Efficient RNS Reverse Converter using New Chinese Remainder Theorem I and the Moduli Set $\{2^{4n}, 2^{2n}+1, 2^n+1, 2^n-1\}$

Constant Akama
Ho Technical University
P.O.Box 217
Ho, Ghana

ABSTRACT
Residue number system (RNS) is a promising technology for high speed, power efficient and fault tolerant hardware design. The reason is that in RNS, arithmetic operations are performed in parallel, thus reducing delays due to carry operations. Additionally, RNS computations are faster than binary computations because of the reduced wordlength due to modulo operations. Despite the advantages of RNS, its performance depends on the moduli set and the reverse conversion algorithm used to convert the residue numbers back to binary form. There is therefore the need to select the moduli set and the reverse conversion algorithm carefully so that the performance of the RNS hardware is not overshadowed by reverse conversion overheads. This paper proposes an 8n bit moduli set $\{2^{4n}, 2^{2n}+1, 2^n+1, 2^n-1\}$ and a reverse conversion hardware architecture based on the new Chinese remainder theorem I(new CRT I). Compared to existing hardware architecture, the proposed architecture shows good balance between the hardware cost efficiency and speed efficiency. Secondly, the proposed architecture outperforms existing systems in terms of cost-delay square ($\Delta\tau^2$).

General Terms
Residue Number Systems, Digital Signal Processing, Hardware Architecture.

Keywords
Residue Number Systems, Reverse Converter, Moduli Set, Chinese Remainder Theorem.

1. INTRODUCTION
Residue number system is characterised by carry free operation, modularity, parallelism and fault tolerance [1],[2], [3]. This makes RNS a promising technology for high speed, power efficient and fault tolerant hardware design. In RNS, the binary number is first converted to a set of residues using a moduli set. This is called the forward conversion. Once converted, all arithmetic operation are performed as modulo operations based on the moduli set. After the operations are completed, the non-weighted residues are converted back to binary numbers. This process is called reverse conversion. Figure 1 shows the RNS processor.

Despite the advantages of RNS, performance of RNS hardware depends on the reverse conversion algorithms. There is therefore the need to select the moduli sets used for RNS reverse conversion as well as the reverse conversion algorithms carefully so that the advantages of RNS are not overshadowed by the conversion overheads. The works in [4],[5],[6] proposed the moduli sets $\{2^n-1, 2^n, 2^n+1\}$,$\{2^{n-1}-1, 2^n-1, 2^n\}$, $\{2^{n-1}-1, 2^n-1, 2^n+1\}$ and $\{2^n-1, 2^n, 2^{n+1}-1\}$ respectively, however these moduli sets have a dynamic range(DR) of $3n$ bits and are not very suitable for digital signal processing(DSP) applications which involve large wordlengths. [7] proposed vertical and horizontal extensions which enabled it to increase the dynamic range of the moduli set $\left\{2^n, 2^n-1, 2^n+1, 2^n-2^{\frac{(n+1)}{2}}+1, 2^n+2^{\frac{(n+1)}{2}}+1\right\}$ from $5n$ bits to $8n+1$ bits. [1] and [8] also proposed $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$, $\{2^n-1, 2^n+1, 2^n, 2^{2n}+1\}$ and $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$ respectively. The moduli sets in [1] have a dynamic range of $5n$ bits and that of [8] has a dynamic range of $8n$ bits. Because of the high dynamic range, the moduli set in [7] and [8] are more appropriate for DSP applications.

The work of [8] proposed two hardware architectures based on the Chinese remainder theorem and the moduli set$\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$. One architecture is optimized for cost efficiency (CE) and the other is optimized for speed efficiency (SE). The problem with these architectures is that the CE architecture trades speed for cost efficiency whereas the SE architecture trades cost for speed efficiency. This does not lend itself to real time DSP applications where high speed as well as low power and cost are needed.

This paper proposes an $8n$ bit moduli set $\{2^{4n}, 2^{2n}+1, 2^n+1, 2^n-1\}$ and a balanced hardware architecture based on the new Chinese remainder theorem I. This hardware architecture is designed to balance both the cost efficiency and the speed efficiency of the reverse converter. The proposed architecture performs better than the architecture in [7]. Compared to [8], the proposed scheme achieves better speed efficiency than the CE scheme as well as better cost efficiency than the SE scheme. Due to the balanced nature of the architecture its cost-delay ($\Delta\tau^2$) performance is better than both the CE and the SE architecture in [8].

The rest of the paper is organized as follows. In section 11, a background of residue number system is presented. The proposed reverse converter as well as the hardware realization and performance analysis is presented in sections 3 and 4 respectively. Section 5 presents the conclusions drawn from the research.

2. BACKGROUND
In residue number system, numbers are represented as residues of a moduli set $\{m_i\}_{i=1,k}$ which is chosen such that the $gcd(m_i, m_j) = 1$ for $i \neq j$ where $gcd(m_i, m_j)$ is the greatest common divisor of m_i and m_j. The residues are derived as $x_i = |X|_{m_i}$ such that the decimal number X becomes $X = x_1, x_2, x_3 \cdots x_k, 0 \leq x_i \leq m_i$. Such a system has a dynamic range (DR) of $M = \prod_{i=1}^{k} m_i$. For the moduli set $\{m_i\}_{i=1,k}$, the representation $x_1, x_2, x_3 \cdots x_k$ is unique and can be converted back to X using reverse conversion.

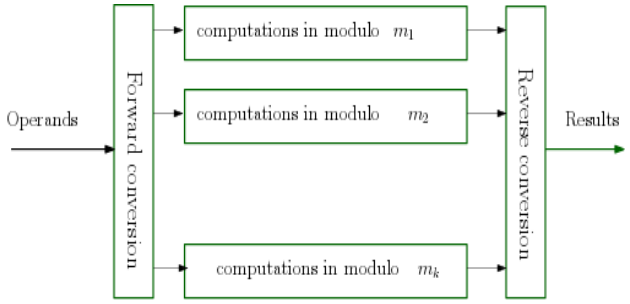

Figure 1: RNS Processor

The major reverse conversion algorithms used in RNS are; the Chinese remainder theorem (CRT), the new Chinese remainder theorem (new CRT)[4] and the mixed radix conversion(MRC)[9] algorithm. The reverse converter proposed in this paper is based on the new CRT I which is given as [4]

$$X = x_1 + m_1 \left| \begin{matrix} K_1(x_2 - x_1) + K_2 m_2 (x_3 - x_2) + \\ K_3 m_2 m_3 (x_4 - x_3) \end{matrix} \right|_{m_2 m_3 m_4} \quad (1)$$

where

$|K_1 m_1|_{m_2 m_3 m_4} = 1$

$|K_2 m_1 m_2|_{m_3 m_4} = 1$

$|K_3 m_1 m_2 m_3|_{m_4} = 1$

3. PROPOSED REVERSE CONVERTER

In this section, the proposed converter is introduced. The following lemmas adopted from [8] and [10] are used to derive the hardware architecture.

Lemma 1. *Modulo 2^k of a number is equivalent to the k least significant bits(LSBs) of the numbe*r.[8]

Lemma 2. *Modulo $2^k - 1$ of a negative number is equivalent to the one's complement of the number which is obtained by subtracting it from $2^k - 1$*

Lemma 3. *Modulo $2^k - 1$ multiplication of a number by 2^t is equivalent to t bit circular shift of the number where both t and k are positive integers*.

Lemma 4. *The sum of a number a and $2^k b$ can be computed by concatenating a and $2^k b$ if and only if a is a k bit number*.

Using the proposed moduli set,

$m_1 = 2^{4n}, m_2 = 2^{2n} + 1, m_3 = 2^n + 1$ and $m_4 = 2^n - 1$.

The parameters in equation [1] can be computed using the theorem below.

Theorem. For the moduli set $\{2^{4n}, 2^{2n} + 1, 2^n + 1, 2^n - 1\}$, the following mathematical statements are true.

$$|m_1|_{m_2 m_3 m_4} = 1 \quad (2)$$

$$|2^{2n-1} m_1 m_2|_{m_3 m_4} = 1 \quad (3)$$

$$|2^{2n-2} m_1 m_2 m_3|_{m_4} = 1 \quad (4)$$

Proof. Equation 2, can be written as

$$|2^{4n}|_{2^{4n}-1} = 1 \quad (5)$$

The left hand side(LHS) of 5 can be shown to be equal to 1. This completes the proof for 2.

Also 3 can be written as

$$|2^{2n-1} 2^{4n} (2^{2n} + 1)|_{2^{2n}-1} = 1 \quad (6)$$

Simplifying equation 6, we get

$$|2^{6n}|_{2^{2n}-1} = 1 \quad (7)$$

thus completing the proof of 3

$$|2^{2n-2} 2^{4n} (2^{2n} + 1)(2^n + 1)|_{2^n-1} = 1$$

By simplifying, we have

$$|2^{6n-2} (2^{2n} + 1)\{(2^n - 1) + 2\}|_{2^n-1} = 1$$

which reduces to

$$|2^{6n-1} (2^{2n} + 1)|_{2^n-1} = 1 \quad (8)$$

By writing 8 as

$$|2^{6n-1}\{(2^n + 1)(2^n - 1) + 2\}|_{2^n-1} = 1$$

we have

$$|2^{6n}|_{2^n-1} = 1$$

completing the proof for 4

Using the proposed theorem, we have $K_1 = 1$, $K_2 = 2^{2n-1}$ and $K_3 = 2^{2n-2}$. Substituting these values in 1, we have

$$X = x_1 + 2^{4n} \left| \begin{matrix} (x_2 - x_1) + 2^{2n-1}(2^{4n} + 1)(x_3 - x_2) + \\ 2^{n-2}(2^{4n} + 1)(2^{2n} + 1)(x_4 - x_3) \end{matrix} \right|_{2^{4n}-1} \quad (9)$$

This is further simplified as

$$X = x_1 + 2^{4n} \left| \begin{matrix} -x_1 + (x_2 + 2^{2n} x_3) - 2^{2n} x_2 + \\ (2^{3n-1} x_4 + 2^{n-1} x_4) - \\ (2^{3n-1} x_3 + 2^{n-1} x_3) \end{matrix} \right|_{2^{4n}-1} \quad (10)$$

Let $= |-x_1|_{2^{4n}-1}$, $U_0 = |(x_2 + 2^{2n} x_3)|_{2^{4n}-1}$,

$U_1 = |-2^{2n} x_2|_{2^{4n}-1}$, $U_2 = |(2^{3n-1} x_4 + 2^{n-1} x_4)|_{2^{4n}-1}$ and

$U_3 = |-(2^{3n-1} x_3 + 2^{n-1} x_3)|_{2^{4n}-1}$

Equation 10 can be written as

$$X = x_1 + 2^{4n} Z \quad (11)$$

where $Z = |A + U_0 + U_1 + U_2 + U_3|_{2^{4n}-1}$

3.1 Evaluating A

$$x_1 = \underbrace{x_{1,4n-1} x_{1,4n-2} \cdots x_{1,0}}_{4n} \quad (12)$$

therefore

$$A = \underbrace{\bar{x}_{1,4n-1} \bar{x}_{1,4n-2} \cdots \bar{x}_{1,0}}_{4n} \quad (13)$$

3.2 Evaluating U_0

$$x_2 = \underbrace{00 \cdots 0}_{2n} \underbrace{x_{2,2n-1} x_{2,2n-2} \cdots x_{2,0}}_{2n} \quad (14)$$

$$x_3 = \underbrace{00 \cdots 0}_{3n} \underbrace{x_{3,n-1} x_{3,n-2} \cdots x_{2,0}}_{n} \quad (15)$$

therefore using lemma 3,

$$|2^{2n} x_3|_{2^{4n}-1} = \underbrace{00 \cdots 0}_{n} \underbrace{x_{3,n-1} x_{3,n-2} \cdots x_{3,0}}_{n} \underbrace{00 \cdots 0}_{2n} \quad (16)$$

$$U_0 = \underbrace{00\cdots0}_{n}\ \underbrace{x_{3,n-1}x_{3,n-2}\cdots x_{3,0}}_{n}\ \underbrace{x_{2,2n-1}x_{2,2n-2}\cdots x_{2,0}}_{2n} \quad (17)$$

3.3 Evaluating U_1

$$U_1 = |-2^{2n}x_2|_{2^{4n}-1} \quad (18)$$

Using equation 14 with lemma 2 and 3, we can write 18 as

$$U_1 = \underbrace{\bar{x}_{2,2n-1}\bar{x}_{2,2n-2}\cdots\bar{x}_{2,0}}_{2n}\ \underbrace{11\cdots1}_{2n} \quad (19)$$

3.4 Evaluating U_2

$$x_4 = \underbrace{00\cdots0}_{3n}\ \underbrace{x_{4,n-1}x_{4,n-2}\cdots x_{4,0}}_{n} \quad (20)$$

Using lemma 3 and 4 , we can write U_2 as

$$U_2 = 0\ \underbrace{x_{4,n-1}\cdots x_{4,0}}_{n}\ \underbrace{0\cdots0}_{n}\ \underbrace{x_{4,n-1}\cdots x_{4,0}}_{n}\ \underbrace{0\cdots0}_{n-1} \quad (21)$$

3.5 Evaluating U_3

Using similar approach as in 21 we can derive U_3 as

$$U_3 = 1\ \underbrace{\bar{x}_{3,n-1}\cdots\bar{x}_{3,0}}_{n}\ \underbrace{1\cdots1}_{n}\ \underbrace{\bar{x}_{3,n-1}\cdots\bar{x}_{3,0}}_{n}\ \underbrace{1\cdots1}_{n-1} \quad (22)$$

4. HARDWARE REALIZATION AND PERFORMANCE ANALYSIS

The hardware realization is based on equations 11, 13, 17, 19 , 21 and 22 as shown in figure 2 . The operand preparation unit(OPU)1 prepares the operands A, U_0, U_1, U_2 and U_3 according to equations 13, 17, 19 , 21 and 22. A, U_0 and U_1, are added using carry-save adder (CSA)1. The sum S_1 and carry C_1 of CSA1 are then added to U_2 by CSA2 to yield S_2 and C_2. These are also added to U_3 by CSA3 to yield S_3 and C_3. S_3 and C_3 are then added by carry propagate adder(CPA)1 to get Z, which is right shifted by $4n$ bits using OPU2. The results of OPU2 are concatenated with x_1 to get X.

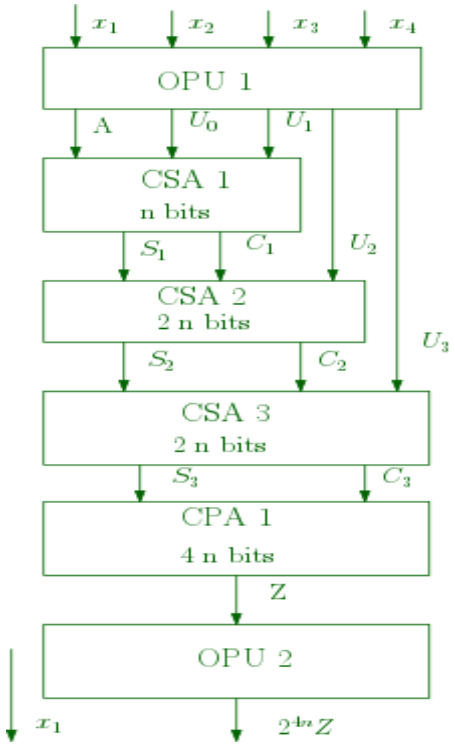

Figure 2 : Proposed Architecture

The carry save adder(CSA1) is a $4n$ bit adder, however because U_0 contains n 0s and U_1 contains $2n$ 1s, we can implement it using n Full Adders(FA) and $3n$ Half Adders(HA). Similarly, U_2 contains $2n$ 0s and U_3 contains $2n$ 1s, therefore we can implement both CSA2 and CSA3 with $2n$ FAs and $2n$ HAs. The overall cost of the proposed hardware is $9nA_{FA} + 7nA_{HA}$, where A_{FA} is the Area of a Full Adder and A_{HA} is the area of a Half Adder. The delay in terms of Full Adders is $4n + 3$ and the cost-delay square $(\Delta\tau^2)$ is $400n^3$.

Compared to existing schemes in [7] and [8] , the proposed scheme outperforms the speed efficient(SE) scheme and in terms of cost and also outperforms the cost efficient(CE) scheme in terms of delay. Additionally, the proposed scheme outperforms[7] in terms of speed and cost efficiency as well as cost-delay square $(\Delta\tau^2)$. Even though the speed efficiency optimized (SE) scheme performs better than the proposed scheme by two bits in terms of speed and the cost optimized (CE) scheme performs better than the proposed scheme by one FA in terms of cost, the proposed scheme balances both cost and delay efficiently compared to the CE and SE schemes which trade speed for cost and cost for speed respectively. As a result, the proposed scheme outperforms both the SE and the CE schemes in terms of cost-delay square$(\Delta\tau^2)$ as shown in figure 3. This shows that the proposed scheme is more efficient than the existing schemes

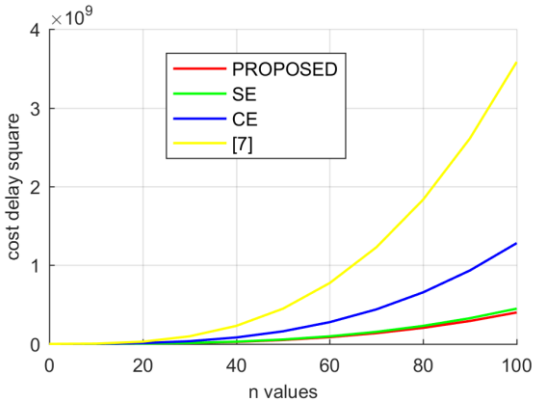

Figure3: Cost-Delay Square

5. CONCLUSION

This paper proposes an efficient 8n bit reverse converter based on the moduli set$\{2^{4n}, 2^{2n}+1, 2^n+1, 2^n-1\}$. Analysis shows that the proposed scheme is more efficient than existing schemes in [7] and [8]. Unlike the SE and CE schemes in [8] which trade cost efficiency for speed efficiency and speed efficiency for cost efficiency respectively, the proposed scheme balances both cost and delay in an efficient manner leading to a better performance in cost-delay square. It also performs better than the SE and CE schemes in terms of cost and speed efficiency respectively.

6. REFERENCES

[1] A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, "Efficient Reverse Converter Designs for the New 4-Moduli Sets $2^n-1, 2^n, 2^n+1, 2^{2n+1}-1$ and $2^n-1, 2^n+1, 2^n, 2^{2n}+1$ Based on New CRTs," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 57, no. 4, pp. 823–835, Apr. 2010.

[2] E. K. Bankas and K. A. Gbolagade, "A residue to binary converter for a balanced moduli set $2^{2n+1}-1, 2^{2n}, 2^{2n}-1$", in: *Awareness Science and Technology*

and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on, 2013, pp. 211–216.

[3] K. A. Gbolagade, R. Chaves, L. Sousa, and S. D. Cotofana, "Residue-to-binary converters for the moduli set $2^{2n}+1, 2^{2n}, 2^n-1$," in *2009 2nd International Conference on Adaptive Science Technology (ICAST)*, 2009, pp. 26–33.

[4] Y. Wang, X. Song, M. Aboulhamid, and H. Shen, "Adder based residue to binary number converters for (2/sup n/-1, 2/sup n/, 2/sup n/+1)," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1772–1779, Jul. 2002.

[5] W. Wang, M. N. S. Swamy, M. O. Ahmad, and Y. Wang, "A high-speed residue-to-binary converter and a scheme for its VLSI implementation," in *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, 1999, vol. 6, pp. 330–333 vol.6.

[6] A. Hiasat and A. Sweidan, "Residue number system to binary converter for the moduli set (2n- 1, 2n- 1, 2n+ 1)," *J. Syst. Archit.*, vol. 49, no. 1–2, pp. 53–58, 2003.

[7] H. Pettenghi, R. Chaves, and L. Sousa, "RNS Reverse Converters for Moduli Sets With Dynamic Ranges up to," vol. 60, no. 6, pp. 1487–1500, 2013.

[8] S. Abdul-Mumin, P. A. Agbedemnab, and M. I. Daabo, "New Efficient Reverse Converters for 8n-bit Dynamic Range Moduli Set," *Int. J. Comput. Appl.*, vol. 161, no. 9, 2017.

[9] N. S. Szabo and R. I. Tanaka, *Residue number system and its application to computer technology*. McGraw-Hill, New York, NY, 1967.

[10] A. Hariri, K. Navi, and R. Rastegar, "A new high dynamic range moduli set with efficient reverse converter," *Comput. Math. with Appl.*, vol. 55, no. 4, pp. 660–668, 2008.