An Extensive Review on Residue Number System for Improving Computer Arithmetic Operations

Satyanarayan Shukla[1], Prof. Divya Jain[2]
[1]Master of Technology Research Scholar, [2]Research Guide, TIT Bhopal

Abstract- Residue Number System (RNS) is a non weighted number framework. In RNS, the number juggling operations are split into littler parallel operations which are autonomous of one another. There is no conveying proliferation between these operations. Consequently gadgets working in this guideline acquire property of fast and low power utilization. In any case, this property makes flood discovery is exceptionally troublesome. In this review paper we have presented a brief survey about the most recently achievements in the RNS, for improving the system performance. We concern the different proposed moduli sets that provide different dynamic ranges, the common means and structures to perform forward and reverse conversion, universal structures of residue arithmetic units and application where using the RNS is beneficial.

Keywords- Computer arithmetic, residue number system (RNS), restricted moduli set, sign detection.

I. INTRODUCTION

In general, numbers may be signed, and for binary digital arithmetic there are three standard notations that have been traditionally used for the binary representation of signed numbers. These are *sign-and-magnitude*, *one's complement*, and *two's complement*. Of these three, the last is the most popular, because of the relative ease and speed with which the basic arithmetic operations can be implemented. Sign-and-magnitude notation has the convenience of having a sign-representation that is similar to that used in ordinary decimal arithmetic. And one's complement, although a notation in its own right, more often appears only as an intermediate step in arithmetic involving the other two notations.

The sign-and-magnitude notation is derived from the conventional written notation of representing a negative number by pretending a sign to a magnitude that represents a positive number. For binary computer hardware, a single bit sources for the sign: a sign bit of 0 indicates a positive number, and a sign bit of 1 indicates a negative number. For example, the representation of the number positive-five in six bits is 000101, and the corresponding representation of negative-five is 100101. Note that the representation of the sign is independent of that of the magnitude and takes up exactly one bit; this is not the case both with one's complement and two's complement notations.

Sign-and-magnitude notation has two representations, 000. . . 0 and 100. . . 0, for the number zero; it is therefore redundant. With one exception (in the context of floating-point numbers) this existence of two representations for zero can be a nuisance in an implementation. Addition and subtraction are harder to implement in this notation than in one's complement and two's complement notations; and as these are the most common arithmetic operations, true sign-and-magnitude arithmetic is very rarely implemented.

The main RNS advantage is the absence of carry propagation between digits, which results in high-speed arithmetic needed in embedded processors. Another important feature of RNS is the digits independence, so an error in a digit does not propagate to other digits, which results in no error propagation, hence providing fault-tolerance systems. In addition, the RNS can be very efficient in complex-number arithmetic, because it simplifies and reduces the number of multiplications needed. All these features increase the scientific tendency toward the RNS especially for DSP applications. However, the RNS is still not popular in general-purpose processors, due the aforementioned difficulties.

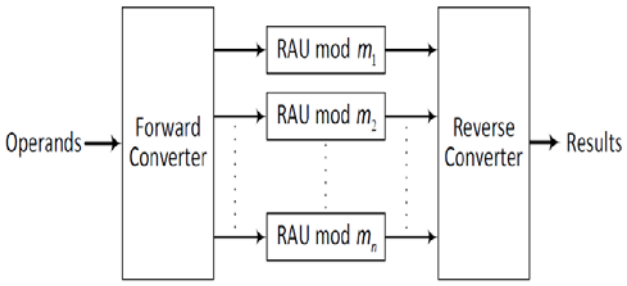

Fig. 1.1: The architecture of the residue number system (RNS)

The basic RNS processor's architecture is shown in Fig. 1.1. It consists of three main components; a forward converter (binary to residue converter), that converts the binary number to *n* equivalent RNS residues, corresponding to the *n* moduli. The *n* residues are then processed using *n* parallel residue arithmetic units (RAUs); each of them corresponds to one Residue Number System Based Building Blocks for Applications in Digital Signal Processing modulo. The *n* outputs of these units represented in RNS are then converted back into their binary equivalent, by utilizing the reverse converter (residue to binary converter).

In one's complement notation, the representation of the negation of a number is obtained by inverting the bits in its binary representation; that is, the 0s are changed to 1s

and the 1s are changed to 0s. For example, the representation of the number positive-five in six bits is 000101 and negative- five therefore has the representation 111010. The leading bit again indicates the sign of the number, being 0 for a positive number and 1 for a negative number. Therefore refer to the most significant digit as the *sign bit*, although here the sign of a negative number is in fact represented by an infinite string of 1s that in practice is truncated according to the number of bits used in the representations and the magnitude of the number represented. It is straightforward to show that the n-bit representation of the negation of a number N is also, when interpreted as the representation of an unsigned number, that of $2n - 1 - N$. (This point will be useful in subsequent discussions of basic residue arithmetic.) The one's complement system too has two representations for zero—00. . . 0 and 11. . . 1—which can be a nuisance in implementations. A similar problem occurs with certain residue number systems. Addition and subtraction in this notation are harder to implement than in two's complement notation (but easier than in sign-and-magnitude notation) and multiplication and division are only slightly less so. For this reason, two's complement is the preferred notation for *implementing* most computer arithmetic.

Negation in two's complement notation consists of a bit-inversion (that is, a translation into the one's complement) followed by the addition of a 1, with any carry from the addition being ignored. Thus, for example, the result of negating 000101 is 111011. As with one's complement notation, the leftmost bit here too indicates the sign: it is 0 for a positive number and 1 for a negative number; but again, strictly, the sign is actually represented by the truncation of an infinite string. For n-bit representations, representing the negation of the number N may also be viewed as the representation of the positive number $2n - N$.

In contrast with the first two conventional notations, the two's complement has only one representation for zero, i.e. 00. . . 0. The two's complement notation is the most widely used of the three systems, as the algorithms and hardware designs required for its implementation are quite straightforward. Addition, subtraction, and multiplication are relatively easy to implement with this notation, and division is only slightly less so.

All of the notations above can be readily extended to non-binary radices. The extension of binary sign-and-magnitude to an arbitrary radix, r, involves representing the magnitude in radix-r and using 0 in the sign digit for positive numbers and $r - 1$ for negative numbers. An alternative representation for the sign is to use half of the permissible values of the sign digit (that is, $0 . . . r/2 - 1$, assuming r is even) for the positive numbers and the other half (that is, $r/2 . . . r - 1$, for an even radix) for the negative numbers.

The generalization of one's complement to an arbitrary radix is known as *diminished-radix complement*, the name being derived from the fact that to negate a number in this notation, each digit is subtracted from the radix diminished by one, i.e. from $r - 1$. Alternatively, the representation of the negation may also be viewed as the result of subtracting the number from $rn - 1$, where n is the number of digits used in the representations. Thus, for example, the negation of 01432 in radix-8 is 76345, i.e. 77777 − 01432. The sign digit will be 0 for a positive number and $r - 1$ for a negative number. The generalization of two's complement to an arbitrary radix is known as *radix complement notation*. In radix complement notation, the radix-r negation of a number is obtained, essentially, by subtracting from rn, where n is the number of digits used in the representations. Alternatively, negation may also be taken as the formation of the diminished-radix complement followed by the addition of a 1. Thus, for example, the radix-8 negation of 01432 is 76346, i.e. 100000 − 01432 or 77777 − 01432 + 1. The determination of sign is similar to that for the radix-r diminished-radix complement.

II. RESIDUE NUMBER SYSTEMS

Residue number systems are based on the *congruence* relation, which is defined as follows. Two integers a and b are said to be *congruent modulo* m if m divides exactly the difference of a and b; it is common, especially in mathematics tests, to write $a \equiv b \pmod{m}$ to denote this. Thus, for example, $10 \equiv 7 \pmod 3$, $10 \equiv 4 \pmod 3$, $10 \equiv 1 \pmod 3$, and $10 \equiv -2 \pmod 3$. The number m is a *modulus* or *base*, and assume that its values exclude unity, which produces only trivial congruences.

If q and r are the quotient and remainder, respectively, of the integer division of a by m—that is, $a = q.m + r$—then, by definition, have $a \equiv r \pmod m$. The number r is said to be the *residue* of a with respect to m, and usually denote this by $r = |a|m$. The set of m smallest values, $\{0, 1, 2, . . . , m - 1\}$, that the residue may assume is called the set of *least positive residues modulo* m. Unless otherwise specified, assume that these are the only residues in use.

Suppose have a set, $\{m1 , m2 , . . . , mN \}$, of N positive and pair wise relatively prime moduli5 . Let M be the product of the moduli. Then every number $X < M$ has a unique representation in the residue number system, which is the set of residues $\{|X |mi \ : 1 \le i \le N\}$. A partial proof of this is as follows. Suppose $X1$ and $X2$ are two different numbers with the same *residue-set* . Then $|X1 |mi = |X2 |mi$, and so $|X1 - X2 |mi = 0$. Therefore $X1 - X2$ is the least common multiple (lcm) of mi. But if the mi are relatively prime, then their lcm is M, and it must be that $X1 - X2$ is a multiple of M. So it cannot be that $X1 < M$ and $X2 < M$. Therefore, the set $\{|X |mi \ : 1 \le i \le N \}$ is unique and may be taken as the representation of X. shall write such a representation in the form $hx1 , x2 , . . . , xN i$,

where *xi* = |*X* |*mi* , and shall indicate the relationship between *X* and

its residues by writing *X* ∼= *hx*1 , *x*2 , . . . , *xN i*. The number *M* is called the *dynamic range* of the RNS, because the number of numbers that can be represented is *M* . For unsigned numbers, that range is [0, *M* − 1].

RNS to MRS Conversion

From MRS Definition have

$$Y = z_{k-1}(m_{k-2}\cdots m_2 m_1 m_0) + \cdots + z_2(m_1 m_0) + z_1(m_1) + (1)z_0$$

Easy to See that $z_0 = y_0$, Subtracting This Value From RNS and MRS Values Results in:

$$Y - y_0 = (y'_{k-1}|\cdots|y'_2|y'_1|0)_{\text{RNS}} = (z_{k-1}|\cdots|z_2|z_1|0)_{\text{MRS}}$$

$$y'_j = \left\langle y_j - y_0 \right\rangle_{m_j}$$

Next, Divide Both Representations by m_0:

$$(y''_{k-1}|\cdots|y''_2|y''_1)_{RNS} = (z_{k-1}|\cdots|z_2|z_1)_{MRS}$$

Thus, if Can Divide by m_0, Have an Iterative Approach for Conversion. Dividing *y*' (a Multiple of m_0) by m_0 is SCALING Easier Than Normal RNS Division Accomplished by Multiplying by Multiplicative Inverse of m_0.

III. LITERATURE RVIEW

Xu, M. and Bian, Z. [1] investigated a fast sign detection algorithm for the residue number system moduli set ${2^{n+1}-1, 2^{n}-1, 2^{n}}$. First, a sign detection algorithm for the restricted moduli set is described. The new algorithm allows for parallel implementation and consists exclusively of modulo 2^{n} additions. Then, a sign detection unit for the moduli set ${2^{n+1}-1, 2^{n}-1, 2^{n}}$ is proposed based on the new sign detection algorithm. The unit can be implemented using one carry save adder, one comparator and one prefix adder. The experimental results demonstrate that the proposed circuit unit offers 63.8%, 44.9%, and 67.6% savings on average in area, delay and power, respectively, compared with a unit based on one of the best sign detection algorithms.

Maji, P. and Rath, G.S. [2] presented a RNS is generally an integer number system. The foremost canonical reason for implementation of filter in residue arithmetic is the inherent property of carry-free addition, subtraction and multiplication. As a result add, subtract and multiply in unison regardless to the numbers. Hereby, devices operating in this principle are fast and ingest low power. However, principal limitation of Residue Number System is the slow and complex nature for arithmetic operations viz. division, comparison, sign detection and overflow detection and rejection. In this paper have described some novel approaches to grapple with the limitations of comparison, sign detection and averting overflow. The selection of moduli in RNS is most important in attaining to solutions of problems as described earlier. Accordingly, a set of moduli is selected. Further in this paper have used this set of moduli to successfully depict a design approach for 32-bit lowpass finite impulse response (FIR) filter.

Daikpor, M.N. and Adegbenro, O. [3] proposed an overview of design implementation of a Symmetrical Multiple Valued Logic (SMVL) arithmetic circuit based on the use of restricted moduli Symmetrical Signed Residue Number System (SSRNS). Restricted radix-7 Symmetrical quaternary Signed digit (Rr7SqSd) T-gate based interconnections and full adders are used to implement sign detection, overflow detection and magnitude comparison without recourse to Mixed Radix number System (MRS) converters design or Chinese Remainder Theorem (CRT) computation.

Tomczak, T. [4] worked a fast algorithm for sign-extraction of a number given in the Residue Number System (2^n-1,2^n,2^n+1). The algorithm can be implemented using three *n*-bit wide additions, two of which can be done in parallel. It can be used in a wide variety of problems, i.e., in algorithms for dividing numbers in the RNS, or in evaluating the sign of determinant in computational geometry, etc.

Rejeb, B.; Henkelmann, H. and Anheier, W. [5] analyzed the division; sign detection and number comparison are the more difficult operations in residue number systems (RNS). These shortcomings limited most RNS implementations to additions, subtractions and multiplications. In this paper, a high level description of a RNS division algorithm is proposed. A general hardware architecture of the algorithm for division by a constant as well as its application to fractal image coding are also presented.

Hiasat, A. A. and Abdel-Aty-Zohdy, H.S. [6] presented a new algorithm for one of the longstanding problems in residue number system, namely division, is presented. The algorithm is very simple. It approaches the paper-and-pencil division procedure where the quotient is selected to guarantee a non-negative remainder. This algorithm does not require sign and overflow detection, scaling, or redundant moduli. Based on computer simulation results, the algorithm is four times faster than the most recent and competitive published work by Lu and Chiang (see IEEE Trans. Compu., vol. C-41, no. 8, p. 1026-32, 1992).

IV. PROBLEM DESCRIPTION

Sign detection plays an essential role in branching operations, magnitude comparisons, and overflow detection. Because the sign information is concealed in each residue digit in a residue number system (RNS), sign detection in an RNS is more difficult than that in the weighted number system, in which the sign is the most significant bit (MSB). Furthermore, sign detection in an RNS is not as efficient as

modular operations, such as addition, subtraction, and multiplication, because of its complexity.
The sign detection problem has been investigated by many researchers. A general theorem is derived by establishing the necessary conditions for sign detection [1].

V. PROPOSED METHODOLOGY

A standard RNS is defined exclusively for positive integers in the range [0, M). To accommodate negative integers, an implicit signed number system may be considered to be split into a positive half of the range and a negative half of the range. The dynamic range M of the moduli set {m1, m2,...,m N-1, mN= 2nn- 1, 2- 1, 2n} is even. After conversion from the residue number to the weighted number, the resulting non integer X in the interval [0, M/2) carries an implicit representation of the sign of the actual result Y, which can be obtained in its range [-M/2, M/2 - 1) as follows

$$Y = \begin{cases} X & if\ 0 \trianglelefteq X < \frac{M}{2} \\ X - m, & if \frac{M}{2} \trianglelefteq X < M \end{cases} \qquad (1)$$

The mixed-radix CRT is presented in [8] as follows.

Given $\{m_1, m_2, \dots\dots, m_N\}$, the magnitude of a residue number $X = (x_1, x_2, \dots x_N)$ is calculated as follows:

$$X = \sum_{j=1}^{N-2} \left(a_{j+1} \prod_{i=1}^{j+1} m_i \right) + a_1 m_1 + a_0 \qquad (2)$$

Where $a_{j+1} = |\ |\sum_{j=1}^{j+2} \frac{\gamma_i x_i}{\prod_{i=2}^{j+1} m_i}|_{mj+2,}$

$$a_1 = |\gamma_1 x_1 + \gamma_2 x_2|_{m2,}$$

$$a0 = x1, N > 1, \gamma_1 = \frac{N1|N_1^{-1}|_{m_1} - 1}{m1},,$$

$$\gamma_1 = M|N_i^{-1}|m_i$$

$$M = m1\ m2 \dots . mN, Ni = \frac{M}{mi} and$$

$$the\ multiplicative\ inverse\ |N_i^{-1}|_{mi}\ is defined\ by$$

$$||N_i^{-1}|_{mi}\ N_i|_{mi} = 1, for\ i = 1, 2, 3, \dots\dots\dots, N. The\ floor\ function\ is\ denoted\ by\ \lfloor \bullet \rfloor.$$

VI. CONCLUSION

Residue number systems are more complex than the other standard notations reviewed. Thus, the residue arithmetic is often realized in terms of lookup-tables (to avoid the complex combinational-logic circuits) and conventional arithmetic. The sign-and- magnitude approach may be convenient for representing signed numbers in RNS, but actual arithmetic operations might be best realized in terms of radix-complement arithmetic. We have analyzed certain choices of representational parameters in RNS naturally lead to diminished-radix complement (one's complement) arithmetic. The dynamic range then consists of a "legitimate" range, defined by the non- redundant moduli and an "illegitimate" range; for arithmetic operations, initial operands and results should be within legitimate range. RNS of this type are especially useful in fault-tolerant computing. The redundant moduli mean that digit-positions with errors may be excluded from computations while still retaining a sufficient part of the dynamic range. Furthermore, both the detection and correction of errors are possible: with k redundant moduli, it is possible to detect up to k errors and to correct up to bk/2c errors. A different form of redundancy can be introduced by extending the size of the digit-set corresponding to a modu- lus, in a manner similar to RSDs.

REFRENCES

[1] Xu, M.; Bian, Z.; Yao, R., "Fast Sign Detection Algorithm for the RNS Moduli Set {$2^{\wedge n+1}$ -1, 2^n-1, 2^n}," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.23, no.2, pp.379,383, Feb. 2015.

[2] Maji, P.; Rath, G.S., "A novel design approach for low pass finite impulse response filter based on residue number system," *Electronics Computer Technology (ICECT), 2011 3rd International Conference on* , vol.3, no., pp.74,78, 8-10 April 2011.

[3] Daikpor, M.N.; Adegbenro, O., "Restricted moduli Symmetrical quaternary signed-digit addition: A design implementation overview," *Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on* , vol., no., pp.1,4, 16-18 June 2011.

[4] Tomczak, T., "Fast Sign Detection for RNS ($2^n = 1, 2^n, 2^n + 1$)," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.55, no.6, pp.1502,1511, July 2008.

[5] Rejeb, B.; Henkelmann, H.; Anheier, W., "Integer division in residue number system," *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on* , vol.1, no., pp.259,262 vol.1, 2001.

[6] Hiasat, A.A.; Abdel-Aty-Zohdy, H.S., "A high-speed division algorithm for residue number system," *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on* , vol.3, no., pp.1996,1999 vol.3, 30 Apr-3 May 1995.

[7] Parhami, B.; Hung, C.Y., "Optimal table lookup schemes for VLSI implementation of input/output conversions and other residue number operations," *VLSI Signal Processing, VII, 1994., [Workshop on]* , vol., no., pp.470,481, 1994.

[8] Dimauro, G.; Impedovo, S.; Pirlo, G., "The `diagonal function' in non-redundant residue number system," *EUROMICRO 94. System Architecture and*

Integration. Proceedings of the 20th EUROMICRO Conference. , vol., no., pp.590,596, 5-8 Sep 1994.

[9] Mi Lu; Chiang, J.-S., "A novel division algorithm for the residue number system," *Computers, IEEE Transactions on* , vol.41, no.8, pp.1026,1032, Aug 1992.

[10] Ray, G.A., "Core-based RNS ALU with customized instructions," Computers and Communications, 1990. Conference Proceedings., Ninth Annual International Phoenix Conference on , vol., no., pp.891,, 21-23 Mar 1990.

[11] F. Barsi and P. Maestrini. Error correcting properties of redundant residue number systems. IEEE Transactions on Computer, Vol. c-22, No. 3 pp.307-315, March, 1973.

[12] M. Bhardwaj, A.B. Premkumar, and T. Srikanthan. Breaking the 2n-bit carry propagation barrier in residue to binary conversion for the IEEE Trans. on Circuits and Syst. II, Vol. 45, pp. 998-1002, September, 1998.

[13] M. Bhardwaj, T. Srikanthan, and C.T. Clarke. A reverse converter for the 4-moduli superset. IEEE Symp. Computer Arithmetic, pp. 168-175, April, 1999.

[14] G. Bi and E.V. Jones. Fast conversion between binary and residue numbers. Electronic Letters, vol. 24, no. 19, pp. 1195-1197, September, 1988.

[15] S. Bi and W.J. Gross. The mixed-radix chinese remainder theorem and its applications to residue comparison. IEEE Trans. on Computers, vol. 57, No. 12, pp. 1624-1632, December, 2008.

[16] S. Bi,W.Wang, and A. Al-Khalili. Modulo deflation in converters. Proc. IEEE International Symposium on Circuits and Systems (ISCAS'04), Vol. 2, pp.429-432, 2004.

[17] S. Bi, W. Wang, and A. Al-Khalili. New modulo decomposed residueto- binary algorithm for general moduli sets. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04), Vol. 5, pp.141-144, 2004.

[18] A. D. Booth. A signed binary multiplication technique. Quaterly J. Math. Appl. Math., Vol. 4, part 2, pp. 236-240, 1951.