Journal of Engineering and Applied Sciences 6 (2): 174-179, 2011
ISSN: 1816-949X
© Medwell Journals, 2011

Bioinformatics: An Important Application Area of Residue Number System

[1, 2]E.Y. Baagyere, [1]K.O. Boateng and [2]K.A. Gbolagade
[1]Department of Computer Engineering,
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
[2]Department of Computer Science, University for Development Studies, Navrongo, Ghana

Abstract: In this study, we present a comprehensive survey of both Residue Number System (RNS) and bioinformatics with emphasis on the application of RNS to Smith Waterman Algorithm (SWA). The limitations of SWA are highlighted. We suggest RNS as an important tool for addressing the SWA limitations by posing interesting open questions. We believe that addressing this issue will open new avenues for both RNS and bioinformatics researchers.

Key words: Residue number system, Chinese remainder theorem, moduli selection, bioinformatics, Smith waterman algorithm, limitations

INTRODUCTION

The origin of Residue Number System (RNS) can be traced to the puzzle given by Sun Tzu, a Chinese mathematician and is illustrated as follows; how can we determine a number that has the remainders 2, 3 and 2 when divided by the numbers 7, 5 and 3, respectively? This puzzle written in the form of a verse in the 3rd century book, Suan-ching by the Chinese scholar Sun Tzu is perhaps, the 1st documented use of number representation using multiple residues. The answer to this puzzle, 23 is outlined in Sun Tzu's historic work. The puzzle, essentially asks us to convert the residues $(2|3|2)_{RNS(7|5|3)}$ into its decimal equivalent.

Sun Tzu formulated a method for manipulating these remainders (also known as residues) into integers. This method is regarded today as the Chinese Remainder Theorem (CRT). The CRT as well as the theory of residue numbers was set forth in the 19th century by Carl Friedrich Gauss in his celebrated Disquisitiones arithmetical (Soderstrand *et al*., 1986).

This >1700 years old number system is making waves in computing recently. Digital systems implemented on residue arithmetic units may play an important role in ultra speed, dedicated, real time systems that support pure parallel processing of integer value data due to its inherent features such as carry free addition, borrow free subtraction, single step multiplication without partial product, parallelisms and fault tolerant.

These interesting properties of RNS have lead to its widespread usage in Digital Signal Processing (DSP) applications such as digital filtering, convolution, correlation, Fast fourier transform, Discrete cosine transform, image processing, cryptography, communications and other highly intensive arithmetic applications. However, RNS has not found a widespread usage in general purpose processors due to difficulties associated with magnitude comparison, sign representation, overflow detection, data conversion, moduli selection, division and other complex arithmetic operations.

LITERATURE REVIEW

RNS is defined in terms of a relatively prime moduli set $\{m_1, m_2, m_3, ..., m_n\}$ that is GCD $(m_i, m_j) = 1$ for $i \neq j$ where, GCD means greatest common divisor. A binary number X can be represented by the residues $(x_1, x_2, x_3, ..., x_n)$, where, $x_i = X \bmod m_i$, $0 \leq x_i < m_i$. Such a representation is unique for any integer $X \epsilon [0, M - 1]$, where:

$$M = \prod_{i=0}^{n-1} m_i$$

is the dynamic range of the system. For a signed number system, any integer in (-M/2, M/2) has a RNS n-tuple representation where, $x_i = X \bmod m_i$ if $X>0$ and $(M-|X|) \bmod m_i$ otherwise. The signed RNS system is often referred to as a symmetric system.

Addition, subtraction and multiplication in RNS are very efficient since, digit by digit computations are allowed. Additionally, there is no ordering significance

Corresponding Author: E.Y. Baagyere, Department of Computer Engineering,
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

between the digits. However, division in RNS is rather complex since, it is not a closed operation. For example, given that X, Y and Z have RNS representations (Taylor, 1984):

$$X \xrightarrow{\text{RNS}} (X_1, X_2, \cdots, X_n) \tag{1}$$

$$Y \xrightarrow{\text{RNS}} (Y_1, Y_2, \cdots, Y_n) \tag{2}$$

$$Z \xrightarrow{\text{RNS}} (Z_1, Z_2, \cdots, Z_n) \tag{3}$$

and supposing that # denotes the operation +, - or *, then Z = X # Y, means:

$$Z_i = (X_i \# Y_i) \bmod m_i \tag{4}$$

if Z belongs to ZM; this means that no carry information need be communicated between residue digits. This explains why RNS is applicable in high performance computing and thus widely used in highly intensive DSP applications. In order to fully exploit these RNS parallelisms, arithmetic units that efficiently implement the modular statement (X_i # Y_i) mod m_i must be found. RNS has not found a widespread usage in general purpose computing due to the numerous disadvantages of RNS discussed. Data conversion and moduli selection are the greatest challenges of RNS.

DATA CONVERSION

Data conversion is one of the greatest challenges of RNS because the input operands are provided in either standard binary or decimal format and must be converted to RNS before the computation can be performed. Similarly, the final results must be represented in the same way as the input operands thus, RNS to binary/decimal conversion is very essential to asuccessful RNS design. This implies that RNS based processors make heavy use of data conversions which are slow processes. For an RNS processor to compete favorable with a conventional processor efficient data converters must be developed so that the RNS speedup will not be nullified by the conversion overhead.

Data conversion can be divided into two categories, namely forward and reverse conversion. Relatively, the reverse conversion is more complex but the forward conversion is not simple either. In this survey, we provide simple explanations on each of these two categories.

Forward conversion: For any n-bit nonnegative integer X in the range $0 \le X \le 2^n - 1$ can be represented in the weighted binary system as:

$$X = \sum_{i=1}^{n-1} b_i 2^i \tag{5}$$

where, b∈{0, 1}. The binary value of X can be converted into a set of n residues as (x_0, x_1, ..., x_{n-1}) where, x_i = X mod m_i. The values of x_i can be found by the following steps. From Eq. 5:

$$X = \sum_{i=1}^{n-1} b_i 2^i$$

Let:

$$X \bmod m_i = |X|_{m_i}$$

This implies:

$$|X|_{m_i} = \left| \sum_{i=0}^{n-1} b_i \left|2^i\right|_{m_i} \right|_{m_i} \tag{6}$$

The term $|2^i|_{m_i}$ can be pre-computed and stored in a Look Up Table (LUT). Also for any n-bit signed integer X in the range $0 \le X \le 2^n - 1$, the residues of X can be represented in the 2's-complement form as:

$$X_{2's-compl} = (b_n b_{n-1} \ldots b_2 b_1 b_0) = -b_n 2^n + \sum_{i=0}^{n-1} b_i 2^i \tag{7}$$

$$x_i \equiv X \bmod m_i \tag{8}$$

$$x_i = \left| b_n \left(m_i - \left|2^n\right|_{m_i} \right) + \sum_{i=0}^{n-1} b_i \left|2^i\right|_{m_i} \right|_{m_i} \tag{9}$$

Again the value of $|2^i|_{m_i}$ can be pre-computed and stored in a LUT.

The reverse conversion: Several reverse conversion techniques have been proposed in literature based on either the traditional Chinese Remainder Theorem (CRT) or the Mixed Radix Conversion (MRC) which may or may not rely on LUTs. The CRT is desirable because the data conversion can be parallelized while MRC is a sequential process by its very nature. However, many up to date RNS to binary/decimal converters are based on MRC due to the complex and slow modulo-M operation (where M is the system dynamic range thus, a rather large constant) required by CRT.

The CRT: The decimal equivalent of any RNS number can be obtained by using the traditional CRT given as:

$$X = \left(x_{L-1} \middle| \ldots \middle| x_2 \middle| x_1 \middle| x_0\right)_{RNS} = \left| \sum_{i=0}^{L} S_i \left| \left(x_i S^{-1}\right) \right|_{m_i} \right|_M \tag{10}$$

Where:

$$S_i = \frac{M}{m_i} \tag{11}$$

and S^{-1} is the multiplicative inverse of $|s_i|_{m_i}$ which implies that:

$$\left| S_i \; S_i^{-1} \right|_{m_i} = 1$$

Next, we give the schematic diagram of the CRT. Figure 1 shows the inherent parallelism feature of the CRT (Taylor, 1984). The traditional CRT can be further simplified when certain moduli sets (whether relatively prime or not) are utilized (Gbolagade and Cotofana, 2008a, b; Wang *et al.*, 2002).

Recently, CRT that requires mod-s_2 s_3 ... s_L instead of mod-s_1 s_2 s_3 ... s_L required by the CRT has been reported. This is called the New CRT and is presented by (Yu *et al.*, 2003). Based on the New CRT, many efficient reverse converters have been presented (Gbolagade and Cotofana 2008c; Wang *et al.*, 2002; Wang, 1998, 2000). We briefly review the New CRT as follows (Wang, 2000). Given the residue number (x_1, x_2, ..., x_n) with respect to the moduli set {s_1, s_2, s_3, ... s_L}, the corresponding decimal number X is computed as:

$$X = x_1 + \left| \begin{matrix} k_1 S_1 (x_2 - x_1) + k_2 S_1 S_2 (x_3 - x_2) + \ldots + \\ k_{(L-1)} \, S_1 S_2 \ldots S_{L-1} \left(x_L - x_{L-1}\right) \end{matrix} \right|_{S_2 S_3 \ldots S_{L-1} S_L} \tag{12}$$

However, conversion from RNS to decimal is relatively fast using MRC since, it does not involve the large modulo-M calculations which is required by the CRT.

The MRC: Suppose that we have an RNS number (x_1, x_2, x_3, ..., x_n) with the corresponding set of moduli {m_1, m_2, m_3, ..., m_n} and a set of digits {a_1, a_2, a_3, ..., a_n} which are the Mixed Radix Digits (MRDs), the decimal equivalent of the residues can be computed as follows:

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \ldots + a_n m_1 m_2 m_3 \ldots m_{n-1} \tag{13}$$

where, the mixed radix digits are given as follows (Gbolagade and Cotofana, 2008a):

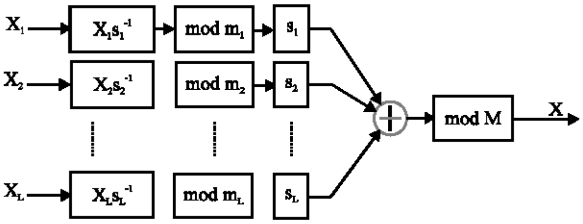

Fig. 1: A schematic diagram of the CRT

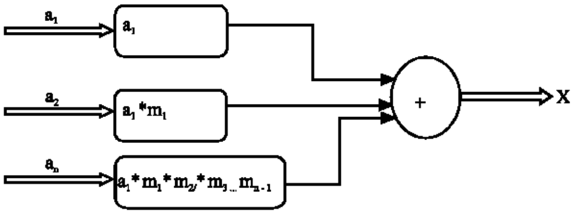

Fig. 2: The shematic diagram of the MRC

$$\begin{aligned} a_1 &= 1 \\ a_2 &= \left| (x_2 - a_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} \\ a_3 &= \left| \left((x_3 - a_1) \left| m_1^{-1} \right| \right) \left| m_2^{-1} \right|_{m_3} - a_2 \right|_{m_3} \cdots \\ a_n &= \left| \left(\begin{matrix} \left(\ldots (x_n - a_1) \left| m_1^{-1} \right|_{m_2} - a_2 \right) \\ \left| m_2^{-1} \right|_{m_3} - \ldots - a_{n-1} \end{matrix} \right) \left| m_{n-1}^{-1} \right|_{m_n} \right|_{m_n} \end{aligned} \tag{14}$$

For the MRD a_i, $0 \leq a_i < m_i$ any positive number in the interval:

$$\left[0, \prod_{i=1}^{N} m_i - 1 \right]$$

can be uniquely represented. Diagrammatically, MRC can be shown in Fig. 2. The only obstacle with the MRC is that by its very nature, it is a sequential process. Several attempts have been made to address this short-coming (Yassine and Moore, 1991; Gbolagade and Cotofana, 2008a, b, 2009). Data conversion and moduli selection are the two most important issues for a successful RNS processor realization. Thus, this review will be incomplete without discussing moduli selection.

MODULI SELECTION

The forms and the number of moduli selected determine the speed, the dynamic range and the hardware complexity of the resulting RNS architecture. The magnitude of the largest modulus dictates the speed of the arithmetic operations. The moduli selected should therefore be made comparable in magnitude since, there is no advantage in further fragmentation as the speed is already being dictated by the magnitude of the largest modulus. We present the types of moduli selection in the following subsections. Unrestricted moduli selection which is one of the categories of moduli selection is presented 1st while the restricted moduli selection.

Unrestricted moduli selection: In an unrestricted moduli selection, prime numbers are chosen in sequence, until the desired dynamic range M is obtained. Thus, relatively prime integer moduli are the topmost among the RNS researchers' design considerations. However in some cases, unrestricted moduli selection does not support simple conversions and simple RNS arithmetic computations. The solutions for realizing, all arithmetic operations are based on ROMs in order to speed up execution. The cost of implementing ROM based RNS data converters is generally very high. Thus, the need for restricted moduli selection (Abdallah and Skavantzos, 1995; Premkumar, 1995; Parhami, 2000; Wang *et al*., 2003).

Restricted moduli selection: These restricted moduli sets are usually based on powers of two and powers-of-two related moduli. This class of moduli set eliminates the need for ROMs in building RNS data converters (Premkumar *et al*., 2006).

Additionally with the restricted moduli sets, the basic building blocks such as multipliers, adders, binary-to-RNS converters and RNS-to-binary converters can also be easily realized using logic gates. Again using restricted moduli sets, several adder based data converters have been proposed. For example: $\{2^n - 1, 2^n, 2^n + 1\}$ (Jenkins and Leon, 1977) $\{2^n - 1, 2^n, 2^n - 1\}$ (Hiasat and Abdel-Aty-Zohdy, 1998) $\{2n - 1, 2n, 2n + 1\}$, $\{2n, 2n+1, 2n + 2\}$ (Parhami, 2000; Wang *et al*., 1999). Moduli sets must have some unique features (Abdallah and Skavantzos, 1995, 2005).

Expected features of the moduli sets: In selecting the moduli set $\{m_i s\}_{i = 1\text{-}L}$, the following general rules are considered:

- They must be relatively prime
- The moduli m_is should be made as small as possible so that operations modulo m_i require minimum computational time
- The moduli m_is should imply simple weighted to RNS and RNS to weighted conversions as well as simple RNS arithmetic. Sets with all their moduli being of the forms $2^{n1} + 1$ and $2^{n2} - 1$ and one of the form 2^{n3} satisfy the requirement of simple conversions and simple arithmetic
- The dynamic range should be large enough in order to avoid overflow
- The moduli m_is should create a balanced decomposition of the dynamic range. This implies that the differences between the number of bits of the different moduli should be very small

APPLICATION OF RNS TO BIOINFORMATICS

First, we present preliminary information as to why applying RNS to bioinformatics is very essential. Then a schematic representation of a RNS SWA based sequence alignment processor is given. Bioinformatics is becoming an increasingly important field of research. With the ability to rapidly sequence DNA information, biologists have the tools to among other things, study the structure and function of DNA, study evolutionary trends and correlate DNA information with disease. For example, two genes were identified to be involved in the origins of breast cancer (Miki *et al*., 1994). Such a research is only possible through the help of high speed sequence comparison.

All the cells of an organism consist of some kind of genetic information. They are carried by a chemical known as the Deoxyribonucleic Acid (DNA) in the nucleus of the cell. The DNA contains the instructions needed by the cell to carry out its functions. It consists of two long interwoven strands that form the famous double helix. Each strand is built from a small set of constituent molecules called nucleotides.

There are four kinds of nucleotides and each has different bases, namely adenine, cytosine, guanine and thymine. Their abbreviated forms are A, C, G and T, respectively. It is possible to deduce the original sequencing in DNA which codes for a particular amino acid. By finding the similarity between a number of amino acid producing DNA sequences and a genuine DNA sequence of an individual, one can identify the protein encoded by the DNA sequence of the individual.

In addition if biologists succeed in finding the similarity between DNA sequences of two different species, they can understand the evolutionary trend between them. Again, another important usage is that the relationship between disease and inheritance can also be studied. This is done by aligning specific DNA sequences of individuals with disease to those of normal people. If correlations can be found which can be used to identify those susceptible to certain diseases, new drugs may be made or better techniques invented to treat the disease.

There are many other applications of bioinformatics and this field is expanding at an extremely fast rate. A human genome contains approximately 3 trillion DNA base pairs. In order to discover which amino acids are produced by each part of a DNA sequence, it is necessary to find the similarity between two sequences. This is done by finding the minimum string edit distance between the two sequences and the process is known as sequence alignment. There are many algorithms for doing sequence alignment. The most commonly used ones are FASTA (Altschul *et al*., 1990) and BLAST (Lipman and Pearson, 1985). BLAST and FASTA are fast algorithms which prune the search involved in a sequence alignment using heuristic methods.

These methods are extremely very fast but at the expense of accuracy. The most accurate sequence alignment algorithm available is the Smith-Waterman Algorithm (SWA) (Smith and Waterman, 1981). However, the SWA is computationally very expensive for particularly long sequences. 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. We believe RNS as a tool can be used to address the computational time limitation of SWA. The description of SWA has been extensively presented (Hasan *et al*., 2007; Hasan and Al-Ars, 2007; Yu *et al*., 2003).

Figure 3 shows the RNS-SWA architecture. The inputs at the front end are the binary/decimal values of the H(i, j) of the SWA. These H(i, j) entries are converted into their respective residues by the RNS forward converter. The residue processors do parallel computations on these residues and the final conversion is made at the back end by the RNS reverse converter and then the final value of the SWA, H(i, j) is obtained after the magnitude comparison is done.

We intend to improve on the computational challenge of the SWA using RNS as a tool in the future. We shall implement the RNS-SWA architecture using VHDL and then synthesize the results to determine the gain in speed and area using MAX+Plus II tools. By this, we shall be exploring the RNS advantages in order to address the following open issues:

- The theoretical analysis have shown the superiority of RNS over the traditional method which is based on Binary number system. What percentage speed-up is RNS going to provide
- Can we further simplify the SWA to include the possibility of parallel computation thus, eliminating this data dependency syndrome

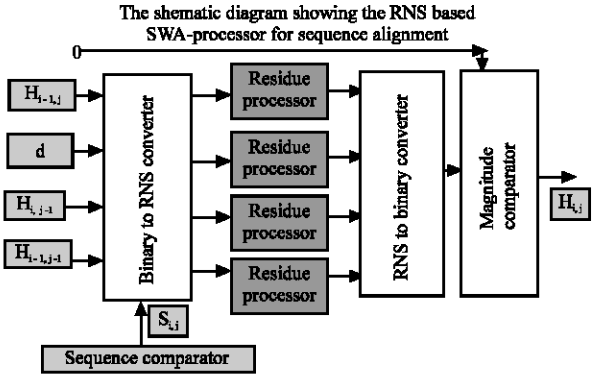

Fig. 3: The RNS-SWA architecture

- With further simplification can we obtain a RNS-based architecture with lower area cost when compared to the traditional/conventional approach
- Can we build a RNS-based architecture with the capability to detect and correct errors

CONCLUSION

In this study, we have presented a comprehensive survey of Residue Number System (RNS). We have shown that RNS has an application in Bioinformatics. As an example, we have shown some serious limitations that affect the practical application of SWA and we suggested some possible ways in which RNS can be applied to overcome those limitations. Issues such as RNS data conversions and moduli selection are thoroughly reviewed. We believe that addressing these issues will open new avenues for both RNS and bioinformatics researchers.

REFERENCES

Abdallah, M. and A. Skavantzos, 1995. A systematic approach for selecting practical moduli sets for residue number systems. Proceedings of the 27th Southeastern Symposium on System Theory, March 12-14, Starkville, MS. USA., pp: 445-449.

Abdallah, M. and A. Skavantzos, 2005. On MultiModuli residue number systems with moduli of forms ra, rb-1, rc+1. IEEE Trans. Circuits Syst. I: Regular Papers, 52: 1253-1266.

Altschul, S.F., W. Gish, W. Miller, E.W. Myers and D.J. Lipman, 1990. Basic local alignment search tool. J. Mol. Biol., 215: 403-410.

Gbolagade, K.A. and S.D. Cotofana, 2008a. Residue Number System operands to decimal conversion for 3-moduli sets. Proceedings of the 51st Midwest Symposium on Circuits and Systems, Aug. 10-13, Knoxville, Tennessee, pp: 791-794.

Gbolagade, K.A. and S.D. Cotofana, 2008b. Generalized matrix method for efficient residue to decimal conversion. IEEE Asia Pacific Conference on Circuits and Systems, Nov. 30-Dec. 3, Macao, pp: 1414-1417.

Gbolagade, K.A. and S.D. Cotofana, 2008c. A residue to binary converter for the {2 n +2, 2 n +1, 2 n }moduli set. Proceedings of the 23rd International Conference on Design of Circuits and Integrated Systems, November 2008, Grenoble, France, pp: 1785-1789.

Gbolagade, K.A. and S.D. Cotofana, 2009. An O(n) residue number system to mixed radix conversion technique. Proceedings of the IEEE International Symposium on Circuits and Systems, May 2009, Taiwan, China, pp: 521-524.

Hasan, L. and Z. Al-Ars, 2007. Performance improvement of the smith-waterman algorithm. Proceedings of the Annual Workshop on Circuits, Systems and Signal Processing, Nov. 29-30, Veldhoven, The Netherlands, pp: 211-214.

Hasan, L., Z. Al-Ars and S. Vassiliadis, 2007. Hardware acceleration of sequence alignment algorithms-an overview. Proceedings of International Conference on Design and Technology of Integrated Systems in Nanoscale Era, Sept. 2-5, Rabat, pp: 92-97.

Hiasat, A.A. and S.H. Abdel-Aty-Zohdy, 1998. Residue-to-binary arithmetic converter for the moduli set (2k, 2k-1, 2k-1-1). IEEE Trans. Circuits Syst. II: Analog Digital Signal Process., 45: 204-209.

Jenkins, W. and B. Leon, 1977. The use of residue number systems in the design of finite impulse response digital filters. IEEE Trans. Cirtuitry Syst., 24: 191-201.

Lipman, D.J. and W.R. Pearson, 1985. Rapid and sensitive protein similarity searches. Science, 227: 1435-1441.

Miki, Y., J. Swensen, D. Shattuck-Eidens, P.A. Futreal and K. Harshman *et al*., 1994. A strong candidate for the breast and ovarian cancer susceptibility gene BRCA1. Science, 266: 66-71.

Parhami, B., 2000. Computer Arithmetic and Hardware Designs. Oxford University Press, New York.

Premkumar, A.B., 1995. An RNS to binary converter in a three moduli set with common factors. IEEE Trans. Circuits and Syst. II Analog Digital Signal Proc., 42: 298-301.

Premkumar, A.B., E.L. Ang and E.M.K. Lai, 2006. Improved memoryless RNS forward converter based on the periodicity of residues. IEEE Trans. Circuits Syst. II: Express Briefs, 53: 133-137.

Smith, T.F. and M.S. Waterman, 1981. Identification of common molecular subsequences. J. Mol. Biol., 147: 195-197.

Soderstrand, M.A., W.K. Jenkins, G.A. Jullien and F.J. Taylor, 1986. Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. IEEE Press, New York.

Taylor, F.J., 1984. Residue arithmetic a tutorial with examples. Computer, 17: 50-62.

Wang, W., M.N.S. Swamy, M.O. Ahmad and Y. Wang, 1999. A comprehensive study of three moduli sets for residue arithmetic. Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, May 9-12, Alberta, Canada, pp: 513-518.

Wang, W., M.N.S. Swamy, M.O. Ahmad and Y. Wang, 2003. A study of the residue-to-binary converters for the three-moduli sets. IEEE Trans. Circuits Syst. I: Fundamental Theory Appl., 50: 235-243.

Wang, Y., 1998. New Chinese remainder theorems. Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers, Nov. 1-4, Pacific Grove, CA. USA., pp: 165-171.

Wang, Y., 2000. Residue-to-binary converters based on new Chinese remainder theorems. IEEE Trans. Circuits Systems II: Analog Digital Signal Process., 47: 197-205.

Wang, Y., X. Song, M. Aboulhamid and H. Shen, 2002. Adder based residue to binary number converters for (2n-1, 2n, 2n+1). IEEE Trans. Signal Process., 50: 1772-1779.

Yassine, H.M. and W.R. Moore, 1991. Improved mixed-radix conversion for residue number system architectures. IEEE Proc. Circuits Devices Syst., 138: 120-124.

Yu, C.W., K.H. Kwong, K.H. Lee and P.H.W. Leong, 2003. A Smith-waterman systolic cell. Proceedings of the 13th International Conference on Field-Programmable, September 2003, Springer-Verlag, pp: 375-384.