Research Journal of Recent Sciences _____ ISSN 2277-2502
Vol. **1(7),** 83-86, July **(2012)** *Res.J.Recent Sci.*

Short Communication

An Efficient Residue to Binary Converter for the New Two-Level Moduli Set $\{2^{2n}\{2^n, 2^{n+1}-1\}, 2^n-1, 2^n+1\}$

Safi Seyyed Mohammad[1], Rashno Meysam[1], Abedi Parvin[2], Kaboli Mina[1] and Safi Fatemeh Sadat[1]
[1]Department of Computer Engineering, Ahvaz branch, Islamic Azad University, Ahvaz, IRAN
[2]Department 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[1]. 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 $\{2^{n+1}-1, 2^n, 2^n-1\}$[2], $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$[3] and $\{2^n, 2^{2n}-1, 2^{2n}+1\}$[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 $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$[5] and $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$[6] 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 $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$[7]. 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 $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$[5] and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$[7] .

Material and Methods

The RNS[1] is based on a moduli set $\{m_1, m_2, \ldots, m_n\}$ which consists of pairwise relatively prime numbers. The dynamic range is defined as $M = m_1 m_2 \ldots m_n$. Each weighted number $X < M$ has a unique representation in RNS as $(x_1, x_2, \ldots, x_n)$ where:

$$x_i = X \mod m_i = \left|X\right|_{m_i} \quad , 0 \le x_i < m_i \qquad (1)$$

By using CRT, the RNS number $(x_1, x_2, \ldots, x_n)$ can be converted into its equivalent weighted number as

$$X = \left|\sum_{i=1}^{n} \hat{m}_i \left|k_i \times x_i\right|_{m_i}\right|_M \qquad (2)$$

Where: $M = \prod_{i=1}^{n} m_i$, $\hat{m}_i = \frac{M}{m_i}$, $k_i = \hat{m}_i^{-1}$, $\left|k_i \times \hat{m}_i\right|_{m_i} = 1$

and $x_i = \left|X\right|_{m_i}$.

By using CRT-1, the reverse conversion can be done as

$$X = x_1 + m_1\left|k_1(x_2 - x_1) + k_2 m_2 (x_3 - x_2) + \cdots + k_{n-1} m_2 m_3 \cdots m_{n-1}(x_n - x_{n-1})\right|_{m_2 m_3 \cdots m_n} \qquad (3)$$

Where: $\left|k_1 \times m_1\right|_{m_2 m_3 \cdots m_n} = 1$,

$$\left|k_2 \times m_1 \times m_2\right|_{m_3 \cdots m_n} = 1, \ldots, \left|k_{n-1} \times m_1 \times m_2 \times \cdots \times m_{n-1}\right|_{m_n} = 1.$$

In two-level RNS for each desired modulo at the first-level a moduli set at the second-level must be chosen in such a way that its dynamic range be equal or greater than the desired modulo at the first-level. In two-level RNS, arithmetic operations are performed on the residues of second-level moduli. Afterward,

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 $\{2^{2n}, 2^n, 2^{n+1}-1\}$, $\{2^n, 2^{n+1}-1, 2^n-1, 2^n+1\}$: the CRT-1 for these two moduli requires only one multiplicative inverse as

$$\left|k \times 2^n\right|_{2^{n+1}-1} = 1 \rightarrow k = 2 \quad (4)$$

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

$$\begin{aligned} T &= x_{11} + 2^n \left|2(x_{12} - x_{11})\right|_{2^{n+1}-1} \\ &= x_{11} + 2^n \left|-2x_{11} + 2x_{12}\right|_{2^{n+1}-1} \end{aligned} \quad (5)$$

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

$$x_1 = \begin{cases} T & if \quad 0 \le T < m_1 \\ T - m_1 & if \quad m_1 \le T < m_{11} \times m_{12} \end{cases} \quad (6)$$

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

$$T = x_{11} + 2^n H \quad (7)$$

$$H = \left|s_1 + s_2\right|_{2^{n+1}-1} \quad (8)$$

$$s_1 = \left|-2x_{11}\right|_{2^{n+1}-1} = \left|-2(0\underbrace{x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{n})\right|_{2^{n+1}-1} = \underbrace{\bar{x}_{11,n-1}\ldots\bar{x}_{11,1}\bar{x}_{11,0}}_{n}1 \quad (9)$$

$$s_2 = \left|2x_{12}\right|_{2^{n+1}-1} = \left|2(\underbrace{x_{12,n}\ldots x_{12,1}x_{12,0}}_{n+1})\right|_{2^{n+1}-1} = \underbrace{x_{12,n-1}\ldots x_{12,1}x_{12,0}x_{12,n}}_{n+1} \quad (10)$$

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.

$$T = H_n \underbrace{H_{n-1}\ldots H_1 H_0 x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{2n} \quad (11)$$

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

$$0 \le T < m_1 \qquad if \quad H_n = 0 \quad (12)$$

$$m_1 \le T < m_{11} \times m_{12} \quad if \quad H_n = 1 \quad (13)$$

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

$$x_1 = 0\underbrace{H_{n-1}\ldots H_1 H_0 x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{2n} \quad (14)$$

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

$$T = 1\underbrace{H_{n-1}\ldots H_1 H_0 x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{2n} \quad (15)$$

The binary representation of $m_1 = 2^{2n}$ can be shown as

$$m_1 = 1\underbrace{0000\ldots0000}_{2n} \quad (16)$$

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

$$x_1 = 0\underbrace{H_{n-1}\ldots H_1 H_0 x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{2n} \quad (17)$$

Since x_1 has the same value for $0 \le T < m_1$ and $m_1 \le T < m_{11} \times m_{12}$, the most significant bit of x_1 can be ignored as shown below

$$x_1 = \underbrace{H_{n-1}\ldots H_1 H_0 x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{2n} \quad (18)$$

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

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

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

$$M = 2^{2n}(2^{2n}-1) \quad (19)$$

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

$$\left|k_1 \times (2^{2n}-1)\right|_{2^{2n}} = 1 \rightarrow k_1 = -1 \quad (20)$$

$$\left|k_2 \times 2^{2n}(2^n+1)\right|_{2^n-1} = 1 \rightarrow k_2 = 2^{n-1} \quad (21)$$

$$\left|k_3 \times 2^{2n}(2^n-1)\right|_{2^n+1} = 1 \rightarrow k_3 = 2^{n-1} \quad (22)$$

The binary vectors x_1, x_2 and x_3 can be represented in bit-level as $x_1 = (H_{n-1},\ldots, H_1, H_0, x_{11,n-1},\ldots, x_{11,1}, x_{11,0})$, $x_2 = (x_{2,n-1},\ldots, x_{2,1}, x_{2,0})$ and $x_3 = (x_{3,n},\ldots, x_{3,1}, x_{3,0})$. Now, (2) can be rewritten as

$$X = \left|\sum_{i=1}^{n} \hat{m}_i \left|k_i \times x_i\right|_{m_i}\right|_M = \sum_{i=1}^{n} \hat{m}_i \left|k_i\right|_{m_i} \times x_i - M \times l \quad (23)$$

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

$$X = \begin{pmatrix} (2^{2n}-1)\times(-1)\times x_1 + \\ 2^{2n}\times(2^n+1)\times 2^{n-1}\times x_2 + \\ 2^{2n}\times(2^n-1)\times 2^{n-1}\times x_3 \end{pmatrix} - 2^{2n}\times(2^{2n}-1)\times l \quad (24)$$

By dividing both sides of (24) by 2^{2n} we have

$$\frac{X}{2^{2n}}=\left((-1+2^{-2n})\times x_1+(2^n+1)\times 2^{n-1}\times x_2+(2^n-1)\times 2^{n-1}\times x_3\right)-(2^{2n}-1)\times d \quad (25)$$

and calculating the floor values in modulo $(2^{2n}-1)$ results in the following

$$\left\lfloor \frac{X}{2^{2n}} \right\rfloor = \left| \begin{array}{l} \left|-1\times x_1\right|_{(2^{2n}-1)} + \left|(2^n+1)\times 2^{n-1}\times x_2\right|_{(2^{2n}-1)} \\ + \left|(2^n-1)\times 2^{n-1}\times x_3\right|_{(2^{2n}-1)} \end{array} \right|_{(2^{2n}-1)} \quad (26)$$

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

$$X = \left\lfloor \frac{X}{2^{2n}} \right\rfloor \times 2^{2n} + x_1 \quad (27)$$

Eq. (26) can be rewritten as

$$\left\lfloor \frac{X}{2^{2n}} \right\rfloor = \left| S_3 + S_4 + S_{51} + S_{52} \right|_{(2^{2n}-1)} \quad (28)$$

$$S_3 = \left|-x_1\right|_{(2^{2n}-1)} = \left| -(\underbrace{H_{n-1}\ldots H_1 H_0 x_{11,n-1}\ldots x_{11,1}x_{11,0}}_{2n}) \right|_{2^{2n}-1} =$$

$$\underbrace{\overline{H}_{n-1}\ldots\overline{H}_1\overline{H}_0\overline{x}_{11,n-1}\ldots\overline{x}_{11,1}\overline{x}_{11,0}}_{2n} \quad (29)$$

$$S_4 = \left|(2^{2n-1}+2^{n-1})\times x_2\right|_{2^{2n}-1} = \left|(2^{2n-1}+2^{n-1})\times(\underbrace{00\ldots00}_{n}\underbrace{x_{2,n-1}\ldots x_{2,1}x_{2,0}}_{n})\right|_{2^{2n}-1} =$$

$$\underbrace{x_{2,0}x_{2,n-1}\ldots x_{2,1}x_{2,0}}_{n+1}\underbrace{x_{2,n-1}\ldots x_{2,1}}_{n-1} \quad (30)$$

$$S_{51} = \left|2^{2n-1}\times x_3\right|_{2^{2n}-1} = \left|2^{2n-1}(\underbrace{00\ldots00}_{n-1}\underbrace{x_{3,n}\ldots x_{3,1}x_{3,0}}_{n+1})\right|_{2^{2n}-1}$$

$$= x_{3,0}\underbrace{00\ldots00}_{n-1}\underbrace{x_{3,n}\ldots x_{3,1}}_{n} \quad (31)$$

$$S_{52} = \left|-2^{n-1}\times x_3\right|_{2^{2n}-1} = \left|-2^{n-1}(\underbrace{00\ldots00}_{n-1}\underbrace{x_{3,n}\ldots x_{3,1}x_{3,0}}_{n+1})\right|_{2^{2n}-1}$$

$$= \underbrace{\overline{x}_{3,n}\ldots\overline{x}_{3,1}\overline{x}_{3,0}}_{n+1}\underbrace{11\ldots11}_{n-1} \quad (32)$$

The hardware implementation of residue to binary converter for the two-level moduli set $\{2^{2n}\ \{2^n,\ 2^{n+1}-1\},\ 2^n-1,\ 2^n+1\}$ 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 $\{2^{2n}\ \{2^n,\ 2^{n+1}-1\},\ 2^n-1,\ 2^n+1\}$ has been compared with converters for $\{2^n-1,\ 2^n,\ 2^n+1,\ 2^{n+1}-1\}$[5] and $\{2^n-1,\ 2^n,\ 2^n+1,\ 2^{n+1}-1\}$[7] 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 CSA 1 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 gate 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 $\{2^n-1,\ 2^n,\ 2^n+1,\ 2^{n+1}-1\}$[5] and $\{2^n-1,\ 2^n,\ 2^n+1,\ 2^{n+1}-1\}$[7] 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 $\{2^{2n}\ \{2^n,\ 2^{n+1}-1\},\ 2^n-1,\ 2^n+1\}$ 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

1. Omondi A. and Premkumar B., Residue Number Systems: Theory and Implementations, Imperial College Press, London **(2007)**

2. Mohan P.V.A., RNS-To-Binary Converter for a New Three-moduli Set $\{2^{n+1}-1, 2^n, 2^n-1\}$, *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 $\{2^{2n}, 2^n-1, 2^{n-1}-1\}$, *IEICE TRANS. INF. & SYST*., **92(9),** 42-51 **(2009)**

4. Hariri A., Navi K. and Restegar R., A new high dynamic range moduli set with efficient reverse converter, *Elsevier J. com and Math*, **55(4),** 660-668 **(2008)**

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)**

6. Mohan P. V. A., New reverse converters for the moduli set $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$, *Elsevier J. Electron. Commun.*, **62(9),** 643-658 **(2008)**

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)**

8. Mewada Shivlal and Singh Umesh Kumar, Performance Analysis of Secure Wireless Mesh Networks, *Researh J. Recent Sci.,* **1(3),** 80-85 **(2012)**

9. Molahosseini A., Navi K., Hashemipour O. and A. Jalali, An efficient architecture for designing reverse converters based on a general three moduli set, *Elsevier J. Systems Architecture*, **54(10),** 929-934 **(2008)**

10. Wang W., Swamy M. N. S., Ahmad M. O. and Wang Y., A Study of the Residue-to-Binary Converters for the Three-Moduli Sets, *IEEE Trans. Circuits and Syst-II*, **40(2),** 235-243 **(2003)**

11. Piestrak S.J., A high speed realization of a residue to binary converter, *IEEE Trans. Circuits and Syst-II*, **42(10)**, 661-663 **(1995)**

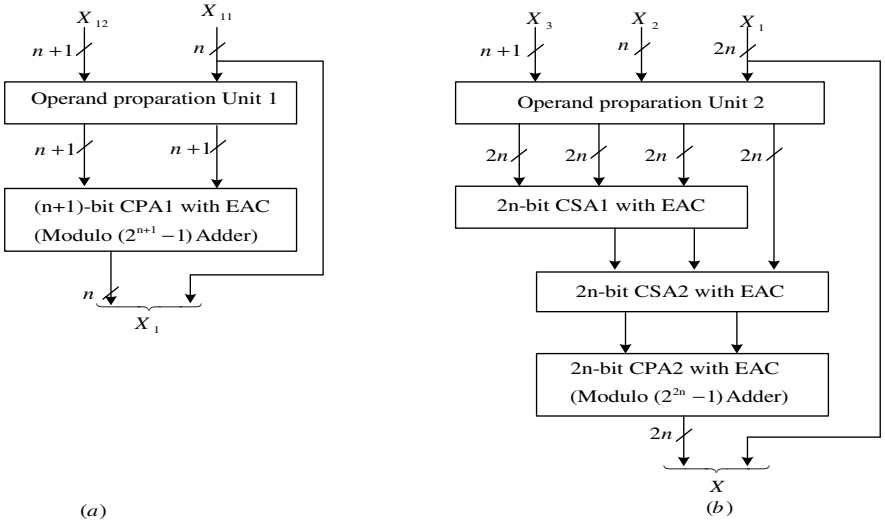

Figure- 1
The proposed residue to binary converter: (a) second-level (b) first-level

Table-1
Performance Comparison

Converter	Area	Unit Gate Area	Delay	Unit Gate Delay
[5]-CI	$(9n+5+k^*)A_{FA}+(2n)A_{XNOR}$ $+(2n)A_{OR}+(6n+1)A_{NOT}$	$(129n+7n^2)/2$ $+4$	$((23n+12)/2)t_{FA}$	46n
[7]	$(10n+6+k^*)A_{FA}+(6n+2)A_{XNOR}$ $+(6n+2)A_{OR}+(7n+2)A_{NOT}$ $+(n+3)A_{MUX21}+(2n+1)A_{MUX3-1}$	$(193n+7n^2)/2$ $+50$	$((15n+22)/2)t_{FA}$	30n
Proposed	$(5n+3)A_{FA}+(n+1)A_{XOR}$ $+(n+1)A_{AND}+(n+1)A_{XNOR}$ $+(n+1)A_{OR}+(4n+1)A_{NOT}$	$(47n+30)$	$(6n+4)t_{FA}+2t_{NOT}$	24n

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