New Efficient Reverse Converters for 8n-bit Dynamic Range Moduli Set

S. Abdul-Mumin
Department of Computer Science
University for Development Studies
Navrongo campus

P. A. Agbedemnab
Department of Computer Science
University for Development Studies
Navrongo campus

M. I. Daabo
Department of Computer Science
University for Development Studies
Navrongo campus

ABSTRACT
This paper proposes two efficient residue to binary converters on a new three-moduli set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$ using the Chinese Remainder Theorem. The proposed reverse converters are adder based and memoryless. In comparison with other moduli sets with similar dynamic range, the new schemes out-perform the existing schemes in terms of both hardware cost and relative performance.

General Terms
Residue Number System, Circuits and Systems, Computer Arithmetic, Computer Architecture, Converters, Digital Signal Processing.

Keywords
Residue to binary converter, reverse converter, residue number system (RNS), Chinese remainder theorem, moduli set.

1. INTRODUCTION
Residue Number System (RNS) is a non-weighted number system that utilizes remainders to represent numbers[1],[2],[3]. The residues are as a result of a decomposition of integer numbers using a selected moduli; each modulus is a channel thus making it possible for this number system to supporting parallel arithmetic as well as carry-free computations. By its nature, the RNS possessed inherent features that makes it useful in the fields of Digital Signal Processing (DSP) intensive computations like digital filtering, convolutions, correlations, Discrete Fourier Transform (DFT) computations, Fast Fourier Transform (FFT) computations and Direct Digital Frequency (DDF) synthesis. However, RNS has not found a widespread usage in general purpose computing due some challenges including overflow detection, magnitude comparison, sign detection, moduli selection, and conversion from decimal/binary to RNS and most especially the vice visa, [3], [4]. Out of these numerous RNS challenges, Data conversion is very critical; for successful application of RNS, data conversion must be very fast so that the conversion overhead does not nullify the RNS advantages. Data Conversion, which is usually based on either the Chinese Remainder Theorem (CRT) [5],[6] [7] or the Mixed Radix Conversion (MRC) [8] can be categorized into forward and reverse conversions. The forward conversion involves converting a binary or decimal number into its RNS equivalent while the reverse conversion involves converting the RNS number into binary or decimal [9]. Relatively, reverse conversion is more complex. Many algorithms have been designed for performing the reverse conversion with different choices of moduli sets such as $\{2^n, 2^n-1, 2^n+1\}$, $\{2^n, 2^n-1, 2^{n-1}-1\}$, $\{2^n, 2^n-1, 2^n+1\}$, $\{2^{n+1}, 2^n-1, 2^{n+1}+1\}$, $\{2^{n+1}-1, 2^n, 2^n-1\}$, $\{2^n, 2^n-1, 2^n+1\}$, $\{2^n-1, 2^n1, 2^{2n+1}-1\}$, $\{2^n-1, 2^n, 2^n+1, 2^{2n}-1\}$,and $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ to mention just a few.

In this paper, firstly we proposed the three-new moduli set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$. This moduli set is balanced and well-formed, which can result in efficient implementation of the residue to binary converter. Then, we present two efficient designs of the reverse converter for these three-moduli set based on the traditional CRT. The proposed converters have better performance, compared to other similar state-of-the-art reverse converters for three-moduli set with similar dynamic range, where the dynamic range is defined in terms of product of the moduli.

The rest of this paper is organized as follows. In Section 2, we present the necessary background information. Section 3 presents the proposed converters. The hardware implementation of the proposed converters is presented in section 4. Section 5 evaluates the performance of our converters while the paper is concluded in Section 6.

2. BACKGROUND
RNS is defined by a set *S* of *N* integers that are pair-wise relatively prime. That is

$$S = \{m_1, m_2, \dots, m_N\}$$

Where gcd $(m_i, m_j) = 1$ for $i, j = 1 \dots N$ and $i \neq j$ and gcd means the greatest common divisor [ ].

Every integer X in $[0, M-1]$ can be uniquely represented with a *N*-tuple where,

$$M = \prod_{i=1}^{N} m_i \ , X \rightarrow (x_1, x_2, \dots, x_N)$$

and $x_i = |X|_{m_i} = (X \bmod m_i)$; for $i = 1$ *to* N

The set S and the number x_i are called the moduli set and residue of *X* modulo m_i respectively.

Here, we propose the new moduli set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$ and first, we show that this set meets the requirements of an RNS moduli set.

Theorem 1: The set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$ is a moduli set for RNS.

Proof: Given the moduli set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$ where $m_1 = 2^{2n}-1$, $m_2 = 2^{4n}$ and $m_3 = 2^{2n}+1$. We should show that the moduli are pair-wise relatively prime for any natural number *n*. m_1 and m_3 have been proven to pair-wise relative prime in [10], [4] and it is also obvious that m_2 is relatively prime to the other moduli.

So our proposed moduli set can be used in RNS and we can consider its reverse converter.

3. PROPOSE CONVERTERS

In this section, we present a reverse conversion algorithm for the moduli set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$.

The following lemmas are important in the design of the proposed converter:

Lemma 1: Modulo 2^s of a number is equivalent to *s* LSBs of the number

Lemma 2: Modulo (2^s-1) of a negative number is equivalent to the one's complement of the number, which is obtained by subtracting the number from (2^s-1) [4].

Lemma 3: 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

Lemma 4: Modulo *a* of an integer *b* of higher length that a (where a in *n*-bit long) can be expressed as modulo *a* of the sum of integers gotten by partitioning *b* into *n*-bit fields.

Lemma 5: The sum of *a* and $2^n b$ is computed as *b* concatenation *a* if *a* is an *n*-bit number.

Now, to calculate the number *X* from its residues, we can apply the CRT. The CRT is formulated as;

$$X = \left| \sum_{i=1}^{N} \ell_i \left| k_i x_i \right|_{m_i} \right|_M \qquad (1)$$

where,

$$M = \prod_{i=1}^{N} m_i \; ; \quad \ell_i = \frac{M}{m_i} \; ; \; |k_i \times \ell_i|_{m_i} = 1$$

Given $m_1 = 2^{2n}-1$, $m_2 = 2^{4n}$ and $m_3 = 2^{2n}+1$, we have $\ell_1 = 2^{4n}(2^{2n}+1)$; $\ell_2 = (2^{4n}-1)$; $\ell_3 = 2^{4n}(2^{2n}-1)$ (2)

Theorem: For the proposed moduli set, we have

$$|k_1|_{m_1} = |2^{n-1}|_{m_1} \qquad (3)$$

$$|k_2|_{m_2} = |-1|_{m_2} \qquad (4)$$

$$|k_3|_{m_3} = |-2^{n-1}|_{m_3} \qquad (5)$$

Proof: For (3), if it can be demonstrated that $2^{n-1} \times \ell_1$ modulus m_1 is equals 1, then 2^{n-1} is the multiplicative inverse of ℓ_1 with respect to m_1. Thus,

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

Similarly, for (4) we have,

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

and finally, for (5) we have,

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

Equation (1) can be rewritten as

$$X = \left| \sum_{i=1}^{N} \ell_i \left| k_i x_i \right|_{m_i} \right|_M =$$

$$\sum_{i=1}^{N} \ell_i \left| k_i \right|_{m_i} \times x_i - M \times K \qquad (6)$$

where *K* is an integer number and depends on the value of *X*.

By replacing (2)-(5) in (6) we have:

$$X = \begin{pmatrix} 2^{4n} \times (2^{2n}+1) \times 2^{n-1} \times x_1 \\ (2^{4n}-1) \times (-1) \times x_2 \\ 2^{4n} \times (2^{2n}-1) \times (-2^{n-1}) \times x_3 \end{pmatrix} - M \times K \qquad (7)$$

By dividing both sides of (7) by $2^{4n}$and calculating the floor values in modulo $(2^{4n}-1)$ we have

$$\left\lfloor \frac{X}{2^{4n}} \right\rfloor = \left| \begin{matrix} |(2^{2n}+1) \times 2^{n-1} \times x_1|_{2^{4n}-1} + |-x_2|_{2^{4n}-1} \\ +|(2^{2n}-1) \times (-2^{n-1}) \times x_3|_{2^{4n}-1} \end{matrix} \right|_{2^{4n}-1} \qquad (8)$$

In this case the number *X* can be computed by

$$X = \left\lfloor \frac{X}{2^{4n}} \right\rfloor \times 2^{4n} + x_2 \qquad (9)$$

Equation (8) can be written as

$$\left\lfloor \frac{X}{2^{4n}} \right\rfloor = |\mathbb{Z}_1 + \mathbb{Z}_2 + \mathbb{Z}_3|_{2^{4n}-1} \qquad (10)$$

where

$$\mathbb{Z}_1 = |2^{n-1} \times (2^{2n} x_1 + x_1)|_{2^{4n}-1} \qquad (11)$$

$$\mathbb{Z}_2 = |-x_2|_{2^{4n}-1} \qquad (12)$$

$$\mathbb{Z}_3 = |-2^{3n-1} x_3 + 2^{n-1} x_3|_{2^{4n}-1} \qquad (13)$$

Now, we consider (11)-(13) and simplify them for implementation in a VLSI system. It is necessary to note that $x_{i,j}$ means the *j*-th bit of x_i.

Evaluation of $\mathbb{Z}_1$

The residue x_1 can be represented in 4*n* bits as follows;

$$x_1 = \overbrace{00\ldots00}^{2n\ Bits} x_{1,(2n-1)} \ldots x_{1,1} x_{1,0} \qquad (14)$$

applying Lemma 5 we have

$$\gamma = \gamma_{4n-1} \cdots \gamma_1 \gamma_0, \qquad (15)$$

where

$$\gamma = 2^{2n} x_1 + x_1 = x_1 \bowtie x_1$$

$$= \underbrace{x_{1,(2n-1)} \ldots x_{1,1} x_{1,0} \acute{x}_{1,(2n-1)} \ldots x_{1,1} x_{1,0}}_{4n} \qquad (16)$$

now applying Lemma 3 in modulo $(2^{4n}-1)$ we have

$$\mathbb{Z}_1 = |2^{n-1} \times \gamma|_{2^{4n}-1} =$$

$$\underbrace{x_{1,n} \ldots x_{1,1} x_{1,0}}_{n+1} \overbrace{x_{1,(2n-1)} \ldots x_{1,1} x_{1,0}}^{2n} \underbrace{x_{1,(2n-1)} \ldots x_{1,n+2} x_{1,n+1}}_{n-1} \qquad (17)$$

Evaluation of $\mathbb{Z}_2$*:*

The residue x_2 can be represented in 4*n* bits as follows;

$$x_2 = x_{2,4n-1} \ldots x_{2,1} x_{2,0} \qquad (18)$$

by applying Lemma 2 we have

$$\mathbb{Z}_2 = |-x_2|_{2^{4n}-1} = \overline{x}_{2,4n-1} \ldots \overline{x}_{2,1} \overline{x}_{2,0} \qquad (19)$$

where $\overline{x}$ means the complement of *x*.

Evaluation of $\mathbb{Z}_3$*:*

The residue x_3 can be represented in 4*n* bits as follows;

$$x_3 = \overbrace{00\ldots00}^{2n-1\ Bits}\ x_{3,2n}\ldots x_{3,1}x_{3,0} \quad (20)$$

for the two parts of $\mathbb{Z}_3$ we use Lemma 3 and we write

$$|2^{3n-1}x_3|_{2^{4n}-1} =$$

$$\underbrace{x_{3,n}\ldots x_{3,1}x_{3,0}}_{n+1\ bits}\ \overbrace{00\ldots00}^{2n-1\ Bits}\ \underbrace{x_{3,2n}\ldots x_{3,1}x_{3,(n+1)}}_{n\ Bits} \quad (21)$$

$$|2^{n-1}x_3|_{2^{4n}-1} = \overbrace{00\ldots00}^{n\ Bits}\ \underbrace{x_{3,2n}\ldots x_{3,1}x_{3,0}}_{2n+1\ Bits}\ \overbrace{00\ldots00}^{n-1\ Bits} \quad (22)$$

for (20) we apply Lemma 2 and we have

$$|-2^{3n-1}x_3|_{2^{4n}-1} =$$

$$\underbrace{\bar{x}_{3,n}\ldots\bar{x}_{3,1}\bar{x}_{3,0}}_{n+1\ bits}\ \overbrace{11\ldots11}^{2n-1\ Bits}\ \underbrace{\bar{x}_{3,2n}\ldots\bar{x}_{3,1}\bar{x}_{3,(n+1)}}_{n\ Bits} \quad (23)$$

therefore,

$$\mathbb{Z}_{3,1} = \underbrace{\bar{x}_{3,n}\ldots\bar{x}_{3,1}\bar{x}_{3,0}}_{n+1\ bits}\ \overbrace{11\ldots11}^{2n-1\ Bits}\ \underbrace{\bar{x}_{3,2n}\ldots\bar{x}_{3,1}\bar{x}_{3,(n+1)}}_{n\ Bits} \quad (24)$$

$$\mathbb{Z}_{3,2} = |2^{n-1}x_3|_{2^{4n}-1} = \overbrace{00\ldots00}^{n\ Bits}\ \underbrace{x_{3,2n}\ldots x_{3,1}x_{3,0}}_{2n+1\ Bits}\ \overbrace{00\ldots00}^{n-1\ Bits} \quad (25)$$

so, $\mathbb{Z}_3$ includes two 4*n*-bit numbers that are $\mathbb{Z}_{3,1}$ and $\mathbb{Z}_{3,2}$

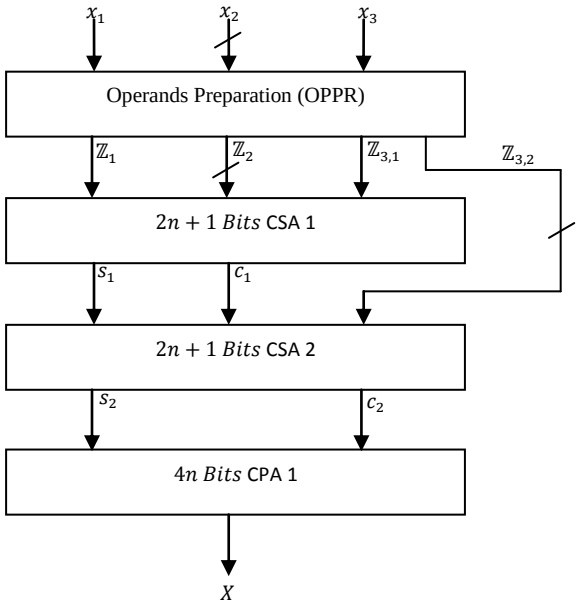

Fig 1: Block diagram of proposed cost-efficient (CE)

5. PERFORMANCE EVALUATION

The performance of the proposed schemes is compared to the schemes in [4] and [6]. The scheme in [4] is 5*n*-bit wide and as a result requires lesser hardware resources as compared to the proposed schemes but relatively slower than the proposed-SE. The complexity of RNS arithmetic depends largely on the dynamic range; note that, [6] and the proposed schemes have the same dynamic range. The proposed schemes perform considerably better than the scheme in [6]. In terms of speed, the proposed SE is much faster. From the table, [4] requires less hardware resources compared to proposed-SE but more resources than the proposed-CE. The proposed schemes also outperform the scheme in [6] in terms of both area and delay. Thus the proposed-CE is cheaper than the compared schemes

4. PROPOSE ARCHITECTURE

We present two converters; Cost Efficient (CE) and Speed Efficient (SE) so that the designer will have a preferred choice. The schematic diagram for the CE is shown in *figure1* and that for the SE in *figure 2*. Both architectures begin with an Operands Preparation Unit (OPPR) which prepares the operands in (15), (18) and (20) by simply manipulating the routing of the bits of the residues and proceeding to compute the parameters in (17), (19), (24) and (25). CSA 1 is $(2n+1)$ bit wide and computes according to equations (17), (19) and (24), CSA 2 is also $(2n+1)$ bit wide and takes in the save (s_1) and carry (c_1) from CSA 1 as well as the parameter in (24); CSAs 3 and 4 respectively work in the same manner. The carry and save from CSA 2 (s_2) and (s_2) are then added using CPA 1 in the case of the CE to get the decimal/binary number X. In the case of the SE in figure 2, the save (s_2) and carry(c_3) from CSA 4 are computed by CPA 2 if the carry bit is zero or CPA 3 if the carry in bit is one and based on that an n-bit $2:1$ Multiplexer (MUX 1) is then used to select the computed value of X either from CPA2 or CPA3.

Regarding the hardware and delay requirements of the proposed schemes: CSAs 1, 2, 3, and 4 require and area each of $(2n+1)_{FA}$ and CPAs 1, 2 and 3 require $(4n)_{FA}$ each of hardware resources. Thus the total hardware requirement for the CE is $(8n+2)_{FA}$ and that of the SE is $(12n+2)_{FA}$. The Delay imposed by CSAs 1, 2, 3, and 4 is each unity and CPA 1 imposes a delay of $(8n)_{FA}$ whilst CPAs 2 and 3 impose a delay of $(4n)_{FA}$ each. Thus the total delay for the CE is $(8n+2)_{FA}$ and that of the SE is $(4n+2)_{FA}$.

The schematic diagrams for the proposed converters are shown below:

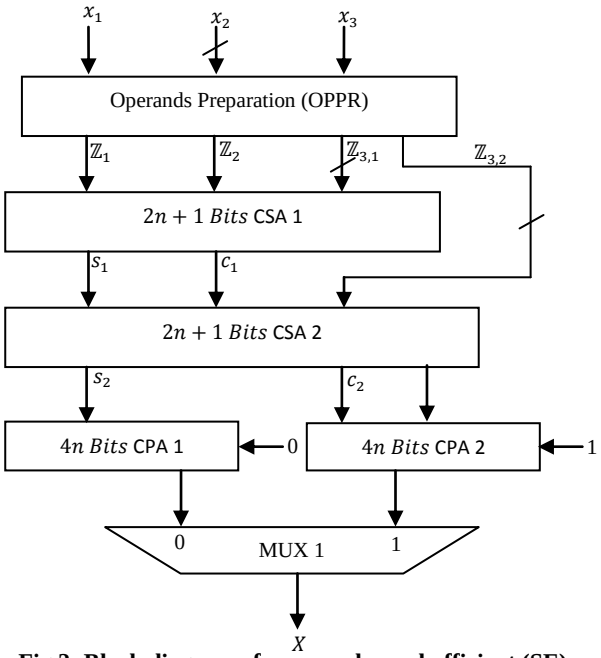

Fig 2: Block diagram of proposed speed-efficient (SE)

such that if the designer desires a scheme that is cheaper then, the proposed-CE will be ideal, on the other hand, the proposed-SE scheme is faster than the compared schemes which can also affect the choice of the designer if a fast schemed is desired. Figure 3 is a graph for the Area-Delay square ($\Delta\tau^2$) which shows how efficient the various schemes are. From the graph, it is observed that only [4] tends to be efficient for larger values of n than the proposed schemes, but this is actually as a result of how wide it is with regards to its range compared to the proposed schemes. In this light, it is obvious that the proposed-SE scheme is a very efficient scheme as shown in the graph.

Table 1: Area, Delay Comparison

Converter	DR (bit)	Area (Δ_{FA})	Delay (τ_{FA})
[4]	$5n$	$11n+1$	$4n+2$
[6]	$8n$	$28n$	$8n+4$
Propose-CE	$8n$	$8n+2$	$8n+2$
Proposed-SE	$8n$	$12n+2$	$4n+1$

In the comparison, we assume an n-bit CSA to require n-bit and a unit bit for the area and delay respectively, whilst the CPA requires an area similar to that of the CSA, but the delay is twice if there is not carry in of bits.

6. CONCLUSION

Two different hardware realisation techniques for the moduli set $\{2^{2n}-1, 2^{4n}, 2^{2n}+1\}$ which is an $8n$-bit dynamic range wider was presented. This implies that larger numbers can be represented using this moduli set and the conversion process achieved by either techniques depending on the preference of the designer. In the final analysis, the schemes demonstrate considerable amount of gains in terms of the area and delay requirements when compared to similar existing schemes. All analysis in this paper is done theoretically; therefore, any future work will focus on practically implementing the algorithms on Field Programmable Gate Array (FPGA) boards.

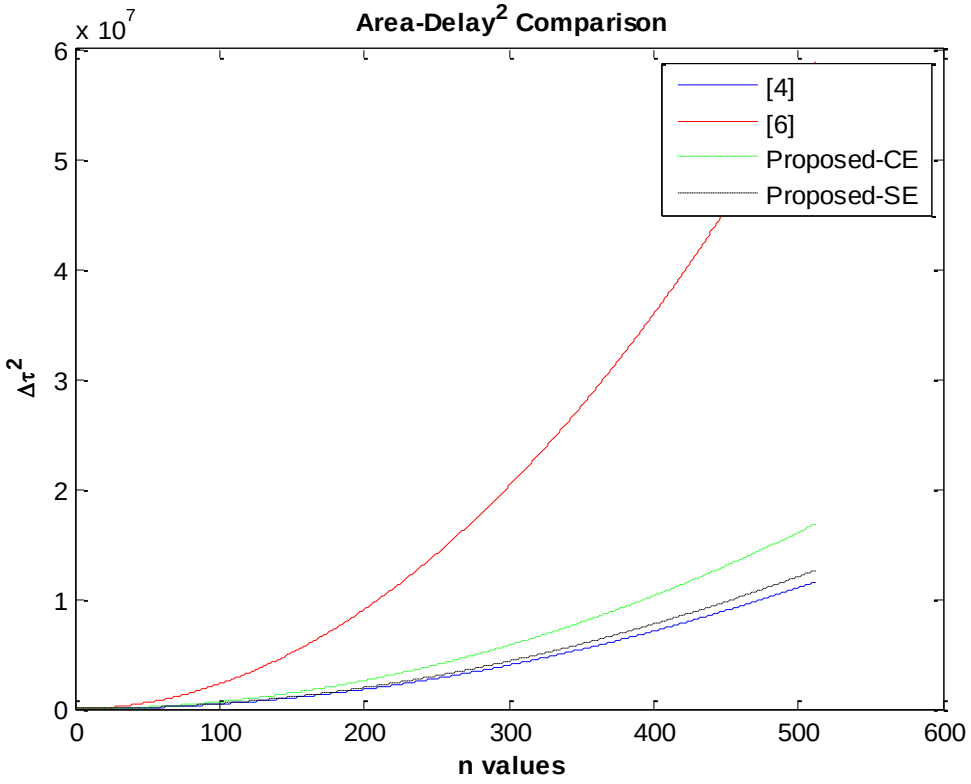

Fig 3: Graph of Area-Delay^2 analysis for the various schemes

7. REFERENCES

[1] A. Omondi and B. Premkumar, *Residue Number Systems: Theory and Implementation*, vol. 2. Published By Imperial College Press And Distributed By World Scientific Publishing Co., 2007.

[2] P. A. Agbedemnab and E. K. Bankas, "A Novel RNS Overflow Detection and Correction Algorithm for the Moduli Set {2^n-1,2^n,2^n+1}," *Int. J. Comput. Appl.*, vol. 110, no. 16, pp. 30–34, Jan. 2015.

[3] M. Bhardwaj, T. Srikanthan, and C. T. Clarke, "A reverse converter for the 4 moduli super set {2^n-1, 2^n, 2^n+1, 2^(n+1)+1}," *IEEE Conf. Comput. Arith.*, 1999.

[4] A. Hariri, R. Rastegar, and K. Navi, "High Dyanamic Range 3-Moduli Set with Efficient Reverse Converter," *Int. J. Comput. Math. Appl.*

[5] E. K. Bankas and K. A. Gbolagade, "A New Efficient FPGA Design of Residue-To-Binary Converter," *Int. J. VLSI Des. Commun. Syst. VLSICS*, vol. 4, no. 6, Dec. 2013.

[6] H. Pettenghi, R. Chaves, and L. Sousa, "RNS Reverse Converters for Moduli Sets With Dynamic Ranges up to -bit," *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 60, no. 6, pp. 1487–1500, Jun. 2013.

[7] M. I. Daabo and K. A. Gbolagade, "RNS Overflow Detection Scheme for the Moduli set {M − 1, M}," *J. Comput.*, vol. 4, no. 8, pp. 39–44, 2012.

[8] K. A. Gbolagade, "New Adder-Based RNS-Binary Converters for the {2^(n+1)+1,2^(n+1)-1,2^n } Moduli Set.," *Int. Sch. Res. Netw.*

[9] G. Jaberipur and H. Ahmadifar, "A ROM-less reverse RNS converter for moduli set 2q ?? 1, 2q ?? 3," *IET Comput. Digit. Tech.*, vol. 8, no. 1, pp. 11–22, Jan. 2014.

[10] E. K. Bankas and K. A. Gbolagade, "A New Efficient RNS Reverse Converter for the 4-Moduli Set {2^n, 2^n+1, 2^n-1, 2^(2n+1)-1}," *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol. 8, no. 2, pp. 318–322, 2014.