# Residue-To-Decimal Conversion with Overflow Detection for a Moduli Set of the Form $\{m_1, m_2, m_3\}$

Mohammed I. Daabo, PhD
Department of Computer Science
C.K. Tedam University of Technology and Applied Sciences
Navrongo, Ghana

## ABSTRACT
Reverse Conversion and Overflow Detection are some of the limiting factors that affect the full implementation of RNS-Based processors in general purpose computing. In this paper, a novel Reverse Converter with overflow detection scheme has been proposed. The Algorithm utilizes the Remainder Theorem and has the property that for any given moduli set $\{m_1, m_2, m_3\}$, the residue number ($x_1$, $x_2$, $x_3$) can be converted into their decimal equivalent X using $m_1\alpha + x_1$, $|m_1\alpha + x_1|_{m_2} = x_2$ and $|m_1\alpha + x_1|_{m_3} = x_3$ for α = 0, 1, 2, 3, …. The Algorithm detects overflow in RNS operations if $m_1\alpha + x_1 \geq$ M. The Algorithm was fully implemented on both moduli sets with common factors and moduli sets with non-coprime factors. Theoretical analysis and simulated results showed that the architecture is built with lesser hardware and has low delay.

## Keywords
Residue Number System, Reverse Converter, Remainder Theorem, Overflow Detection, MRC, CRT.

## 1. INTRODUCTION
An arithmetic carry progression based on the weighted number system is one of the reasons for performance delay in digital computing. To enhance system performance, Residue Number System (RNS), which is considered as an alternative candidate to weighted number system, has been widely used in addition and multiplication dominated digital signal processing applications [3,6,11,15]. In addition, RNS has been established as one of the most popular techniques for reducing power dissipation and computation load in Very Large-Scale Integrated Circuits (VLSI) [18]. Thus, the advantages of RNS over the conventional binary number system include: parallelism, fault tolerance, low power dissipation and high-speed computations and are well stated in [4], [9], [12], [14], [17]. It is, however, interesting to know that despite the above interesting inherent features of RNS, it has not found a widespread usage in general-purpose computing due to the following difficult RNS arithmetic operations: sign detection, magnitude comparison, overflow detection, moduli selection and data conversion [13, 16, 19].

Residue to Binary conversion is one major drawback of RNS, in that a number represented in residue notation does not have a magnitude and sign information and thus should be converted to a weighted number to extract the needed information. Conversion from residue notation to weighted notation and vice versa are needed in almost all applications employing residue arithmetic [2]. Most existing residue-to-binary converters are built based on either the Chinese Remainder Theorem (CRT) [1, 8, 10], or the Mixed Radix Conversion (MRC) technique [ 5, 7, 8, 11]. However, CRT based converters have to depend on the large modulo-M (M being the dynamic range), thus making computations generally slow. In the case of MRC, the computations are done in sequential manner to obtain the Mixed Radix Digits ($MRD_s$). This also involves a lot of arithmetic operations.

In this paper, a novel reverse converter with overflow detection scheme for any three moduli set that involves magnitude comparison and fewer arithmetic operations has been proposed. The technique is an iterative algorithm that uses the Remainder Theorem's (RT) concept in polynomials. The proposed scheme is characterized by low area cost and high conversion speed.

## 2. PROPOSED ALGORITHM
From Fig 1, if X is the decimal equivalent of the RNS number ($x_1$, $x_2$, $x_3$) with respect to the moduli set $\{m_1, m_2, m_3\}$, then we perform the following:

(i) Compute the decimal number $m_1\alpha + x_1$ where α = 0, 1, 2, 3, …

(ii) Test the condition for overflow. If $m_1\alpha + x_1 \geq$ M, then overflow has occurred, (where $M = \prod_{i=1}^{n} m_i$.).

(iii) Compute the following;

$$|m_1\alpha + x_1|_{m_2} = x_2$$

&

$$|m_1\alpha + x_1|_{m_3} = x_3$$

(iv) If the conditions in (iii) hold true, then X = $m_1\alpha + x_1$ is the decimal equivalent of the RNS number ($x_1$, $x_2$, $x_3$) with respect to the moduli set $\{m_1, m_2, m_3\}$

## 3. REVERSE CONVERSION

### 3.1 The Remainder Theorem
**Theorem 1.** Remainder Theorem states that if a polynomial f(x) is divided by the factor (x-b), then the remainder is the value of f(x), at x=b. i.e., f(b) is the remainder.

**Proof:** Let f(x) be a polynomial divided by (x-b). Let q(x) be the quotient and R be the remainder. By division algorithm,

Dividend = (Divisor)(quotient)+ Remainder

i.e. f(x) = q(x)(x-b) +R (1)

Substitute x=b in (1)

implies f(b)= q(b)(b-b) +R

f(b)= 0+R

f(b)=R.

Hence the remainder, R=f(b).

## 3.2 The Reverse Converter

**Proposition 1.** If X is a decimal number representing the RNS number $(x_1, x_2, x_3)$ with moduli set $\{m_1, m_2, m_3\}$, then X = $m_1\alpha + x_1$
if and only if $|m_1\alpha + x_1|_{m_2} = x_2$ and $|m_1\alpha + x_1|_{m_3} = x_3$ where α = 0, 1, 2, 3, …

**Proof:** From Equation (1), we have f(x) = q(x)(x-b)+R. If we substitute X for f(x), $m_1$ for q(x) and α for (x-b),

We have $X = m_1\alpha_1 + x_1$ (2)

Similarly, $X = m_2\alpha_2 + x_2$ (3)

And $X = m_3\alpha_3 + x_3$ (4)

But

$$|X|_{m_1} = x_1 \quad (5)$$

$$|X|_{m_2} = x_2 \quad (6)$$

$$|X|_{m_3} = x_3 \quad (7)$$

Implies $|m_1\alpha_1 + x_1|_{m_1} = x_1$ (8) $|m_2\alpha_2 + x_2|_{m_2} = x_2$ (9)

$|m_3\alpha_3 + x_3|_{m_3} = x_3$ (10)

To determine a unique value of α that will satisfy the three Equations, we choose $m_1\alpha_1 + x_1$ such that $m_1 > m_2$ and $m_3$. Thus since $|m_1\alpha_1 + x_1|_{m_1} = x_1$ is always true, therefore X = $m_1\alpha_1 + x_1$ if and only if $|m_1\alpha_1 + x_1|_{m_2} = x_2$ and $|m_1\alpha_1 + x_1|_{m_3} = x_3$.

For convenience, we simply use α instead of $\alpha_1$.

Hence $X = m_1\alpha + x_1$ if only

$$|m_1\alpha + x_1|_{m_2} = x_2$$

&

$$|m_1\alpha + x_1|_{m_3} = x_3$$

## 3.3 Overflow Detection

**Proposition 2.** If $(m_1\alpha)$ and $x_1$ are decimal numbers and $M = \prod_{i=1}^{n} m_i$, then overflow occurs in the addition of $(m_1\alpha_1)$ and $x_1$ if $m_1\alpha + x_1 \geq M$.

**Proposition 3.** If X = $m_1\alpha + x_1$ is the decimal number representation of the RNS number $(x_1, x_2, x_3)$ with moduli set $\{m_1, m_2, m_3\}$ and $M = \prod_{i=1}^{n} m_i$, then $|M\lambda + X|_M = X$ and $|M\lambda + X|_{m_2} = x_2$ and $|M\lambda + X|_{m_3} = x_3$ is always true with other values of X within the overflow region given by $X_F = M\lambda + X$ where λ = 1, 2, 3, …

**Proof**

Consider $|M\lambda + X|_M = X$

$|M\lambda + X|_M = |M\lambda|_M + |X|_M$

$= |0|_M + |X|_M$

$= |X|_M$

$= X$

Consider $|M\lambda + X|_{m_2} = x_2$ $|M\lambda + X|_{m_2} = |M\lambda|_{m_2} + |X|_{m_2}$

$= |0|_{m_2} + |X|_{m_2}$

$= |X|_{m_2}$

$= x_2$

Consider $|M\lambda + X|_{m_3} = x_3$ $|M\lambda + X|_{m_3} = |M\lambda|_{m_3} + |X|_{m_3}$

$= |0|_{m_3} + |X|_{m_3}$

$= |X|_{m_3}$

$= x_3$

# 4. NUMERICAL ILLUSTRATIONS

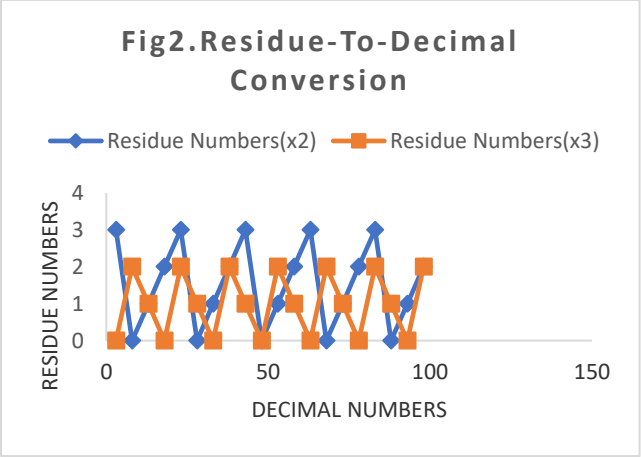
## 4.1 Example 1: Moduli set with pair-wise factors $\{2^n+1, 2^n, 2^n-1\}$

In order to give a fair illustration of the proposed algorithm, we consider imposing it on the RNS number $(3,3,2)_{RNS}\langle 5|4|3\rangle$. Substituting the values of $x_2 = 3, x_3 = 2$ and $m_2 = 4, m_3 = 3$ for n=2 into the equations $X = m_1\alpha + x_1$, $|m_1\alpha + x_1|_{m_2} = x_2$ and $|m_1\alpha + x_1|_{m_3} = x_3$ as in Table 1.

**Table 1. Numerical Illustration on Moduli set with Pair-Wise Factors $\{2^n+1, 2^n, 2^n-1\}$ n=3**

| **Calculate X with α values** | | **Is $|5\alpha + 3|_4 = 3$ and $|5\alpha + 3|_3 = 2$** | |
|---|---|---|---|
| **α** | **X = 5 α + 3** | **$|5\alpha + 3|_4$** | **$|5\alpha + 3|_3$** |
| 0 | 3 | 3 | 0 |
| 1 | 8 | 0 | 2 |
| 2 | 13 | 1 | 1 |
| 3 | 18 | 2 | 0 |
| 4 | 23 | 3 | 2 |

From Table 1, it can be observed that, as values of X = 5 α + 3 are computed for the iterative values of α, the condition for conversion is met when $x_2 = 3$, and $x_3 = 2$ and the decimal equivalent for the given RNS number is thus 23. A simulation of the conversion process is shown in Fig 2




In Fig 2, it can be seen that the conversion process oscillates between two residue classes given by $0 \leq x_2 \leq 2^n - 1$ and $0 \leq x_3 \leq (2^n - 1)$-1. For every decimal number X, there exists two unique residues $(x_2, x_3)$ that define it such that $(X, x_2)$ and $(X, x_3)$ are unique points. In the illustrated example, the unique points are (23,3) and (23,2) for the unique residues (3,2). Thus, the residue classes are (0,1,2,3) and (0,1,2).

**Other values of X within the overflow region $X_F$ can be calculated as follows:**

The equation $X_F = M\lambda + X$ is used in this case with values of M= 60, X=23 and λ = 1,2,3, ...

That is $X_F = 60\lambda + 23$, λ = 1,2,3, ...

$X_1 = 83$

$X_2 = 143$

$X_3 = 203$

⋮ ⋮

$X_F = M\lambda + X$

## 4.2 Example 2: Moduli Set with Common Factors {2n+2, 2n-1, 2n}

Similarly, as in example1, a representation of the RNS number $(6, 0, 0)_{RNS}\langle 8|5|6\rangle$ in its decimal form is shown in Table2. This is achieved by substituting the values of $x_1 = 6$, $x_2 = 0$, $x_3 = 0$, $m_1 = 8$, and $m_2 = 5$, $m_3 = 6$ into the

equations $X=m_1\alpha + x_1$, $|m_1\alpha + x_1|_{m_2} = x_2$ and $|m_1\alpha + x_1|_{m_3} = x_3$ which satisfies the condition for the reverse conversion.

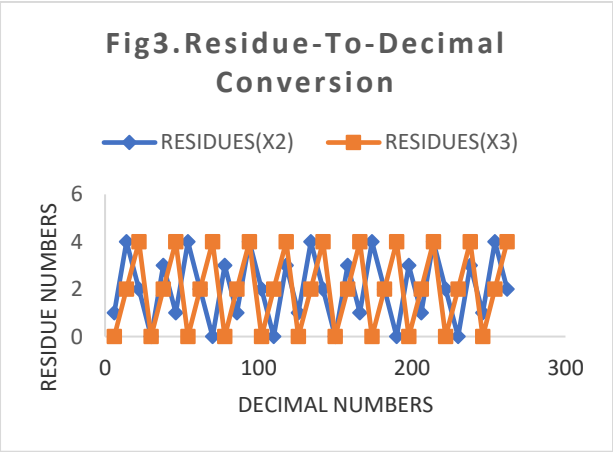
**Table 2. Numerical Illustration on Moduli Set with Common Factors {2n+2, 2n-1, 2n} n=3**

| **Calculate X with α values** | | **Is $\|8\alpha + 6\|_5 = 0$ & $\|8\alpha + 6\|_6 = 0$** | |
|---|---|---|---|
| **α** | **X= 8 α + 6** | **$\|8\alpha + 6\|_5$** | **$\|8\alpha + 6\|_6$** |
| 0 | 6 | 1 | 0 |
| 1 | 14 | 4 | 2 |
| 2 | 22 | 2 | 4 |
| 3 | 30 | 0 | 0 |

In Table 2, the decimal number computed is 30. This is set up by the residue pair (0,0) stated in the condition of the algorithm. The iteration, however, can continue to go even beyond the overflow region as illustrated in equation $X_F = M\lambda + X$ and shown in Fig 3.

**Table 3. Characterization of each Part of the Proposed Reverse Converter**

| Component | Area ($\Delta_{FA}$) | Delay ($t_{FA}$) |
|---|---|---|
| CPA 1 | $2n$ | - |
| CPA 2 | $2n+2$ | $2n+2$ |
| CPA 3 | $2n-2$ | - |
| ACCOM | - | 1 |




Similarly in Fig 3, the simulated results showed that the unique points are (30,0) and (30,0) for the given unique residues (0,0). Thus, the residue classes are (1,4,2,0,3) and (0,2,4).

**Other values of X within the overflow region $X_F$ can be calculated as follows:**

The equation $X_F = M\lambda + X$ is used in this case with values of M= 240, X=30 and $\lambda = 1,2,3, \ldots$

That is $X_F = 240\lambda + 30, \lambda = 1,2,3, \ldots$

$X_1 = 270$
$X_2 = 510$
$X_3 = 750$
$\vdots \quad \vdots$
$X_F = M\lambda + X$

## 5. HARDWARE IMPLEMENTATION

We implement the algorithms on the traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$ for the purposes of hardware realization.

Let $X = Z + x_1$, where $Z = m_1\alpha$; (11)

then

$$x_2 = |\lambda_1 + \lambda_2|_{m_2}, \quad \lambda_1 = |Z|_{m_2}, \quad \lambda_2 = |x_1|_{m_2} \quad (12)$$

and

$$x_3 = |\gamma_1 + \gamma_2|_{m_3}, \quad \gamma_1 = |Z|_{m_3}, \quad \gamma_2 = |x_1|_{m_3} \quad (13)$$

The binary representation of (12) will be

$$\underbrace{\lambda_{1,n-1}\lambda_{1,n-2}\ldots\lambda_{1,1}\lambda_{1,0}}_{n\ bits} + \underbrace{\lambda_{2,n-1}\lambda_{2,n-2}\ldots\lambda_{2,1}\lambda_{2,0}}_{n\ bits} \quad (14)$$

which implies

$$x_2 = \underbrace{x_{2,2n-1}x_{2,2n-2}\ldots x_{2,1}x_{2,0}}_{2n\ bits} \quad (15)$$

also, for (13),

$$\underbrace{\gamma_{1,n-2}\gamma_{1,n-3}\cdots\gamma_{1,1}\gamma_{1,0}}_{n-1\ bits} + \underbrace{\gamma_{2,n-2}\gamma_{2,n-3}\cdots\gamma_{2,1}\gamma_{2,0}}_{n-1\ bits} \quad (16)$$

which implies

$$x_3 = \underbrace{x_{3,2n-3}x_{3,2n-4}\ldots x_{3,1}x_{2,0}}_{2n-2\ bits} \quad (17)$$

also,

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

Our Architecture is based on equations (11), (12), (13), (15), (17) and (18)

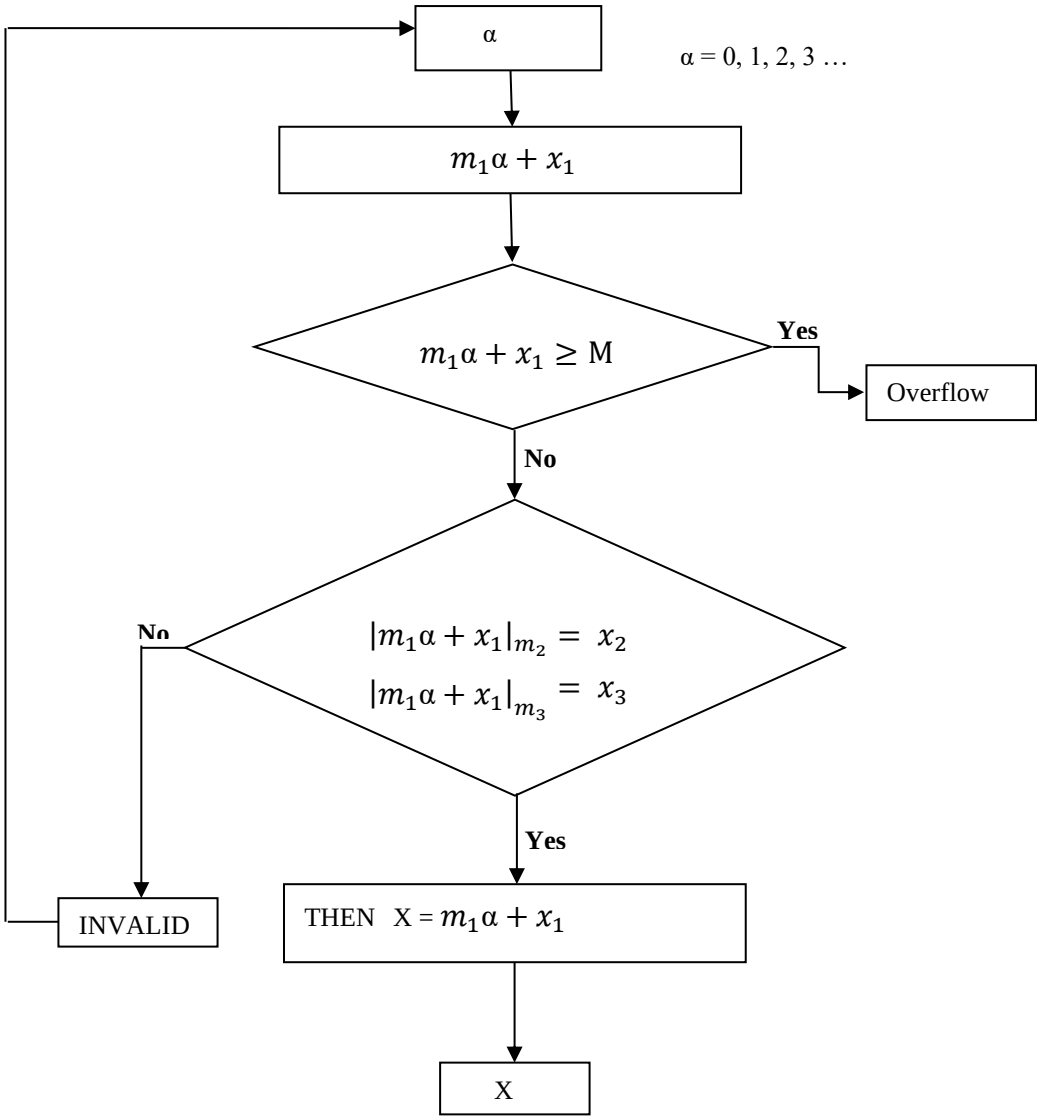



**Fig 1. Proposed Reverse Converter with Overflow Flowchart**

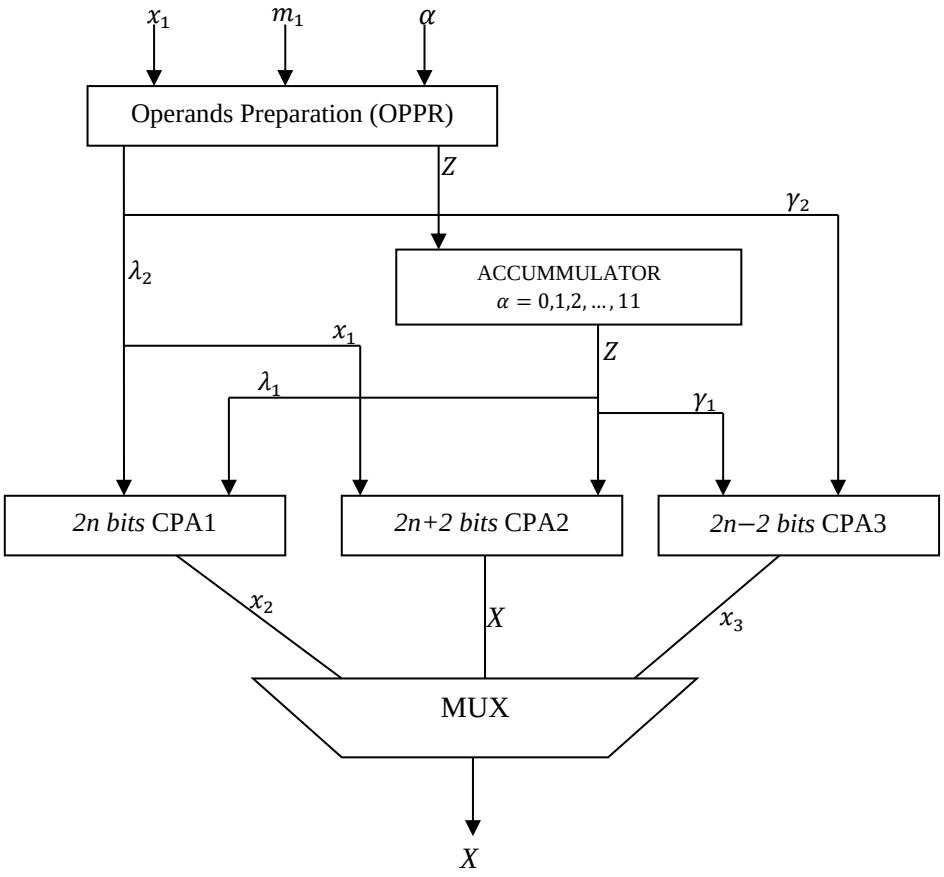



**Fig. 4. Schematic Diagram of Proposed Reverse Converter**

## 6. PERFORMANCE ANALYSIS

Fig.4 is the schematic diagram of the proposed scheme. It is a simple architecture that utilizes modular computations in an iterative manner supported by carry propagation adders (CPAS). As can be seen in the diagram, the hardware requirement includes 1 accumulator with delay of 1 and no area, 3CPAS and a MUX. The accumulator keeps track of the iterative results and the CPAS perform the summations. CPA 1 has no delay with an area requirement of 2n. Similarly, CPA 2 has delay of 2n+2 and also 2n+2 area. CPA 3 area requirement is 2n-2 with no delay. The overall effect is a design of an efficient and high-speed converter with overflow detection in terms of area cost and delay.

## 7. CONCLUSION

The paper presented an efficient Reverse Converter with Overflow Detection scheme. Modular computations with simple iterations defined the conversion process. The scheme was successfully implemented on moduli set with non-coprime factors and moduli set with common factors. Analysis and simulated results showed that the converter detects Overflow and has low delay and small area requirement. These findings can help prepare RNS for use in general purpose computing.

## 8. ACKNOWLEDGEMENT

I am thankful to the numerous authors and friends who have contributed knowledge to this work

## 9. REFERENCES

[1] Skanvantzos, A. and Wang, Y. 1999 Implementation issues of the two-level residue number system with pair of conjugate moduli". IEEE Transactions on Signal Processing. Vol.47, No.3, March 1999.

[2] Koc, C.K. 1991. An Inproved Algorithm for Mixed-Radix Conversion of Residue Numbers. Computers and Maths. Applic., Vol.22, no.8, pp.63-71.

[3] Papocheristou, C.R. 1977 Characteristic Measures of Switching Function. Inform. Sci., vol.13, pp.51-75.

[4] Taylor, F.J 1984. Residue Arithmetic: A tutorial with examples. IEEE Computer Magazine, vol.17, pp.50-62..

[5] Yassine, H.M. 1992. Matrix mixed-radix conversion for RNS arithmetic architecture. 34th Midwest Symposium on Circuits and Systems, pp.273-278.

[6] Igarashi. 1979. An Improved Lower Bound on the Maximum Number of Prime Inplicants. Trans. IECE, Japan, vol. E-62, pp.389-394.

[7] Gbolagade, K.A and Cotofana, S.D. 2008 MRC Technique for RNS to Decimal Conversion for the moduli set {2n+2, 2n+1, 2n}". 16th Annual Workshop on Circuits, Systems and Signal Processing, pp.318-321, Veldhoven, The Netherlands.

[8] Gbolagade, K.A. 2010. Effective Reverse Conversion in Residue Number System Processors. PhD Thesis, Delft University of Technology The Netherlands, PP. 15.

[9] Gbolagade, K.A and Cotofana, S.D. 2009. Residue-to-decimal converters for moduli set with common factors". 52nd IEEE International Midwest Symposium on Circuits and Systems (MINSCAS, 2009), PP.624-627.

[10] Abdallah, M. and Skavantzos, A. 1997 (On the binary

quadratic residue number system with non-coprime moduli". IEEE Transactions on Signal Processing, Vol.45,No.8.

[11] Sheu, M., Lin, S., Chen, C. and Yang, S. 2004. An Efficient VLSI Design for Residue to Binary Converter for General Balance Moduli ($2^n$-3, $2^n$-+1, $2^n$-1, $2^n$+3)". IEEE Transactions on Circuits and Systems- II. Express Briefs, Vol.51, no.3.

[12] Soderstand, M.A., Jenkins, W.K., Jullien G.A and Taylor, F.J. 1986 Residue Number System Arithmetic: Modern Application in Digital Signal Processing". IEEE press, New York, 1986.

[13] Daabo, M.I and Gbolagade, K.A. 2012 RNS Overflow Detection Scheme for the Moduli Set {M-1, M}. Journal of computing, Vol. 4, Issue 8 pp.39-44. ISSN (Online) 2151-9617.

[14] M.I. Daabo and Gbolagade, K.A 2012 Overflow Detection Scheme in RNS Multiplication Before Forward Conversion. Journal of computing, Volume 4, Issue 12, pp. 13-16 ISSN (Online) 2151-9617.

[15] Chakraborti, N.B., Soundararajan, S and Reddy, A.L.N. 1986 An Implementation of Mixed Radix Conversion for Residue Number Application, IEEE Transactions on Computers, Vol. c-35, no. 8.

[16] Szabo, N.S. and Tanaka, R.I. 1967 Arithmetic and Its Application to Computer Technology. New York: McGraw-Hill.

[17] Ananda. P.V., Mohan, 2002 Residue Number system: Algorithms and Architecture. Kluwer Academic New York.

[18] Stouratitis, T. and V. Paliouras, V. " Considering the Alternatives in Low-Power Design". IEEE

[19] Chren, W.A Jr. 1990 A new Residue Number System Division Algorithms". Comput.Math. Appl., Vol.19, no.7, pp.13-29.