Short Communication

An Efficient Residue to Binary Converter for the New Two-Level Moduli Set
{2^{2n} \{2^n, 2^{n+1}-1\}, 2^n -1, 2^n +1}\n
Safi Seyed Mohammad, Rashno Meysam, Abedi Parvin, Kaboli Mina and Safi Fatemeh Sadat

1Department of Computer Engineering, Ahvaz branch, Islamic Azad University, Ahvaz, IRAN
2Department of Computer Engineering, Ahvaz branch, Islamic Azad University, Shoushtar, IRAN

Available online at: www.isca.in
Received 9th May 2012, revised 12th May 2012, accepted 19th June 2012

Abstract

In this paper a new two-level four moduli set \{2^{2n} \{2^n, 2^{n+1}-1\}, 2^n -1, 2^n +1\} is introduced and an efficient residue to binary converter is proposed for it. This moduli set contains the moduli set \{2^{2n}, 2^n -1, 2^n +1\} in its first-level and the moduli set \{2^n, 2^{n+1}-1\} in its second-level for the modulo 2^{2n}. The reverse converter for this moduli set is implemented in two-level structure, which is designed based on Chinese remainder theorem (CRT) and the new CRT-1 methods. The proposed residue to binary converter for this moduli set improves the hardware cost and delay significantly in comparison to the similar previously presented moduli sets.

Keywords: Reverse converter, residue arithmetic, VLSI architecture.

Introduction

The residue number system (RNS) is a carry-free number system, which can be used as a method for high-speed and low-power implementation of digital signal processing (DSP) computation algorithms. The residue to binary conversion is very important and complex part of an RNS system. The complexity of the residue to binary converter is mainly based on moduli set. Up to now, many moduli sets have been presented with various dynamic ranges (DR) such as \{2^{2n+1}-1, 2^n, 2^n -1\}^2, \{2^{2n}, 2^n -1, 2^{n+1}-1\}^3 and \{2^n, 2^{2n} -1, 2^{2n+1} +1\}^4, which have dynamic ranges equal to 3n, 4n and 5n-bits respectively. Some applications require large dynamic ranges with high parallelism. Therefore, four-moduli sets \{2^n -1, 2^n, 2^n +1, 2^{n+1}-1\} and \{2^{n-3}, 2^n -1, 2^n +1, 2^{n+3}\} have been presented. Since moduli set \{2^n -1, 2^n, 2^n +1, 2^{n+1}-1\} has appropriate moduli, it has a more efficient RNS arithmetic unit compared to moduli set \{2^n -3, 2^n -1, 2^n +1, 2^{n+3}\}.

Hosseinzadeh et al have decreased the delay of reverse converter for moduli set \{2^{n-1}, 2^n, 2^n +1, 2^{n+1}-1\}. However, a little more hardware has been applied.

In this paper, an improved residue to binary converter is proposed for moduli set \{2^n -1, 2^n, 2^n +1, 2^{n+1}-1\} by converting it into a two-level moduli set in the form of \{2^{2n} \{2^n, 2^{n+1}-1\}, 2^n -1, 2^n +1\} such that its residue to binary converter has lower delay and hardware cost in comparison to the proposed converters for \{2^{n-1}, 2^n, 2^n +1, 2^{n+1}-1\}^2 and \{2^n -1, 2^n, 2^n +1, 2^{n+1}-1\}^3.
for converting from RNS to binary system, the residues of the second-level are converted to corresponding residues at the first-level. Then, recently obtained residues are converted to binary system.

Residue to binary converter for the two-level moduli set \( \{2^n, 2^n+1\} \): the CRT-1 for these two moduli requires only one multiplicative inverse as

\[
|k \times 2^n|_{2^{n+1}-1} = 1 \rightarrow k = 2
\]

(4)

The \( T = (x_{11}, x_{12}) \) can be obtained by substituting the value of \( k \), and moduli \( m_{11} = 2^n \), \( m_{12} = 2^{n+1} \) in (3) as shown below

\[
T = x_{11} + 2^n \left[ 2(x_{12} - x_{11}) \right]_{2^n+1-1}
\]

(5)

To calculate \( x_1 \) which is the corresponding residue of \( m_1 = 2^n \), since the value of \( x_1 \) is in \( 0 \leq x_1 < m_{11} \) and the value of \( T \) is in \( 0 \leq T < m_{11} \times m_{12} \) span and with respect to that reality that \( m_{11} \leq m_{11} \times m_{12} \), therefore below equation is used.

\[
x_1 = \begin{cases} T & \text{if } 0 \leq T < m_{11} \\ T - m_{11} & \text{if } m_{11} \leq T < m_{11} \times m_{12} \end{cases}
\]

(6)

The simplification of (5) can be performed with considering the point that, by expressing \( x_1 \) in \( k \) bits, \( x_1 \times 2^n \) \( \mod 2^{n+1} \) and \( -x_1 \) \( \mod 2^{n+1} \) are equivalent to \( p \) bits circular left shifting of \( x_1 \), and one’s complement of \( x_1 \), respectively.\(^1\) The residues can be represented at bit-level by: \( x_{11} = (x_{11,0}, \ldots, x_{11,1}, x_{11,0}) \) and \( x_{12} = (x_{12,0}, \ldots, x_{12,1}, x_{12,0}) \). Therefore, (5) can be rewritten as

\[
T = x_{11} + 2^n H
\]

(7)

\[
H = \left[ s_1 + s_2 \right]_{2^{n+1}-1}
\]

(8)

\[
s_1 = -2x_{11} \left[ 2^n \right]_{2^{n+1}-1} = \left[ -2 \left( n \times x_{11,0} \right) \right]_{2^{n+1}} = \left[ \overline{x_{11,0} \ldots x_{11,0}} \right]_{2^{n+1}}
\]

(9)

\[
s_2 = 2x_{12} \left[ 2^n \right]_{2^{n+1}-1} = \left[ 2 \left( 2^n x_{12,0} \right) \right]_{2^{n+1}} = \left[ x_{12,0} \ldots x_{12,0} \right]_{2^{n+1}}
\]

(10)

By substituting (9) and (10) in (8), \( H \) is obtained as a \((n+1)\)-bits number. For calculating \( T \), it is sufficient to concatenate \( x_{11} \) to \( H \).

\[
T = H_n H_{n-1} \ldots H_1 H_0 x_{11,n-1} \ldots x_{11,1} x_{11,0}
\]

(11)

According to (11) equations (12) and (13) are concluded.

\[
0 \leq T < m_{11} \quad \text{if} \quad H_n = 0
\]

(12)

\[
m_{11} \leq T < m_{11} \times m_{12} \quad \text{if} \quad H_n = 1
\]

(13)

With respect to (12) and (6) \( x_1 \) is obtained as follow

\[
x_1 = 0H_{n-1} \ldots H_1 H_0 x_{11,n-1} \ldots x_{11,1} x_{11,0}
\]

(14)

For the values greater than \( m_1 = 2^n \) and based on (11) and (13), \( T \) is equal to

\[
T = H_n \ldots H_1 H_0 x_{11,n-1} \ldots x_{11,1} x_{11,0}
\]

(15)

The binary representation of \( m_1 = 2^n \) can be shown as

\[
m_1 = \overline{10000 \ldots 0000}
\]

(16)

By substituting (15) and (16) in (6), \( x_1 \) is obtained as

\[
x_1 = 0H_{n-1} \ldots H_1 H_0 x_{11,n-1} \ldots x_{11,1} x_{11,0}
\]

(17)

Since \( x_1 \) has the same value for \( 0 \leq T < m_{11} \) and \( m_{11} \leq T < m_{11} \times m_{12} \), the most significant bit of \( x_1 \) can be ignored as shown below

\[
x_1 = H_n \ldots H_1 H_0 x_{11,n-1} \ldots x_{11,1} x_{11,0}
\]

(18)

By calculating \( x_1 \) and using residues \( x_2 \) and \( x_3 \), the residue to binary converter for the first-level moduli set \( \{2^n, 2^n-1, 2^n+1\} \) is designed.

Residue to Binary converter for the moduli set \( \{2^n, 2^n-1, 2^n+1\} \) based on CRT: according to (2) and by assuming \( m_1 = 2^n \), \( m_2 = 2^n-1 \) and \( m_3 = 2^n+1 \) we have

\[
\hat{m}_1 = (2^n - 1), \quad \hat{m}_2 = 2^n (2^n + 1), \quad \hat{m}_3 = 2^n (2^n - 1) \quad \text{and} \quad M = 2^n (2^n - 1)
\]

(19)

Considering (19) the required multiplication reverses for (2) are computed as follows

\[
\begin{align*}
|k_1 \times (2^n - 1)|_{2^{2n}} &= 1 \rightarrow \hat{k}_1 = 1 \\
|k_2 \times 2^n (2^n + 1)|_{2^{2n+1}} &= 1 \rightarrow \hat{k}_2 = 2^{n-1} \\
|k_3 \times 2^n (2^n - 1)|_{2^{2n+1}} &= 1 \rightarrow \hat{k}_3 = 2^{n-1}
\end{align*}
\]

(20)-(22)

The binary vectors \( x_1, x_2 \) and \( x_3 \) can be represented in bit-level as

\[
x_1 = (H_{n-1}, \ldots, H_1, H_0, x_{11,n-1}, \ldots, x_{11,1}, x_{11,0}), \quad x_2 = (x_{2n-1}, \ldots, x_{2,1}, x_{2,0}) \quad \text{and} \quad x_3 = (x_{3n-1}, \ldots, x_{3,1}, x_{3,0}).
\]

Now, (2) can be rewritten as

\[
X = \sum_{i=1}^{n} \hat{m}_i [k_i]_{m_i} x_i - M \times l
\]

(23)

Where: \( l \) is an integer number and depends on the value of \( X \). By replacing (25)-(22) in (23) we have

\[
\begin{align*}
X &= \left( 2^n - 1 \right) \times \left( 2^n + 1 \right) \times 2^{n-1} x_2 + 2^n \times \left( 2^{2n} - 1 \right) \times l \\
&= 2^n \left( 2^n + 1 \right) \times 2^{n-1} x_2 + 2^n \times \left( 2^{2n} - 1 \right) \times l
\end{align*}
\]

(24)

By dividing both sides of (24) by \( 2^n \) we have
and calculating the floor values in modulo \((2^{2n} - 1)\) results in the following

\[
\left\lfloor \frac{X}{2^{2n}} \right\rfloor = \left\lfloor -1 \times x_1 \right\rfloor_{2^{2n-1}} + \left\lfloor 2^{n+1} \times x_2 \right\rfloor_{2^{2n-1}} + \left\lfloor 2^{n-1} \times x_3 \right\rfloor_{2^{2n-1}}
\]

(26)

In this case, the number \(X\) can be computed by the following

\[
X = \left\lfloor \frac{X}{2^{2n}} \right\rfloor \times 2^{2n} + x_1
\]

(27)

Eq. (26) can be rewritten as

\[
\left\lfloor \frac{X}{2^{2n}} \right\rfloor = [S_3 + S_4 + S_{51} + S_{52}]_{(2^{2n-1})}
\]

(28)

The hardware implementation of residue to binary converter for the two-level moduli set \(\{2^n, 2^{n+1} - 1\}\) is illustrated in Figure-1. The required hardware consists of \(n\) NOT gates in operand preparation unit 1 (opu1) which are used for calculating equation (9). To implement (8), a modulo \(2^{n+1} - 1\) adder is required. In this paper a \((n+1)\)-bits carry propagate adder (CPA) with end around carry (EAC) is used to satisfy it. Opu2 contains \((3n+1)\) NOT gates to calculate equations (29) and (32). Equation (28) is implemented by applying two \(2n\)-bits carry-save adders (CSA) with EAC and one \(2n\)-bits CPA with EAC. Some of the used full adders (FA) in CSA1 and CSA2 are reduced with pair of XOR/AND and XNOR/OR gates, because equations (31) and (32) have some bits with constant values 0 or 1. Equation (27) is computed by concatenating \(x_1\) with \(\left\lfloor \frac{X}{2^{2n}} \right\rfloor\) without any extra hardware.

### Results and Discussions

In Table-1 the performance of the proposed residue to binary converter for the moduli set \(\{2^n, 2^{n+1} - 1\}\) has been compared with converters for \(\{2^n - 1, 2^n + 1\}\) and \(\{2^n - 1, 2^n, 2^{n+1} - 1\}\) from both hardware cost and delay viewpoints. As shown in figure-1, the delay of opu1 and opu2 are equal to one NOT gate and CSA1 and CSA2 have the delay of one full adder. In addition, the delay of CPA1 and CPA2 is equal to \((2n+2)t_{FA}\) and \((4n)t_{FA}\) respectively, where \(t_{FA}\) denotes the delay of a full adder (FA). For a better comparison, the unit gate model is considered to obtain total area and delay estimations. Based on this model, each two-input monotonic gate counts as one gate in area and delay, an XOR/XNOR gate counts as two gates in area and delay, and an FA has area of seven gates and delay of four gates. The corresponding total unit area and delay are presented in Table-1. According to the results of Table-1, our proposed residue to binary converter has significant reduction in both delay and hardware cost in comparison to the converters presented for \(\{2^n - 1, 2^n, 2^{n+1} - 1\}\) and \(\{2^n - 1, 2^n, 2^{n+1} + 1, 2^{2n+1} - 1\}\) moduli sets.

### Acknowledgement

This paper has been derived from Seyyed Mohammad Safi research plan in Islamic Azad university Ahvaz branch.

### Conclusion

This paper presents an efficient two-level design of reverse converter for the new two-level moduli set \(\{2^n, 2^{n+1} - 1\}\) based on combination of CRT and New CRT-1. Comparison with the similar four-moduli residue to binary converters show that the proposed design is faster and requires less hardware area.

### References


2. Mohan P.V.A., RNS-To-Binary Converter for a New Three-moduli Set \(\{2^{n+1} - 1, 2^n, 2^{n+1} - 1\}, IEEE trans. Circuits Syst. 54(9), 775-779 (2007)

3. Sabbagh A., Dadkhah C.H., Navi K. and Eshghi M., Efficient MRC-Based Residue to Binary Converters for the New Moduli Sets \(\{2^{2n}, 2^n - 1, 2^{n+1} - 1\}\) and \(\{2^{2n}, 2^n - 1, 2^{n+1} - 1\}\), JEICE TRANS. INF. & SYST., 92(9), 42-51 (2009)

5. Mohan P.V.A. and Premkumar A.B., RNS-to-Binary Converters for Two Four-Moduli Set \(\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}\) and \(\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}\), *IEEE Trans. Circuits syst. I.*, 54(6), 1245-1254 (2007)


7. Hosseinzadeh M., Sabbagh A. and Navi K., An improved reverse converter for the moduli set \(\{2^n -1, 2^n, 2^n +1, 2^{n+1} - 1\}\), *IEICE Elect. Exp.*, 5(17), 672-677 (2008)


---

**Figure- 1**

The proposed residue to binary converter: (a) second-level (b) first-level

**Table 1**

<table>
<thead>
<tr>
<th>Converter</th>
<th>Performance Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]-CI</td>
<td>((9n+5+k^*)A_{FA}+(2n)A_{XNOR}+(2n)A_{OR}+(6n+1)A_{NOT})</td>
</tr>
<tr>
<td>[7]</td>
<td>((10n+6+k^*)A_{FA}+(6n+2)A_{XNOR}+(6n+2)A_{OR}+(7n+2)A_{NOT}+(n+3)A_{MUX2}+(2n+1)A_{MUX3-1})</td>
</tr>
<tr>
<td>Proposed</td>
<td>((5n+3)A_{FA}+(n+1)A_{XOR}+(n+1)A_{AND}+(n+1)A_{XNOR}+(n+1)A_{OR}+(4n+1)A_{NOT})</td>
</tr>
</tbody>
</table>

* \(k= (n-4)*(n+2)/2\)