DOI: http://dx.doi.org/10.26483/ijarcs.v9i3.6113

Volume 9, No. 3, May-June 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A SINGLE BIT ERROR DETECTION AND CORRECTION BASED ON THEMRC AND THE MP TECHNIQUES IN RRNS ARCHITECTURE

Yaw Afriyie
Department of Accountancy and Commerce
School of Business and Law
University for Development Studies
Wa, Ghana.

M.I. Daabo
Department of Computer Science
Faculty of Mathematical Sciences
Universityf for Development Studies
Navrongo, Ghana.

Abstract: This paper presents some results on single error detection and correction based on the Redundant Residue Number System (RRNS).The proposed technique utilizes the Mixed Radix Conversion (MRC) and the Modulus Projection (MP) algorithms that significantly simplifies the error correction process for integers. The MP considerably reduces the computational steps and hardware architecture and further improve the processing speed. This results in a considerable improvement in the speed by **97**% and tends to require about **96**% less hardware resources in the proposed scheme when compared with the existing scheme used in this work. The proposed scheme is built on simple adders in the design of the architecture which saw a considerable improvement in both area and speed in as compared to the work by Yangyang*et. al* [6] which used ROMs and latches for the design of their architecture.

Keywords: MP, MRC, Mixed Radix Digits, Residue Number System, Redundant Residue Number System (RRNS)

I. INTRODUCTION

The increasing demand for speed and accuracy in digital communication has led to the introduction of parallel computing. RNS is a form of parallel computing that was first introduced by Garner [1]. RNS provides a very fast arithmetic due to its capability of performing the carry-free operations, i.e. addition, subtraction and multiplication. RNS also possesses parallel and fault tolerant features, which are seen to be helpful for hardware implementation. Barsi and Maestrini [2] in their work posited that RNS offers a great speed as a result of its carry-free nature. Because of this, these have led to the increase in the development of a number of error detection and error correction algorithms based on RNS. When some redundant residues are added, the RNS has the possibility of error detection and correction, hence, the term Redundant Residue Number System (RRNS). Redundancy is achieved through various schemes in RNS. Behrouz [3] indicated that the ratio of the redundant bits to information bit is important to any scheme. There are significant works by Mandelbaum [4] and Szabo [5] concerning RRNS in the detection and correction of errors. Some concepts such as legitimate and illegitimate range for consistency checking related to error techniques are also studied. An algorithm that is based on the detection and correction of single bit errors by Szabo and Tanaka [4] allowed for scaling by a product moduli from the RNS based
on n clock cycles. The main difficulty in their work was that the scaling reduced the processing speed. The clock cycles denotes the time required for the elementary operation. In the work by Yangyang *et. al* [6], a discussion of a single bit error correction algorithm that implements ROMs and latches were used. The cost of ROMs and latches are however expensive to build affecting both the area and delay of the architecture. Some useful investigations were conducted by Jenkins *et. al* [7] to detect and correct single bit errors based on the Mixed Radix Conversion (MRC) and the Base Extension (BEX) techniques to RRNS application in digital filters and residue number error checkers due to efficient pipeline architectures. Goldreich*et. al.* [8], proposed a performance evaluation of Residue number based on the Chinese Remainder Theorem (CRT) in the detection and correction of errors in RRNS. The resulting effects of the schemes in [5]-[8] when compared to the MRC and the MP proposed in this work offer a low complexity and detects and corrects single bit errors faster. To this effect, the proposed scheme offers a great advantage in terms of area cost, delay and would be able to detect and correct single bit errors faster.

In this paper, we propose a new scheme that will show the effectiveness of the RRNS based on MRC and the MP that will detect and correct bit single errors.

II. RESIDUE ARITHMETIC FUNDAMENTALS

RNS is characterized by a set of k pairwise relatively prime positive integers, i.e. the greatest common divisor gcd(m_i, m_j) =1 with $i \neq j$, $m_1, m_2 \ldots m_{k-1}, m_k$ called the moduli, that is formed in increasing, i.e., $m_1 < m_2 < \cdots < m_{k-1} < m_k$[3]. Their products represent the interval [0, M) called the legitimate range that defines the useful computational range of the number system, that is,

$$M = \prod_{i=1}^{N} m_i \qquad \ldots\ldots\ldots\ldots\ldots (1)$$

To represent positive and negative numbers, the dynamic range is defined as $[-((M-1)/2, ((M-1)/2]$ if M is odd and $M/2$ if M is even. Every natural integer X in the legitimate range can be represented by a set of residues $r_1, r_2 \ldots, r_{k-1}, r_k$ where

$$r_i \equiv X(mod)m_i \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

With $i \in$ [1, k] and |X $|m_i$ denotes X modulo m_i. Due to the carry-free property, the three operations namely addition, subtraction and multiplication can be operated with respect to the moduli independently, i.e.

$$x_1, x_2 \ldots x_k * \ x_k * y_1, y_2 \ldots y_k = z_1, z_2 \ \ldots \ z_k, \ z_i \equiv |x_i * y_i \ \ |m_i \quad (3)$$

With * denotes the three operations. Consequently, RNS is able to provide a fast arithmetic.

© 2015-19, IJARCS All Rights Reserved

III. CONVERSION

It is well known that MRC and CRT are approaches that are often applied in conversion. This can be seen in the work of Mandelbaum [4]. This study will be limited to the MRC and the MP techniques because the real time implementation of the CRT involves a modular operation with a large integer M which results in large complexities. Daabo [9] indicated that to prevent the computations with such larger M, the CRT satisfies the real-time signal processing time due to its parallel means of computation and there is a constant limit to this approach. The process of converting from conventional representations to RNS is known as forward conversion whilst converting from the RNS to the conventional representations is known as the reverse conversion. The residue to conventional number representation is done mainly by the MRC or the CRT as seen in the work of Mohahosseini [10]. The MRC is carried out by a weighted approach. The MRC is expressed by the following equations;

X = $a_1 + a_2m_1 + a_3m_1m_2 + a_nm_1m_2m_3 \dots m_{k-1}$(4)

where $a_{i,i=1,k}$ is the Mixed Radix Digits (MRDs) can be computed as:

$$a_1 = x_1$$
$$a_2 = \left|(x_2 - a_1)|m_1^{-1}|_{m_2}\right|_{m_2}$$
$$a_3 = \left|((x_3 - a_1)|m_1^{-1}|_{m_3} - a_2)|m_2^{-1}|_{m_3}\right|_{m_3}$$
$$\vdots$$
$$a_k = \left|(((x_k - a_1)|m_1^{-1}|_{m_k} - a_2)|m_2^{-1}|_{m_k} - \dots - a_{k-1})|m_{k-1}^{-1}|_{m_k}\right|_{m_k} \quad (5)$$

This paper presents an efficient algorithm for detection and correction of single bit errors for the moduli set $\{2^n - 1, 2^n, 2^n + 1$, 22n−3, 22n+1+1.

The rest of this paper is organized as follows: Section 4 presents the proposed method. In Section 5, the hardware implementation of the proposed scheme is presented, a simplified algorithm with numerical illustrations are also presented. The performance of the proposed scheme is evaluated in Section 6 whiles the paper is concluded in Section 7.

IV. PROPOSED METHOD

This section provides a new method for detecting and correcting single bit errors in RRNS in the given moduli set.

a. Proposed Algorithm

The algorithm for the proposed scheme is given below;

1. Compute the integer message X using the MRC.
2. Perform iterations using $C_t^n = \frac{n!}{(n-t)!t!}$ by discarding a residue at time
3. An error occurs if the integer message X falls within the illegitimate range but not found within the legitimate range.
4. Declare the error in the residue digit

In the course of computing the MP into integers, the decoding algorithm is used. The algorithm is premised on the MP and the MRC. For the MP, we have;

$$X_i = X\left(mod\frac{M_{m_r}}{M_i}\right) \quad (6)$$

For the given moduli set $S = \{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 3, 2^{2n+1} + 1$ where *m1=2n−1, m2=2n, m3=2n+1, m4=22n−3 and* $m_5 = 2^{2n+1} + 1$the respective $m_i - projections$ are;

$$\widehat{M}_1 = (2^n)(2^n + 1)(2^{2n} - 3)(2^{2n} + 1) \quad (7)$$
$$\widehat{M}_2 = (2^n - 1)(2^n + 1)(2^{2n} - 3)(2^{2n} + 1) \quad (8)$$
$$\widehat{M}_3 = (2^n - 1)(2^n)(2^{2n} - 3)(2^{2n} + 1) \quad (9)$$
$$\widehat{M}_4 = (2^n - 1)(2^n)(2^n + 1)(2^{2n} + 1) \quad (10)$$
$$\widehat{M}_5 = (2^n - 1)(2^n)(2^n + 1)(2^{2n} - 3) \quad (11)$$

The projections for the respective moduli are given as;

$$X_1 = |X|_{(2^n)(2^n+1)(2^{2n}-3)(2^{2n}+1)} \quad (12)$$
$$X_2 = |X|_{(2^n-1)(2^n+1)(2^{2n}-3)(2^{2n}+1)} \quad (13)$$
$$X_3 = |X|_{(2^n-1)(2^n)(2^{2n}-3)(2^{2n}+1)} \quad (14)$$
$$X_4 = |X|_{(2^n-1)(2^n)(2^n+1)(2^{2n}+1)} \quad (15)$$
$$X_5 = |X|_{(2^n-1)(2^n)(2^n+1)(2^{2n}-3)} \quad (16)$$

The multiplicative inverses for the MRC based on the same moduli set are computed as follows;

$$|m_1^{-1}|m_2 = |(2^n - 1)^{-1}|_{2^n} = -1 \quad (17)$$
$$|m_1^{-1}|m_3 = |(2^n - 1)^{-1}|_{2^n+1} = -2^{n-1} \quad (18)$$
$$|m_2^{-1}|m_3 = |(2^n)^{-1}|_{2^n+1} = 1 \quad (19)$$
$$|m_4^{-1}|m_5 = |(2^{2n} - 3)^{-1}|_{2^{2n}+1} = 2^{2n} \quad (20)$$

Now the $a_{i's}$ can be computed using the MRC as follows;

$$a_1 = x_1 \quad (21)$$
$$a_2 = |(x_2 - x_1)|m_1^{-1}|m_2|m_2 = |(x_2 - x_1)(-1)|m_2 = |x_1 - x_2|_{2^n} \quad (22)$$
$$a_3 = |(x_3 - x_1)|m_1^{-1}|m_3 - a_2)|m_2^{-1}|m_3|m_3 = |(x_3 - a_1)(-2^{n-1}) - a_2|_{2^n+1} \quad (23)$$

The decimal equivalent for the non-redundant part is thus;

$$X = a_1 + a_2m_1 + a_3m_1m_2 \quad (24)$$
$$= a_1 + a_2(2^n - 1) + a_3(2^n - 1)(2^n) \quad (25)$$

The redundant part is;

$$a_4 = x_4 \quad (26)$$
$$a_5 = |(x_5 - x_4)|m_4^{-1}|m_5|m_5 = |(x_5 - x_4)(2^{2n})|m_5 = |2^{2n}(x_5 - x_4)|_{2^{2n}+1} \quad (27)$$

The decimal equivalent for the redundant part is thus;

$$\bar{X} = a_4 + a_5m_4 \quad (28)$$
$$= a_4 + (2^{2n}(a_5 - x_4)(2^{2n} - 3)) \quad (29)$$

V. HARDWARE IMPLEMENTATION

For the considered moduli set for the non-redundant part $m = \{2^n - 1, 2^n, 2^n + 1\}$ with its corresponding residues (x_1, x_2, x_3). We let the binary representations of the residues be;

$$x_1 = x_{1,n-1} \dots x_{1,1}x_{1,0} \quad (30)$$
$$x_2 = x_{2,n-1} \dots x_{2,1}x_{2,0} \quad (31)$$
$$x_3 = x_{3,n} \dots x_{3,1}x_{3,0} \quad (32)$$

and the redundant part as;

$$x_4 = x_{4,2n-1}, x_{4,2n-2}, \dots x_{4,0} \quad (33)$$
$$x_5 = x_{5,2n,2n-1,2n-2} \dots x_{5,0} \quad (34)$$

Thus by the MRC technique,

$$a_1 = x_1 \quad (35)$$
$$a_2 = |x_1 + \bar{x}_2|_{2^n}$$
$$= \left|x_{1,n-1}x_{1,n-2} \dots x_{1,1}x_{1,0} + \bar{x}_{2,n-1}\bar{x}_{2,n-2} \dots \bar{x}_{2,1}\bar{x}_{2,0}\right|_{2^n} = \left|\underbrace{x_{1,n-1}x_{1,n-2} \dots x_{1,1}x_{1,0}}_{n\ bits} + \underbrace{\bar{x}_{2,n-1}\bar{x}_{2,n-2} \dots \bar{x}_{2,1}\bar{x}_{2,0}}_{n\ bits}\right|_{2^n}$$

(36)

$$a_3 = |2^{n-1}(a_1 + \bar{x}_3) - a_2|_{2^n+1}$$
$$= |2^{n-1}a_1 + \bar{x}_3 2^{n-1} - a_2|_{2^n+1}$$
$$= \left|2^{n-1}\underbrace{(x_{1,n-1} \dots x_{1,0})}_{n\ bit} + \right.$$

2n−1(x3,n...x3,1x3,0)n+1−(x1,n−1...x1,1x1,0)n bits−(x2,n−1... x2,1x2,0)n bits2n+1

(37)

Implementation of equations (30) – (37) gives the correct output of a_3 whenever an error occurs in the non-redundant part. The decimal representation of the architecture for the non-redundant from equations (28) and (29) is;

$$X = x_1 + 2^na_2 - a_2 + 2^{2n}a_3 - 2^na_3 \quad (38)$$
$$= \bar{\tau} + 2^{2n}a_3 \quad (39)$$
$$= \bar{\tau}_{2n-1} \dots \bar{\tau}_0 \underbrace{000 \dots 0}_{n+1\ bit} + \underbrace{a_{3,n} \dots . a_{3,0} \dots \overbrace{000 \dots 0}^{2n}}_{3n+1\ bit} \quad (40)$$

where

© 2015-19, IJARCS All Rights Reserved

$\bar{\tau} = \rho - 2^n a_3 - a_2$

$=$

$\underbrace{\rho_{,2n-1} \ldots \rho_0}_{2n\ bit} + \underbrace{\bar{a}_{3,n} \bar{a}_{3,n-1} \ldots \bar{a}_{3,0}}_{2n\ bit} + \underbrace{111 \ldots 1}_{n\ bit} +$

$$\underbrace{a_{2,n-1}, a_{2,n-2} \ldots a_{2,0}}_{2n\ bit} \underbrace{111 \ldots 1}_{n\ bit} = \bar{\tau} \qquad (41)$$

and,

$$\rho = \underbrace{000 \ldots 0}_{n\ bit} \underbrace{x_{1,n-1} \ldots x_{1,0}}_{n\ bit} \bowtie \underbrace{a_{2,n-1}, a_{2,n-2} \ldots a_{2,0}}_{n\ bit} \underbrace{000 \ldots 0}_{n\ bit}$$

$$= \underbrace{x_{1,n-1} \ldots x_{1,0} a_{2,n-1}, a_{2,n-2} \ldots a_{2,0}}_{n\ bit} \qquad (42)$$

VI. Proposed Architecture

The residue number is converted to the Mixed Radix System (MRS) in parallel with the computation of the MRS, which detects and corrects primarily based on the non-redundant part. In the event of an error in any of the channels, the redundant part will be employed in the detection and correction of the residue digit error.The Mixed Radix Digits (MRDs) are computed according to equation (24) where all the MRDs $a_1, a_2$and a_3 are computed individually in equations (21) to (23). The a_{is} which are the MRDs for the non-redundant part are computed separately which is seen in Figure 1. As shown in Figure 1, a_3 is computed using Carry Save Adders (CSAs) 1, 2 and 3 and two regular $(n+1)$ bit Carry Propagate Adders (CPAs) 1 and 2 respectively. All the CSAs require an area of $(n+1)\Delta_{FA}$ each whilst CPAs 1, 2 and 3 require an area of n each. In order to obtain the MRD a_3 will require a total area of $(11n+4)\Delta_{FA}$. Regarding the delay, CSA (i.e. CSAs 1, 2 and 3) impose a delay of D_{FA} each in the reverse convertor. CPAs 1 in the reverse convertor require a delay of $(4n+2)D_{FA}$. CPAs 2 and 3 in the reverse convertor require a delay of $(6n + 2DFA$ each. The reverse convertor for the MRC also has one CSA that also impose a delay on the system. The total delay needed for the proposed scheme is $(10n+7)D_{FA}$. The schematic diagram for the proposed scheme is shown in Figure 1.
The schematic diagrams for the proposed scheme are shown below.

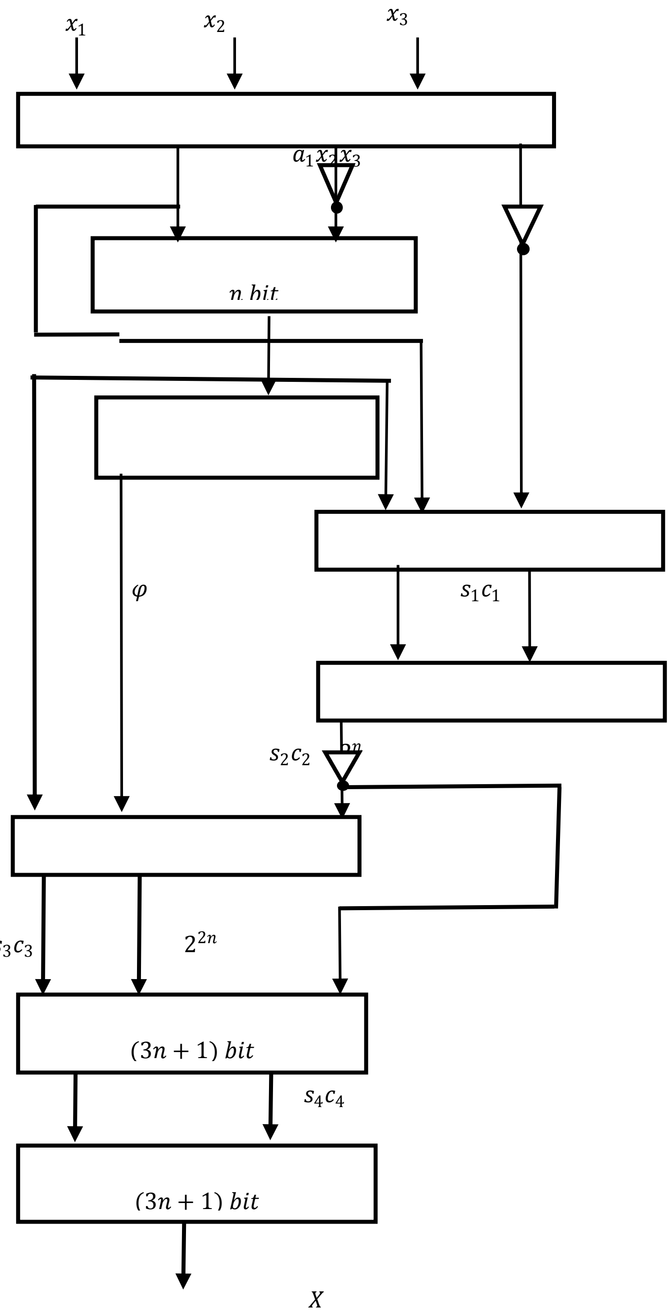

Figure 1: Block Diagram of RC for the non-redundant part

VII. Numerical Results

Let us now consider some numerical illustrations with the proposed scheme.
Consider an (n,k) code where n is the length of the code and k is the dimension of the code with the moduli set $m_1, m_2, m_3, m_4, m_5 = (3,4,5,13,17)$ where m_1, m_2 and m_3 are non-redundant moduli, $m_4,$ *and* m_5 are the redundant moduli. We consider the integer message X=57, for its residue digits are $x_i = (0, 1, 2, 5, 6)$. The legitimate range = M_R= 3*4*5=60 and the illegitimate range = M_K= 13*17 =221. Assume that during storage or computation, an error occurs in the second residue digit such that r_2=3. Therefore, the received codevector will be $\bar{x}_i = (0, \bar{3}, 2, 5, 6)$.

$6687_3 = |6687|_{4420} = 2267$

$6687_4 = |6687|_{3315} = 57*$

$6687_5 = |6687|_{2652} = 1383$

© 2015-19, IJARCS All Rights Reserved

$6687_{13} = |6687|_{1020} = 567$
$6687_{17} = |6687|_{780} = 447$
From the above calculation, it can be observed that there is only one legitimate projection with respect to the second moduli $m_2 = 4$ which falls within the legitimate range and it is the word most likely sent. It can be seen and concluded from the above projections that the second residue (r_2) is in error and hence, anytime it is involved in computation an erroneous output will be executed.
Using the same illustration given above, we now consider detecting and correcting the error using the moduli set.
The result for the iterative processes are shown below;
$r_1, r_2, r_3, r_4 = X_{1234} = 447$
$r_1, r_2, r_3, r_5 = X_{1235} = 747$
$r_1, r_2, r_4, r_5 = X_{1245} = 1383$
$r_1, r_3, r_4, r_5 = X_{1345} = 57*$
$r_2, r_3, r_4, r_5 = X_{1345} = 2272$
From these results, it could be observed that whenever r_2 is involved in the computation it gives an illegitimate value i.e. X_{1234}, X_{1235}, X_{1245} and X_{1345}. When r_2 was discarded in the X_{1345}, the recovered data is 57 which clearly indicates that r_2 is the erroneous digit. The conclusion is that the correct result is 57 and the error, which occurred in, r_2 can be corrected by computing $r_2 = 57 mod 4 = 1$.

VIII. PERFORMANCE EVALUATION

The ability to detect and correct errors in a digital system improves its reliability and integrity. In this work, the MRC decoding technique is compared with the MP. It is found that the MRC decoding processes result in more computational steps in detecting and correcting errors in RRNS by dropping a residue at a time. The MP on the other hand detects and corrects single bit errors without going through more iterative steps in its implementation. A theoretical analysis shows that the MP recovers integer messages faster and improves computational speed than the MRC decoding processes as shown in Figure 2.
To evaluate the performance of the proposed scheme, the work is compared with Yangyang et. al [6]. Theoretically, the proposed scheme has less delay and computational complexity without compromising on accuracy as shown in Figures 3 and 4. Also, the proposed design employs simple adders (CSAs and CPAs) for its implementation. In the scheme of Yangyang et. al [6], twenty (20) latches which has an area of $9n$ and nine (9) ROMs with an area of $4n^2$ were employed and that resulted into a total area requirement of $(40n^2 + 20n)\Delta_{FA}$.

Table 1: Area, Delay Comparison

Scheme	**Area**(Δ_{FA})	**Delay**(D_{FA})
Yangyang et. al [6]	$40n^2 + 20n$	$72n^2 + 40n$
Proposed	$11n + 4$	$10n + 7$

Figures 3 and 4 show the graphical illustrations of the area and delay comparisons.

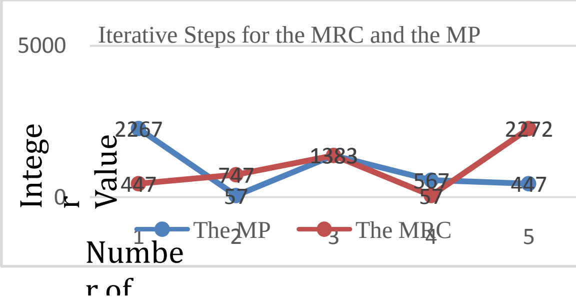

Figure 2: Graph of Comparison between Modulus Projection and MRC Iterative Steps

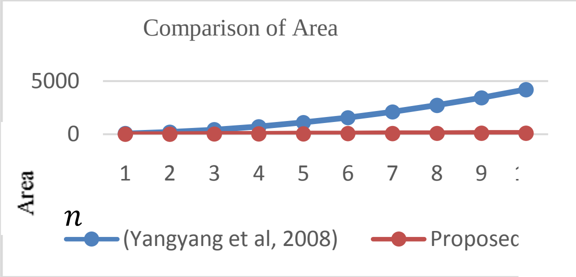

Figure 3: Graph of area comparison of proposed scheme with Yangyang et. al [6] scheme

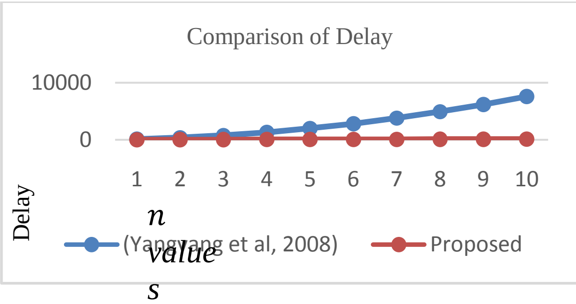

Figure 4: Graph of delay comparison of proposed scheme with Yangyang et. al [6] scheme

IX. CONCLUSION

The need to have efficient and faster algorithm in detecting and correcting errors cannot be ignored in digital systems. The algorithm presented is premised on both the Modulus Projection and the Mixed Radix Conversion. The MP considerably reduced the iterative steps and improved the speed of the architecture. Generally, the proposed scheme was built using simple adders instead of ROMs and latches that are associated with high cost of implementation. An area/delay analyses showed that there is a considerable improvement in the speed by 97% and tends to require about 96% less hardware resources in the proposed technique.

X. REFERENCES

[1] H. L. Garner, (1959), "The residue number system," *IRE Trans. Electronic Computers*, Vol. 8, pp. 140-147.
[2] F. Barsi & P. Maestrini, (1973), "Error Correcting Properties of Redundant Residue Number Systems. *IEEE Transactions of Computers*, Vol. 22, No. 3, pp. 307–315.
[3] A.F Behrouz, (2008). Data Communications and Networking. Fourth Edition. *Error Detection and Correction Computer*, Vol. 23, No. 7, pp. 267-306.
[4] D. M. Mandelbaum, (1972), "Error correction in residue arithmetic," *IEEE Trans. Comput.*, Vol. 21, No. 6, pp. 538-545.

© 2015-19, IJARCS All Rights Reserved

[5] Szabo, N. & Tanaka, R. (1967). *Residue Arithmetic and its Application to Computer Technology*. MC-Graw-Hill, New York.

[6] Yangyang T., Boutillon E., Jégo C., & Jézéquel M. (2008). A new single-error correction scheme based on Self-Diagnosis Residue Number Arithmetic. *Conference on Design and Architectures for Signal and Image Processing (DASIP)* pp. 27–33. doi: 10.1109/DASIP.2010.5706242

[7] W. K. Jenkins, C. Radhakrishnan, & S. Pal (2007), Fault Tolerant Signal Processing for Masking Transient Errors in VLSI Signal Processors," *Proceedings IEEE International Symposium on Circuits and System (ISCAS)*, pp. 2570 – 2573

[8] O. Goldreich, D. Ron & M. Sudan (2000). Chinese Remaindering with Errors", *IEEE Trans. Infor. Theory*, Vol. 46, pp.1330-1338.

[9] M.I. Daabo, & K.A. Gbolagade (2014). An Overflow Detection Scheme with a Reverse Converter for the Moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. *Journal of Emerging Trends in Computing and Information Sciences,* ISSN 2079-8407, Vol. 5, No. 12, pp. 931-935.

[10] A.S. Molahosseini, & K. Navi (2007). New Arithmetic residue to binary Converters. *International Journal of Computer Sciences and Engineering Systems*, Vol. 1, No.4, pp. 295-299.

© 2015-19, IJARCS All Rights Reserved