Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I,
IMECS 2012, March 14 - 16, 2012, Hong Kong

# Non-iterative RNS Division Algorithm

Mansoureh Labafniya[1], Mohammad Eshghi[2]

**Abstract: Until now many algorithms for division operation in residue number systems are presented but almost all of them have an overall loop, conversion from RNS to binary or mixed radix system or using LUT and exclude some numbers in the range of acceptable inputs as a denominator in division operation. In this paper, a non-iterative algorithm for division in RNS system is presented in which all numbers as denominator are accepted. Comparison based on more time consuming operation , modular multiplication , between proposed algorithm in this paper and [12] shows more than 2 times increase in speed and efficiency during divide operation.**

***Index Terms*— residue number system; division; multiplicative inverse; Euclidean algorithm; base extension**;

## 1. INTRODUCTION

Residue number system is used to present natural numbers using their remainders. In this system, some operations, such as addition and multiplication, are performed in parallel on remainders of inputs. Division, sign detection and number comparison are difficult operations in residue number systems. This weakness has limited the RNS to add, subtract and multiply operations.

As a result, RNS is not used in many applications that need a divide operation. [1-4]

Many algorithms for division in RNS are presented.

Some of these iterative algorithms work by subtracting denominator from numerator in a major loop, until numerator gets less than denominator. Quotient is equal to the number of iterations of this major loop. Some of them use Newton iteration to compute reciprocal and then compute quotient. [5,6]

Another common way for division is using the definition of division. In this algorithm, first the position of the most-significant non-zero bit in the divisor and dividend is determined. Then, according to difference between these two positions, divisor is shifted to the left and is subtracted from dividend. These actions are repeated in a major loop until the result is smaller than divisor .

In some methods for dividing $X$ to $Y$, first the proper $2^k$ is detected such that $Y.2^k \leq X \leq Y.2^{k+1}$. In the next iterations, these two margins varied until quotient obtained. [7-10]

There is another method in which, instead of dividing two proposed numbers, X and Y, two different numbers which have the same ratio and are less than X and Y, are chosen. For doing this, some new moduli are introduced, and at last, in a major loop, the division result is calculated [11, 12]

These algorithms have three major deficiencies. First, all of them have an overall loop which increases the complexity and delay of the algorithm. Second, some of these methods exclude some numbers in the range of acceptable inputs as a denominator in division operation. Third, they have some operations in the binary or mixed radix system or use a lookup table to perform an RNS division.

In order to solve problems one and two, in this paper a non-iterative division algorithm is proposed.

The rest of paper is organized as follows. In Section *2*, the non-iterative division algorithm is described. A comparison between non-iterative algorithm in this paper and iterative division algorithms in [12], is presented in Section 3.

## 2. A NON-ITERATIVE AND PURE RNS DIVISION ALGORITHM

During previous section many methods for division operation in RNS system introduced. These methods have some problems such as complexity, time consuming and nested loops for inputs. In this section a new method is introduced that solves these problems.

Consider $X=(x_1,x_2,\dots,x_L)$ and $Y=(y_1,y_2,\dots,y_L)$ as inputs of the algorithm1. Modulo set is $\{m_1,m_2,\dots,m_L\}$ and output of algorithm is $K=(k_1,k_2,k_3,\dots,k_L)$ as quotient and $R=(r_1,r_2,\dots)$ as remainder.

Algorithm 1: the proposed pure RNS division algorithm

Input: $X=(x_1,x_2,\dots,x_L)$ , $Y=(y_1,y_2,\dots,y_L)$ and modulo set $\{m_1,m_2,\dots,m_L\}$

Output: K=X/Y and R=(X mod Y) in RNS

Condition: all moduli $\{m_1,m_2,\dots,m_l\}$ are prime numbers but 2.

Begin

1. Calculate $Y^{-1}$= multiplicative inverse of Y
2. Calculate R = X mod Y
3. Calculate K= (X – R). $Y^{-1}$
4. If any $y_i$= 0,then set $k_i = (k_n \bmod m_i)$ in which $k_n \neq 0$.

End.

In Section 2.1 overall view of the proposed algorithm is presented. Modified multiplicative inverse algorithm is presented in Section 2.2. Section 2.3 discussed about calculating $|X|_Y$ in pertinent details. In Section 2.4 special problem of having 'zero' in $y_i$ is solved.

[1] Young Researchers Club, Tehran North Branch, Islamic Azad University, Tehran, Iran, m.labafniya@srbiau.ac.ir

[2] Electrical and Computer Engineering Faculty, Shahid Beheshti University, Tehran, Iran, eshghi@sbu.ac.ir

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

2.1 Overall views

Consider $X$ and $Y$ are two positive numbers. If $X>Y$ and if $X$ is dividable to $Y$, then $K=X/Y=X.Y^{1}$. Variable $Y^{1}$ is the multiplicative inverse of $Y$, that is $Y.Y^{1}=1$. It must be noticed that equation $X/Y=X\ Y^{1}$ is true when $X$ is dividable to $Y$. Equation (1) through (4) proof this.

$$X = Y.K + R \qquad (1)$$

Where $K$ is natural number as quotient and $R$ is the remainder of $X/Y$ operation.

$$X.Y^{1}=K+R.Y^{1} \qquad (2)$$

According to Eq.*(2)*, if $R=0$ then:

$$K=X/Y=X.Y^{1} \qquad (3)$$

When $X$ is not dividable to $Y$ then, we must subtract $R.Y^{1}$ from $X.Y^{1}$ which is shown in equation *(4)*.

$$K=(X-R).Y^{1} \qquad (4)$$

Example 1: Consider division of 20/4 in a RNS system which its modulo set is $\{7, 17\}$. Inputs to this RNS system are $X=20= (6, 3)$ and $Y=4 = (4, 4)$. In order to compute $X/Y$, first multiplicative inverse of $Y$ is calculated.

$Y^{1} = 4^{-1} = (2, 13)$ where $4.4^{-1} = (1, 1)$
$R= (20)_4 = 0 = (0, 0)$

Because $R=0$, $K$ is calculated according to equation (3).

$K =X/Y= X.Y^{1} = (6, 3). (2, 13) = (5, 5) = 5$ ◆

2.2 Multiplicative inverse algorithm

Multiplicative inverse of a number $Y$ which is denoted by $Y^{1}$ can be calculated using Euclidian algorithm which is used in [9] and is shown in algorithm (2).
Suppose input is in range $[0, M)$ where $M$ is the product of modulo set $\{m_1,m_2,m_3,..,m_L\}$.
Modular multiplication of $Y^{1}$ and $Y$ is according to equation *(5)*.

$Y.Y^{1} = M.c +1$ *(5)*, where c is a positive number

If one of the $m_i$ is even, then $M$ is even and $M.c +1$ is odd. If $Y$ is even there is not any $Y^{-1}$ to multiply it to $Y$ to have an odd result. In this situation there are some $Y$'s without $Y^{1}$. As a result, even moduli eliminate numbers to have a multiplicative inverse.
If all the moduli are odd, $M$ is odd. In this system the previous problem is solved, however there are some numbers without multiplicative inverse. The following example clarifies this situation.

Example 2: consider an RNS system with modulo set of $\{11, 9\}$ multiplicative inverse of $Y=21= (10, 3)$ is $21^{-1}= (10, u)$. The letter 'u' denotes an undefined component in RNS presentation of a number. In this case 'u' indicates that for *3* in modulo *9* the multiplicative inverse is undefined because $m_i=9$ is multiple of $y_i=3$. ◆

To solve these problems the RNS system has to have modulo set of prime numbers. In this condition, all numbers in the range of RNS have multiplicative inverse.

Lemma1: In a RNS system with modulo set of $\{m_1,m_2,m_3,\ldots,m_L\}$, all numbers in the range of $[0, M)$ in which $M= m_1m_2m_3\ldots.m_L$ has multiplicative inverse if and only if all moduli $m_i$ are prime numbers but 2.

Proof: According to above discussion and examples, if all $m_i$ are prime number but 2, then $M$ is an odd number. Since all $m_i$ are prime number, there is not any $y_i$ in $m_i=A.y_i$ where A is an integer number. Therefore, all numbers have multiplicative inverse.
If there is a $m_i$ which is not prime number, or 2, then there is a $y_i$ for $m_i= A.y_i$ and $y^{-1}_i$='u'. ◆

Based on lemma 1, the multiplicative inverse algorithm can be used for all numbers in the RNS system with prime moduli but 2.

Algorithm 2: the proposed modified Euclidean based multiplicative inverse algorithm

Input: $y_j$ , $m_i$
Output: $inv_j$
Condition: no condition
Begin
  $g_0 = m_i$, $g_1 = y_j \bmod m_i$, $v_0 = 0$, $v_1 = 1$;
  Loop
 If $g_1 \neq 0$
  then $h = g_0$ div $g_1$, $g_2 = g_0 \bmod g_1$
 else $inv_j = v_0$, output $inv_j$ and halt;
 $v_2 = h \times v_1$;
 $v_0 = v_0 + m_i \times ((v_2 - v_0)/m_i)$;
 $v_2 = v_0 - v_2$, $g_0 = g_1$, $g_1 = g_2$, $v_0 = v_1$, $v_1 = v_2$;
  End Loop
End

2.3 X mod Y

In order to calculate $|X|_Y$, base extension method is used. Suppose the primary modulo set is $\{m_1, m_2, m_3\}$ and $X=(x_1, x_2, x_3)$. Based on the Chinese remainder theorem, we have:

$$\begin{aligned}|X+ m_1.m_2.m_3.a|_{m4} = ||&x_1.\ m_2^{-1}.\ m_3^{-1}|_{m1}.m_2.m_3 \\ +&|x_2.\ m_1^{-1}.\ m_3^{-1}|_{m2}.m_1.m_3 \\ +&|x_3.\ m_1^{-1}.\ m_2^{-1}|_{m3}.m_2.m_1|_{m4} \\ =&x_4 \end{aligned} \qquad (6)$$

Secondary modulo set is $\{m_1, m_2, m_3, m_4\}$ in which $m_4=Y$. In this new RNS system $X=(x_1, x_2, x_3, x_4)$, and $x_4 =|X|_Y$ is unknown, which can be calculated using Eq. *7*.

$$x_4 = ||x_1.\ m_2^{-1}.m_3^{-1}|_{m1}.\ m_2.m_3 + |x_2.\ m_1^{-1}.m_3^{-1}|_{m2}.\ m_1.\ m_3 + |x_3.\ m_1^{-1}.m_2^{-1}|_{m3}.\ m_1.m_2|_{m4} \qquad (7)$$

Example 3: in a RNS with modulo set equal to $\{3, 7, 13\}$, $X=23$ is presented as $X= (2, 2, 10)$. In order to calculate $(X \bmod 5) =|X|_5$ it is not necessary to convert to binary system. The base extension technique can be used to find (X mod 5) as stated in equation (7) :
$|3^{-1}|_5 = 2;\ |7^{-1}|_5 = 3;\ |13^{-1}|_5= 2$
$|3^{-1} \times 7^{-1} \times 13^{-1} \times |X|_5 + 4|_5 = 0$
$|2\times3\times2\times|X|_5|_5=1$

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$|X|_5=3$
Therefore for modulo set $\{3, 5, 7, 13\}$, $X= (2, 3, 2, 10)$. ◆

It must be noticed that if selected Y is small enough, calculating Eq.7 will be easy but if it is not small , Eq.7 can be simplified . this simplification depends on specific application which used a special structure in binary mode for Y .

2.4 Calculating quotient
According to equation *(4),* quotient is $K= (X – R).Y^{1}$.

Example 4: calculate $X/Y =93/5$ in RNS system in which $M= \{7, 17\}$. In this system we have:

$Y=5= (5, 5)$, and $Y^{1}=5^{-1}= (3, 7)$, then
$R= (X)_Y = (93)_5=3 = (3, 3)$

According to algorithm (2), we have $X\text{-}R= (2, 8) – (3, 3) = (6, 5)$.
Since K= $(X – R). Y^{1}= (6, 5). (3, 7) = (4, 1)$, Then quotient is equal to $(4, 1)$ which in decimal system is 18. ◆

Problem occurs when our number for calculating multiplicative inverse is multiple of one of the moduli. For example if $Y=17$ and moduli are $\{7,17\}$, then $Y=17=(3,0)$ and $Y^{1}=17^{-1}=(5,'z')$. It means that the inverse of 0 is undefined, and it is denoted by 'z'. This undefined value propagates through the algorithm and make the corresponding $k_i=0$, which is not correct. In this situation $k_i$ can be calculated using other nonzero and defined $k_n$ as:
$k_i=k_n \bmod m_i$

Example 5 shows this situation.

Example 5: Consider $X/Y = 80/7$ in RNS system with modulo set of $\{7, 17\}$. In this system, $7= (0, 7)$, $7^{-1}= (z, 5)$ and $(80)_Y = (80)_7= 3$
According to algorithm (1), we have:
$(3, 12) – (3, 3) = (0, 9)$
Then, $K= 75/7= (z, 5). (0, 9) = (z, 11)$.
Since $k_1$='z', we set $k_1= (k_2 \bmod m1)$. That is
$k_1= (11)_7=4$
Therefore, $K= 80 /7= (4, 11)$. ◆

Lemma2: In division $X$ to $Y$ in a RNS system, $K=X/Y$, if $Y$ is multiple of one of the moduli, for example $m_i$, then $y_i=0$ and $k_i$ is undefined. In this situation, the proposed algorithm is executed for all moduli but $m_i$. In last step $k_i$ is calculated using other non-zero $k_n$, as:
$k_i=|k_n|_{mi}$.

Proof: Consider that the selected moduli are $\{m_1, m_2\}$ and $m_2>m_1$. The greatest input of this system is $N=m_1. m_2 -1$.

$X<m_1.m_2$
$Y=n.m_1$
$K=X/Y < (m_1.m_2/m_1) =m_2$
$k_2=|K|_{m2}=K$
$k_1=|K|_{m1}=|k_2|_{m1}$. ◆

Example 6: consider dividing $X/Y = 19/5$ in a RNS system with modulo set of $\{3, 7\}$. In this system, $5= (2, 5)$, $5^{-1}= (2, 3)$ and $(19)_Y = (19)_5=4$.
According to algorithm (1), we have:
$(1, 5) – (1, 4) = (0, 1)$
Then, $Q= 19/5= (0, 1). (2, 3) = (0, 3) =3$

## 3. ANALYSES AND COMPARISON:

In this section we compare the presented algorithm in this paper with algorithm presented in [12]. In [12] there is an iterative algorithm to calculate the division of $X$ to $Y$. All numbers in the range of RNS are not acceptable as denominator, $Y$. However, in the presented non-iterative algorithm, all numbers in the range of system can be set as a denominator.

The analysis is based on the number of modular multiplication, because modular multiplication is the most time consuming operations in this conversion. We suppose that this algorithm is implemented on the parallel processors.

According to the algorithm(1), first ($R=X$ mod $Y$) and ($Y^{-1}$= inverse of $Y$) are to be computed. According to [12] for computing R=($X$ mod $Y$), $m+1$ modular multiplications and for ($Y^{1}$= inverse of $Y$), $d$ modular multiplications are needed. Because these two operations are independent, we can compute them in parallel. Therefore, according to our proposed algorithm, maximum ($m+1$, $d$) time delay is needed. The last modular multiplication is for $K = (X – R).Y^{-1}$. The complexity of our proposed algorithm is shown in table 1.

Suppose the overall loop in the division algorithm of [12] is repeated $n$ times. Table 2 shows the complexity analysis of this algorithm.

Table 1: Complexity analyses for the proposed method

| operation | Time of modular multiplication |
|---|---|
| $R = X \bmod Y$, $Y \rightarrow (Y)^{-1}$ | $max(m+1,d)$ |
| $K = (X – R).Y^{-1}$ | $1$ |
| *Total time* | $max(m+2,d+1)$ |

Table 2: Complexity analyses for the paper [12] method

| operation | Time of modular multiplication |
|---|---|
| $a \rightarrow a'$ ; $b \rightarrow b'$ | $m+1$ |
| $b' \rightarrow (b')^{-1}$ | $d$ |
| $c' \rightarrow c$ ; $(b')^{-1} \rightarrow (b'')^{-1}$ | $m+1$ |
| $(a (a'')^{-1} – c)(p^{-1})$ ; $(b (b'')^{-1})(p^{-1})$ | 2 |
| *Total time* | $(2m+d+4).n$ |

The overall delay time of the algorithm presented in paper [12] is $(2m+d+4).n$ where our proposed algorithm has the overall delay time of $max (m+2, d+1)$. Suppose $d+1$ is greater than $m+2$. The ratio of these delays, called $s$, is:

$$s= ((2m+d+4).n) / (d+1) \tag{8}$$

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Figure 5 shows this delay ratio versus number of bits, *d*, which is varied from 2 to 32. In this figure, number of major loop iterations, *n*, is considered to be 2 and figures are for different number of moduli of the RNS system, *m*. According to figure 5, proposed algorithm in this paper has more than 2 times increase in speed and efficiency to paper [12] during divide operation.

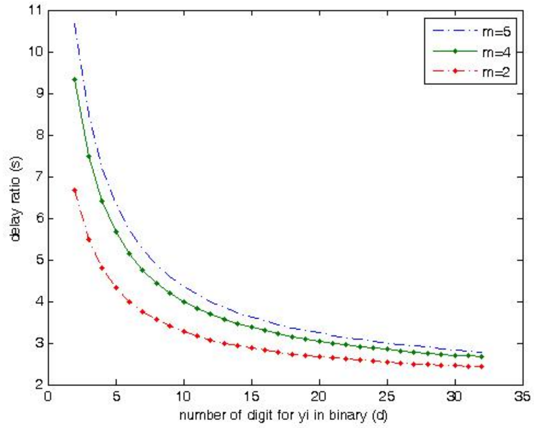


Figure 5: Ratio of delays of algorithm presented in [12] for n=2 and delay of our proposed algorithm

## 4. CONCLUSION

Many division algorithms which proposed for residue number systems, have three problems: an overall loop and not supporting all numbers in the range as a denominator.
In this paper, we proposed a non-iterative and pure RNS division algorithm which solves these problems. Comparison between presented algorithm and [12], showed more than 2 times increase in speed and efficiency during divide operation.

## REFERENCES

[1] Omondi, B.Premkumar,, ―RESIDUNEU MBER SYSTEM :theory and implementation‖, imperial college press, 2007
[2] m. labafniay, m.eshghi, "an efficient adder/subtracter circuit for one-hot residue number system", 2010 International Conference on Electronic Devices, Systems and Applications (ICEDSA2010), Kuala lumpor, Malaysia, 978-1-2010 IEEE
[3] Parhami, " computer arithmetic: algorithms and hardware designs", oxford university press, New York, 2000
[4] Somayeh Timarchi, Keivan Navi, ―Arithmetic Circuits of Redundant SUT-RNS‖, IEEE Transactions on instrumentation and measurement, VOL. 58, NO. 9, September 2009
[5] M. Hitz, E. Kaltofen, "Integer division in residue number systems", IEEE Transactions on Computers 44 (8) (1995) 983– 989
[6] E. Hussein, M. A. Hasan and M. I. Elmasry, " alow power algorithm for division in residue number system", 1998 IEEE.
[7] Hiasat, H. Abdel-Aty-Zohdy, "Design and implementation of an RNS division algorithm", Proceedings of 13th IEEE Symposium on Computer Arithmetic, Asilomar, California, USA, 1997, pp. 240–249.
[8] Jen-Hoyang , Chin-Chen Chang ,And Chien-Yuan Chen,"a high speed division algorithm in residue number system using parity checking technique",International journal of computer mathematics( 2006).
[9] Jen-Shiun Chiang, Mi Lu, "a general divisin algorithm for residue nmber systems", 1991 IEEE
[10] Mi Lu, "a novel division algorithm for the residue number system", IEEE transaction on computers, vol.41, 1992
[11] Gamberger," New approach to integer division in residue number systems", Proceedings of 10th IEEE symposium on Computer Arithmetic, Grenoble, France, 1991, pp. 84–91.
[12] Chin-Chen Chang a,*, Yeu-Pong Lai ," A division algorithm for residue numbers", applied mathematics and computation in elsevier ,2006 .


ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)