www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 9 Issue 07 July 2020, Page No. 25116-25124
ISSN: 2319-7242 DOI: 10.18535/ijecs/v9i07.4507

RNS Bases in Computer Architecture for DNA Sequence Application

[1]L.O. Olatunbosun[*], [2]A.A. Adam, [3]K.A. Gbolagade

[1]I. C. T, Department of Computer Science Federal University of Agriculture, Abeokuta, Nigeria
[2]I. C. T, Department of Computer Science Cresent University Abeokuta, Nigeria
[3]I. C. T, Department of Computer Science Kwara state University Malete. Nigeria

Abstract:
In this paper we present an RNS bases algorithm with architecture implementation for gene sequence applications. Based on the existing RNS arithmetic algorithm, investigations were made on RNS application and its inherent arithmetic advantages; data conversion algorithm from Decimal/Binary to RNS; the forward conversion; Conversion from RNS to Binary/Decimal; the reverse conversion using the Chinese remainder theorem CRT, conversion from RNS to mixed radix form with capability for effective computation performance, analysis of Smith Waterman Algorithm based on DNA sequence computing. Its limitations and open issues for future research were highlighted.

Keywords: Residue Number System,, SWA, DNA, Bioinformatics, CRT , MRC. MMA, Moduli Set.

1. Introduction

The residue number system (RNS) is an integer system capable of supporting high speed concurrent arithmetic [5]. In RNS, an integer is decomposed into a set of smaller integers i.e. with shorter binary representations, which can be processed independently and in parallel [9],[12]. This basis of RNS is set of pair wise prime integers S = {m1,m2, ,mn}, where gcd (mi,mj) = 1 for i = j. The set S is called the moduli set and the dynamic range of the number system is [0,], where M is the product of all moduli mi in S [13]. Any integer X within the dynamic range has a unique RNS representation given by an ordered set of residues X → {$x1$, $x2$. . . xn}, xi = ⎢ X ⎢ mi .where ⎢ X⎢ mi denotes X mod mi. The most important characteristic of the RNS representation is that it is a non-weighted number system, which facilitates parallel computing. If integers **A** and **B** have RNS representations {a_1, a_2. . . a_n} and {b_1, b_2, . . . , b_n} respectively, then the RNS representation of C = A ® B is C → {C_1, C_2. . . C**n**}, **Ci** = (a_i ® b_i) m_i .Where ® denotes addition, subtraction, multiplication or any combination of the three. The computation of **ci** depends upon **ai, bi** and **mi** only. Hence, each **ci** can be computed using a separate arithmetic unit, often called a channel**.** The reconstruction X from {x_1, x_2. . . x_n} is based on the Chinese Remainder Theorem (CRT) The main interest of the Residue Number Systems is to distribute integer operations on evaluations with the residues values [9],[11] where an operation with large integers is made on the residues which are small numbers and such that computations can be executed independently for each modulo allowing a complete parallelization of the arithmetic**.**

Example 1. Let the co-prime n-turple be (255, 256, 257), thus M = 16776960, **a** = 10000 is represented by (55, 16, 234) and **b** = 300 is represented by (45, 44, 43).

We can verify that: a × b = 3000000 are represented by (180, 192, 39). Where 180 = 55 * 45 mod 255, 192 = 16 * 44 mod 256 and 39 = 234 * 43 mod 257.

2. Background

Residue Number Systems (RNS) permit the representation of large dynamic range computations over small modular rings, which accelerate arithmetic computation speed. [**2],[5**]This modular arithmetic work was originated in the ancient day of the fifth century. The ancient study begins with a mathematical riddle from a third-century book, by the Chinese mathematician Sun Tsu in which puzzle

was illustrated as follows: such that how can we determine a number that has the remainders 2, 3, and 2 when divided by the numbers 7, 5, and 3, respectively? The riddle goes like: We have things of which we do not know the number. If we count them by three, the remainder is 2; if we count them by five, the remainder is 3; if we count them by seven, the remainder is 2; how many things are there? Sun Tzu gave the solution to this riddle with a rule called Tai Yen, which was later generalized in 1247 by another Chinese Mathematician Qin Jiushao. The procedure of obtaining the solution to the puzzle was first proposed and known as the Chinese Remainder Theorem.**[2], [4], [5]** but the use of this arithmetic to represent number was introduced only in 1959 by H.I. Garner.

3. Basic Rns Mathematical Model

3.1 Description of the residue representation

RNS is defined by a set of relatively prime integers called the moduli. The moduli set is denoted as $\{m_1, m_2, m_3 \ldots, m_n\}$ where n^{th} is the modulus [9]. Each integer X can be denoted as a set of smaller integers called residues. The set of residues are represented as $\{r_1, r_2, r_3, \ldots, r_n\}$ where r_n is the n^{th} residue. The residue r_n is defined as the least positive remainder of an integer value X is divided by the modulus, m_n [10]. This notation based on congruence is written as; X Mod $m_i = r_i$. RNS has a large active range can be divided into smaller dynamic ranges having the capacity to perform operation like addition, subtraction and multiplication independently with no carry propagation among modular paths.

The RNS which is an unweighted number can uniquely represent all integer numbers, X, that lie in its dynamic range. The dynamic range of an RNS is determined by the moduli set and it is denoted as M, where;

$$M = \prod_{n_i}^{n} m_i \qquad (1)$$

However, RNS provides unique representation for all integers in the range between 0 and M − 1. In a situation where the integer X is greater than M − 1, RNS repeats itself. The concept of legitimate and illegitimate range was first introduced by Barsi and Maestrini **[13]** in the course of investigating the error-correcting features of RRNS.

All RNS complexities are given in number of basic modulo operations on small moduli m_i or ($\acute{m}_j$). The costs of RNS algorithms depend also on the efficiency of such operators [12]. We present in this section the addition and the multiplication modulo mi (or ḿj)

3.2. The addition operator and operand:

We consider two integers a and b in which $0 \trianglelefteq a, b < mi$ and their sum such that we want to reduce modulo mi.We have the following identities: $a + b = \mathbf{s = s1\ 2^k + s0} = s_{imi} + \mathbf{cis1 + s0}$ when considering the reduction modulo mi we obtain $\mathbf{a + b \equiv cis1 + s0}$ (mod mi). Furthermore, as $0 \trianglelefteq a, b < mi$, such that $a + b < 2mi - 1$, i.e. $a + b < 2^k + (2^k - 2ci - 1)$. Hence, $s1 \trianglelefteq 1$. The addition modulo $\mathbf{m_i}$ is given by the following algorithm**:**

Algorithm 1:

mod add(a, b, mi)

Data: $0 \trianglelefteq a, b < mi$

Result: s = a + b mod p

d0 ← a + b;

d1 ← d0 + ci;

If $d1 \geq 2^k$ **then**

| s ← d1 mod 2^k;

else

| s ← d0;

end|

To resume algorithm 1, the moduli addition is equivalent to two additions, the comparison $d1 > 2^k$ corresponds

to the overflow which is used for selecting the result.

3.3. The multiplication operator and operand.

The addition operands are: a and b such that $0 \trianglelefteq a, b < mi$. We note p = a × b the product of those two values, and we will reduce p modulo mi. For this, we decompose p such that: $p = p12^k + p0 = p1mi + cip1 + p0$. Thus, $a \times b \equiv cip1 + p0$ (mod mi).We note $p^I = cip1 + p0$ .The condition $0 \trianglelefteq a, b < mi$, gives that $p < m_1^2 - 1 < 2^{2k}$ and $p1 < 2^k$. Now, if suppose that $ci < 2^t$, then we obtain with this first reduction $p^I = cip1 + p0 < 2^k(2^t - 1) + 2^k = 2^{k+t}$

A second reduction is needed. by decompose p^I such that $p^I = P_1^I\ 2^k + p^I_O$. Like previously, we obtain $a \times b \equiv Cip_1^I + p0 \pmod{mi}$. Noting that $p^{II} = ciP^I_1 + p^I_0$ and that, as $p^I < 2^{k=t}$, $P^I_1 < 2^t$. Now, if we consider that $2t < k - 1$ then we assume that $p^{II} < 2^k + 2^{2t} < 2mi$. Thus we just need to reduce p^{II} like in the addition algorithm to find the final result.

Algorithm 2:

mod prod (a, b,mi)

Data: $0 \trianglelefteq a, b < mi$

With $mi = 2k - ci$ and $ci < 2^t$ and

$1 < t < (k - 1)/2$

Result: $r = a \times b \bmod p$

$p \leftarrow a \times b;$

$p1 \leftarrow p \div 2^k$; $p0 \quad p \bmod 2^k$;

$p^I \leftarrow cip1 + p_o$;

$p^I_1 \leftarrow \quad p0 \div 2^k$; $P^I_O \leftarrow p^1 \bmod 2^k$;

$p^{II} \quad cip^I_1 + p^I_o$;

$p \leftarrow p^{II} + ci$;

If $p \geq 2^k$ **then**

$\quad r \leftarrow p \bmod 2^k$;

else

$\quad r \leftarrow p^{II}$;

End

The observation in the above algorithms with $mi = 2^k - ci$ such that $ci < 2^t$ and $1 < t < (k - 1)/2$, indicates that addition and multiplication modulo mi are very efficient.

In a residue number system (RNS), a number x is represented by the list of its residues with respect to k pair wise relatively prime moduli $M_{k-1} > \ldots > m_1 > m_o$ [11]. The residue x_i of x with respect to the ith modulus m_i is significant to a digit, and the entire k- residue representation of x can be viewed as a k-digit number, where the digit set for the ith position is $[0, m_{i-1}]$ $X_i = x \text{ Mod } m_i = (x)\ m_i$

The product M of the k-pairwise relatively prime moduli is the number of different representable value in the RNS (Dynamic Range)

$M = m_{k-1} x \ldots x\ m_i\ x\ m_o$

Let RNS (8/7/5/3).

$M = 8*7*5*3$

$M = 840$

$(-x)\ m_i = (M-x)\ m_i$

The value 840 can be used to represent numbers 0 through 839 – 420 through +419 consecutive intervals. Given RNS representation of x, the representation of –x can be found by complementing each of digits x with respect to its modules m_i with 0 digits unchanged.

Let 21 = $(5/0/1/0)_{RNS}$, then;

$-21 = (8-5/0/5-1/0)_{RNS}$

$= (3/0/4/0)_{RNS}$

4. Division Operation In Rns

Division is one of the main obstacles that hinder the use of **RNS**. In RNS representation, division is not a simple task operation. The technique between division in conventional representation and RNS representation is tedious and is base on the following limitations:

- There is an overall loop which increases the complexity and delay of the RNS division algorithm.
- Some numbers in the range of acceptable inputs as a denominator are excluded in division process..
- A lookup Table is used to perform an RNS division with operations in the binary or mixed radix system

4.1. Division Algorithm with a non-iterative operation. [8]

Input: $X = (x_1, x_2 \ldots x_n)$, $Y = (y_1, y_2, \ldots y_n)$ and input : $(m_1, m_2 \ldots m_n)$. The modulo set

Output: P = X / Y and S = (X mod Y) in RNS,

If all moduli are relatively prime numbers,

1) Calculate Y^{-1} = multiplicative inverse of Y
2) Calculate S = X mod Y
3) Calculate P= (X – S). Y-1
4) If any, y = 0, then set

$P_i = P_n \bmod m_i$ for $P_n \neq 0$

End.

4.2. Division in conventional representation.

We defined this as;

$x/y = q$ **(2)**

Or otherwise as:

$y * q = x$ **(3)**

where q = quotient.

4.3. The congruent in RNS is defined as:

$y * q = x \bmod m$ **(4)**

by the multiplicative inverses of y operation on both sides such that:

$$q = x * y^{-1} \bmod m \quad \textbf{(5)}$$

Then $_{Eq}$ **2** = $_{Eq}$ **5** only with an integer value, otherwise, multiplying by the multiplicative inverse in RNS representation will not be ≠ to division in conventional representation

Example 2: Consider an RNS with m=7, we want to compute the following quotient: **(i)** $6_{/2}$ **(ii)** $6/4$

Case1: $6_{/2 = q} \rightarrow 6 \times 2^{-1} \bmod 7 = 3$ {Conventional representation division equivalent}

We notice in part (b), that division in RNS is not equivalent to that in conventional representation when the quotient is a non-integer value. Due to this fact, division in RNS is usually done by converting the residues to conventional representation, performing the division, and then converting back to RNS representation. Tedious and complex conversion steps result in undesired overhead. This is one of the main drawbacks of RNS representation.

$_{6/4 = q} \rightarrow 6 \times 4^{-1} \bmod 7 = 5$. However, in gene sequencing this could serves as an advantage to security in bit error detection and correction.

5. Rns as a Weighted Representation

Given that RNS (8/7/5/3), the weight associated 105,120,336,280 $(1/2/4/0)_{RNS}$

$= \left| ((105*1) + (120*2) + (336*4) + (280*0)) \right|_{840}$

$= \left| (1689) \right|_{840}$ **= 9**

The sign of an RNS number can be changed by independently complementing each of its digits with respect to its modulus where Addition. Subtraction and Multiplication can be performed independently operating on each digit.

Example 2: Given RNS (8/7/5/3) illustrates the process $(5/5/0/2)_{RNS}$ represents x = +5 $(7/6/4/2)_{RNS}$ represents y = - 1

$(4/4/4/1)_{RNS}$ represents x+y; ($\left|5+7\right|_8 = 4$, $\left|5+6\right|_4 = 3$

$(6/6/1/0)_{RNS}$ represents x-y ; $\left|5-7\right|_8 = 6$, $\left|5-6\right|_7 = 6$

$(3/2/0/1)_{RNS}$ represents x*y; $\left|5*7\right|_8 = 3$, $\left|5*6\right|_7 = 2$

The simplicity and speed are the advantages of RNS arithmetic and is paramount in DNA sequencing. The Fig.1 below depicts the structure of an adder, subtractor and multiplier for RNS arithmetic [14]. This representation solve the problem of carry propagation with 6-bits residues, each operation require a 4k*6, look up table can be implemented with 9kb of memory.

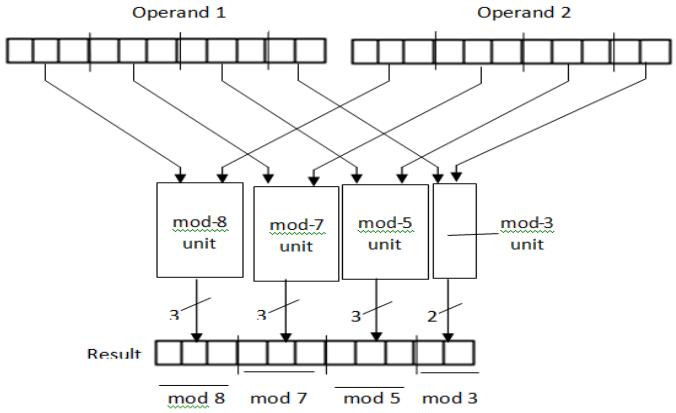

Fig.1. Operands operation for RNS (8/7/5/3)

6. Data Conversion in Residue Number System Arithmetic.

The achievement of hardware realization depends on both data conversion and choice of moduli set In this section, we look into the conversion from Decimal/Binary to RNS, Conversion from RNS to mixed radix form and Conversion from RNS to Binary/Decimal using CRT and MRC. [6],[11].

6.1. The conversion from Decimal/Binary to RNS

Given a number y with respect to the moduli m_i $0 \leq i \leq k_{-1}$ If y is an unsigned binary number and to

avoid time consuming division, we take the advantage of the following equality: $((y_{k-1}\ldots y_1y_0)_{two})\ m_i = ((2^{k-1}y_{k-1})\ m_i + \ldots + (2y_1)_{mi} + (y_0)m_i)\ m_i$ By pre computing $(2^j)_{mi}$ for each **i** and **j**, then , the residue x_i of y (Mod $_{mi}$) can be computed by modulo-m_i.

Example 3:

Let $y = (1010\ 0100)_2$

$= (164)_{10}$ in RNS (8/7/5/3) yMod8

$= x_3 = (y_2\ y_1\ y_0)_2$

$= (100)_2 =$ **4**. But, $y = 2^7 + 2^5 + 2^2$ and Mod7, Mod5, Mod3 respectively

For j = 7, 5, 2.

$X_2 = |y|_7 = |2+4+4|_7 = \mathbf{3}$

$X_1 = |y|_5 = |3+2+4|_5 = \mathbf{4}.$

$X_0 = |y|_3 = |2+2+1|_3 = \mathbf{2}$

The RNS $(8/7/5/3) = (164)_{10} \rightarrow (\mathbf{4/3/4/2})_{RNS}$ .The worst case requires k modular addition by a k- bit number.

6.2. Conversion from RNS to Mixed Radix.

Given RNS bases with **RNS** $(M_{k-1} | \ldots | m_2 | m_1)$ is a mixed-Radix number system.**MRS** $(M_{k1} | \ldots | m_2 | m_1 | m_0)$ kdigit,$M_{k2./}m_2m_1m_2\ldots.m_2m_1m_0 \quad m_1m_0\ m_0)$ k-digit

$Y=(X_{k1} | .. | x_2 | x_1 | x_0)_{RNS}=(Z_{k1} | .. | z_2 | z_1 | z_0)_{MRS}$ .....

(6)

$Y = Z_{k-1}(m_{k-2}\ldots m_2m_1m_0) + \ldots + z_2(m_1m_0) + z_1(m_0) + z_0$

(7)

$Z_0 = x_0$, subtracting **$z_0 = x_0$** from both **RNS and MRS** representation above

Yield: $y - x_0 = (x'_{k-1} | \ldots | x'_2 | x'_1 | 0)_{RNS} = (z_{k-1} | \ldots | z_2 | z_1 | 0)_{MRS}$

(8)

Where $x'_j = (x_j - x_0)\ m_j.$

If we divide both RNS and MRS by m_0 yield: $(x''_{k-1} | \ldots | x''_2 | x''_1)_{RNS} = (z_{k-1} | \ldots | z_2 | z_1)_{MRS}$

Example 4: convert $y = (0 | 6 | 3 | 0)_{RNS}$ to mixed radix representation $z_0 = x_0 = 0$

$(0 | 6 | 3 | 0)_{RNS} = (0 | 6 | 3 | 0)_{RNS} * (3 | 5 | 2 | = (0 | 2 | 1 | -)_{RNS}$

If $z_1 = 1$.replace 1 and divide by 5

$(7 | 1 | 0 | -)_{RNS} = (7 | 1 | 0 | -)_{RNS} * (5 | 3 | - | -) = \mathbf{(3 | 3 | - | -)_{RNS}}$

If $z_2 = 3$, replace 3 and divide by 7

$(0 | 0 | - | -)_{RNS} = (0 | 0 | - | -)_{RNS} * (7 | - | - | -)_{RNS}$

7

$= \mathbf{(0 | - | - | -)_{RNS}}$

If $z_3 = 0$. Then $y = (0 | 6 | 3 | 0)_{MRN} = (0 | 3 | 1 | 0)_{MRS}$

$= 48$

$(000/011/001/00)_{MRS}$: $(000/011/000/00)_{MRS}$

If $Y' = (0 | 6 | 3 | 0)_{RNS} =$ **48** and $(5 | 3 | 0 | 0)_{RNS} =$ **45** by comparing the binary coded equivalent of MRS with RNS thus: the equivalent mixed Radix Representations $(0/3/1/0)_{MRS}$ and $(0/3/0/0)_{MRS}$ **or** $(000/011/001/00)_{MRS}$ and $(000/011/000/00)_{MRS}$ when coded in binary can be compare as ordinary numbers

Remark: The mixed-Radix representation allows us to detect the sign of a number and compare the magnitudes of two RNS numbers. This feature is very useful in molecular sequence computation when there likelihood mutation in the cells sequences.

6.3. Conversion From RNS to Binary/Decimal

Residue to decimal conversion has some speed limitations, and is derived from

- The weights for the RNS directly based on the Chinese Remainder Theorem CRT. The magnitude of an RNS number can be obtained from the CRT formula.
- The Mixed Radix representation of the RNS number and use the weight radix of the mixed radix position to complete the conversion.

6.3.1. Chinese Remainder Theorem

$\underline{X} = \underline{(x_{k-1}/\ldots/x_2/x_1/x_0)_{RNS}} =$

$$\left| \sum_{i=0}^{k-1} M_i \left| \alpha_i x_i \right| \right| m_i \left| M_i \right| \alpha_i \left| x_i \right| m_i \left| M \right. \quad (9)$$

Where

$$M_i = \frac{M}{m_i} \text{ and } \alpha_i = \left| M_i^{-1} \right| m_i \quad (10)$$

is the inverse of Mi modulo mi

The conversion from $^{\alpha}$ representation to RNS can be done in K iterations [1].If we consider a full parallelization, with n modular multiplier-adders (MMA), this conversion can be logarithmic (ie O(log(K))) such that the modular multiplier-adder (MMA) forms the basic operator of RNS computing.

Note two RNS bases use $^{\alpha} = (m_1, m_2, . , m_n)$ and $^{\alpha^i} = (m^i_1, m^i_2, . . m^i_n)$.

From the Chinese Remainder Theorem

In the proof of the Chinese Remainder Theorem we can show, that a solution of the following system: $X = x_1 \pmod{m_1}$, $x_2 \pmod{m_2}$, $x_3 \pmod{m_3}$ … $x_n \pmod{m_n}$ can be constructed from the formula [CRT]

$$X = x_1 |M_1|_{m1}^{-1} M_1 + x_2 |M_2|_{m2}^{-1} M_2 + \ldots + xn |M_n|_{mn}^{-1} Mn|_M$$

…. **(11)**

Multiplicative inverse of M_i with respect to m_i is the multiplicative inverse of X of modulus m_i is denoted as Xi^{-1} and should satisfy $(Xi^{-1}X) \bmod mi = 1$.

Simplify the equation11. We consider $^{a}_1 = xi |Mi |mi -1$

mod mi, so we obtain a new expression for X:

$$X = \left(\sum_{i=1}^{n} {}^{\alpha_i} M_i \right) \bmod M \quad (12)$$

Hence we deduce that the cost of the conversion from RNS to binary representation is equivalent to n modular products

I	3								
m_i	8								
x_i	0	1	2	3	4	5	6	7	0
$\Theta = (M_i^{-1} \mid \alpha_i x_i \mid_{mi})$	.0000	.1250	.2500	.3750	.5000	.6250	.7500	.8750	.0000
$\mu = \left\| M_i \right\| \alpha_i x_i \left\| mi \right\| M$	0	105	210	315	420	525	630	735	0
I	2								
m_i	7								
x_i	0	1	2	3	4	5	6	0	
$\Theta = (M_i^{-1} \mid \alpha_i x_i \mid_{mi})$	.0000	.1429	.2857	.4286	.5714	.7143	.8571	.0000	
$\mu = \left\| M_i \right\| \alpha_i x_i \left\| mi \right\| M$	0	120	240	360	480	600	720	0	
I	1					0			
m_i	5					3			
x_i	0	1	2	3	4	0			
$\Theta = (M_i^{-1} \mid \alpha_i x_i \mid_{mi})$	.0000	.4000	.8000	.2000	.6000	.0000			
$\mu = \left\| M_i \right\| \alpha_i x_i \left\| mi \right\| M$	0	336	672	168	504	0			
i	0								
m_i	3								
x_i	0	1	2						
$\Theta = (M_i^{-1} \mid \alpha_i x_i \mid_{mi})$	.0000	.3333	.6667						
$\mu = \left\| M_i \right\| \alpha_i x_i \left\| mi \right\| M$	0	280	560						

Table 1. Look up Table values for CRT Decoding approximation $(8 \mid 7 \mid 5 \mid 3)_{RNS}$ where μ is the value needed in applying CRT

The maximum error (ϵ) to 4 decimal digits in each entry is 0.00005 as compare to the scaled 4(ϵ) = 0.0002 shows that RNS number value x > y is effective. We can compare the magnitudes of two RNS numbers through appropriate CRT decoding by dividing the equality in the CRT equation above by M as expressed for the value of x in [0,1]

$$\frac{X}{M} = \left(\frac{x_{k-1} \left| \ldots \right| x_2 \left| x_1 \right| x_0}{M} \right)_{RNS} \quad (13)$$

$$= \left| \sum_{i=0}^{K=1} m_i^{-} \left| \alpha_i x_i \right| mi \right|_M \quad (14)$$

Example 5: Let $x = (0/6/3/0)_{RNS}$ and $y = (5/3/0/0)_{RNS}$

$$\frac{X}{M} \equiv (.0000 + .8571 + .2000 + .0000)_1 \rightarrow .0571$$

$$\frac{Y}{M} \equiv (.6250 + .4286 + .0000 + .0000)_1 \rightarrow .0536$$

7. Moduli Selection

The choice of moduli selection plays the paramount roles and dictates the speed up of computations, the more the dynamic range of the moduli set the faster the system and the slower its reverse conversion. This feature is paramount molecular sequences MS, with inherent properties like fault tolerance etc. [6], [7]

- They must be relatively primed i.e they should be pairwise prime. With gcd (mi,mj) = 1 for all mi ≠ mj.
- The moduli *mis* should imply simple binary to RNS and RNS to binary conversions as well as simple RNS arithmetic.
- The smaller the moduli, the faster the arithmetic operations:
- The higher the dynamic range of the moduli set, the faster its forward conversion and the slower its reverse conversion.
- To avoid overflow, the dynamic range should be large enough.
- Efficiency of RNS moduli should be considered and high efficiency is more desirable, example the RNS **(15|13|11)** require 12 bits, it can represent 2^{12} = 4096, whereas only 2145 numbers are presented - the efficiency is **52%**
- Select prime numbers in sequence until a desired dynamic range is obtained i.e. the moduli product should be large enough to implement the desired dynamic range.
- Each moduli **mi** should be as small as possible so that operations modulo *mi* require minimum computational time.
- Moduli numbers can be restricted to power of **2**.
- The moduli should provide a well balanced decomposition of the dynamic range. This means that the difference in word length between the moduli should be as small as possible.

7.1. Limitations of residue number system.

RNS has limitation in applications involving arithmetic operation like scaling, overflow, division, magnitude comparisms, sign detection, overflow detection, division, reverse conversion and the complexity of the system to implement etc High speed is achieve in the operation with addition, subtraction, multiplication by supporting carry free addition, borrow free subtraction and digit to digit multiplication without partial product option for processing data in the advancing Biotechnology [4],[6].[10]

8. Rns-Swa Based Implementation In Gene Sequence Alignment

In 1981, T. F. Smith and M. S.Waterman described a method, commonly known as the Smith-Waterman (SW) algorithm **[4],[15]** for finding common regions of local similarity**.** The SWA is an optimal method for homology searches and sequence alignment in genetic databases and makes all pair wise comparisons between two strings of DNA. It achieves high sensitivity as all the matched and near-matched pairs are detected; however, the computation time required strongly limits its use. .

In calculating the local alignment, the matrix H(i ,j) is used to keep track of the degree of similarity between the two sequences **(A_i;B_j)** that are aligned. Each element of the matrix H(i; j) is calculated accordingly as in Equation 10 below:

$$H_{(I,J)} = \max \begin{cases} 0 \\ H(i-1,j-1) + S(i,j) \\ H(i-1,j) - d \\ H(i,j-1) - d \end{cases} \quad (15)$$

Where

H (i, j) is the maximum similarity score between the two sequences compared, S **(i, j)** is the similarity score in comparing sequence $\mathbf{A_i}$ to sequence $\mathbf{B_j}$ and **d** is the gap penalty for a mismatch in the comparison.

The algorithm consists of three main steps

- The initialization step:-Matrix H is initialized by setting H(0,j) = 0 and H(i,0) = 0 for all i and j.
- Matrix fill step**:-** This step is done to fill in all the entries of the matrix using equation x^{I}_{1}, x^{II}_{1} and x^{III}_{1} where X = ($x_1, x_{2 \ldots} x_n$) and n > 0 computationally intensive where hardware acceleration runs.
- Trace back step: **-** Where the matrix scores entered are trace back to inspect optimal score local alignment.

9. Application of Rns In Dna Sequence

These goals are achievable with the RNS conversion capability, and the availability of high speed RNS processor for simplification of detected error and correction as applicable in DNA .Sequences and many other fields that involve a complex arithmetic algorithm [4][15].

- Allows high speed correlation of DNA information which enhances computation and relational studies between DNA diseases and their inheritance. Example genes identified to be involved in breast cancer.
- The trade-offs between word length and hardware size, and between propagation delay and accuracy, various types of number representation have been adopted in which the Residue Number System (RNS) provide a very high speed sequence comparison with emphasis on its arithmetic advantages in real life application in bioinformatics and successfully used in many applications that involve high computations because of the carry propagation absence in the RNS embedded processor.
- Allows us to trace evolutionary trend.: Example, if Bioinformatics are able to find the similarity between any two sequences, they will be able to trace and understand evolutionary trend between them **[4]**

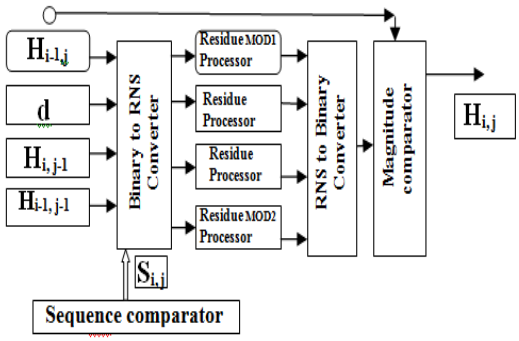

Fig.3: Schematic architecture showing RNS-SWA Based Processor for DNA Sequences

Fig.3: Schematic architecture showing RNS-SWA Based Processor for DNA Sequences

10. Rns-Swa Based Architecture

The architecture of hardware realization is achieved base on the number of defined moduli set e.g. $2^{2n} - 1, 2^n, 2^{2n} + 1$. Input from the SWA local alignment in **eq 15.**

- Equation 15 **:** H (i-1,j), H (I,j-1), H (i-1,j-1), d and S (i,j) is input and the binary to residue converter is assigned and executed by forward conversion to RNS processor .MOD 1
- The result is sent to the RNS processor/converter MOD2 where arithmetic computation is done in the embedded processor taking the advantage of the inherent properties of the carry free propagation in RNS and enableling the realization of high-speed and low-power consumption.
- The corresponding final result X (i,j) is then Converted from residue to binary converter as binary/decimal number.

11. Conclusion

The arithmetic advantages associated with RNS bases are employed to address the computational challenges of the SWA. The availability of embedded RNS processor and the absence of carry propagation in RNS enhanced the possibility of realization of high-speed and low-power consumption, these has effectively accelerated the Speed and accuracy of arithmetic operations which are the major challenges in the field of bioinformatics computing. The next generation research could employ the use of RNS polynomial and FPGA tools for further implementation and make it possible, to build a specialized RNS based architecture, having the capability to address mutation with detection and correction of errors in molecular sequence computing.

References

[1.] C. N. Zhang, B. Shirazi, and D. Y. Y. Yun**, [1993]** "An efficient algorithm and parallel implementations for binary and residue number systems," Journal of Symbolic Computation 15(4), pp. 451–462.

[2.] Bin Cao, Chip-Hong Chang, S.Thambipillai. **[1994]** "Design of residue generators and multi operand modulo adders using carry-save Adders,"IEEE Trans. Comp., vol. 43, pp.68-77.

[3.] A.B.Premkumar, **[2013]** "Improved memory less RNS forward converter based on periodicity of residues" IEEE Trans. Circuits and Systems-II, express Briefs, vol. 53, no.2,5 .

[4.] Hassan Kehinde Bello and Kazeem Alagbe Gbolagade. **[2017]** "A Survey of Human Deoxyribonucleic Acid" British Journal of Applied Science & Technology. 21(5): 1-10.

[5.] K.A Gbolagade, R. Chares, L. Sousa, S.D Cotofana. **[2009]** "Residue –to- binary converters for the {22n+1-1, 22n, 2n-1} moduli set. 2nd IEEE international conference in adaptive science & technology. Pp 26 – 33.

[6.] R. Conway and J. Nelson**, [2003]** "New CRT-based RNS converter using restricted moduli set," IEEE Transactions on Computers 52(5), pp. 572–578.

[7.] C. Y. Hung and B. Parhami**, [1994]** "An approximate sign detection method for residue numbers and its application to RNS division," Computers and Mathematics with Applications 27, pp. 23–35.

[8.] Mansoureh and Mohammed **[2012].** Non iterative RNS division Algorithm, IMECS Vol. I. ISSN;2078-0966 online

[9.] K.A. Gbolagade **[2009c].** "A shorter algorithm for efficient residue to decimal conversion". Advances in computer Science and Engineering, 3(2), pp. 147-156.

[10.] K.A. Gbolagade **[2011].** New adder-base RNS-to-binary converters for the $\{2^{n-1}+1,2^{n-1}-1,2^{n}\}$modoli set. ISRN Signal Processing. Journal, 7, 1-7.

[11.] K.A. Gbolagade **[2013]** An efficient MRC base RNS-to-binary converters for the $\{2^{2n}-1,2^{n}, 2^{2n+1}-1\}$modoli set. International Journal of Advance Research in Computer Engineering and Technology.2 (10) pp,2661-2664.

[12.] A.S. Madhu kumar,., and F. Chin**, [2002],** "Performance of a Residue Number system based CDMA system over wireless personal communication", Springer. Netherlands, Vol. 22, No. 1, pp. 89 – 102.

[13.] F. J. Taylor, **[1990]**,"An RNS Discrete Fourier Transform implementation,"IEEE Trans. Acoust. Speech, Signal Process, Vol. 38, No. 8. Pp. 1386 – 1394.

[14.] B. Parhami and H.F. Lai **[1993]** "Alternate Memory Compression Schemes for modular Multiplication "IEEE Trans. Signal Processing Vol. 41, pp.1378-1385.

[15.] Dan Gasified, **[1997].** Algorithms on Strings. Trees and Sequences: Computer Science and Computational Biology Cambridge University Press.