An Extensive Review on Fast Sign Detection Algorithm for the RNS Moduli

Vanshri Kamble[1], Prof. Suresh Gawande

[1]*Research Scholar,* [2]*Research Guide*

Department of Electronics and Communication, Bhabha Bhopal

Abstract - the brief present is review on fast sign detection algorithm for RNS moduli RNS is mostly used in VLSI implementation of DSP architecture for achieving low power and high speed. Among the application RNS, implementation of FIR filters, IIR filters, adaptive filters, digital frequency synthesis, two dimensional filters, image encryption and coding are most significant. As an integral part of high-speed computing, parallel hardware architecture plays an important role in fast computers. There is always a need for new devices and new techniques to do faster computation. the speed of arithmetic operations in binary number system is limited by the time needed for carry propagation. In RNS, the arithmetic operations are split into smaller parallel operations which are independent of each other. There is no carry propagation between these operations. Hence devices operating in this principle inherit property of high speed and low power consumption. But this property makes overflow detection is very difficult. Hence the moduli set is chosen such that there is no carry generated.

Keywords - RNS, Fast Sign Detection, 32-Bit, FPGA.

I. INTRODUCTION

The RNS is a very old number system. It was found 1500 years ago by a Chinese scholar Sun Tzu. Since the last five decades, RNS's features have been rediscovered and thus the interest in this system has been renewed. The researchers have used the RNS in order to benefit from its features in designing high-speed and fault-tolerance applications.

The fundamental idea of the RNS is based on uniquely representing large binary numbers using a set of smaller residues, which results in carry-free, high-speed and parallel arithmetic. This system is based on modulus operation, where the divider is called modulo and the remainder of the division operation is called residue. The basic notation in the RNS is,

$$x_i = X \bmod m_i = \langle x_i \rangle_{m_i} \; ; 0 \le x_i < m_i$$

$$X \xrightarrow{RNS} \left(\langle x_1 \rangle_{m_1}, \langle x_2 \rangle_{m_2}, \cdots, \langle x_n \rangle_{m_n} \right) \quad ; \quad GCD(m_i, m_j) = 1$$

The RNS uniquely represents any integer X that locates in its dynamic range M, which is the product of the moduli within the moduli set. Any interval of M consecutive integers can be uniquely represented in the RNS.

The principal aspect that distinguishes the RNS from other number systems is that the standard arithmetic operations; addition, subtraction and multiplication are easily implemented, whereas operations such as division, root, comparison, scaling and overflow and sign detection are much more difficult. Therefore, the RNS is extremely useful in applications that require a large number of addition and multiplication, and a minimum number of comparisons, divisions and scaling. In other words, the RNS is preferable in applications in which additions and multiplications are critical. Such applications are DSP, image processing, speech processing, cryptography and transforms, 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. The basic RNS processor's architecture is shown in Figure 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 (RAU); each of them corresponds to one 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).

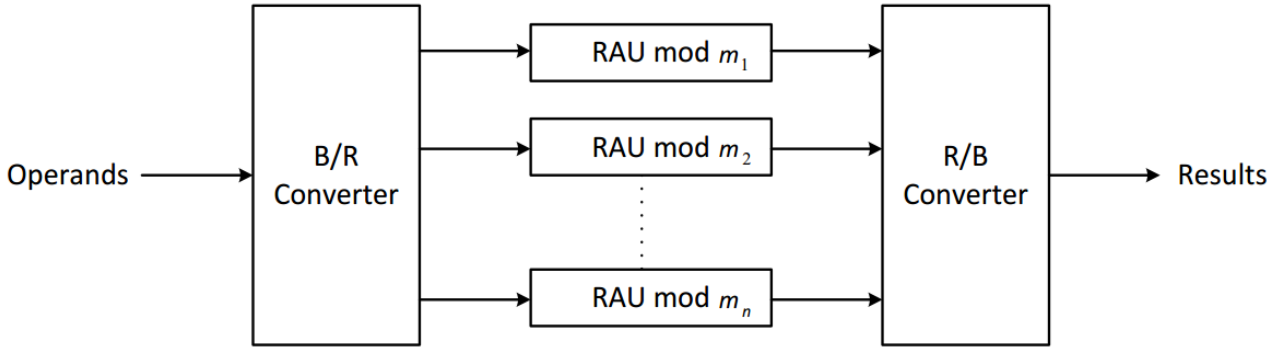

Figure 1.1 The basic architecture of the residue number system.

II. RESIDUE ARITHMETIC SYSTEM

consider two numbers 43 and 29 consider a moduli set $P = [3,5,7]$. then RNS representation of these two numbers are as follows:43 →{1,3,1} 29 →{2,4,1}

Multiplicative Inverse

$$M_i . M_i^{-1} = 1(mod\ p_i)$$

where $M_i = \frac{R}{P_i}$. Thus for the above case $R = 3x5x7 = 105$ and $M_1 = 35, M_1 = 21$ and $M_3 = 15$. Hence $M_1^{-1} = 2$, $M_2^{-1} = 1$ and $M_3^{-1} = 1$.

Reverse Conversion

$$\sum_{i=0}^{k} | \ M_i . M_i^{-1} n_i \ | R$$

where R is range p_i is i^{th} number of moduli set P.

Thus for RNS number {1,3,1},

$|(2\ x\ 35\ x\ 1) + (1\ x\ 21\ x\ 3) + (1\ x\ 15\ x\ 1)|_{105} = |70 + 63 + 15|_{105} = |148|_{105} = 43$ Similarly for {2,4,1} gives 29.

Addition and Multiplication

- Addition

	43 ||	1	3	1
+	29 ||	2	4	1
=	72 ||	0	2	2

- Multiplication

	43 ||	1	3	1
*	29 ||	2	4	1
=	92 ||	2	2	1

Multiplication gives 92 as the result is $|1247|_{105} = 92$.

Most important advantage of residue arithmetic over conventional arithmetic is the ab¬sence of carry propagation, in the two main operations of addition and multiplication, and the relatively low precisions required ranging to individual prime or co-prime number of the moduli set, which enables LUT implementations in various operations However, the fields of Communication and Signal Processing are yet to be explored thoroughly for application on RNS.

Basic advantages of residue arithmetic are:

- High Speed
- Low Power
- Superior Fault Tolerance
- Reduction of Computational Load

III. RNS ALGORITHM

The RNS implementations can provide speedup for addition, subtraction and multiplication. In RNS, a decimal number is represented by an n- tuple of its remainders with respect to each modulus in the moduli set. A remainder called r, of a number X with respect to a modulus m is denoted by r = X mod m and is calculated by r = X- m q, where q is the largest integer which yields a non-negative r RNS can

be the moduli set for a residue number system. This RNS can represent any number from 0 to $(M - l)$, where M is the product of all the moduli in the set.

Number X in this system will be represented by n-tuple $(r_0, r_l, r_2, r_{(n-l)})$,

$$where\ r_i = X\ mod\ m_i\ 0 <= i <= n-1.$$

As for the number of modulus in a set, no limitation has been set. On one hand, the choice of moduli set affects the performance of the algorithms. Most of RNS applications today use 3 or 4 numbers of modules.One of the major attractive features of RNS is that each of the residue digits is independent of each other and hence, there is no carry propagation from one residue digit to another.

Choice Of Moduli Set

There are no fixed rules regarding moduli set used in RNS. The only requirement for a modulus to be in a set is that it has to be a pair-wise relatively prime to any other moduli in the set (Freking.W.L and Parhi.K.K(2000)).However, that guideline is not always necessary. A moduli set can have moduli that have common factor, hence they are not relatively prime to one another. The example for this type of moduli set shown in the research work is $\{2^n + 1, 2^n, 2^n - 1\}$.The popular moduli set used in most of research works are $\{2^n + 1, 2^n, 2^n - 1\}$., because of the following reasons.

1. The dynamic range for the set is the closest to the dynamic range of its 3n binary counterpart.

2. As all the moduli are almost equal, number of operations in the set is spread out almost evenly throughout the moduli set.

3. Implementation of RNS can be accomplished using conventional hardware.

A.Binary To Rns And Rns To Binary Conversion

The basic technique in converting binary to RNS has been presented in where a binary number is represented by a series of terms ai*2i. The residue of any number to the power of 2 with respect to a modulus is fixed and depends on the exponent.

B.RNS To Binary Conversion

The classical approach to RNS to binary conversion is the Chinese Remainder Theorem (CRT). However, this method requires large modulus adders. A more effective way has been shown to convert the residue number to a Mixed- Radix (MR) representation, from which the binary number may be obtained fairly easy. Conversion from residue to mixed-radix representation is, however still a complicated process.

C.Chinese Remainder Theorem (CRT)

The CRT is the basic and the commonly used technique. This technique requires large modulo adders, which causes a serious hardware complexity.

D.Mixed Radix Conversion (MRC)

The MRC technique is very popular, but it is very complicated and slow (Preethy.A.P and Radhakrishnan),(Sangyun Hwang and Sungho Kang. It is relatively better than CRT, because it does not require as many big adders as in CRT

IV. RELATED WORK

S. Kumar and C. H. Chang,[1] ign arithmetic is ineluctable in digital signal processing algorithms. Due to the way signed integer is represented in Residue Number System (RNS), sign detection has been a difficult operation similar to the residue-to-binary conversion. In this paper, a new sign detection method for the three moduli set RNS $\{2^n, 2^n - 1, 2^n + 1\}$ is proposed. It leverages the one complement and circular left shift properties of modulo 2n and $2n \pm 1$ arithmetic in the new modified Chinese Remainder Theorem for sign detection. Compared with the sign detector based on the most efficient reverse conversion algorithm, our proposed sign detector saves on average 24.5% of area and accelerates the computation by $45.6\%\ for\ n = 5, 10, 15, 20.$

T. Tomczak s,[2] In this paper, we propose 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.

S. Bi and W. J. Gross,[3] The Chinese remainder theorem (CRT) and mixed-radix conversion (MRC) are two classic theorems used to convert a residue number to its binary correspondence for a given moduli set {P n,...,P 2, P 1}. The MRC is a weighted number system and it requires operations modulo P i only and hence magnitude comparison is easily performed. However, the calculation of the mixed-radix coefficients in the MRC is a strictly sequential process and involves complex divisions. Thus the residue-to-binary (R/B) conversions and residue comparisons based on the MRC require large delay. In contrast, the R/B conversion and residue comparison based on the CRT are fully parallel processes. However, the CRT requires large operations modulo $M = P_n, \ldots, P_2\ P_2$. In this paper, a new mixed-radix CRT is proposed which possesses both the advantages of the CRT and the MRC, which are parallel processing, small operations modulo P i

only, and the efficiency of making modulo comparison. Based on the proposed CRT, new residue comparators are developed for the three-moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. The FPGA implementation results show that the proposed modulo comparators are about 20% faster and smaller than one of the previous best designs.

P. V. A. Mohan and A. B. Premkumar,[4] In this paper, reverse converters for two recently proposed four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^n+1 - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1}+ 1\}$ are described. The reverse conversion in the three-moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ has been optimized in literature. Hence, the proposed converters are based on two new moduli $sets\ \{(2^n(22^n - 1)), 2^n + 1 - 1\}\ and\ \{(2^n(22^n - 1)), 2^n + 1 + 1\}$ and use mixed radix conversion. The resulting designs do not require any ROM. Both are similar in their architecture except that the converter for the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^n+1 + 1\}$ is slightly complicated due to the difficulty in performing reduction modulo (2^n+1+1) as compared with modulo (2^n+1-1). The proposed conversion techniques are compared with earlier realizations described in literature with regard to conversion time as well as area requirements.

Thu Van Vu,[5] Two conversion techniques based on the Chinese remainder theorem are developed for use in residue number systems. The new implementations are fast and simple mainly because adders modulo a large and arbitrary integer M are effectively replaced by binary adders and possibly a lookup table of small address space. Although different in form, both techniques share the same principle that an appropriate representation of the summands must be employed in order to evaluate a sum modulo M efficiently. The first technique reduces the sum modulo M in the conversion formula to a sum modulo 2 through the use of fractional representation, which also exposes the sign bit of numbers. Thus, this technique is particularly useful for sign detection and for any operation requiring a comparison with a binary fraction of M. The other technique is preferable for the full conversion from residues to unsigned or 2's complement integers. By expressing the summands in terms of quotients and remainders with respect to a properly chosen divisor, the second technique systematically replaces the sum modulo M by two binary sums, one accumulating the quotients modulo a power of 2 and the other accumulating the remainders the ordinary way. A final recombination step is required but is easily implemented with a small lookup table and binary adders.

Z. D. Ulman, [6] A new method of sign detection is proposed. The advantage of this method is a possibility of simultaneous execution of two operations: residue to mixed-radix conversion of the number magnitude and sign detection in one and the same circuit (implicit-explicit conversion).

V. PROBLEM IDENTIFICATION

The reviwed algorithmin a fast sign detection algorith is presented for moduli set include the modulo 2. Allows for parallel implementation. the circuit achieves significant improvements in terms of area, delay, and power. using the different modullo of the SRN algorithm the performnce and of the system in terms of delay and powe can be furter improved in better manner .

VI. PROPOSED METHODOLOGY

As goes throuh the brief study and review of RNS Moduli set . the proposed work is to emplement the RNS moduli on the 64 bit to execute fast sign detection mechanism.

VII. CONCLUSION

The basics of residue arithmetic and the operations like multiplication and addition are approached in ’break and process’ methodology. This reduces complexity and computational load and processes things faster. However the undiscovered part of RNS is the power of its wide range as specified by the moduli set. This system has a non-linear distribution of numbers. in the proposed work RNS Moduli can be implemented for the 64 bit also to perform fastedst sign identification mechanism.

REFERENCES

[1] S. Kumar and C. H. Chang, "A high-speed and area-efficient sign detector for three moduli set RNS {2n, 2n-1, 2n+1}," 2015 IEEE 11th International Conference on ASIC (ASICON), Chengdu, 2015, pp. 1-4.

[2] S. Bi and W. J. Gross, "The Mixed-Radix Chinese Remainder Theorem and Its Applications to Residue Comparison," in IEEE Transactions on Computers, vol. 57, no. 12, pp. 1624-1632, Dec. 2008.

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

[4] P. V. A. Mohan and A. B. Premkumar, "RNS-to-Binary Converters for Two Four-Moduli Sets {2n−1,2n,2n+1,2n+1−1} and {2n−1,2n,2n+1,2n+1+1}," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, no. 6, pp. 1245-1254, June 2007.

[5] Thu Van Vu, "Efficient Implementations of the Chinese Remainder Theorem for Sign Detection and Residue Decoding," in IEEE Transactions on Computers, vol. C-34, no. 7, pp. 646-651, July 1985.

[6] Z. D. Ulman, "Sign Detection and Implicit-Explicit Conversion of Numbers in Residue Arithmetic," in IEEE Transactions on Computers, vol. C-32, no.

[7] N. Szabo, “Sign detection in nonredundant residue systems,” IRE Trans. Electro1”. Comput., vol. EC-11, no. 4, pp. 494-500, Aug. 1962.

[8] E. Al-Radadi and p. Siy, “RNS sign detector based on Chinese remainder theorem II (CRT II),” Comput. Math. Appl., vol. 46, nos. 10-11, pp. 1559-1570, 2003.

[9] M. Akkal and P. Siy, “Optimum RNS sign detection algorithm using MRC-II with special moduli set,” J. Syst. Arch., vol. 54, no. 10, pp. 911-918, Oct. 2008.

[10] S. Piestrak, “Design of residue generators and multioperand modular adders using carry-save adders,” IEEE Trans. Comput., vol. 43, no. 1, pp. 68-77, Jan. 1994.

[11] R. Zimmermann, “Efficient VLSI implementation of modulo (2n ± 1) addition and multiplication,” in Proc. 14th IEEE Symp. Comput. Arith¬metic, 1999, pp. 158-167.

[12] K. Furuya, “Design methodologies of comparators based on parallel hardware algorithms,” in Proc. 10th ISCIT, Oct. 2010, pp. 591-596.