in the range \([0,M-1]\), where \( M=M_1 M_2 \ldots M_n \) is the DR of the moduli set \( \{M_1, M_2, \ldots, M_n\}\) [1].

New Chinese Remainder Theorem I: For a three moduli set \( \{M_1, M_2, M_3\} \), the number \( X \) can be converted from its residue representation \( (x_1 x_2 x_3) \) by New CRT-I as follows [13]

\[
X = x_1 + M_1 k_1 (x_2 - x_1) + k_2 M_2 x_3 - x_2 \mid M_2 M_3 (2)
\]

Where \( [k_2 M_2 x_3 - x_2] \mid M_3 = 1 \) (3)

\[
[k_2 x_3 M_2] \mid M_3 = 1 \quad (4)
\]

Where \( k_2 \) and \( k_3 \) are multiplicative inverses.

3. PROPOSED BACKWARD CONVERTER DESIGN

In this section the New Chinese Remainder Theorem-I (New CRT-I) is applied to derive backward conversion algorithm for proposed moduli set \( \{2^{2n+1}, 2^n - 1, 2^{2n} + 1\} \) and adder-based hardware implementation of the conversion technique is presented.

Conversion Algorithm: For the proposed moduli set New CRT-I theorem is employed to design efficient backward conversion algorithm. The following theorems and properties are needed for the derivation of conversion algorithm.

First we prove that the moduli set consists if pair-wise relatively prime numbers.

**Theorem 1**: The moduli \( 2^{2n+1}, 2^n - 1, 2^{2n} + 1 \) are pair-wise relatively prime numbers.

Proof: from the Euclidean theorem, we have \( \gcd(a,b) = \gcd(b, [a]_b) \), therefore \( \gcd(2^{2n+1}, 2^n - 1) = \gcd(2^{2n+1} - 2^n - 1, 2^{2n} - 1) = \gcd(2^n - 1, 2^{2n} - 1) = 1 \) Similarly \( \gcd(2^{2n} - 1, 2^n + 1) = \gcd(2^n + 1, 2^n - 1) = 1 \) and \( \gcd(2^{2n+1} - 1, 2^{2n} + 1) = \gcd(2^{2n} + 1, 2^{2n} + 1) = 1 \). Thus from these results it can be concluded that the moduli set contains relatively prime moduli and it is a valid RNS moduli set.

**Theorem 2**: For the moduli set \( \{2^{2n+1}, 2^n - 1, 2^{2n} + 1\} \) the following holds true:

\[
\begin{align*}
|2^{2n+1} - 1| & = 2^{2n-1} \quad (5) \\
|2^n - 1| & = 2^n - 2 \quad (6)
\end{align*}
\]

Therefore \( k_1 = 2^{2n-1} \)

Proof: if it can be demonstrated that \( ([2^n - 1] \times [2^{2n} + 1]) \mid 2^n - 1 = 1 \) then \( 2^{2n+1} - 1 \) is the multiplicative inverse of \( 2^{2n+1} \) with respect to \( 2^n - 1 \). \( 2^{2n+1} \times [2^n - 1] \mid 2^n - 1 = 1 \). Thus Eq (5) holds true and \( k_1 = 2^{2n-1} \).

In the same way \( 2^n - 2 \times [2^{2n+1} + 1] \mid 2^n - 1 \). Thus Eq (6) holds true and hence \( k_2 = 2^n - 2 \).

**Theorem 3**: For the given three moduli set \( \{M_1, M_2, M_3\} = \{2^{2n+1}, 2^n - 1, 2^{2n} + 1\} \), the number \( X \) can be derived from its corresponding residues \( (x_1 x_2 x_3) \) by

\[
X = x_1 + 2^{2n-1} [x_2 - x_1] + (2^n + 1) 2^{2n-2} [x_2 - x_1] \quad (7)
\]

Proof: By substituting \( M_i = 2^{2n+1}, M_1 = 2^n + 1, M_3 = 2^{2n} - 1 \) and values of \( k_1, k_3 \) from theorem 1 into Eq(2) we get Eq (7).

The following properties are required for the derivation of backward converter and used for further simplification to decrease hardware complexity.

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

**Property 2**: A negative number in modulo \( (2^p - 1) \) is equivalent to the one’s compliment of the number, which is obtained by subtracting the number from \( (2^p - 1) \).

Proposed moduli set

\[
\begin{align*}
x_1 &= x_1 2^n \ldots x_{2n}, \quad (8) \\
x_2 &= x_2 2^{n-1} \ldots x_{2n}, \quad (9) \\
x_3 &= x_3 2^{n-1} \ldots x_{2n} \quad (10)
\end{align*}
\]

From Eq(7) we know that
We simplify Eq(7) as follows
\[ X = x_1 + 2^{2n}Z \]  
(11)

Where
\[ Z = \lfloor 2^{2n-1}(x_2 - x_1) + 2^{2n-1}(2^{2n} + 1)(x_3 - x_2) \rfloor_{2^{2n-1}} \]
(12)

Consider
\[ Z = |v_1 + v_2 + v_3|_{2^{2n-1}} \]  
(13)

Where
\[ v_1 = |2^{2n-1}(-x_1)|_{2^{2n-1}} \]
(14)

\[ v_2 = |(-2^{2n-2}(2^{2n} + 1) + 2^{2n-1})(x_2)|_{2^{2n-1}} \]
(15)

\[ v_{21} = |2^{2n-1}(x_2)|_{2^{2n-1}} \]
(16)

\[ v_{22} = |(-2^{2n-2}(2^{2n} + 1)(x_2,2)_{2^n} + 2^{2n-1}x_2)|_{2^{2n-1}} \]
(17)

\[ v_{22}' = |2^{2n-1}[(2^{2n} + 1)x_2]|_{2^{2n-1}} \]
(18)

\[ v_{22}'' = |(-2^{2n-2}(2^{2n} + 1)(x_2,2)_{2^n} + 2^{2n-1}(2^{2n} + 1)x_2)|_{2^{2n-1}} \]
(19)

\[ v_3 = |(2^{2n-2} \times (2^{2n} + 1)(x_3)|_{2^{2n-1}} \]
(20)

Consequently, \( Z \) in Eq(13) can be calculated as
\[ Z = |v_1 + v_2 + v_3|_{2^{2n-1}} \]
(21)

Then from Eq(11) we get the final equation as follows
\[ X = x_1 + 2^{2n}Z \]
(22)

**Hardware implementation:** The hardware structure of the proposed reverse converter is shown in Fig.1. Implementation is based on equations Eq (11), Eq(13), Eq(21) and Eq(22). The operand preparation unit prepares the required operands in equations Eq(14), Eq(16), Eq(18), Eq(19) and Eq(20) by simple manipulation of the routing of the bits of residues. We require (6n+3) Not gates for performing the inversions in Eq(14), Eq(18) and Eq(19).

![Fig.1. Hardware Architecture of Proposed Converter](image-url)
For the five operands it requires three levels of 4n-bit Carry Save Adders (CSA) with End Around Carry (EAC) followed by a 4n bit Carry Propagate Adder (CPA) with End Around Carry (EAC). The computation of equation Eq(22) requires just concatenation of (2n+1) bits of $x_j$ with 4n bits of Z. The description of different parts of the proposed reverse converter is given in Table 1.

### TABLE 1
**HARDWARE REQUIREMENTS OF PROPOSED CONVERTER**

<table>
<thead>
<tr>
<th>Parts</th>
<th>FA</th>
<th>NOT</th>
<th>XOR/AND PARTS</th>
<th>XNOR/OR PARTS</th>
<th>DELAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPU1</td>
<td>-</td>
<td>6n+3</td>
<td>-</td>
<td>-</td>
<td>$t_{OFA}$</td>
</tr>
<tr>
<td>CSA1</td>
<td>2n+1</td>
<td>-</td>
<td>-</td>
<td>2n-1</td>
<td>$t_{CPA}$</td>
</tr>
<tr>
<td>CSA2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2n-2</td>
<td>$t_{CPA}$</td>
</tr>
<tr>
<td>CSA3</td>
<td>2n+1</td>
<td>-</td>
<td>-</td>
<td>2n-1</td>
<td>$t_{CPA}$</td>
</tr>
<tr>
<td>CPA4</td>
<td>4n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$(8n)t_{FA}$</td>
</tr>
</tbody>
</table>

### 4. PERFORMANCE EVALUATION

The performance of the proposed moduli set is evaluated by performing both theoretical and practical analysis by implementing it using application specific integrated circuit. The results of theoretical analysis are presented in Table 2. This table suggests that in terms of area, delay the proposed backward converter based on New CRT-I is on par with the other backward converters. In order to perform an accurate comparison and analysis, all the converters are described in Verilog HDL and simulated with ISEv14.3. All the conversion circuits are implemented using cadence RTL compiler. The converters are implemented using 180nm standard cell technology libraries of TSMC.

The practical results presented in Table 3 for $n=6,8,10,14$ and 32 suggests that proposed backward converter achieved about 20% power reduction over the other backward converters in the same class of 6n-dynamic range. The efficiency of the proposed converter is much better with increase in the value of ‘n’. The area and delay of the proposed converter is on par with converter in [10], whereas when compared with converter in [16], it shows slight better performance.

The practically analysis of the proposed converter suggests that in aspects of Area x Power, Power x Delay and Area x Power x Delay, it is capable of 20%, 22% and 22% better performance respectively over the equivalent state of art converters.

### TABLE 2
**HARDWARE REQUIREMENTS AND CONVERSION DELAYS OF THE DIFFERENT REVERSE CONVERTERS**

<table>
<thead>
<tr>
<th>Moduli set</th>
<th>Area</th>
<th>Power(nW×105)</th>
<th>Delay(ns)</th>
<th>Area × Power(μm²×nW×105)</th>
<th>Power ×Delay(nW×ns×107)</th>
<th>Area × Power ×Delay(μm²×nW×ns×1011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>19234</td>
<td>15.7</td>
<td>13.03</td>
<td>3.01</td>
<td>2.04</td>
<td>3.93</td>
</tr>
<tr>
<td>[16]</td>
<td>22543</td>
<td>16.7</td>
<td>17.43</td>
<td>3.76</td>
<td>2.91</td>
<td>6.56</td>
</tr>
<tr>
<td>PROPOSED</td>
<td>19328</td>
<td>12.98</td>
<td>12.9</td>
<td>2.50</td>
<td>1.67</td>
<td>3.23</td>
</tr>
</tbody>
</table>

### TABLE 3
**AREA, POWER, DELAY RESULTS FOR PROPOSED AND OTHER BACKWARD CONVERTERS**

#### For n=6

<table>
<thead>
<tr>
<th>Moduli set</th>
<th>Cell Area(μm²)</th>
<th>Power(nW×105)</th>
<th>Delay(ns)</th>
<th>Area × Power(μm²×nW×105)</th>
<th>Power ×Delay(nW×ns×107)</th>
<th>Area × Power ×Delay(μm²×nW×ns×1011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>19234</td>
<td>15.7</td>
<td>13.03</td>
<td>3.01</td>
<td>2.04</td>
<td>3.93</td>
</tr>
<tr>
<td>[16]</td>
<td>22543</td>
<td>16.7</td>
<td>17.43</td>
<td>3.76</td>
<td>2.91</td>
<td>6.56</td>
</tr>
<tr>
<td>PROPOSED</td>
<td>19328</td>
<td>12.98</td>
<td>12.9</td>
<td>2.50</td>
<td>1.67</td>
<td>3.23</td>
</tr>
</tbody>
</table>

#### For n=8

<table>
<thead>
<tr>
<th>Moduli set</th>
<th>Cell Area(μm²)</th>
<th>Power(nW×105)</th>
<th>Delay(ns)</th>
<th>Area × Power(μm²×nW×105)</th>
<th>Power ×Delay(nW×ns×107)</th>
<th>Area × Power ×Delay(μm²×nW×ns×1011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>21349</td>
<td>16.12</td>
<td>14.08</td>
<td>3.44</td>
<td>2.27</td>
<td>4.84</td>
</tr>
<tr>
<td>PROPOSED</td>
<td>21468</td>
<td>13.21</td>
<td>14.07</td>
<td>2.83</td>
<td>1.85</td>
<td>3.99</td>
</tr>
</tbody>
</table>

#### For n=10

<table>
<thead>
<tr>
<th>Moduli set</th>
<th>Cell Area(μm²)</th>
<th>Power(nW×105)</th>
<th>Delay(ns)</th>
<th>Area × Power(μm²×nW×105)</th>
<th>Power ×Delay(nW×ns×107)</th>
<th>Area × Power ×Delay(μm²×nW×ns×1011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>26680</td>
<td>18.91</td>
<td>17.16</td>
<td>5.04</td>
<td>3.24</td>
<td>8.65</td>
</tr>
<tr>
<td>[16]</td>
<td>26796</td>
<td>22.25</td>
<td>24.12</td>
<td>6.78</td>
<td>5.36</td>
<td>16.34</td>
</tr>
<tr>
<td>PROPOSED</td>
<td>30470</td>
<td>16.26</td>
<td>17.15</td>
<td>4.35</td>
<td>2.78</td>
<td>7.47</td>
</tr>
<tr>
<td>Moduli set</td>
<td>Cell Area (μm²)</td>
<td>Power(nW×10^5)</td>
<td>Delay(ns)</td>
<td>Area × Power (μm²×nW×10^5)</td>
<td>Power ×Delay (nW×ns×10^7)</td>
<td>Area × Power ×Delay (μm²×nW×ns×10^11)</td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>[16]</td>
<td>42483</td>
<td>32.25</td>
<td>33.32</td>
<td>13.7</td>
<td>10.75</td>
<td>45.67</td>
</tr>
<tr>
<td>PROPOSED</td>
<td>37532</td>
<td>22.32</td>
<td>23.31</td>
<td>8.37</td>
<td>5.2</td>
<td>19.53</td>
</tr>
</tbody>
</table>

For n=32

<table>
<thead>
<tr>
<th>Moduli set</th>
<th>Cell Area (μm²)</th>
<th>Power(nW×10^5)</th>
<th>Delay(ns)</th>
<th>Area × Power (μm²×nW×10^5)</th>
<th>Power ×Delay (nW×ns×10^7)</th>
<th>Area × Power ×Delay (μm²×nW×ns×10^11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>85351</td>
<td>61.2</td>
<td>51020</td>
<td>52.2</td>
<td>31.2</td>
<td>266.55</td>
</tr>
<tr>
<td>[16]</td>
<td>96541</td>
<td>73.1</td>
<td>74875</td>
<td>70.5</td>
<td>54.6</td>
<td>527.95</td>
</tr>
<tr>
<td>PROPOSED</td>
<td>85723</td>
<td>49.8</td>
<td>51010</td>
<td>42.7</td>
<td>25.4</td>
<td>217.93</td>
</tr>
</tbody>
</table>

**CONCLUSION**

In this paper we proposed a new novel 3-moduli set with backward converter based on New CRT-I. The proposed backward converter is memory less and purely adder based. Hardware architecture of the proposed converter employs only Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs).

The performance of the proposed converter is evaluated both theoretically and practically. The experimental results suggest that proposed backward converter is 20% better in terms of power when compared to the converter in [10]. Further, in aspects of Area x Power, Power x Delay and Area x Power x Delay, it is capable of 20%, 22% and 22% better performance respectively over the other equivalent state of art converters. Performance will be much better for large values of n.

**ACKNOWLEDGMENT**

The authors would like to thank the management of Dr. B.V. Raju Institute of Technology, Narsapur for providing Cadence EDA tools and support.

**REFERENCES**


[10] A. S. Molahosseini, K. Navi and C. Dadkhah, O. Kavehei and S. Timarchi. “Efficient Reverse Converter Designs for the New 4-Moduli sets \( \{2^n - 1, 2^n + 1, 2^{n+1} - 1\} \) and \( \{2^n - 1, 2^n + 1, 2^{n+1}, 2^{n+2} + 1\} \) based on New CRTs”. IEEE Transactions on Circuits and Systems I, Vol. 57, No. 4, 823-835, April 2010.


