Vol. 5, No. 12 December 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences
©2009-2014 CIS Journal. All rights reserved.

http://www.cisjournal.org

An Over Flow Detection Scheme with a Reverse Converter for the Moduliset $\{(2^n -1, 2^{n)}, 2^n +1\}$

[1] M.I. Daabo, [2] K. A. Gbolagade
[1, 2] Department of Computer Science, Faculty of Mathematical Sciences, University for Development Studies Navrongo, Ghana.

[1] daabo2005@yahoo.com, [2] gkazy1@yahoo.com

ABSTRACT

The quest to make Residue Number System (RNS) based processors a reality, has made researchers to continue seeking solutions to some of the limiting factors of RNS based processor realization. Overflow detection and Reverse Conversion are some of the limiting factors in RNS implementation. In this paper, we propose an overflow detection scheme with a residue to binary converter for the moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. Our proposal detects overflow for the sum of operands by distributing the numbers in the system dynamic range into groups. In the conversion segment, the proposal utilizes the Remainder Theorem (RT) in order to convert any RNS number into its decimal equivalent. Our scheme performs dual function, i.e., it detects overflow and does reverse conversion. Theoretical analysis shows that the proposed scheme performs reverse conversion and detects overflow faster than previously proposed schemes and requires lesser hardware resources.

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

1. INTRODUCTION

Carry propagation constitutes the main reason 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 (WNS), has been widely used in addition and multiplication dominated digital signal processing applications [4, 10]. RNS has also been established as one of the most popular number systems suitable for reducing power dissipation and computation load in Very Large Scale Integrated Circuits (VLSI). RNS thus have the following advantages over the conventional WNS: parallelism, fault tolerance, low power dissipation and high computational speed [8, 9]. However RNS 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 conversions [7, 8].

Conversion from RNS to WNS and vice versa is required in almost all applications employing residue arithmetic [3]. Most of the existing RNS to binary converters are built based on either the Chinese Remainder Theorem (CRT) [1] or the Mixed Radix Conversion (MRC) technique [6, 7]. However, the CRT approach involves the large modulo-M operation which makes computations generally slow. In the case of MRC, the computations are done in a sequential manner to obtain the Mixed Radix Digits (MRD$_s$). This sequential approach is a speed limiter in RNS.

This paper proposes residue to binary converter using the overflow detection scheme proposed in [10] and the Remainder Theorem (RT) approach. The new method also detects overflow in the moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ for the sum of operands by distributing the numbers in the system dynamic range into several groups. Theoretical analysis indicates that the proposed scheme outperforms known schemes.

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.

2. PROPOSED ALGORITHMS

Algorithm I: Overflow Detection Scheme

Given an RNS number (x_1, x_2, x_3) with respect to the moduli set $\{2^n - 1, 2^n, 2^n + 1\}$, then we can compute the group number of the corresponding integer X as follows;

Compute ω = $|x_1 - x_3|_{2^n-1}$;
Compute ώ by giving the value of ω 1 bit right rotation;
Compute β = $|x_2 - x_3|_{2^n}$;
Compute α = $|ώ - β|_{2^n-1}$;

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^n that is $g(X) + g(Y) > 2^n$. Overflow will not occur in the sum of X and Y if $g(X) + g(Y) < 2^n$.

If $g(X) + g(Y) = 2^n$, then we further determine g(Z) and compare the results with 2^{n-1}.2^{n-1} is 1bit right rotation of 2^n. In this circumstance, if g (Z) $>2^{n-1}$ 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 = $\beta + \alpha\,(2^n)$

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

The schematic diagram of the proposed scheme is given inFig.1below.

From Fig.1, the number of groups required for the distribution is γ and is given by the expression

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

We can compute the length of any group L, that will make the distribution, using Equation (2) below.

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

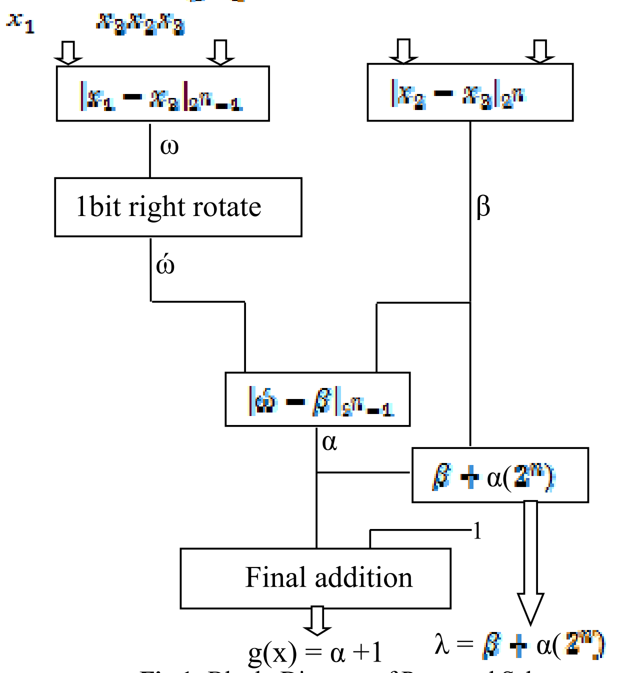

Fig 1: Block Diagram of Proposed Scheme

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

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

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

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

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:

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

From (5), the values of ω can be obtained as follows:

Even values of *ω*:0, 2, 4, …,2^n-2

Odd values of *ω*:1, 3, …,2^n-3

The values of *ω* is then given 1 bit right rotate to perform modular subtraction and that gives ώ = 0,1,2,…., 2^n-3, 2^n-2.

Now with the values of β and ώ known, the group number of any residue number in RNS is defined as:

$$\alpha = |ώ - \beta|_{2^n-1} \quad 0 \le \alpha \le 2^n - 2 \qquad (6)$$

For purposes of implementation of the proposed algorithm, we simply add one (1) to the obtained group number from (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

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

Table 1 below shows the distribution of the numbers in dynamic range $[0, 2^{3n} - 2^n)$ which is given as a product of the elements in the moduli set $\{2^n-1, 2^n, 2^n+1\}$.

Table 1: Distribution of Numbers in Dynamic Range

Number	Group
$0 \longrightarrow 2^n(2^n+1) - 1$	1
$2^n(2^n+1) \longrightarrow 2[2^n(2^n+1)] - 1$	2
$\vdots$	
$(2^n-2)[2^n(2^n+1)] \longrightarrow (2^n-1)[2^n(2^n+1)] - 1$	γ

Theorem:

If X and Y are two operands in the process of addition, Z=X + Y such that g(X) and g(Y) are the group numbers of the operands respectively, then the following overflow conditions are true:

i. If g(X) + g(Y) <2^n, no overflow will occur.
ii. If g(X) + g(Y) >2^n, overflow must occur.
iii. If g(X) + g(Y) = 2^n, 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

$(2^n-2)\,[2^n\,(2^n+1)] \le Z \le 2^n\,[2^n\,(2^n+1)] - 2$ (8)

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

$(2^n-2)\,[2^n\,(2^n+1)] \le M \le 2^n\,[2^n\,(2^n+1)] - 2$ (9)

In order to proof $g(X) + g(Y) = 2^n$, we replace the values of (2^n-2) and 2^n in terms of $g(X)+g(Y)$.

Therefore, the final form of (9) is

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

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

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

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

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

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

$$\left\{\begin{array}{l} g(X)+g(Y) < 2^n+1 \\ \qquad \Rightarrow g(X)+g(Y) = 2^n \\ g(X)+g(Y) > 2^n-1 \end{array}\right. \qquad (13)$$

Therefore, overflow can be detected by comparing the sum of the groups of operands with2^n. If the sum of groups of operands exceeds2^n, overflow must occur otherwise no overflow will occur. Overflow possibility should be further investigated in the third mode. In this case, $g(X) + g(Y) = 2^n$ is given 1-bit right rotate as $2^n/2 = 2^{n-1}$. The results is subsequently compared with the group number of sum of operands g(Z). In this case, if $g(Z) > 2^{n-1}$ then overflow has not occurred otherwise there is overflow.

3. REVERSE CONVERSION

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.

$f(x) = q(x)(x-b)+R$ (14)

Substitute x =b in implies $f(b) = q(b)(b-b)+Rf(b)=R$.

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$\{2^n - 1, 2^n, 2^n + 1\}$, then $X = \lambda(2^n - 1) + x_3$, where λ is the floor of X defined by $\lambda = \beta + \alpha(2^n)$.$\beta$ 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), $(2^n + 1)$ for q(x) and $\lambda$for (x-b) in Equation (14). Hence

$X = \lambda(2^n + 1) + x_3$ (15)

4. HARDWARE IMPLEMENTATION

The following Lemmas are important in the implementation of the proposed scheme:

Lemma 1:

Modulo 2^s of a number is equivalent to s Least Significant Bit LSBs of the number [7].

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 $(2^s - 1)$ [7].

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 [7].

Let $X = z_1 + z_2 + z_3$ (16)

where

$z_1 = 2^n\lambda$, $z_2 = \lambda$ and$z_3 = x_3$ (17)

But

$\lambda = \beta + 2^n\alpha$, (18)

$\beta = |x_2 - x_3|_{2^n}$, $\alpha = |\acute{\omega} - \beta|_{2^n-1}$,

$\omega = |x_1 - x_3|_{2^n-1}$ (19)

Where $\acute{\omega}$ is a one-bit right rotate of $\omega = |x_1 - x_3|_{2^n-1}$

The binary representation of equations (16) – (19)

Is

$$\beta = \left| \overbrace{x_{2,n-1}x_{2,n-2}\cdots x_{2,1}x_{2,0}}^{n-bits} + \underbrace{\bar{x}_{3,n-1}\bar{x}_{3,n-2}\cdots\bar{x}_{3,1}\bar{x}_{3,0}}_{n-bits} \right|_{2^n}$$

Vol. 5, No. 12 December 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences
©2009-2014 CIS Journal. All rights reserved.

http://www.cisjournal.org

$$= \beta_{n-1}\beta_{n-2}\cdots\beta_1\beta_0 \quad (20)$$

$$\omega = \left|\overbrace{x_{1,n-1}x_{1,n-2}\cdots x_{1,1}x_{1,0}}^{n-bits} + \underbrace{x_{3,n-1}x_{3,n-2}\cdots x_{3,1}x_{3,0}}_{n-bits}\right|_{2^n-1}$$
$$= \omega_{n-1}\omega_{n-2}\cdots\omega_1\omega_0$$

(21)

$\acute{\omega}$is a one-bit right rotate of ω modulo 2^n-1 which gives;

$$\acute{\omega} = \left|\omega_{n-2}\omega_{n-3}\cdots\omega_0\omega_{n-1}\right|_{2^n-1} = \acute{\omega}_{n-1}\acute{\omega}_{n-2}\cdots\acute{\omega}_1\acute{\omega}_0$$

(22)

Therefore,

$$\alpha = \left|\overbrace{\acute{\omega}_{n-1}\acute{\omega}_{n-2}\cdots\acute{\omega}_1\acute{\omega}_0}^{n-bits} + \underbrace{\beta_{n-1}\beta_{n-2}\cdots\beta_1\beta_0}_{n-bits}\right|_{2^n-1}$$
$$= \alpha_{n-1}\alpha_{n-2}\cdots\alpha_1\alpha_0$$

(23)

For (17), since β is an n-bit number, it will concatenate with $(2^n)\alpha$ which implies no hardware needed to implement:

That is $\lambda = \beta + 2^n\alpha$

$$= \underbrace{\alpha_{n-1}\alpha_{n-2}\cdots\alpha_1\alpha_0}_{n-bits}\overbrace{00\cdots0}^{n-bits} \bowtie \underbrace{\beta_{n-1}\beta_{n-2}\cdots\beta_1\beta_0}_{n-bits} =$$
$$\underbrace{\underbrace{\alpha_{n-1}\alpha_{n-2}\cdots\alpha_1\alpha_0}_{n-bits}\underbrace{\beta_{n-1}\beta_{n-2}\cdots\beta_1\beta_0}_{n-bits}}_{2n-bits}$$

(24)

For Equation (17),

$z_1 = 2^n\lambda$ implies

$$z_1 = \underbrace{\underbrace{\lambda_{2n-1}\lambda_{2n-2}\cdots\lambda_1\alpha_0}_{2n-bits}\underbrace{00\cdots0}_{n-bits}}_{3n-bits} \quad (25)$$

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

$z_3 = x_3$ implies

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

Now, for Equation (16),

$$X =$$
$$\underbrace{\underbrace{z_{1,3n-1}z_{1,3n-2}\cdots z_{1,1}z_{1,0}}_{3n-bits} + \underbrace{z_{2,3n-1}z_{2,3n-2}\cdots z_{2,1}z_{2,0}}_{3n-bits} + \underbrace{z_{3,n-1}z_{3,n-2}\cdots z_{3,1}z_{3,0}}_{n-bits}}_{3n+1-bits}$$

(28)

The architecture of the scheme is based on equations (20) – (29) and described in Fig.2 below.

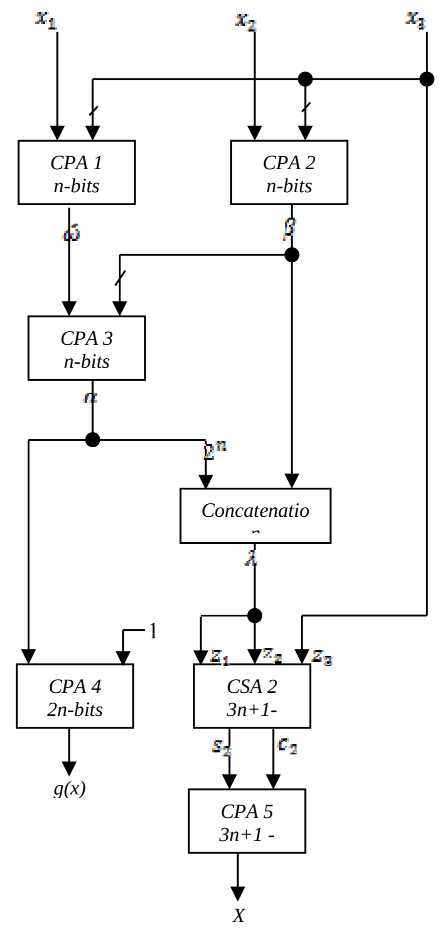

Fig 2: Architecture of the Scheme

5. PERFORMANCE ANALYSIS

Table 2: Comparison of Area and Delay

Design	Area	Delay
[5]	37n + 18	16n + log n + 13
[10]	76n + (33/2)n +log2 n	6log2 n + 23
Our Proposal	10n + 2	$11n + 2$

In Table 2, our proposed scheme is compared with the schemes proposed in [5, 10] in terms of area and delay. As can be seen from Table II, our proposal is faster than the previously proposed schemes with significant reduction in hardware cost. Also, our scheme detects overflow and performs reverse conversion which is a unique feature that makes it different from other existing schemes.

Vol. 5, No. 12 December 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences
©2009-2014 CIS Journal. All rights reserved.

http://www.cisjournal.org

6. CONCLUSION

Detecting overflow and performing reverse conversion are of most importance in realizing the full implementation of RNS. In this paper, we proposed a new scheme for performing reverse conversion for the moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. The proposal is based on using group numbers and the Remainder Theorem to perform the reverse conversion. Our proposed scheme also detects overflow using the group number approach. The scheme, which performs dual function, has significant reduction in delay and area compared to the schemes proposed in [5] and [10].

REFERENCES

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

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

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

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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