International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 9747-9756

http://www.ripublication.com

Efficient Design Of 64 Bit Floating Point RNS-MAC Unit

C.M.Saru Preethi[1] and I.Mary Sajin Sanju[2]

Department of Electronics and Communication Engineering,
Sathyabama University, Chennai, Tamil Nadu, India
Email: [1] sarupreethi@gmail.com

ABSTRACT

This paper presentsthe design and implementation of floating point RNS-MAC unit. The residue number systemprovides encoding of large numbers into a group of small numbers which results in significant speed up of the overall processing of data. This fact sways the implementation of RNS in some applications where intensive processing is inevitable. This paper focuses on MAC unit which consists of three basic units namely multiplier, the conversion unit for forward and reverse conversion and an accumulator unit .The floating point number can be represented by $M \times B^E$,where M is the mantissa ,E is the exponent and B is the base.The choice of moduli and the process of converting decimal to residue number, which is forward conversion and the inverse conversion with the help of Chinese reminder theorem (CRT) is done here.

Keywords: Residue, Moduli, Chinese reminder theorem (CRT), Mixed radix conversion (MRC), Residue Number System (RNS), Multiply Accumulate unit (MAC), Moduli.

INTRODUCTION:

The portable computing and wireless communication systems require high speed computation, low power consumption, complex functionality, and real-time processing capabilities.The main components used in Digital signal processor (DSP) are multiplier, adder and accumulator. Hence the performance of MAC unit plays an important role in the design of filters. The performance can be increased by optimized design of multiplier and adder. Residue number system (RNS) has beenproposed by A. Omondi and B. Premkumar(2007), allowing high-accuracy integer-values arithmetic operations to be decomposed into independent (carry-free) low-accuracy computations that can be performed in parallel. The RNS provides an attractive

alternative to traditional weighted number systems for high speed digital signal processing (DSP) and communication applications. To interface with the digital system, where the binary numbers are employed, the RNS-based processors require the conversions between binary form to the residue representation. S.Piestrak (1995)proposed the conversion from binary number to residue representation is relatively easier, converting the other way, however, is much more difficult, in which the Chinese Remainder Theorem (CRT) or Mixed Radix Conversion (MRC) is generally employed.

The floating point number can be represented by $M \times B^E$, where M is the mantissa , E is the exponent and B is the base.The floating point representation chosen here in half precision.The input is in 16 bit floating point representation (half precision) and the output is in 32 bit floating point representation (single precision).

In this paper the first section is the introduction about the floating point MAC and the second section is about the concepts of the residue number system.In the third section the existing system of floating point MAC unit is discussed. The fourth and fifth section describes the proposed MAC units and conclusions.

RNS REPRESENTATION:

RNS is defined by a set of relatively prime integers called the *moduli*. The moduli-set is denoted as $\{m_1,m_2,\ldots..m_n\}$ where m_i is i^{th} modulus. Each integer X can be represented as a set of smaller integers called the residues. The residue-set is denoted as $\{r_1,r_2,\ldots\ldots r_n\}$ where r_i is the i^{th} residue. The residue r_i is defined as the least positive remainder when X is divided by the modulus m_i. This relation can be notationally written based on the following equation .

$$Xmodm_i = r_i$$

The RNS is capable of uniquely representing all integers X that lie in its *dynamic range*. The dynamic range is determined by the moduli-set $\{m_1,m_2,\ldots..m_n\}$ and denoted as M as shown in the equation.

$$M = \pi_{i=1}^{n} m_i$$

The RNS provides unique representation for all integers in the range between 0 and *M-1* . If the integer X is greater than *M-1*, the RNS representation repeats itself. Therefore, more than one integer might have the same residue representation.

It is important to emphasize that the moduli have to be relatively prime to be able to exploit the full dynamic range M.

In the above section the RNS scheme for unsigned numbers is discussed. However, some applications require representing negative numbers. To achieve that, we can partition the full range [0:M-1] into two approximately equal halves: the upper half represents the positive numbers, and thelower half represents the negative numbers. The numbers X that can be represented using this new convention have to satisfy the following relations as shown in following equations.

$$-\frac{M-1}{2} \leq X \leq \frac{M-1}{2} \; if \; M \; is \; odd$$

$$-\frac{M}{2} \leq X \leq \frac{M}{2} - 1 \; if \; M \; is \; even$$

CHOICE OF MODULI:

The residue to binary converters based on the form $\{2^{n-1}, 2^n, 2^{n+1}\}$, have been widely used in RNS architectures, asthey offer efficient circuits. However,to comply with the RNS requirement for pairwise relatively prime moduli in the base, no more than one modulus of the form 2^ncan be utilized. Therefore,not all residue channelsmay benefit from the efficiency of architectures for operations modulo 2^n; thus the performance of the system is determined by the remainder of the channels.

The basic requirement for the moduli is to havebalance or it can be put as, the two moduli should havethe difference as small as possible. A common choiceof prime modulus that does not complicate arithmetic and which has good representational efficiency is m_i=$2^i - 1$. Not all pairs of numbers of the form $2^i - 1$arerelatively prime, but it can be shown that$2^j - 1$ and$2^k - 1$ are relatively prime if and only if *j* and *k* arerelatively prime. Many moduli sets are based on thesechoices, but themoduli sets of the form $\{2^{n-1}, 2^n, 2^{n+1}\}$are amongthe most popular in use. If the moduli are small, then alarge number of them may be required to ensure a sufficient dynamic range. Of course, ultimately thechoices made, and indeed whether RNS is useful or not,depend on the particular applications and technologiesat hand.

EXISTING SYSTEM:

The modules that are required for implementing a Floating Point RNS MAC are Binary to RNS Converter, Modulo Adder, Modulo Multiplier, RNS to Binary Converter and Accumulator.Basically floating point inputs are given as Mantissa and Exponent. The input to the MAC is in half precision (16-bit) notation.The output of MAC is in single precision (32-bit) notation. The sign bit indicates whether the number is positive (sign bit=0) or negative (sign bit=1). Exponent is biased exponent and the mantissa is in normalized form.

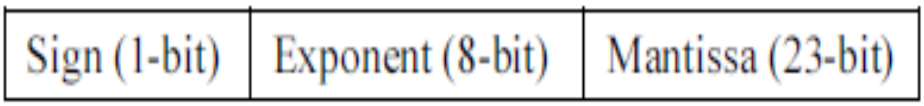

Figure 2. Floaing Point Representaion

The block diagram of the residue number system based MAC unit is shown in figure 1.

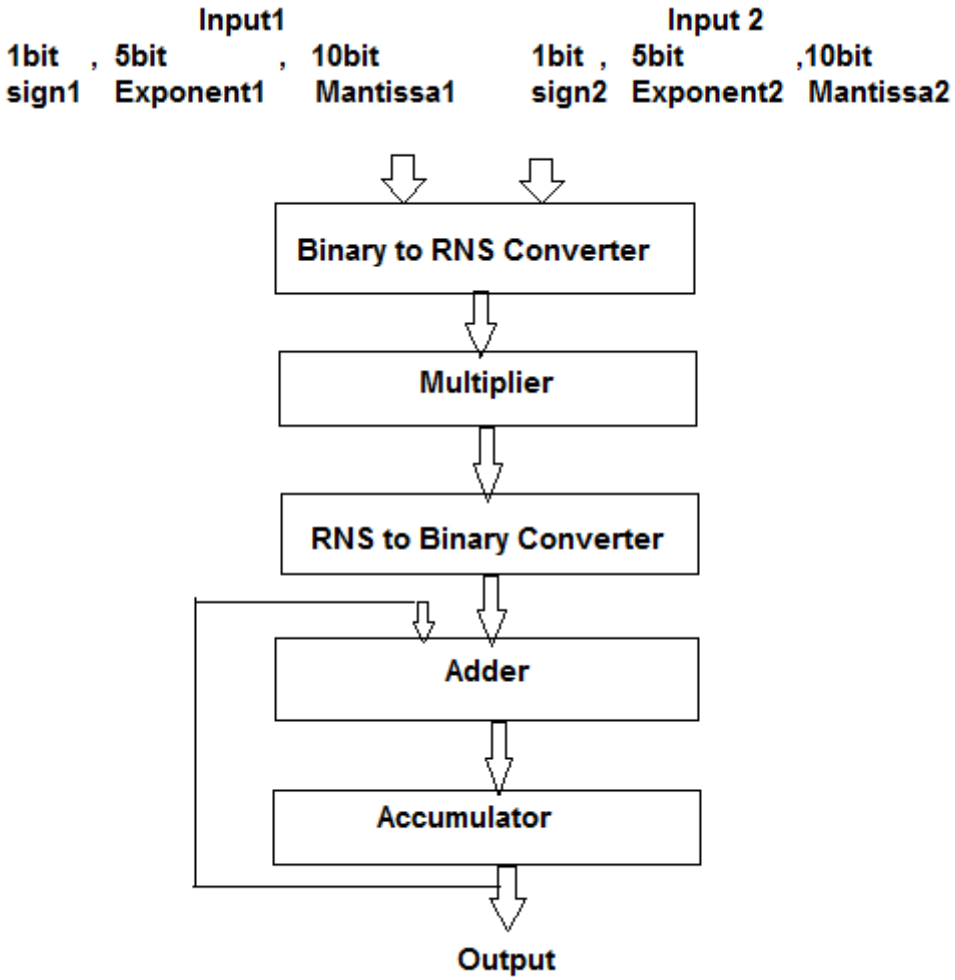

Figure1. Block diagram of floating point MAC unit.

The flow of operations for Floating point RNS MAC unit is as follows:

- The unbiased Exponent is converted tobiased by adding 15 (this value depends on number of bits used to represent exponent). This biasing is done to ensure that the Exponent is unsigned.
- The Mantissa and biased Exponent isconverted to Residue Number System. In RNS, based on the moduli, residues are obtained.
- For multiplication, the Mantissa should be
- Multiplied and Exponent should be added. For this, an RNS Mantissa modulo multiplier and RNS Exponent modulo adder are used.
- Using accumulator the products are addedand saved.

PROPOSED SYSTEM:

The performance of the MAC unit can be increased by the optimized design of adder.In the proposed system the existing kogge-stone adder is replaced by a low power adder.The power consumption can be reduced by modifying the low power fused Adder/Subtracor.

The floating point multiply and accumulate unit consists of binary to RNS convertor, floating point multiplier, RNS to binary convertor, adder and accumulator circuitry.

FORWARD CONVERSION:

The forward conversion stage is considered as an overhead in the overall RNS. Forwardconverters are usually classified into two categories based onthe moduli used. The first category includes forwardconverters based on arbitrary moduli-sets. These convertersare regularly built using Look Up Tables (LUTS) whichconsists of ROM's. The second category includes forwardconverters based on special moduli-sets. The use of thesespecial moduli-sets simplifies the forward conversionalgorithms and architectures.Usually, the special moduli-sets are referred to as low-costmoduli-sets. A typical architecture for the implementation of aforward converter from binary to RNS representation usingthe special moduli-set is shown in Figure 2.The forward converter has to be efficient in terms of area, speed, and power.The use of special moduli-sets simplifies the forward conversion algorithms and architectures. The special moduli-set converters are usually realized using pure combinational logic.

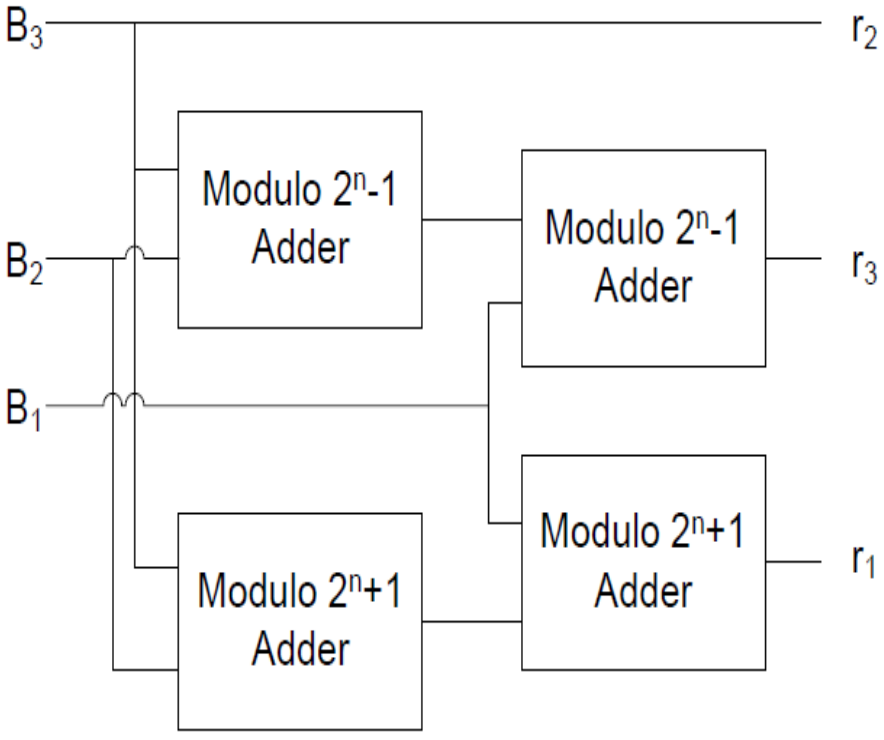

Figure 3.$\{2^{n-1}, 2^n, 2^{n+1}\}$forward converter

The input binary number is divided into 3 blocks each of 'n' bits. Thus, if a given binary number X is divided into blocks B_1,B_2,B_3.

$$B_1 = \sum_{j=2n}^{3n-1} 2^{j-2n} x_j$$

$$B_2 = \sum_{j=n}^{2n-1} 2^{j-n} x_j$$

$$B_3 = \sum_{j=0}^{n-1} 2^{j} x_j$$

Thus, the input binary number, X can be given in terms of the blocks as,

$$X = B_1 2^{2n} + B_2 2^n + B_3$$

Then the residues can be obtained as follows,

$$r_1 = |B_1 + B_2 + B_3|_{2^{n-1}}$$

$$r_2 = |B_3|_{2^n}$$

$$r_3 = |B_1 - B_2 + B_3|_{2^{n+1}}$$

The residues r_1,r_2,r_3 are calculated for the binary floaing point input number X.

FUSED ADDER/SUBTRACTOR:

The adder structure can be any conventional adder that can be used like a ripple carry adder (RCA), a carry look ahead adder or any parallel prefix tree. Here we make use of the fused low power adder/subtractor architechture. Figure 3 shows the low power fused adder/Subtractor architechture.

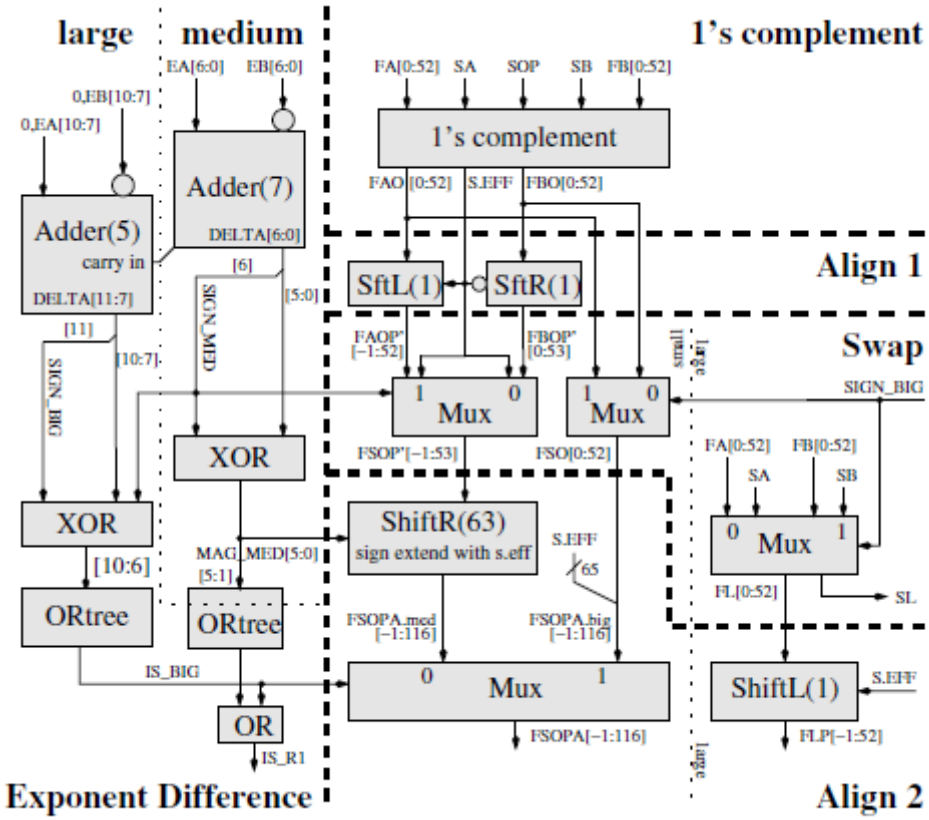

Figure 4. Fused Adder/Subtracor

The first MSB bit of input A and B are Xor ed. Exponent of A is compared with the exponent of B and the greatest of two is chosen .The Mantissa of the greatest exponent is taken, for addition add 256 and for subtraction 256 is subtracted. The performance of the MAC unit can be increased by the optimized design of adder.

REVERSE CONVERSION:

Reverse conversion algorithms in the literature are all based on either Chinese Remainder Theorem (CRT) or Mixed-Radix Conversion (MRC). The MRC is an inherently sequential approach. On the other hand, the CRT can be implemented in parallel. The main drawback of the CRT based R/B reverse converter, is the need of a large modulo adder in the last stage. All the converters proposed in the literature have this problem. The reverse conversion is one of the most difficult RNS operations and has been a limiting factor to a wider use of RNS.In general, the realization of a VLSI implementation of R/B converters is still complex and costly.

The new Chinese Remainder Theorems (NCRT) makes thecomputation faster and efficient without any extra overheads.A new high-speed ROM-less residue to binary converter forthe three moduli residue number system of the

form$\{2^{n-1}, 2^n, 2^{n+1}\}$ is used. Unlike any other converter, its delay includes the time of only one 1's complement addition of two 2n-bit numbers which is only 2/3 of the binary range of the RNS equal to as 3n. Thus, it is potentially the fastest known residue-to-binary converter.

The mathematical equations for CRT are as follows,

For a set of moduli $\{m_1, m_2, m_3 \ldots . m_i\}$ and the residues are $\{r_1, r_2, r_3 \ldots . r_i\}$,then the binary number X is given as,

$$X = |\sum_{i=1}^{n} r_i M_i^{-1} M^i|$$

Where,

$$M_1 = \frac{m_1 * m_2 * m_3}{m_1}$$

$$M_2 = \frac{m_1 * m_2 * m_3}{m_2}$$

$$M_3 = \frac{m_1 * m_2 * m_3}{m_3}$$

RESULTS AND DISCUSSION:

The architechure is designed using verilog HDL and simulated using Altera Modelsim.he simulation resuls for the existing and proposed floating point RNS-MAC units are obtained.Fig 4 shows he simulation ouput for existing 16-bit floaing point residue number MAC unit.

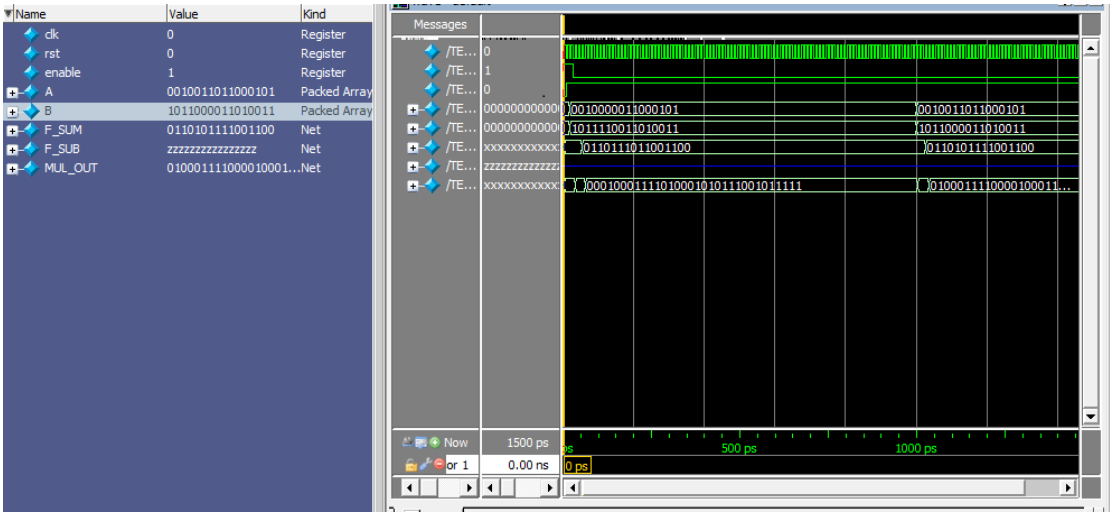

Figure.5 Simulation Result for Existing 16-bit floating point RNS-MAC unit.

Fig 5 shows the output of the proposed floating point residue number MAC using fused Adder /Subtarctor.

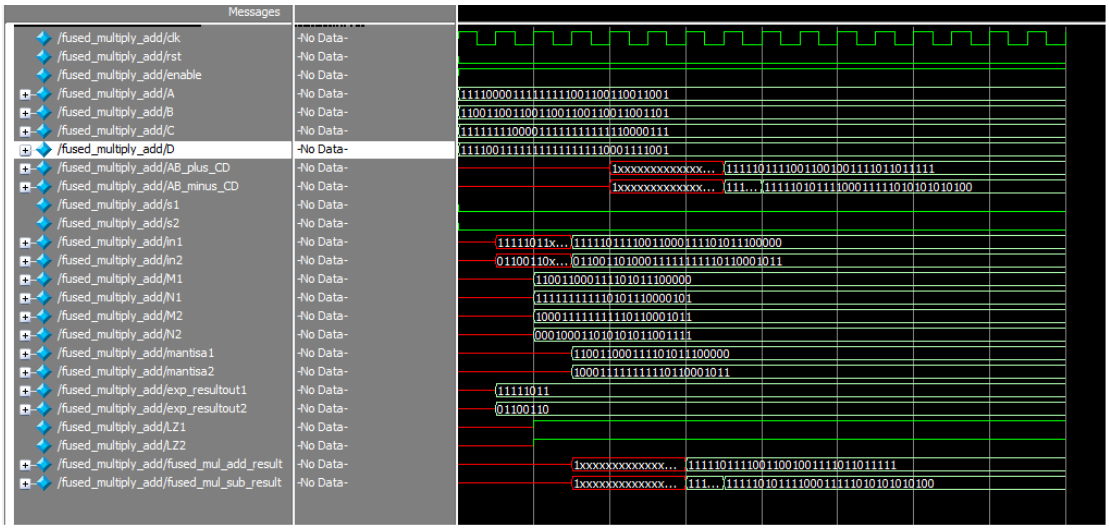

Figure 6. Simulation Result for Existing 64-bit floating point RNS-MAC unit.

PERFORMANCE ANALYSIS:

Table 1.Performance comparision of existing and proposed

Device	Power Consumption	Delay
Existing(16-bit)	71.80mw	2.83ns
Proposed(16-bit)	64.42mw	1.01ns

Table 2 Performance Analysis of 64-bit Floating point RNS-MAC Unit

Device	Power Consumption	Delay
Proposed (64-bit)	64.42mW	10.12ns

Table1 shows the performance comparison of existing and proposed 16 –bit Floating point RNS-MAC Units. It shows the comparison between the power and the delay of the circuits.Table2 shows the performance analysis of proposed 64–bit Floating point RNS-MAC Unit. It shows the comparison between the power and the delay of the circuits.

CONCLUSION:

The 64 bit MAC unit is designed in Verilog HDL and implemented in Altera ModelSim. The large number is represented in the form of residues. So addition and

multiplication is performed parallely on all the residues. Thus arithmetic is performed in a faster rate. Thus 64-bit Floating Point RNS MAC unit is found to be of efficient. Although the basic arithmetic operations like addition and multiplication are easily implemented using RNS. However, there are some operations that are not easily implemented. Some of these unimplementable or difficult to implement functions are division, magnitude comparison, sign determination, scaling and overflow detection. Many works are being done in this area to have these operations done with residue number system, so that the system emerges as a better option for processing data in the growing technology.

REFERENCES:

[1] A. Hiasat and A. Sweidan, "*Residue-to-Binary Decoder for an Enhanced Moduli Set,"* IEE Proceedings, Computers andDigital Tech., Vol. 151, No.2, pp.127-130, March 2004.

[2] A. Omondi and B. Premkumar, *"Residue Number System:Theory and implementation",* Imperial College Press, 2007, (ISBN 978-1-86094-866-4).

[3] C.L.Wey,*"Residue-to-binary Converters for Highspeed Digital Signal Processing,"* Proc. of 6th IEEE International Conference on Electro/Information Technology(EIT), E. Lansing, Michigan, May 2006.

[4] G. C. Cardarilli, A. Nanareelli, and M. Re, *"Reducing power dissipation in FIR filters using the residue number system,"* Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Vol. 2, pp. 320-323, 2000.

[5] N.Mathan, "*CNTFET based Highly Durable Radix-4 Multiplier using an Efficient Hybrid Adder",*Biosciences Biotechnology Research Asia, December 2014,Vol. 11(3), pp 1855-1860.

[6] Neha Singh*"An overview of residue number system"National* seminal on circuits,devices and communication,Nov 2008,B.I.T Mesra, Ranchi.

[7] N.Vivek. K Anusudha *"Design of RNS Based Addition Subtraction and Multiplication Units"*International Journal of Engineering Trends and Technology (IJETT) – Volume 10 Number 12 - Apr 2014

[8] R. Zimmerman, *"Efficient VLSI implementation of modulo* ($2^n \pm 1$) *addition and multiplication,"* in Proc. 14th IEEE Symp. Computer Arithmetic, pp. 158-167, Apr. 1999.

[9] S.Piestrak*, "A high speed realization of a residue to binary number system Converter"*, IEEE Transactions on Circuits and systems-II, vol.42, no. 10, October 1995

[10] S. Yen, S. Kim, S. Lim and S. Moon, *"RSA Speedup with Chinese Remainder Theorem Immune against Hardware Fault Cryptanalysis,"* IEEE Transactions on Computers, vol. 52, no. 4, pp. 461-472, 2003.

[11] Wei Wang, M. N. S. Swamy, and M. O. Ahmad, *"An area-time-efficient residue-to-binary converter,"* Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Vol. 2, pp. 904-907, 2000.