*Transactions of the Nigerian Association of Mathematical Physics*
***Volume 1, (November, 2015*), pp 127 – 132**
***© Trans. of NAMP***

# A Combined Scheme for Overflow Detection And Reverse Conversion for the Moduli Set $\{2^{2n}-1,\ 2^{2n}, 2^{2n}+1\}$

***[1]M. I. Daabo, [2]K. A. Gbolagade and [3]P.A. Agbedemnab***

**[1,3]Department of Computer Science, Faculty of Mathematical Sciences, University for Development Studies Navrongo, Ghana.**
**[2]Department of Computer , Library and Information Science, College of Information and Communication Technology, Kwara State University, Nigeria.**

## *Abstract*

***In this paper, we proposed an overflow detector and a reverse converterfor the sum of operands for the moduli set*** $\{2^{2n}-1,\ 2^{2n},\ 2^{2n}+1\}$***. The numbers in the legitimate range*** $[0, M-1]$ ***are distributed in to groups. The group numbers of operands are used to detect overflow. The proposed scheme utilizes the Remainder Theorem and the group numbers to perform reverse conversion. The new scheme has a larger dynamic range of 6n bits compared with similar schemes. As a result, the proposal has improved delay at the expense of area cost when compared with other similar existing schemes.***

**Keywords:** Residue Number System, Reverse Converter, Remainder Theorem, Overflow Detection, Group Number

## 1.0 Introduction

In digital computing, carry propagation is a serious speed limiter. Unfortunately, this phenomenon grossly dominates the weighted number system (WNS) arithmetic. To enhance system performance, Residue Number System (RNS), which is without carry propagation is considered as an alternative candidate to the WNS and has been widely used in addition and multiplication dominated digital signal processing applications [3, 8, 15, 16]. In addition, RNS has proved to be one of the most popular techniques for reducing power dissipation and computation load in Very Large Scale Integrated Circuits (VLSI) [20]. Hence the advantages of RNS over the WNS include parallelism, fault tolerance, low power dissipation and high speed computations and are well documented in [6, 13, 14, 20]. It is interesting however to know that, RNS has not found widespread usage in general purpose computing due to the difficulties in performing sign detection, magnitude comparison, overflow detection, moduli selection and data conversions [13, 18, 21].

Residue-to-binary conversion and vice versa are required in almost all applications employing residue arithmetic [2]. Traditionally, most of the existing reverse converters are built on principles based on either the Chinese Remainder Theorem (CRT) [10, 11] or the Mixed Radix Conversion (MRC) [7, 9, 10, 16]. However, converters built on the CRT approach involve the large modulo-M operations 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 series of arithmetic operations and can cause delay.

In this paper, we proposed a technique that can perform reverse conversion and also detect overflow in the moduli set $\{2^{2n}-1,\ 2^{2n},\ 2^{2n}+1\}$. This moduli set is an extension of the traditional moduli set proposed in [15] to detect overflow using the group numbers approach. The new method utilizes the group numbers idea and the principles of the Remainder Theorem (RT) to perform reverse conversion. The method detects overflow in the moduli set for the sum of operands by distributing the numbers in the system dynamic range into groups. The group numbers of operands are then used to establish the overflow conditions. Theoretically, the proposed scheme has improved delay with moderate increase in area cost.

The rest of the article is organized as follows: Section 2 presents the proposed algorithms. The reverse conversion is described in Section 3. In Section 4, the hardware implementation of the architecture is described. The proposed scheme is compared with other existing schemes in terms of performance in Section 5, while the paper is concluded in Section 6.

Corresponding author: M. I. Daabo, E-mail: daabo2005@yahoo.com, Tel.: +2348109668798 (K.A.G)

## 2.0 Proposed Algorithms

**Algorithm I: Overflow Detection Scheme**

The algorithms below can be used to detect overflow and perform reverse conversion in an RNS system. Given an RNS number ($x_1$, $x_2$, $x_3$) with respect to the moduli set $\{2^{2n} - 1,\ 2^{2n},\ 2^{2n} + 1\}$, then we can compute the group number of the corresponding integer X as follows;

**Compute ω = $|x_1 - x_3|_{2^{2n}-1}$;**

**Compute ώ by giving the value of ω 1 bit right rotation;**

**Compute β = $|x_2 - x_3|_{2^{2n}}$;**

**Compute α = $|ώ - β|_{2^{2n}-1}$;**

The group number g(X), of any integer X, with residues ($x_1$, $x_2$, $x_3$) is given as g(X) = α + 1;

Given two integers, X and Y the sum (X + Y) will be in the overflow region if the sum of their group numbers is greater than $2^{2n}$ that is g(X) + g(Y)> $2^{2n}$. Overflow will not occur in the sum of X andY if g(X) + g(Y) $< 2^{2n}$.

If g(X) + g(Y) = $2^{2n}$, then we further determine g(Z) and compare the results with $2^{2n-1}$. $2^{2n-1}$is 1bit right rotation of $2^{2n}$. In this circumstance, if g(Z) $> 2^{2n-1}$ then there is no overflow otherwise overflow has occurred.Z =$|X + Y|_M$.

**Algorithm II: Residue to Binary Converter**

**Compute the floor of the number X as λ = $\beta$ + α($2^{2n}$)**

**Compute the decimal number X as X = λ($2^n$+1) + $x_3$**

Fig.1below shows the schematic diagram of the proposed scheme. From Fig.1, the number of groups required for the distribution is γ and is given by the expression

$$\gamma = \left||x_1 - x_3|_{2^{2n}-1} - |x_2 - x_3|_{2^{2n}}\right|_{2^{2n}} = 2^{2n} - 1 \qquad (1)$$

Equation (2) below can be used to compute the length of any group L, that will make the distribution.

$$L = \frac{(2^{2n}-1)(2^{2n})(2^{2n}+1)}{(2^{2n}-1)} = (2^{2n})(2^{2n} + 1) \qquad (2)$$

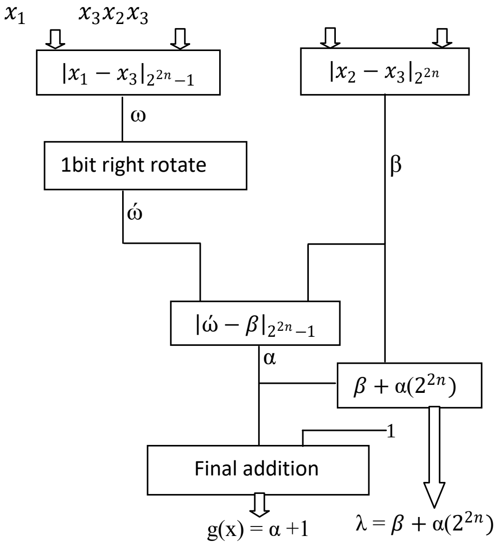

Fig.1:Block Diagram of Proposed Scheme

In any of these groups there are $2^{2n}$ sub groups of β and β takes values in the range below.

$$\beta = |x_2 - x_3|_{2^{2n}} \quad 0 \le \beta \le 2^{2n} - 1 \qquad (3)$$

Examples of values of β for numbers in the first group with range [0,$2^{4n}$+$2^{2n}$) are shown as follows:

$$\beta = |x_2 - x_3|_{2^{2n}} = \begin{cases} 0 \le X < 2^{2n} + 1 & \beta = 0 \\ 2^{2n} + 1 \le X < 2(2^{2n} + 1) & \beta = 1 \\ \vdots & \\ (2^{2n} - 1)(2^{2n} + 1) \le X < 2^{2n}(2^{2n} + 1) & \beta = 2^{2n} - 1 \end{cases} \qquad (4)$$

To determine the group number of any residue number, first we get the value of ω. For clarity, this is exhibited in the range [0,$2^{4n}$+$2^{2n}$) as follows:

$$\omega = |x_1 - x_3|_{2^{2n}-1} = \begin{cases} 0 \le X < 2^{2n}+1 & \omega = 0 \\ 2^{2n}+1 \le X < 2(2^{2n}+1) & \omega = 2 \\ 2(2^{2n}+1) \le X < 3(2^{2n}+1) & \omega = 4 \\ \vdots & \\ (2^{2n-1}-1)(2^{2n}+1) \le X < 2^{2n-1}(2^{2n}+1) & \omega = 2^n - 2 \\ 2^{2n-1}(2^{2n}+1) \le X < (2^{2n-1}+1)(2^{2n}+1) & \omega = 1 \\ \vdots & \\ (2^{2n}-2)(2^{2n}+1) \le X < (2^{2n}-1)(2^{2n}+1) & \omega = 2^n - 3 \\ (2^{2n}-1)(2^{2n}+1) \le X < 2^{2n}(2^{2n}+1) & \omega = 0 \end{cases} \qquad (5)$$

From Equation (5), the values of ω can be obtained as follows:
Even values of $\omega$:0, 2, 4, …,$2^{2n}$-2
Odd values of $\omega$:1, 3, …,$2^{2n}$-3
The obtained values of $\omega$ are then given 1 bit right rotate to perform modular subtraction and that gives $\acute{\omega}$ = 0,1,2,…,$2^{2n}$-3, $2^{2n}$-2.With the values of β and $\acute{\omega}$ known, the group number of any residue number in RNS is defined as:

$\alpha = |\acute{\omega} - \beta|_{2^n-1} 0 \le \alpha \le 2^{2n} - 2$ (6)

For purposes of implementation of the proposed algorithm, we simply add one (1) to the obtained group number in (6). In this case, if X is an integer, with residues ($x_1$, $x_2$, $x_3$), then its group number g(X) is given by :

$g(X) = \alpha+1, \quad 1 \le g(X) \le 2^{2n} - 1.$ (7)

Table 1 below shows the distribution of the numbers in dynamic range [0, $2^{6n}$-$2^{2n}$) which is given as a product of the elements in the moduli set {$2^{2n}$-1, $2^{2n}$, $2^{2n}$+1}.

**Table I**: Distribution of Numbers in Dynamic Range

| Number | Group |
|---|---|
| 0 ⟶ $2^{2n}(2^{2n}+1)$ -1 | 1 |
| $2^{2n}(2^{2n}+1)$ ⟶ 2[$2^{2n}(2^{2n}+1)$] -1 | 2 |
| ⋮ | |
| ($2^{2n}$-2)$2^{2n}(2^{2n}+1)$] ⟶ ($2^{2n}$-1) [$2^{2n}(2^{2n}+1)$] -1 | γ |

**Theorem 1:**Let X and Y represent any two operands in the process of addition, Z=X + Y and g(X) and g(Y) are the group numbers of the operands respectively, then the following hold true:

i. If g(X) + g(Y) <$2^{2n}$, no overflow will occur.
ii. If g(X) + g(Y) >$2^{2n}$, overflow must occur.
iii. If g(X) + g(Y) = $2^{2n}$, overflow may or may not occur. So, it requires further investigations and this will be described later.

**Proof**: In case iii, the range of the sum, X + Y in binary system from Table1is given by

($2^{2n}$-2) [$2^{2n}$($2^{2n}$+1)] ≤ Z ≤ $2^{2n}$[$2^{2n}$($2^{2n}$+1)] – 2 (8)

Since, M is exactly located in the middle of the obtained range, (8) can be written as

($2^{2n}$-2)[$2^{2n}$($2^{2n}$+1)] ≤ M ≤ $2^{2n}$[$2^{2n}$($2^{2n}$+1)] – 2 (9)

In order to proof g(X) + g(Y) = $2^{2n}$, we replace the values of ($2^{2n}$-2) and $2^{2n}$in terms of g(X)+g(Y). Therefore, the final form of (9) is

((g(X) -1) + (g(Y)-1)) $2^{2n}$($2^{2n}$+1)< ($2^{2n}$-1) [$2^{2n}$($2^{2n}$+1)]< (g(X) + g(Y)) [$2^{2n}$($2^{2n}$+1)](10)

From (10), the term $2^{2n}$($2^{2n}$+1) is common in all the sides of the inequality, thus it can be eliminated as follows:

g(X) + g(Y) -2 <$2^{2n}$-1 < g(X) + g(Y) (11)

After adding one to each term in (11), the resulting inequality can be defined as

g(X) + g(Y) - 1<$2^{2n}$ <g(X) + g(Y) +1 (12)

Finally (12) can be divided into two parts, that is

$$\begin{cases} g(X) + g(Y) < 2^{2n} + 1 \\ \qquad \Longrightarrow \; g(X) + g(Y) = 2^{2n} \\ g(X) + g(Y) > 2^{2n} - 1 \end{cases} \qquad (13)$$

We can be detect by comparing the sum of the groups of operands with $2^{2n}$. If the sum of groups of operands exceeds$2^{2n}$, overflow must occur otherwise no overflow will occur. Overflow possibility should be further investigated in the third mode. In this case, g(X) + g(Y) = $2^{2n}$ is given 1-bit right rotate as $2^{2n}/2 = 2^{2n-1}$. The results is subsequently compared with the group number of sum of operands g(Z). In this case, if g(Z) >$2^{2n-1}$ then overflow has not occurred otherwise there is overflow.

## 3.0 Reverse Conversion

**Theorem 2:** 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 (14)
Substitute x = b in
implies f(b) = q(b)(b-b)+R
f(b)=R.
Hence the remainder, R=f(b)

**Proposition1.** If X is a decimal number representing the RNS number $(x_1,\ x_2,\ x_3)$ with respect to the moduli set $\{2^{2n}-1,\ 2^{2n},\ 2^{2n}+1\}$, then X = λ$(2^{2n}-1)$+ $x_3$, where λ is the floor of X defined by λ = $\beta+$ α$(2^{2n})$. $\beta$ is a subgroup number and α is the group number of x.

**Proof:** It can be proved by mere substitution of X for f(x), $(2^{2n}+1)$ for q(x) and λ for (x-b) in Equation (14). Hence X = λ$(2^{2n}+1)$ + $x_3$ (15)

## 4.0 Hardware Implementation

To successively implement the proposed scheme, the following Lemmas are used.

**Lemma 1:** Modulo $2^s$ of a number is equivalent to s Least Significant Bit LSBs of the number [10].

**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 $(2^s-1)$ [10].

**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 [10].

Let $X=z_1+z_2+z_3$ (16)
where
$z_1=2^{2n}$λ,$z_2=\lambda$ and$z_3=x_3$ (17)
But$\lambda=\beta+2^{2n}\alpha$, (18)
$\beta=|x_2-x_3|_{2^{2n}}$, $\alpha=|\acute{\omega}-\beta|_{2^{2n}-1}$; (19)
Where $\acute{\omega}$ is a one-bit right rotate of $\omega=|x_1-x_3|_{2^{2n}-1}$
The binary representation of equation (16) – (19) is

$$\beta=\left|\overbrace{x_{2,2n-1}x_{2,2n-2}\dots x_{2,1}x_{2,0}}^{2n-bits}+\underbrace{\bar{x}_{3,2n-1}\bar{x}_{3,2n-2}\dots\bar{x}_{3,1}\bar{x}_{3,0}}_{2n-bits}\right|_{2^{2n}}=\beta_{2n-1}\beta_{2n-2}\dots\beta_1\beta_0 \quad (20)$$

$$\omega=\left|\overbrace{x_{1,2n}x_{1,2n}\dots x_{1,1}x_{1,0}}^{2n-bits}+\underbrace{\bar{x}_{3,2n-1}\bar{x}_{3,2n-2}\dots\bar{x}_{3,1}\bar{x}_{3,0}}_{2n-bits}\right|_{2^{2n}-1}=\omega_{2n-1}\omega_{2n-2}\dots\omega_1\omega_0 \quad (21)$$

$\acute{\omega}$is a one-bit right rotate of $\omega$ modulo $2^{2n}-1$ which gives;
$\acute{\omega}=|\omega_{2n-2}\omega_{2n-3}\dots\omega_0\omega_{2n-1}|_{2^{2n}-1}$ (22)
$=\acute{\omega}_{2n-1}\acute{\omega}_{2n-2}\dots\acute{\omega}_1\acute{\omega}_0$
Therefore,

$$\alpha=\left|\overbrace{\acute{\omega}_{2n-1}\acute{\omega}_{2n-2}\dots\acute{\omega}_1\acute{\omega}_0}^{2n-bits}+\underbrace{\bar{\beta}_{2n-1}\bar{\beta}_{2n-2}\dots\bar{\beta}_1\bar{\beta}_0}_{2n-bits}\right|_{2^{2n}-1}=\alpha_{2n-1}\alpha_{2n-2}\dots\alpha_1\alpha_0 \quad (23)$$

For (18), since $\beta$ is an2n-bit number, it will concatenate with $2^{2n}\alpha$ which implies no hardware needed to implement: that is
$\lambda=\beta+2^{2n}\alpha$

$$=\underbrace{\alpha_{n-1}\alpha_{n-2}\dots\alpha_1\alpha_0}_{2n-bits}\overbrace{00\dots0}^{2n-bits}\bowtie\underbrace{\beta_{n-1}\beta_{n-2}\dots\beta_1\beta_0}_{2n-bits}$$

$$=\underbrace{\underbrace{\alpha_{n-1}\alpha_{n-2}\dots\alpha_1\alpha_0}_{2n-bits}\underbrace{\beta_{n-1}\beta_{n-2}\dots\beta_1\beta_0}_{2n-bits}}_{4n-bits} \quad (24)$$

For Equation (17),
$z_1=2^{2n}$λ, Implies,

$$z_1=\underbrace{\underbrace{\lambda_{4n-1}\lambda_{4n-2}\dots\lambda_1\alpha_0}_{4n-bits}\underbrace{00\dots0}_{2n-bits}}_{6n-bits} \quad (25)$$

$$z_2 = \underbrace{\lambda_{4n-1}\lambda_{4n-2} \dots \lambda_1\alpha_0}_{4n-bits} \quad ` \qquad (26)$$

$z_3 = x_3$, Implies,

$$z_3 = \underbrace{z_{3,2n-1}z_{3,2n-2} \dots z_{3,1}z_{3,0}}_{2n-bits} \qquad (27)$$

Now, for Equation (16),

$$X = \underbrace{\underbrace{z_{1,3n-1}z_{1,3n-2} \dots z_{1,1}z_{1,0}}_{6n-bits} + \underbrace{z_{2,2n-1}z_{2,2n-2} \dots z_{2,1}z_{2,0}}_{4n-bits} + \underbrace{z_{3,n-1}z_{3,n-2} \dots z_{3,1}z_{3,0}}_{2n-bits}}_{6n+1-bits} \qquad (28)$$

The architecture of the scheme is based on equations (20) – (28)and described in Fig. 2.

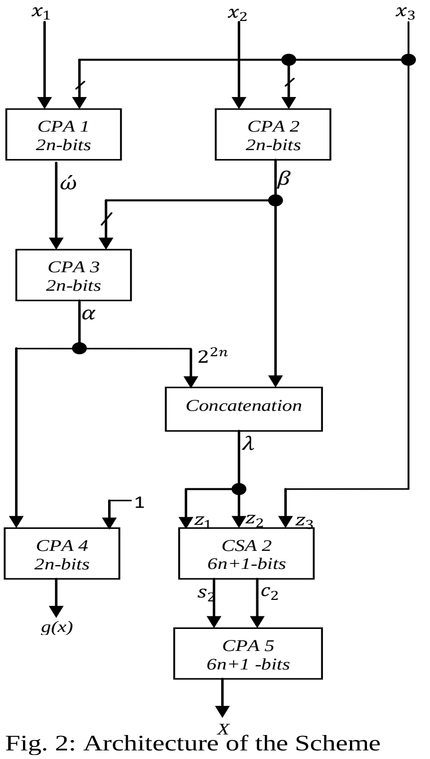

Fig. 2: Architecture of the Scheme

## 5.0 Performance Analysis

**Table II:** Comparison of Area and Delay

| **Design** | **Area** | **Delay** |
|---|---|---|
| [4] | 37n + 18 | 16*n* + log *n* + 13 |
| [15] | 76*n* + (33/2)*n* +log2 *n* | 6log2 *n* + 23 |
| Our Proposal | 20n + 2 | $14n + 2$ |

From Table II, our proposed scheme is compared with the schemes proposed in [4] and[15] in terms of area and delay. As can be seen in Table II, our proposal is faster thanthe previously proposed schemes with a moderate increase in hardware cost. In addition, our scheme detects overflow and performs reverse conversion which is a unique feature that makes it different from other existing schemes.Also, the proposed scheme has a dynamic range of 6n bits which is better than 3n bits dynamic range proposed in [4] and [15].

## 6.0 Conclusion

In this paper, we extended the dynamic range of the proposal suggested in [15] from 3n bits to 6n bits and then proposed an overflow detector and a reverse converter for the moduli set $\{2^{2n} - 1,\ 2^{2n},\ 2^{2n} + 1\}$ for the sum of operands. As in [15], the numbers in the legitimate range $[0, M-1]$ are distributed in to groups. The group numbers of operands are then used to detect overflow. The proposed scheme also utilizes the Remainder Theorem and the group numbers of operands to perform reverse conversion. Increased in the dynamic range in this case has improved the performance of the RNS architecture since more numbers have been captured within the legitimate range. This is the first step normally taken in addressing the problem of overflow in RNS system. As a result, the proposal has achieved improved delay at the expense of moderately increase in area cost compared with similar existing schemes.

## 7.0 References

[1] BI. Shao-quiang and W. J. Groos, "Efficient residue comparison algorithm for general Moduli sets", IEEE International Circuits and Systems, 2005, pp. 1601-1604.

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

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

[4] D. Younes and P. Steffan, " Universal Approaches for Overflow and Sign Detection in Residue Number System Based on $\{2^n– 1, 2^n, 2^n+ 1\}$" The Eighth International Conference on Systems, 2013.

[5] E. Gholami, R. Farshidi, M. Hosseinzadeh and K. Navi, "High speed residue number system comparison for the moduli set $\{2^n-1, 2^n, 2^n+1\}$", Journal of communication and computer, Vol. 6, No. 3, 2009, pp. 40-46.

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

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

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

[9] K. A. Gbolagade and S.D. Cotofana, " 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, November 2008.

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

[11] M. Abdallah and A, Skavantzos, (On the binary quadratic residue number system with non-coprime moduli". IEEE Transactions on Signal Processing, Vol.45,No.8, August 1997.

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

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

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

[15] M. Rouhifar, M. Hosseinzadeh S. Bahanfar and M. Teshnehlab. Fast Overflow Detection in Moduli Set $\{2^n, 2^n-1, 2^n+1\}$. International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011 ISSN (Online): 1694-0814

[16] M. Sheu, S. Lin, C. Chen and S. Yang, " 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, March 2004.

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

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

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

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

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