A NEW SINGLE-ERROR CORRECTION SCHEME BASED ON SELF-DIAGNOSIS RESIDUE NUMBER ARITHMETIC

Yangyang Tang[†], Emmanuel Boutillon[†], Christophe Jégo[‡] and Michel Jézéquel[‡]

[†]Université Bretagne Sud, UMR CNRS 3192 Lab-STICC,
Université Européenne de Bretagne, BP 92116, 56321 Lorient Cedex, France
[‡]Institut TELECOM/TELECOM Bretagne, UMR CNRS 3192 Lab-STICC,
Université Européenne de Bretagne, CS 83818, 29238 Brest Cedex 3, France
[†]{firstname.lastname}@univ-ubs.fr, [‡]{firstname.lastname}@telecom-bretagne.eu

ABSTRACT

With the rapid size shrinking in electronic devices, radiation-induced soft-error has emerged as a major concern to the current circuit manufacturing. In this paper, we present a new error correction scheme based on the residue number arithmetic to cope with the single soft-error issue. The proposed technique called bidirectional redundant residue number system requires the redundant moduli to satisfy some constraints to achieve fast error correction. In this system, both the iterations for decoding the valid number and the error-correcting table that contains all combinations of erroneous digit, are not necessary. The detection and the diagnosis are simultaneously performed in plural parallel consistent-checking that has the capability of locating the corrupt digit. Finally, efficient pipeline architecture for the self-diagnosis decoder is detailed.

Index Terms— Single event transient, fault tolerant, residue number system, self-diagnosis arithmetic, hardware design.

1. INTRODUCTION

Most recently years, due to the rapid development of logic circuit manufacturing the advanced deep submicron or nano-scale electronic devices have been applied in various platforms, especially the one with modern high performance processors. At the same time, the down-scaling technology incurs integrated chips more susceptible to Single Event Transient (SET), called radiation-induced soft-error [1]. In this paper, we turn our attention to Residue Number System (RNS) arithmetic to deal with the single soft-error issue.

The RNS, firstly introduced by Garner [2], provides a very fast arithmetic due to its capability of performing the carry-free operations, i.e. addition, subtraction and multiplication. By adding some redundant residues, the RNS has the error detection and correction property that is called Redundant Residue Number System (RRNS). A method based on RRNS that is able to correct single bit error by encoding the residues was proposed by Cheney [3]. Szabo and Tanaka [4] dedicated their investigations to the properties of the RNS and presented a method for correcting single residue-error correction. The idea of consistency checking was first introduced by Watson and Hastings [5]. They also defined a method that needs a table for single residue error correction. Yau and Liu [6] introduced a method that uses appropriate computations to remove the error-correcting table. Mandelbaum [7] derived that two redundant moduli are necessary for single error correction. Then, Barsi and Maestrini [8] showed the necessary and sufficient condition for the correction of a given error affecting a single residue digit of any legitimate number. They also proposed an RNS product code and further determined that if the error affects an arbitrary legitimate number or a number in overflow, then the single residue can be detected or corrected [9]. Jenkins *et al.* [10-13] proposed residue number error checkers based on Mixed Radix Conversion (MRC) [14] and implemented this method into digital filters due to efficient pipeline architectures. Their method disjoints one residue in each projection to process current error detection and diagnosis. A corrupt number is not actually corrected, but left out a series of m_i weighted projections. Ramachandran [15] showed a particular procedure of modulo projections that discards more than one residue at each time. Su and Lo [16] used the redundant digits of MRC as the entries to build a lookup table. Krishna *et al.* [17-18] bounded the RRNS with coding theory and derived the conditions for the single, double and multiple error corrections. Katti [19] presented a method using a modulo set with common factors that leads to simplified error detection and correction. The former methods detect or correct error using the computational iterations or an error-correcting table for the error detection and the error correction. However, this technique induces considerable hardware or delay overhead.

In this paper, we propose a new error correction scheme based on RRNS without using the computational iterations and the error-correcting table. This technique, called

Bidirectional Residue Redundant Number System (BRRNS), requires redundant moduli that satisfy some constraints to achieve fast error correction. In this system, both the iterations for decoding the valid number and the error-correcting table that contains all combinations of erroneous digit, are not necessary. Actually, the detection and the diagnosis are simultaneously performed in plural parallel consistent-checking to locate the corrupt digit. The error-free number is restored through either the information residue set or the redundant residue set. To deal with the soft-error, BRRNS codes with the capability of single-error correction are presented.

To design an efficient architecture that implemented a BRRNS, two approaches for residue to binary conversion [20], namely Mixed Radix Conversion (MRC) and Chinese Remainder Theorem (CRT), can be applied. The CRT satisfies the real-time signal processing due to its ability of performing the computation in parallel. However, by comparison with the MRC, a modulo operation in large constant limits the major use of the CRT approach. The Base Extension (BEX) based on the CRT is often used for generating additional residue. A fast BEX algorithm [21] reduces latency and hardware resources by using an additional small redundant residue. In our design, for the purpose of less hardware complexity, the MRC is employed in efficient pipeline architectures that can execute a self-diagnosis decoder for single-error correction.

The rest of this paper is organized as follows. Section 2 introduces the fundamentals of residue arithmetic. Section 3 reviews the error detection and error correction properties of conventional RRNS, as well as their requirements to modulo representation. Section 4 presents the proposed algorithm. In section 5, the architecture implemented the proposed algorithm is detailed and then compared with the one based on the conventional RRNS. Finally, a conclusion is drawn in section 6.

2. RESIDUE ARITHMETIC FUNDAMENTALS

A standard RNS [2] is characterized by a set of k pairwise relative prime positive integers, i.e. greatest common divisor $(m_i, m_j)=1$ with $i \neq j$, $m_1, m_2\dots, m_{k-1}, m_k$, called moduli, that is formed in increasing, i.e., $m_1<m_2<\dots<m_{k-1}<m_k$. Their product represents the interval $[0, M)$ called the legitimate range that defines the useful computational range of the number system, that is,

$$M = \prod_{i=1}^{k} m_i \,. \tag{1}$$

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

$$r_i \equiv \left|X\right|_{m_i} \; or \; r_i \equiv X(mod\; m_i) \tag{2}$$

with $i \in [1, k]$, and $\left|X\right|_{m_i}$ 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 moduli independently, i.e.

$$x_1x_2\cdots x_k * y_1y_2\cdots y_k = z_1z_2\cdots z_k,\; z_i \equiv \left|x_i * y_i\right|_{m_i} \tag{3}$$

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

An RRNS [7-8] is defined as an RNS with n-k additional moduli. The moduli, $m_1, m_2\dots, m_{k-1}, m_k$, are called the information moduli. The additional n-k moduli, $m_{k+1}, m_{k+2}\dots, m_{n-1}, m_n$, are called the redundant moduli. Moreover, the product of all the moduli, M_T, is denoted as the total range:

$$M_T = \prod_{i=1}^{n} m_i = MM_R \,,\; M_R = \prod_{i=n-k}^{n} m_i \tag{4}$$

where M_R represents the illegitimate range. Similarly, the residue vector is composed of the information residues $r_1, r_2\dots, r_{k-1}, r_k$, and the redundant residues $r_{k+1}, r_{k+2}\dots, r_{n-1}, r_n$. Any number belonging to the illegitimate range indicates the overflow or error occurrence as explained in the next section.

When the operations were performed in residue vector, a converter of residue to integer or binary (henceforth, termed simply residue/binary, vice versa, binary-residue) is necessary in order to build the number. It is well known that CRT and MRC [4, 14, 20] approaches are often applied for the conversion. According to the CRT, assume an integer X formed in a residue vector r_i and $w_i = M_T/m_i$, then X can be computed by

$$X = \sum_{i=1}^{n} w_i \left|c_i r_i\right|_{m_i} (mod\; M_T)\,. \tag{5}$$

The coefficient c_i is determined by the congruence,

$$c_i w_i \equiv 1 (mod\; m_i)\,. \tag{6}$$

It is often necessary to find the extra residue digits $r_{n+1}\dots, r_{n+p}$, of an integer X for a spare set of relatively prime moduli $m_{n+1}\dots, m_{n+p}$, given the residue vector $r_1, r_2\dots, r_{n-1}, r_n$, relative to a set of relatively prime moduli $m_1, m_2\dots, m_{n-1}, m_n$. This is known as the BEX [21] that is based on the CRT. Actually, X is firstly obtained by (6), and then the extra residue digits can be computed with (2).

Compared with the CRT, the MRC is carried out by a weighted approach. If the same previous example is considered, then the MRC is expressed by the following equations,

$$X = \sum_{i=1}^{n} a_i \prod_{j=1}^{i-1} m_j \tag{7}$$

where $0 \le a_i < m_i$, and $\prod_{j=1}^{0} m_i = 1$.

The digits a_i are called the mixed radix digit. They can be obtained from a computational iteration, that is,

$$a_i \equiv \prod_{j=1}^{i-1}[m_{j,i}^{-1}(r_i - a_j)](mod\ m_i), \quad (8)$$

where $m_{j,i}^{-1}$ is defined as $m_{j,i}^{-1} m_j \equiv |1|_{m_i}$, with $a_1 = r_1$.

In this way, the MRC approach is less complex than the CRT approach due to a serial process. This process can be achieved in pipeline procedure for hardware design, which will be demonstrated in section 5.

3. THE ERROR DETECTION AND CORRECTION

3.1. RRNS Based Fault-Tolerant Model

It is well known that Error-Correcting Codes (ECC) consist of an information part and a redundant part [22]. Having inherited the property of RNS, RRNS codes not only have the capability of performing the fast arithmetic, but also the error detection and correction due to the extra redundant residue. Some research about the combination of RRNS and ECC has been done [23]. Assuming the SET occurs during the computation or in the memories (DRAM, SRAM, etc.), the fault-tolerant models applied the RRNS technique which are showed in Figure 1.

By applying the noisy channel coding theorem developed by Shannon [24], the RRNS codes are composed by the binary/residue conversion as the encoder and the conversion as the decoder. Moreover, the computing function and the memory are such as the noisy channel if they are affected by a SET. For the sake of facility, we note that the residue/binary conversion is assumed as an error-free process in this paper.

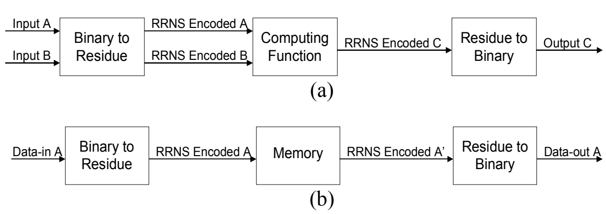

Figure 1. RRNS based fault-tolerant models: (a) Application to a computing function. (b) Application to a memory.

3.2. The error detection

The conventional approach for the error detection and the error correction is performed by the consistent checking or the BEX. The consistent checking was firstly proposed by Watson and Hastings in [5]. It can be defined as: all the residues in the residue vector are correct if the condition

$$0 \le X < M \quad (9)$$

is satisfied. The process used to determine whether or not it holds is called consistent checking.

Theorem 1: The RRNS codes with n-k=1 redundant modulus detect all single errors if the condition $m_1<m_2<\ldots<m_k<m_{k+1}$ is satisfied.

Proof: Suppose a single SET distorts the i^{th} residue digit in the residue vector of an integer X, it is represented as an addition of the number e_i to the residue while $0 < e_i < m_i$, namely,

$$X' \to [r_1, r_2 \ldots, (r_i + e_i) \ldots, r_{n-1}, r_n] \quad (10)$$

if X' is the altered number. According to (3), the X' can be defined as the sum of the X and an integer E formed by the residue vector e_i,

$$X' = X + [0,\ 0\ldots,\ e_i \ldots,\ 0,\ 0],$$
$$[0,\ 0\ldots,\ e_i \ldots,\ 0,\ 0] \equiv E(mod\ m_i). \quad (11)$$

Therefore, with (5) and (7), E can be expressed by

$$E = c \prod_{j=1, j\ne i}^{n} m_j \quad (12)$$

where c is a non-zero multiplicative coefficient. Then, theorem 1 is directly obtained. Since the redundant modulus m_{k+1} is larger than all the information moduli, E is larger than the legitimate range M in (1), namely,

$$\prod_{j=1, j\ne i}^{n} m_j \ge M = \prod_{j=1}^{n-1} m_j, n \ge i.$$

Such that,

$$X' \ge X + c \cdot M. \quad (13)$$

With the expression of the X' (13), the number E leads the decoded integer to appear in the illegitimate range. Hence, the single error is detected.

3.3. The error correction

Theorem 2: The RRNS codes with (n-k) redundant moduli can detect (n-k) errors or correct (n-k)/2 errors.

This theorem has been demonstrated in different manners [6, 8-9, 17-18]. The conventional RRNS codes that perform the error correction can be described in three stages:

STAGE 1: Consistent checking,
Detect all single errors with theorem 1. If no error is detected, then stop here, otherwise, go to the next stage.
STAGE 2: Erroneous digit locating,
Find the location of the erroneous digit or digits. It is generally carried out by two approaches, m_i-projections and based on BEX.
STAGE 3: Recovery,
Remove the corrupt digit or digits, and then restore the integer X from the rest error-free residues.

Example 1: An 8-bit RRNS codeword is generated from the modulo set (m_1, m_2, m_3, m_4, m_5)=(4, 5, 7, 9, 11), where m_4 and m_5 are redundant moduli. The legitimate range is [0, 4×5×7)=[0, 140). The RRNS codeword for input data X=125 is the residue vector (1, 0, 6, 8, 4). Assuming that the third residue of the codeword is corrupted from 6 to **1**

during the storing or the transmission, these results in X=(1, 0, **1**, 8, 4). The following process based on the MRC, that is,

1) consistent checking, discarding r_5, the recovered data is 1205 decoded by (r_1, r_2, r_3, r_4) and an error is detected.
2) erroneous digit is located, and iterations are executed,
3) m_i-projections are applied. Only one decoded integer is smaller than 140, as shown in the table below, so the correct integer X=125 is restored.

Iteration	1	2	3
Discarded Residue	r_1	r_2	r_3
Recovered Data	3095	1709	125

Clearly, if the erroneous digit is r_4 or r_5, then the integer X will not be able to be recovered until the discarding the residue r_4 or r_5. Therefore, the m_i-projections need n cycles to obtain the error-free integer at the worst case.

4. THE PROPOSED ALGORITHM

The proposed technique, bidirectional redundant residue number system requires redundant moduli that satisfy some constraints to achieve fast error correction with minimum hardware overhead. The BRRNS is characterized by n pairwise relative prime positive moduli, $m_1, m_2\ldots, m_{n\text{-}1}, m_n$, that are formed in increasing, i.e., $m_1<m_2<\ldots<m_{n\text{-}1}<m_n$. The first k moduli form a set of information moduli, and the remaining n-k moduli are redundant moduli. BRRNS has the same structure as RRNS, but BRRNS requires that the redundant moduli satisfy the following conditions,

$$M=\prod_{i=1}^{k} m_i \approx M_R=\prod_{j=k+1}^{p} m_j,\ p\le n, \quad (14)$$

$$m_n < m_1\cdot m_2 \quad (15)$$

where the legitimate range of BRRNS is $Min(M, M_R)$ and any redundant modulus is smaller than the product of the two minimum redundant values as (15). As the product of certain redundant moduli is approximately equivalent to the product of the information moduli, any integer X belonging to the legitimate range can be also restored through the redundant residue vector, namely,

$$X \to [r_{k+1}+r_{k+2}\ldots+r_{p-1}+r_p],\ p\le n. \quad (16)$$

Here, we define this property as alternative recovery.

Theorem 3: The BRRNS codes with n-k=2 redundant moduli is able to correct all single errors if the condition $m_1<m_2<\ldots<m_k<m_{k+1}<m_{k+2}$ is satisfied.

By applying the theorem 2, thus, this theorem has no need to be proved. However, the plural consistent-checking and the alternative recovery make the error-correction in the BRRNS codes faster than in the conventional RRNS codes.

With one redundant modulus are able to detect whether or not the single error occurs. Moreover, double consistent-checking can be done with two redundant moduli. The set of results obtained from double consistent-checking includes: one non-consistency and one consistency, both non-consistency and both consistency. The occurrence of single error is expected in two cases; the distortion of an information residue and the distortion of a redundant residue. In other words, the corrupt digit is either an information residue or a redundant residue and the spare digits are error-free. Consequently, with the property of alternative recovery, any integer X belonging to the legitimate range can be either restored from the information residue vector or from the redundant residue vector.

The BRRNS codes perform the error correction in two phases:

PHASE 1: Plural consistent-checking,
All single errors can be detected and located simultaneously.
PHASE 2: Rehabilitation,
The integer X belonging to the legitimate range is restored through the redundant residue vector or the information residue vector with the determination of phase 1.

Example 2: A 8-bit BRRNS codeword is generated from the modulo set (m_1, m_2, m_3, m_4, m_5)=(4, 5, 7, 11, 13). The legitimate range is [0, 4×5×7)=[0, 140), according to $Min(M, M_R)$. If an integer decimal message of X=125 is considered, then the corresponding residue values are X=(1, 0, 6, 4, 8). Assuming an error occurs in r_3, then the received BRRNS codeword becomes (1, 0, 1, 4, 8),

1) Plural consistent-checking,
   r_1, r_2, r_3 and r_4 are involved in the first consistent checking, then, a_4=5≠0, Erroneous.
   r_1, r_2, r_3 and r_5 are involved in the second consistent checking, then, a_5=7≠0, Erroneous.
   Both consistent-checking are faulty, hence, the corrupt digits are in the information residue vector.

Serial Process	1	2	3	4	5
Radix	a_1	a_2	a_3	a_4	a_5
Value	1	1	4	5	7

2) Rehabilitation,
   the integer X is restored by r_4 and r_5, the redundant residue set.

$$a_4^{'}=r_4,$$

$$a_5^{'}=(r_5-a_4^{'})\cdot m_{4,5}^{-1}(mod\ m_5).$$

	a_4’	a_5’	X
Value	4	11	125

5. THE ARCHITECTURES

5.1. Conventional RRNS error checker

In the literature [10-13], Jenkins and his colleagues proposed residue number error checkers based on the RRNS and implemented this method into digital filters with efficient pipeline architecture. For instance, the conventional RRNS code with $k=3$ and $n=5$, the mixed radix digits a_i for each of n projections required for error decoding are computed by the expressions of MRC showed in (8), namely,

$$\begin{aligned} a_1 &= r_1 , \\ a_2 &= (r_2 - a_1) \cdot m_{1,2}^{-1} (mod\ m_2), \\ a_3 &= ((r_3 - a_1) \cdot m_{1,3}^{-1} - a_2) \cdot m_{2,3}^{-1} (mod\ m_3), \\ a_4 &= (((r_4 - a_1) \cdot m_{1,4}^{-1} - a_2) \cdot m_{2,4}^{-1} - a_3) \cdot m_{3,4}^{-1} (mod\ m_4), \\ a_5 &= ((((r_5 - a_1) \cdot m_{1,5}^{-1} - a_2) \cdot m_{2,5}^{-1} - a_3) \cdot m_{3,5}^{-1} - a_4) \cdot m_{4,5}^{-1} (mod\ m_5) \end{aligned} \tag{17}$$

These computations are performed in series, thus, a pipelined architecture for the implementation of the mixed radix converter is mostly applied in digital filters, as shown in Figure 2. For the sake of facility, the Look-up tables formed as ROM take place of the computations for the radixes conversions. Thus, the residues or the former radixes are read as the address to fetch the radix. In the case of five-moduli, the pipeline spends five cycles until a_5 fan-outed, which is one cycle of m_i-projections. In practice, one of the five residues will be dumped in each iteration, in which has the capability of performing the consistent-checking (see also section 2).

However, the functionality of the consistent-checking performed during the iterations is error detection, but not the decoding. Actually, once single soft-error occurred, the m_i-projections requires n iterations at the worst case for obtaining the error-free number.

For this reason, it mostly impacts the real-time decoding achieved at a low latency. In the following subsection, the self-diagnosis decoder based on the BRRNS for single-error is presented.

5.2. BRRNS self-diagnosis decoder

If the same previous instance is considered, a BRRNS code with $k=3$ and $n=5$ has a different structure for the computations of mixed radix digits a_i.

$$\begin{aligned} a_1 &= r_1 , \\ a_2 &= (r_2 - a_1) \cdot m_{1,2}^{-1} (mod\ m_2), \\ a_3 &= ((r_3 - a_1) \cdot m_{1,3}^{-1} - a_2) \cdot m_{2,3}^{-1} (mod\ m_3), \\ a_4 &= (((r_4 - a_1) \cdot m_{1,4}^{-1} - a_2) \cdot m_{2,4}^{-1} - a_3) \cdot m_{3,4}^{-1} (mod\ m_4), \\ a_5' &= (((r_5 - a_1) \cdot m_{1,5}^{-1} - a_2) \cdot m_{2,5}^{-1} - a_3) \cdot m_{3,5}^{-1} (mod\ m_5). \end{aligned} \tag{18}$$

As explained in section 4, the proposed method performs the plural consistent-checking in parallel, that is, the consistent-checking by observing a_4 and the one by observing a_5' are processed during the same cycle. Therefore, here, a pipeline architecture based on this method for single-error correction can be achieved as shown in Figure 3.

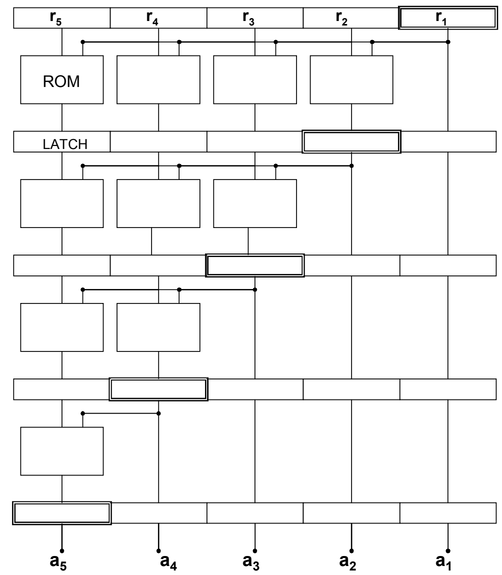

Figure 2. Error checker based on the MRC for a systematic RRNS code with k=3 and n=5.

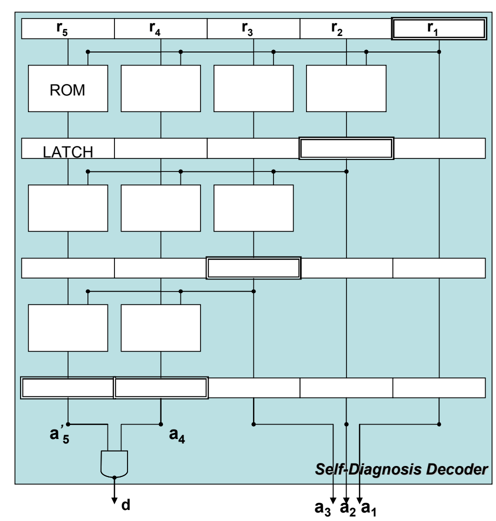

Figure 3. Self-Diagnosis decoder for single-error correction based on the MRC for a systematic BRRNS code with k=3 and n=5.

During the first three cycles, a_1, a_2, and a_3 values are computed respectively. Afterward a_4 and a_5' values are obtained at the fourth cycle with equation (18). To determine the set for the data recovery, a "AND" gate is needed to observe the value of a_4 and a_5'. Moreover, "signal'*d*'=*0*" indicates that the information residues, a_1, a_2, and a_3, are error-free, otherwise, a_4 and a_5' are error-free while "*d*=1". Subsequently, after the self-diagnosis decoder, the addition based on either a_1, a_2 and a_3, or a_4 and a_5' values is required for completing the rehabilitation of the integer X.

6. CONCLUSION

In this paper, we have presented a new single-error correction method based on residue number arithmetic. Firstly, the former algorithms applied the RNS or RRNS to achieve error correction was reviewed. Then, we demonstrated RRNS based single-error detection and single-error correction. The major contribution of this paper, that is, a new method based on residue number arithmetic called Bidirectional Redundant Number System and its proofs relevant to single-error detection and correction were presented in detail. Afterward, the pipeline hardware architecture implemented the BRRNS was demonstrated for the single-error correction.

By using the BRRNS, the fast error correction can be achieved. Moreover, the iterations for decoding the valid number and the error-correcting table that contains all combinations of erroneous digits, are not needed anymore. In the BRRNS, the plural parallel consistent-checking performs the error detection and the error diagnosis. With the determination made in the self-diagnosis decoder, any integer X belonging to the legitimate range is able to be restored either by the carry-free information radixes or the carry-free redundant residues set.

For the future directions, we will turn our attention to extend BRRNS into the double-error correction issue, and also the applications of BRRNS code, as well as their performances.

7. APPENDIX

Corollary: The error is detected if $a_j \neq 0$ with $j \in [k+1, n]$.
Proof: Suppose a single SET distorts the i^{th} residue digit in the residue vector of an integer X. It is represented as an addition of the number e_i to the residue while $0 < e_i < m_i$, namely,

$$X' \rightarrow [r_1, r_2 ..., (r_i + e_i) ..., r_{n-1}, r_n]$$

where X' is the altered number. According to (3), the X' can be defined as the sum of the X and an integer E formed by the residue vector e_i,

$$X' = X + [0, 0..., e_i ..., 0, 0],$$

$$[0, 0..., e_i ..., 0, 0] \equiv E(mod\ m_i).$$

Therefore,

$$E = \sum_{j=1, j\neq i}^{n} a_j \prod_{l=1}^{j-1} m_l$$

such that

$$X' = X + \sum_{j=1, j\neq i}^{n} a_j \prod_{l=1}^{j-1} m_l$$

Since the redundant modulus m_{k+1} is larger than all the information moduli, E is larger than the legitimate range M in (1). The only case that the decoded integer appears in the legitimate range is $a_j=0$, where $j \in [k+1, n]$. Consequently, $a_j \neq 0$ indicates the residue digit affection, otherwise, no error occurs.

8. REFERENCE

[1] J. M. Benedetto, P. H. Eaton, D. G. Mavis, M. Gadlage, and T. Turflinger, "Digital single event transient trends with technology node scaling," *IEEE Trans. Nuclear Science*, vol. 53, no. 6, pp. 234-240, 2006.

[2] H. L. Garner, "The residue number system," *IRE Trans. Electronic Computers*, Vol. EC-8, pp. 140-147, 1959.

[3] P. W. Cheney, *An investigation of residue number theory for digital systems*, Ph. D. dissertation, Stanford Univ., Calif., 1962.

[4] N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and its Applications to computer Technology*, New York: McGraw-Hill, 1967.

[5] R. W. Watson and C. W. Hastings, "Self-Checked computation using residue arithmetic," *Proc. IEEE*, vol. 54, no.12, pp. 1920-1931, 1966.

[6] S. S. Yau and Y. Liu, "Error correction in redundant residue number systems," *IEEE Trans. Computers*, vol. c-22, no. 1, pp. 5-11, 1973.

[7] D. Mandelbaum, "Error correction in residue arithmetic," *IEEE Trans. Computers*, vol. c-21, no. 6, pp. 538-545, 1972.

[8] F. Barsi and P. Maestrini, "Error correcting properties of redundant residue number systems," *IEEE Trans. Computers*, vol. c-22, no. 3, pp. 307-315, 1973.

[9] -------, "Error detection and correction by product codes in residue number systems," *IEEE Trans. Computers*, vol. c-23, no. 9, pp. 915-924, 1974.

[10] M. H. Etzel and W. K. Jenkins, "Redundant residue number systems for error detection and correction in digital filters," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. asp-28, no. 5, pp. 538-544, 1980.

[11] W. K. Jenkins, "The design of error checkers for self-checking residue number arithmetic," *IEEE Trans. Computers*, vol. c-32, no. 4, pp. 388-396, 1983.

[12] M. J. Bell, Jr. and W. K. Jenkins, "A residue to mixed radix converter and erro checker for a five-moduli residue number system," *Proc. IEEE ICASSP'84*, vol. 3, no. 1, pp. 242-245, 1984.

[13] W. K. Jenkins and E. J. Altman, "Self-checking properties of residue number error checkers based on mixed radix conversion," *IEEE Trans. Computers*, vol. 35, no. 2, pp. 159-167, 1988.

[14] F. J. Taylor, "Residue arithmetic: A tutorial with examples," *IEEE Trans. Computers*, vol. 17, no. 5, pp. 50-62, 1984.

[15] V. Ramachandran, "Single residue error correction in residue number systems," *IEEE Trans. Computers*, vol. c-32, no.5, pp. 504-507, 1983.

[16] C. Su and H. Lo, "An algorithm for scaling and single residue error correction in residue number systems," *IEEE Trans. Computers*, vol. 39, no. 8, pp. 1053-1064, 1990.

[17] H. Krishna, K. Lin and J. Sun, "A coding theory approach to error control in redundant residue number systems part I: Theory and single error correction," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 1, pp. 8-17, 1992.

[18] J. Sun and H. Krishna, "A coding theory approach to error control in redundant residue number systems part II: Multiple error detection and correction," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 1, pp. 18-34, 1992.

[19] R. S. Katti, "A new residue arithmetic error correction scheme," *IEEE. Trans. Computers*, vol. 45, no. 1, pp. 13-19, 1996.

[20] A. Omandi and B. Premkumar, *Residue Number Systems: Theory and Implementation*, World Scientific Book, 2007.

[21] A. P. Shenoy and R. Kumaresan, "Fast base extension using a redundant modulus in RNS," *IEEE Trans. Computers*, vol. 38, no. 2, pp. 292-296, 1989.

[22] R.J. McEliece, *The Theory of Information and Coding*, Cambridge Univ. Press, 2002.

[23] L. Yang and Lajos Hanzo, "Redundant residue number system based error correction codes," *Proc. IEEE Vehicular Technology*, pp. 1472-1476, 2001.

[24] C. E. Shannon, "A mathematical theory of communication," *Jour. Bell Systems Technology*, vol. 27, pp. 379-423, 623-656, 1948.