*Wireless Engineering and Technology*, 2013, 4, 198-213
http://dx.doi.org/10.4236/wet.2013.44029 Published Online October 2013 (http://www.scirp.org/journal/wet)

# Parallel Algorithms for Residue Scaling and Error Correction in Residue Arithmetic

**Hao-Yung Lo, Ting-Wei Lin**

Department of Electrical Engineering, National Tsing Hua University, Hsinchu City, Chinese Taipei.
Email: hylo@ee.nthu.edu.tw

Received August 3rd, 2013; revised September 11th, 2013; accepted September 30th, 2013



## ABSTRACT

In this paper, we present two new algorithms in residue number systems for scaling and error correction. The first algorithm is the Cyclic Property of Residue-Digit Difference (CPRDD). It is used to speed up the residue multiple error correction due to its parallel processes. The second is called the Target Race Distance (TRD). It is used to speed up residue scaling. Both of these two algorithms are used without the need for Mixed Radix Conversion (MRC) or Chinese Residue Theorem (CRT) techniques, which are time consuming and require hardware complexity. Furthermore, the residue scaling can be performed in parallel for any combination of moduli set members without using lookup tables.

**Keywords:** Chinese Remainder Theorem (CRT); Error Correction; Error Detection; Parallel Residue Scaling; Residue Number Systems (RNS); Target Race Distance (TRD); Target Residue-Digit Difference

## 1. Introduction

Because the residue number system (RNS) operations on each residue digit are independent and carry free property of addition between digits, they can be used in high-speed computations such as addition, subtraction and multiplication. To increase the reliability of these operations, a number of redundant moduli were added to the original RNS moduli [RRNS]. This will also allow the RNS system the capability of error detection and correction. The earliest works on error detection and correction were reported by several authors [1-12]. Waston and Hasting [1,2] proposed the single residue digit error correction. Yau and Liu [3] suggested a modification with the table lookups using the method above. Mandelbaum [4-6] proposed correction of the AN code. Ramachandran [7] proposed single residue error correction. Lenkins and Altman [8-10] applied the concept of modulus projection to design an error checker. Etzel and Jenkins [11] used RRNS for error detection and correction in digital filters. In [12-16] an algorithm for scaling and a residue digital error correction based on mixed radix conversion (MRC) was proposed. Recently Katti [17] has presented a residue arithmetic error correction scheme using a moduli set with common factors, *i.e.* the moduli in a RNS need not have a pairwise relative prime.

In this study, we developed two new algorithms without using MRD (Mixed-radix digit) or CRT (Chinese remained Theorem) for speeding-up the scaling processes and simplifying the error detection and correction in RNS. The first algorithm is used for these purposes, through the residue digit difference cyclic property (CPRDD) within the range of $0 \le x \le M_t - 1$, where $M_t = m_1 m_2 \cdots m_n m_{n+1} \cdots m_{n+r}$ with $r$ additional moduli. The moduli $\{m_1, m_2, \cdots, m_n\}$ are called the nonredundant moduli; $\{m_{n+1}, m_{n+2}, \cdots, m_{n+r}\}$ are the redundant moduli. The interval, $[0, M-1]$, is called the legitimate range, where $M = \Pi_{i=1}^{n} m_i$, and the interval, $[M, M_t - 1]$, is the illegitimate range, where $M_t = MM_r = M \cdot \Pi_{i=1}^{r} m_{n+i}$, and $M_t$ is the total range. This paper is organized as follows: Section II will describe the scheme the cyclic property of residue digit difference (CPRDD). Section III describes the Target Race Distance (TRD) algorithm and followed by some examples. Section IV discusses residue scaling and error correction using the TRD and CPRDD algorithms. Finally, the conclusion is given in section V.

## 2. Error Detection and Correction Using Residue Digit Difference Cyclic Property

Any residue digit $x_i$ representation in moduli set $(m_1, m_2, \cdots, m_n)$ has its cyclic length with respect to its module number. For example, if the moduli set is (4, 5, 7, 9), then the cyclic lengths of any residue digits $(x_1, x_2, x_3, x_4)$ are 4, 5, 7 and 9, respectively. Since these cyclic lengths are not equal, they are very difficult to use as tools for error detection and correction. Actually, there exists the property of common (uniform) cyclic length in RNS between residue digital-differences (RDD). Consider three moduli set $(m_1, m_2, m_3) = (2, 3, 5)$. The residue representations and their corresponding digit-differences are shown in **Table 1** and defined as the difference in value between two digits, $d_{ij} = \langle x_i - x_j \rangle_{m_i}$, where $d'_{ij}$ s are all modulo to positive values with respect to $m_j$ if the cycle length of $m_j$ is assigned.

Note that the residue digit-differences $\langle d_{ij} \rangle m_i$ in **Table 1** are obtained from $\langle x_i - x_j \rangle_{m_i}$ if $m_i < m_j$, and from $\langle x_j - x_k \rangle_{m_k}$ if $m_j < m_k$. This difference of $(x_i - x_j)$ or $(x_j - x_k)$ in values may be positive or negative, depending upon $x_i \ge x_j$ or $m_j \ge m_k$ and $x_i < x_j$ or $x_j < x_k$, respectively

All negative values must be modulo to positive values. For example, on starred row 28, as shown in **Table 1**, the digit difference in value for $x_1 = 0$ and $x_3 = 3$ is $d_{13} = 0 - 3 = -3$. It results in $d_{13} = \langle -3 \rangle_2 = 1$

From the cyclic property of residue-digit difference (CPRDD) in RNS, we now have the following theorem.

Theorem 1. For a moduli set $(m_1, m_2, \cdots, m_i, \cdots, m_j, \cdots, m_n)$ and residue representation for $x = (x_1, x_2, \cdots, x_i, x_j, \cdots, x_n)$ in RNS, there exists a cyclic property in differences between two residue digits, $d_{ij} = \langle x_i - x_j \rangle_{m_i}$ or $\langle x_i - x_j \rangle_{m_j}$. The cyclic length can be assigned, either to $m_i$ or $m_j$, depending upon modulo operation with respect to $m_i$ or $m_j$.

Proof: Consider the case respective to $m_j$, the residue-digit difference (RDD) between two digits in $X = (x_1, x_2, \cdots, x_i, x_j, \cdots, x_n)$ can be in general expressed by the equation

$$\langle d_{ij} \rangle = \langle x_i - x_j \rangle_{m_i} = \langle \langle x_i + pm_i \rangle - \langle x_j + qm_j \rangle \rangle_{m_i} \quad (2\text{-}1)$$

where $p = 0, 1, \cdots, (m_j - 1)$

$q = 0, 1, \cdots, (m_i - 1)$

and $i, j, p, q$ are integers.

For simplicity, we only consider the case of $m_i < m_j$ and assume $m_j - m_i = r$, and the case of $m_i > m_j$ can be obtained in a similar way.

The related theorem and algorithm are described as follows.

1) In cycle 0, (the initial cycle), we have $X = x_j = 0, 1, \cdots, (m_j - 1)$ with $q = 0$, $\langle d_{ij} \rangle = 0 = \langle x_i - x_j \rangle_{m_i} \stackrel{?}{=} \langle x_i + p\, m_i - x_j \rangle_{m_i}$ As $x_j = x_i + p$

**Table 1. Cyclic property of Residue Digit Difference.**

| Decimal | $m_1 = 2$ $x_1$ | $m_2 = 3$ $x_2$ | $m_3 = 5$ $x_3$ | $\langle d_{13} \rangle_2$ | $\langle d_{23} \rangle_3$ | $m_3$cycle |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 1 | 1 | 1 | 0 | 0 | |
| 2 | 0 | 2 | 2 | 0 | 0 | 0 |
| 3 | 1 | 0 | 3 | 0 | 0 | |
| 4 | 0 | 1 | 4 | 0 | 0 | |
| 5 | 1 | 2 | 0 | 1 | 2 | |
| 6 | 0 | 0 | 1 | 1 | 2 | |
| 7 | 1 | 1 | 2 | 1 | 2 | 1 |
| 8 | 0 | 2 | 3 | 1 | 2 | |
| 9 | 1 | 0 | 4 | 1 | 2 | |
| 10 | 0 | 1 | 0 | 0 | 1 | |
| 11 | 1 | 2 | 1 | 0 | 1 | |
| 12 | 0 | 0 | 2 | 0 | 1 | 2 |
| 13 | 1 | 1 | 3 | 0 | 1 | |
| 14 | 0 | 2 | 4 | 0 | 1 | |
| 15 | 1 | 0 | 0 | 1 | 0 | |
| 16 | 0 | 1 | 1 | 1 | 0 | |
| 17 | 1 | 2 | 2 | 1 | 0 | 3 |
| 18 | 0 | 0 | 3 | 1 | 0 | |
| 19 | 1 | 1 | 4 | 1 | 0 | |
| 20 | 0 | 2 | 0 | 0 | 2 | |
| 21 | 1 | 0 | 1 | 0 | 2 | |
| 22 | 0 | 1 | 2 | 0 | 2 | 4 |
| 23 | 1 | 2 | 3 | 0 | 2 | |
| 24 | 0 | 0 | 4 | 0 | 2 | |
| 25 | 1 | 1 | 0 | 1 | 1 | |
| 26 | 0 | 2 | 1 | 1 | 1 | |
| 27 | 1 | 0 | 2 | 1 | 1 | 5 |
| 28* | 0 | 1 | 3 | 1 | 1 | |
| 29 | 1 | 2 | 4 | 1 | 1 | |
| 30 | 0 | 0 | 0 | 0 | 0 | |
| 31 | 1 | 1 | 1 | 0 | 0 | Out of Range |
| 32 | 0 | 2 | 2 | 0 | 0 | |

$m_i$ with $p = 0, 1, \cdots, \lfloor m_j / m_i \rfloor$, we have $\langle d_{ij} \rangle_{m_i} = 0$ with $m_j$'s 0s in cycle 0, where $\lfloor x \rfloor$ means the largest integer less than or equal to x.

Thus, the RDD has $m_j$'s "0" in the initial cycle for each modulus, *i.e.*, in cycle 0, $\langle d_{ij}\rangle_{m_i} = (0,0,\cdots,0)$ for all $i \neq j$.

2) Next consider each modulus $m_i$,

Since $x_i = X - pm_i$ and $x_j = X - qm_j$, then

$\langle d_{ij}\rangle = \langle x_i - x_j\rangle_{m_i} = \langle\langle x_i + pm_i\rangle - \langle x_j + qm_j\rangle\rangle_{m_i}$ where

$x_i = 0,1,2,\cdots,(m_i-1),0,1,\cdots,r-1$

$x_j = 0,1,2,\cdots,(m_i-1),m_i,m_i+1,\cdots,m_i+r-1 = (m_j-1)$

For RDD = 0 (for cycles $0, m_i, 2m_i,\cdots,\lfloor m_j/m_i\rfloor m_i$) then

$$\langle d_{ij}\rangle_{m_i} = \left((0-0),(1-1),\cdots,\left[(m_i-1)-(m_i-1)\right],\langle m_i-0\rangle_{m_i},\langle(m_i+1)-1\rangle_{m_i},\cdots,\langle m_i+r-1-(r-1)\rangle_{m_i}\right) = (0,0,\cdots,0)$$

with $m_j$'s 0 s.

For RDD = 1 (not necessary in cycle 1),

$$\langle d_{ij}\rangle_{m_i} = \left((1-0),(2-1),\cdots,\left[m_i-(m_i-1)\right],\langle(m_i+1)-m_i\rangle_{m_i},\cdots\right) = (1,1,\cdots,1) \text{ with } m_j\text{'s 1 s.}$$

…

For RDD = $m_i - 1$

$$\langle d_{ij}\rangle_{m_i} = \left(((m_i-1)-0),(m_i-1),\langle(m_i+1)-2\rangle_{m_i},\cdots\right) = \left((m_i-1),(m_i-1),\cdots,(m_i-1)\right)$$

with $m_j$'s $(m_i-1)$ s.

Corollary 1. From the above theorem, we can immediately obtain that each cycle in the residue-digit difference of *x* will start at location 0, and end at location $(m_i m_j m_k - 1) = M_p - 1$.

Corollary 2. It is easily shown that there exists $m_i$ number of cycles with respect to the cyclic length of $M_p$.

Proof. Since the residue-digit difference of $x = (x_1,x_2,\cdots,x_i,x_j,\cdots,x_n)$ representation is pair-wise, the legitimate range of this pair-wise RDD$(x_i,x_j)$ is $m_i m_j$, (from 0 through $m_i m_j - 1$). From corollary 1, the cyclic length is $m_i m_j$. Thus the number of cycles within this cyclic length for $N_i$ is $N_i = \frac{m_i m_j}{m_i} = m_j$, and for $m_j, N_j = \frac{m_i m_j}{m_j} = m_i$.

Theorem 2. The algorithm of theorem 1 and its corollaries can be extended to two or more pair-wise residue-digit differences.

Proof: consider a three moduli set, we have two pair-wise moduli sets, whose RDD (Residue Digital Difference) is

$$\langle d_{jk}\rangle_{m_k} = \left\langle\langle X\rangle_{m_j} - \langle X\rangle_{m_k}\right\rangle_{m_k} = \left\langle\langle x_j + qm_j\rangle_{m_j} - \langle x_k + sm_k\rangle_{m_k}\right\rangle_{m_k}$$

where $m_k$ is again the referenced module.

Follow the same procedure as step (2) as above.

Assume $m_k = m_j + r_2$, and also pair-wise numbers $x_j = 0,1,2,\cdots,(m_j-2)$, $(m_j-1),0,1,\cdots,(r_2-1)$ and

$$x_k = 0,1,2,\cdots,(m_j-2),(m_j-1),m_j,(m_j+1),\cdots,(m_j+r_2-1) = (m_k-1).$$

1) For $q = s = 0, x_j = x_k$, $0 \le x_j, x_k \le m_k - 1, d_{jk} = 0$ thus

$$\langle d_{jk}\rangle_{m_j} = \underbrace{0,0,\cdots,0}_{},\underbrace{\langle -m_j\rangle_{m_j},\langle -m_j\rangle_{m_j},\cdots,\langle -m_j\rangle_{m_j}}_{},$$

$m_j$'s "0" $r_2$'s "$\langle -m_j\rangle_{m_j}$" $(=0)$

This shows that $\langle d_{jk}\rangle_{m_j}$ has also $m_k$ "0"s in cycle 0 of $m_k\ (x_k = 0,\ 1,\ 2,\cdots,m_k-1)$. The cyclic length is $(m_j)\cdot(m_k) = +r_2$, and the number of cycles for $m_j$ is $m_k\left(= m_j + r_2 = m_j + m_k - m_j \text{ or } = m_j \cdot m_k/m_j\right)$.

2) For $q \neq s \neq 0$ and $\langle d_{jk}\rangle_{m_j} = h$ (a constant for any RDD), if $x_j \neq x_k$

$$\langle d_{jk}\rangle_{m_j} = \underbrace{\langle(h-0),\ ((h+1)-1),\cdots,< 0-(m_j-h) >_{m_j},< 1-(m_j-h+1)\rangle_{m_j}}_{m_k\text{'s"}h\text{"}},\cdots$$

This shows that the $\langle d_{jk}\rangle_{m_j} = h$ in any location has also $m_k$'s "*h*" in cycle i of $m_k$. The number of cycles for $m_j$ is still $m_k\left(= m_j \cdot m_k/m_j\right)$. Combining these three moduli $(m_i,m_j,m_k)$ into one set, we have cyclic length $M_p = m_i \cdot m_j \cdot m_k$ (for example, $m_1 \cdot m_2 \cdot m_3 = 2 \cdot 3 \cdot 5 = 30$). The number of cycles for $m_1, m_2, m_3$ are $N_{m_1} = m_2 \cdot m_3 = 3 \cdot 5 = 15$, $N_{m_2} = m_1 \cdot m_3 = 2 \cdot 5 = 10$, and $N_{m_3} = m_1 \cdot m_2 = 2 \cdot 3 = 6$, respectively. As shown in **Table 1**, the RDD pairs of $\underbrace{\left(\langle d_{13}\rangle_2, \langle d_{23}\rangle_3\right) \text{ are } (0,0),(1,2),(0,1),(1,0),(0,2)}_{\text{All 5 } (=m_3) \text{ pairs in each } m_3 \text{ cycle.}}$, and (1, 1)

In general, $M_p = m_1 \cdot m_2 \cdots \cdot m_k \cdots \cdot m_n$ and $N_{m_k} = M_p / m_k$ with $m_k$ rows and $(n-1)$RDD in each row.

This completes the proof.

Example 2-1.

Consider a moduli set $(m_1, m_2, m_3) = (4, 5, 7)$, $X = 9$ and its corresponding residue digits representation set is $(1, 4, 2)$. The cyclic length is $140 (= 4 \cdot 5 \cdot 7)$ and the number of cycles for $m_1, m_2$, and $m_3$ are $N_{m_1} = 35, N_{m_2} = 28$, and $N_{m_3} = 20$, respectively.

Error detection and correction:

Before the CPRDD algorithm used for error detection and correction is described, some basic terms in use must be defined.

Definition 1: Stride distance $S_{ij}$: It is the incremental or decremental distance between moduli $m_i$ and $m_j$ in absolute value from *i*th cycle to $(i+1)$th cycle.

For example: $S_{23} = |5 - 7| = 2$.

(1) Error detection

Let the moduli set be $\{m_1, m_2, \cdots, m_k, m_{k+1}, \cdots, m_{k+r}, \cdots\}$ where $m_1, m_2, \cdots, m_k$ are the nonredundant moduli and $m_{k+1}, \cdots, m_{k+r}$ are the redundant moduli. Since the cyclic lengths of CPRDD $d_{ij}$'s are constant, it is thus easily found that the number of cycles on track $L_{ij}$ from the starting point 0 (or other $d_{ij}$) to its target position. In turn the distance of RDD's can also be found.

**Theorem 3.** The number of cycles on track $L_{ij}$ (column $d_{ij}$) from any starting point (say $\hat{d}_{ij}$) to its target position $d_{ij}$ can be found using the equation below;

$$\left\langle \hat{d}_{ij} + S_{ij} k_{ij} \right\rangle_{m_i} = d_{ij}$$

where $S_{ij} =$ the stride distance between moduli $m_i$ and $m_j$ and k = the number of cycles passing through from starting point $\hat{d}_{ij}$ to the destination, $d_{ij} =$ on track $L_{ij}$ If $\hat{d}_{ij} = 0$, then the number of cycles are equal to the total cycles from the starting point "0" to its target position $d_{ij}$.

Proof: Since $k_{ij}$ is the number of cycles from 0 to $d_{ij}$ with respect to module $m_j$, and $m_j$ is the cyclic length, thus $k_{ij} m_j$ is the total distance from the starting point $\hat{d}_{ij} = 0$ to its target position $d_{ij}$. The remaining distance for $d_{ij}$ on track $L_{ij}$ in the $(k_{ij})$th cycle must be on the same row of $d_{ij}$ on track $L_{ij}$. Thus, $\mathrm{RDD}(x_i) = \mathrm{RDD}(x_j) = (k_{ij} m_j + d_{ij})$.

Once the RDD's of $x_1, x_2, \cdots, x_n, x_r$ are found, the error detection and correction for moduli can be found just by comparing the calculated cycles or RDD with the original residue representation, pair-wise so that the error module can be detected.

The procedure for error detection by using CPRDD algorithm is summarized as follows.

1) Choose two most significant (largest) moduli as the referred moduli among the *n* moduli, say $m_{n-1}$ and $m_n$.

2) Find the skip distance of a cycle $S_{(n-1)(n)} = |m_{n-1} - m_n|$.

3) Find the digit difference $d_{(n-1)(n)} = \langle x_{n-1} - x_n \rangle_{mn}$ from $X = (x_1, x_2, \cdots, x_{n-1}, x_n)$.

4) Create the equation of $\mathrm{RDD}(x_{n-1}, x_n) = \langle d(x_{n-1}, x_n) \rangle_{m_n}$ or

$$\mathrm{RDD}(x_{n-1}, x_n) = \left\langle S_{(n-1)(n)} \cdot k_{(n-1)(n)} \right\rangle_{m_n} = \langle d(x_{n-1}, x_n) \rangle_{m_n} \tag{2-2}$$

5) Solve for $k_{(n-1)(n)}$ from Equation (2-2) as the $S_{(n-1)(n)}$ and $\langle d(x_{n-1}, x_n) \rangle_{m_n}$ ? are known. The value of $k_{(n-1)(n)}$ must be less than or equal to $(m_1 \cdot m_2 \cdots \cdot m_k \cdots \cdot m_{n-2})$.

6) Find the corresponding $\mathrm{RDD}(x_{n-1}, x_n)$ distance from the starting point to $x_{n-1}$.

7) Calculate $(x_1, x_2, \cdots, x_n)$ from $\mathrm{RDD}_1$, $\mathrm{RDD}_2$, $\cdots$, and check the values of $(x_1, x_2, \cdots, x_n)_1, (x_1, x_2, \cdots, x_n)_2$, and …. If these sets' numbers are equal, then no error occurs; otherwise, error exists.

We take the similar numerical as example 2-1 to verify this algorithm. (CPRDD)

Example 2-2. Assume that a moduli set $(m_1, m_2, m_3, m_4) = (4, 5, 7, 9)$ and number *X* whose residue representation is $(x_1, x_2, x_3, x_4) = (1, 2, 6, 7) = (97)_{10}$. If an error occurs at $m_2, X = (1, 3, 6, 7)$, the error detection can be described as follows.

Let us begin our procedures from the $\mathrm{RDD}(x_3, x_4) = d(x_3, x_4)$. Since $S_{34}(\text{skip distance of a cycle}) = |m_3 - m_4| = |7 - 9| = 2$,

$$d_{13} = \langle -5 \rangle_4 = 3, \quad d_{23} = \langle -3 \rangle_5 = 2$$

and $d_{34} = \langle 6 - 7 \rangle_9 = \langle -1 \rangle_9 = 8$. Then $N(d_{34}) = \langle S_{34} \cdot k_{34} \rangle_9 = \langle 2 \cdot k_{34} \rangle_9 = 8$. Solve for $k_{34}$, and let $k_{34} < (m_1 m_2) = 20$ within legitimate range $(4 \cdot 5 \cdot 7) = 140$, then $k_{34} = 4, 13$.

The corresponding $\mathrm{RDD}(x_3, x_4)$ primary distances for these two $k_{34}$ are, respectively,

$$\mathrm{RDD}_1(4) = 4 \cdot 7 + 6 = 34$$
$$\mathrm{RDD}_2(13) = 13 \cdot 7 + 6 = 97$$

Thus, the generated results of the residue representation from $\mathrm{RDD}_1(4)$ and $\mathrm{RDD}_2(13)$ are respectively

$$X_1(34) = (x_1, x_2, x_3, x_4) = (2, 4, 6, 7),$$

$$X_2(97) = (x_1, x_2, x_3, x_4) = (1, 2, 6, 7).$$

Since the calculated results of $X_1$ and $X_2$ are not identical, there must be errors in one of these moduli. We cannot determine which one is erroneous. To locate the module where the error exists, at least one additional (redundant) module must be used.

The procedure for error correction by using CPRDD algorithm is essential the same as the error detection. However, two additional redundant moduli $m_{r_1}$ and $m_{r_2}$ must be added for one error correction. Note that only one redundant modulus added for error detection.

1) Choose $m_{r_1}$ $\left(\text{or } m_{r_2}\right)$ as a referred modulus.

2) Find $k_{(1)(r_1)}, k_{(2)(r_2)}, \cdots, k_{(r_1)(r_2)}$ as the same procedures of error detection steps 2-7.

3) Examine the values of $k_{(1)(r_1)}, k_{(2)(r_2)}, \cdots, k_{(r_1)(r_2)}$. If common value exists among, $k_{(1)(r_1)}, k_{(2)(r_2)}, \cdots, k_{(r_1)(r_2)}$, then no error occurs. If there is one and only one, say $k_{(i)(r_1)}$ that has no common value with all other $k_{(j)(r_1)}$, then an error exits in modulus $m_i$. This completes the error correction procedures.

The following example is illustrated here to verify this algorithm.

Example 2-3. Error correction

As before we can further locate and correct a single error by adding two redundant moduli, $m_{r_1}$ and $m_{r_2}$. Let us use the same example. The moduli 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 $m_{r_1} = 9$ and $m_{r_2} = 11$, and the residue $X$ representation, $(x_1, x_2, x_3, x_4, x_5) = (1, 2, 6, 7, 9) = (97)_{10}$. If a single error occurs at $m_3$, e.g. $X = (1, 2, 5, 7, 9)$, and $m_4$ is assigned as a reference module, then $\langle d_{14} \rangle_4 = \langle -6 \rangle_4 = 2$, $\langle d_{24} \rangle_5 = \langle -5 \rangle_5 = 0$, $\langle d_{34} \rangle = \langle -2 \rangle_7 = 5$,and $\langle d_{45} \rangle_{11} = \langle -2 \rangle_{11} = 9$. From CPRDD algorithm, we can find the number of cycles for these RDD's.

$$\langle S_{14} k_{14} \rangle_4 = \langle 5 k_{14} \rangle_4 = 2,$$

$$\langle S_{24} k_{24} \rangle_5 = \langle 4 k_{24} \rangle_5 = 0,$$

$$\langle S_{34} k_{34} \rangle_7 = \langle 2 k_{34} \rangle_7 = 5,$$

$$\langle S_{45} k_{45} \rangle_{11} = \langle 2 k_{45} \rangle_{11} = 9$$

Since the cycle length is 9, all above $k_{ij}$ values must be less than $\left\lceil \frac{140}{9} \right\rceil = 16$. Thus we have

$$k_{14} = 2, 6, 10, 14$$

$$k_{24} = 0, 5, 10, 15$$

$$k_{34} = 6, 13.$$

$$k_{45} = 10$$

If no errors occur, all $k_{ij}$'s are equal, *i.e.*, $k_{14} = k_{24} = k_{34} = k_{45}$.

Compared to the above results with pairwise moduli, only $k_{14} = k_{24} = k_{45} = 10$ meets this condition. There exists no such value in $k_{34}$.

This shows that the module $m_3$ is faulty, therefore we can correct it as follows: since $k_{14} = k_{24} = k_{45} = 10$, the $\mathrm{RDD} = k_{14} \cdot \text{cycle length} + x_4 = 10 \cdot 9 + 7 = 97$.

Thus $x_1 = \langle 97 \rangle_4 = 1$, $x_2 = \langle 97 \rangle_5 = 2$, $x_3 = \langle 97 \rangle_7 = 6$, $x_4 = \langle 97 \rangle_9 = 7$.

This completes the error correction.

Note that the above CPRDD's for each residue-digit difference, $d_{ij}$, and $k_{ij}$ can be processed in parallel. In addition, if the referenced module is assigned to the erroneous module by chance, e.g., $m_3$ this algorithm will fail to locate the error. In this case, there are no $k_{ij}$'s values that can be found to match this condition. The way to solve the problem is, of course, to assign any other moduli, e.g., $m_1$ or $m_2$.

The hardware design for the proposed algorithm in Example 2-3 is shown in **Figure 1**.

## 3. The Target Race Distance (TRD) Scheme

The conversion or decoding technique from residue representation to $X$ in binary is usually accomplished using the mixed-radix digit (MRD) or Chinese remained theorem (CRT). An optimal matched and parallel converter of this kind can be seen in [13]. The MRD is shown by the following expression with weighted numbers:

$$x = \langle a_1 m_0 + a_2 m_1 + a_3 m_1 m_2 + \cdots + a_n m_1 m_2 \cdots m_{n-1} \rangle_{M_p}$$
$$= \sum_{i=1}^{n} \alpha_i \left( m_0 m_1 \cdots m_{i-1} \right) \text{ with } m_0 = 1,$$

where $M_p = m_1 m_2 \cdots m_n = \Pi_{i=1}^{n} m_i$, and $\alpha_i \in [0, m_{i-1}]$ is the mixed-radix conversion (MRC) of $x$.

Optimization can be obtained using this method, as the accessed table lookup time is exactly equal to the right addition time, after immediate column stage for the tree network of the adders.

 

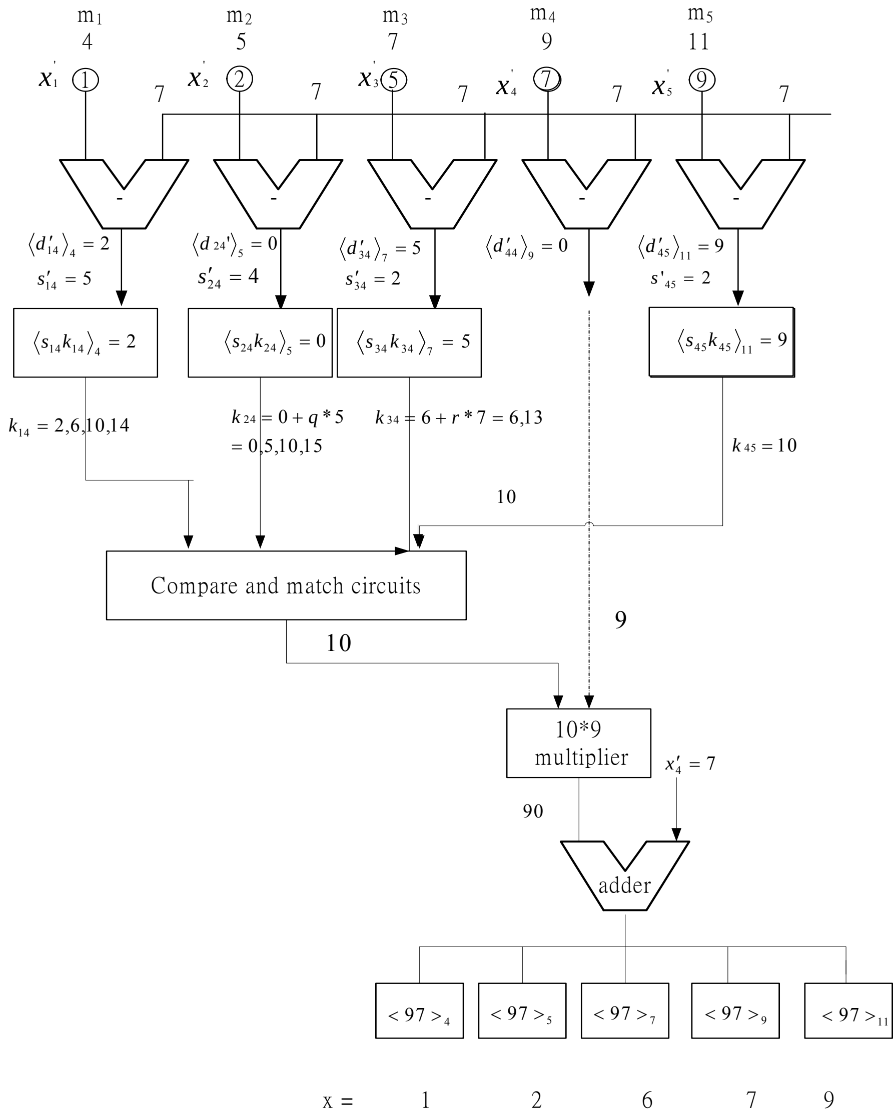


**Figure 1. The hardware implementation for the proposed error and correction location algorithm can be accomplished without using lookup tables.**

However, time is still consumed reading a large number of lookup tables. Additional hardware complexity is required by the adder-tree networks. An algorithm called the target race distance was with a simpler structure was developed for high-speed conversion.

TRD algorithm

Suppose each residue number in the $\mathrm{RNS}\langle X_i\rangle_{m_i}$ has its own track $L_i$, and the distance over track $L_i$ from 0 (starting point) to $X_i$ (end point) through $k_i$ cycles can be expressed using

$$D_i = x_i + k_i m_i,\ k_i = 0,1,\cdots,(m_i - 1),.$$

Obviously, the primary (no multiples of $m_i$) distance of $x_i$ is $(D_i)_{\min} = x_i(k_i = 0)$. To obtain the $X$ from its residue representation of $x_1, x_2, \cdots, x_r$, we must find a target such that $x_1, x_2, \cdots, x_r$ traversing the same distances over tracks $l_1, l_2, \cdots, l_r$ respectively, *i.e.* when the TRD distance of each target $x_i$ is reached, then $D_1 = D_2 = \cdots = D_r$. The TRD distance of $X$ can be found from the following theorem:

Theorem 4. Consider the simple case of two moduli sets $(m_1, m_2)$. Its residue representation and targets are $x_1$ and $x_2$ respectively. Let $(D_1)_p$ be the primary distance of residue $x_1$ from 0 to $x_1$ on the track $L_1$, and $(D_2)_p$ be the primary distance of $x_2$ from 0 to $x_2$ on track $L_2$. Then the TRD distance for these two residues $x_1$ and $x_2$ that have the same TRD distances can be obtained by the following equation.

$$\mathrm{TRD}(x_1, x_2) = (x_1 + k_1 \cdot m_1) = (x_2 + k_2 \cdot m_2) \qquad (3\text{-}1)$$

In addition, $k_1$ can be calculated from the equation

$$\langle x_1 + k_1 \cdot m_1\rangle_{m_2} = (D_2)_p = x_2$$

where $m_1$ is the cyclic length of $x_1$, and $k_1$ is number of cycles, all of the integers,

$$k_1 = 0,1,2,\cdots,m_2-1.$$

Proof: It is easy to show that the above $\mathrm{TRD}(x_i,\ x_{i+1})$ is the common target distance of $x_1$ and $x_2$, Since

$$\langle x_1 + k_1 m_1\rangle_{m_1} = x_1$$

And $\langle x_2 + k_2 m_2\rangle_{m_2} = x_2 = \langle x_1 + k_1 m_1\rangle_{m_2} = X$,

thus $\mathrm{TRD}(x_1,x_2) = \langle x_1 + k_1 m_1\rangle_{m_2} = (x_1 + k_1 m_1) = X$ is the TRD distances for both of $x_1$ and $x_2$.

Corollary: It is evident that the above theorem can be extended to $n$ moduli set $(m_1,m_2,\cdots,m_n)$ and residue number $(x_1,x_2,\cdots,x_n)$. The corresponding TRD of $(x_1,x_2,\cdots,x_n)$ are therefore

$$\begin{aligned}\mathrm{TRD}(x_1,x_2,\cdots,x_n) &= (x_1 + k_1 m_1) + (x_2 + k_2 m_1 m_2)\\ &\quad + (x_3 + k_3 m_1 m_2 m_3)\\ &\quad + \cdots(x_{n-1} + k_{n-1} m_1 m_2 \cdots m_{n-1})\end{aligned}$$

In addition, $k_i$ can be solved from the following equations.

$$\langle x_1 + k_1\cdot m_1\rangle_{m_2} = x_2$$

$$\cdots$$

$$\langle x_i + k_i\cdot m_1 m_2\cdots m_i\rangle_{m_{i+1}} = x_{i+1}$$

where $k_i = 0,1,\cdots,(m_{i+1}-1)$

Note that $x_1,x_2,\cdots,x_n$ are the targets of moduli $m_1,m_2,\cdots,m_n$ respectively and the $\mathrm{TRD}(x_1,x_2,\cdots,x_n)$ is the distance that has equal track lengths, *i.e.* $L_1 = L_2 = \cdots = L_n = L$. That is;

$\langle L\rangle_{m_1} = x_1, \langle L\rangle_{m_2} = x_2, \langle L\rangle_{m_3} = x_3, \cdots, \langle L\rangle_{m_n} = x_n$.

Example 3-1 Let the moduli set be $(m_1,m_2,m_3,m_4) = (4,\ 5,\ 7,\ 9)$ and the residue representation be $(x_1,x_2,x_3,x_4) = (3,\ 1,\ 2,\ 5)$. The procedures to find the TRD distance can be described as follows:

1) Find the primary distance $(D_1)$ of residue $x_1 = (D_1)_p = \langle 3\rangle_{m_2}$ since $m_2 > m_1$ and $\langle 3 + k_1\cdot 4\rangle_5 = 1$ is required , thus $k_1 = 2$, and $\mathrm{TRD}(x_1,x_2) = (3 + 2\cdot 4) = 11$

2) Repeat the procedure 1 to find the number of cycles $k_2$ and $k_3$ and the last TRD distances (destinations), $\mathrm{TRD}(x_1,x_2,x_3)$ and $\mathrm{TRD}(x_1,x_2,x_3,x_4)$.

Since $\hat{x}_3 = \langle 11\rangle_7 = 4$

$$\langle \hat{x}_3 + k_2\cdot 4\cdot 5\rangle_7 = 2$$

$$\langle 4 + k_2\cdot 4\cdot 5\rangle_7 = 2$$

$$\therefore k_2 = 2$$

thus $\mathrm{TRD}(x_3) = 2\cdot 4\cdot 5 = 40$

and $\mathrm{TRD}(x_1 x_2 x_3) = 11 + 40 = 51$

$$\hat{x}_4 = \langle 51\rangle_9 = 6$$

$$\langle 6 + k_3\cdot 4\cdot 5\cdot 7\rangle_9 = 5$$

$$\therefore k_3 = 7$$

thus $\mathrm{TRD}(x_4) = 7\cdot 4\cdot 5\cdot 7 = 980$

and $\mathrm{TRD}(x_1 x_2 x_3 x_4) = 51 + 140\cdot 7 = 1031$

The final TRD distance is the common distinction of this system for targets $x_1,x_2,x_3$ and $x_4$ *i.e.* $\mathrm{TRD}(x_1 x_2 x_3 x_4) = 1031 = X$. This result can be verified as follows:

$\langle 1031\rangle_4 = 3$, $\langle 1031\rangle_5 = 1$, $\langle 1031\rangle_7 = 2$ and $\langle 1031\rangle_9 = 5$

**Figure 2** Shows the TRD's on tracks $l_1,l_2,l_3$ and $l_4$ respectively.

Error detection and correction by TRD algorithm

A redundant residue number system with $r = 1$ redundant moduli will allow detection of any single error [4,14]. Consider the moduli set $(m_1,m_2,m_3,m_4) = (4,5,7,9)$ and the correct residue representation $X(x_1,x_2,x_3,x_4) = (1,2,6,7) = 97$. Let us

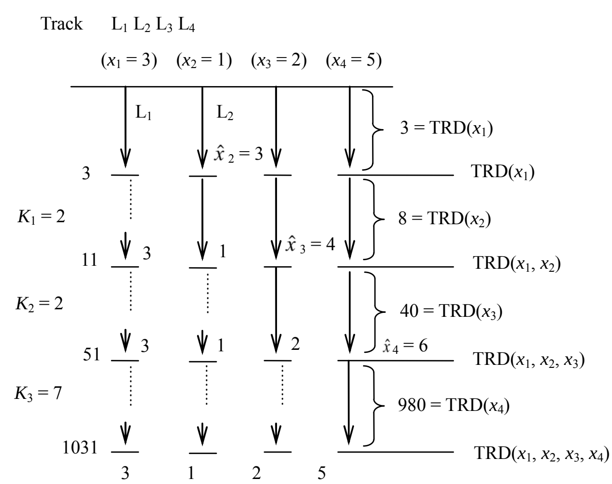


**Figure 2. TRD's on track $L_1$, $L_2$, $L_3$ and $L_4$.**

assume that $m_4 = 9$ is the redundant moduli with a single error $X_F(x_1, x_2, x_3, x_4) = (1,3,6,7)$ residue representation. The TRD theorem can be used to detect this error. We find that final TRD for $x_1, x_2, x_3$ and $x_4$ does not fall into the legitimate range as follows *i.e.* $R_F > (4 \cdot 5 \cdot 7) = 140$

$$\mathrm{TRD}_1(x_1 = 1) = 1$$
$$\mathrm{TRD}_2(x_2 = 3) = 12(k_1 = 3)$$
$$\mathrm{TRD}(x_1 x_2) = \mathrm{TRD}_1 + \mathrm{TRD}_2 = 13$$
$$\hat{x}_3 = \langle 13 \rangle_7 = 6 = x_3$$
$$\mathrm{TRD}_3(x_3 = 6) = 0$$
$$\hat{x}_4 = \langle 13 \rangle_9 = 4$$
$$\langle 4 + k_3 \cdot 140 \rangle_9 = 7$$
$$k_3 = 6$$
$$\therefore \mathrm{TRD}_4(x_4 = 7) = 6 \cdot 140 = 840.$$

The final TRD distance $\mathrm{TRD}(x_1, x_2, x_3, x_4) = 13 + 840 = 853 > 140$. If we need to locate and correct this module error, another redundant module must be added. Let us assume that $m_5 = 11$ for this requirement in the above residue representation.

The current redundant moduli set is $(m_1, m_2, m_3, m_4, m_5) = (4, 5, 7, 9, 11)$ and the correct residue representation is $x = (x_1, x_2, x_3, x_4, x_5) = (1, 2, 6, 7, 9) = (97)$. Let us assume that $m_4 = 9$ and $m_5 = 11$ are the redundant moduli. With a single error $x_F = (x_1, x_2, x_3, x_4, x_5) = (1,3,6,7,9)$. The TRD theorem can again be used to locate and correct this error. We find that final TRD's for $(x_1, x_2, x_4, x_5) = (1,3,7,9)$ dose not fall in the legitimate range, but other final TRD's for $(x_1, x_2, x_3, x_4) = (1,6,7,9)$ do, falls in the legitimate range:

1) TRD for $x_1, x_2, x_4$ and $x_5$

$$\mathrm{TRD}_1(x_1 = 1) = 1$$
$$\mathrm{TRD}_2(x_1, x_2) = 13$$
$$\mathrm{TRD}_4(x_1, x_2, x_4) = 133$$
$$\hat{x}_5 = \langle 133 \rangle_{11} = 1$$
$$\langle 1 + k_4 \cdot 180 \rangle_{11} = 9 = x_5, k_4 = 2$$
$$TRD_5(x_1, x_2, x_4, x_5) = 133 + 360$$
$$= 493 > 140(\text{out of legitimate range}).$$

2) TRD for $x_1, x_3, x_4$ and $x_5$

$$\mathrm{TRD}_1(x_1 = 1) = 1$$
$$\mathrm{TRD}_3(x_3 = 6) = 12; k_2 = 3$$
$$\hat{x}_4 = \langle 13 \rangle_9 = 4$$
$$\langle 4 + k_3 \cdot 28 \rangle_9 = 7, k_3 = 3$$
$$\mathrm{TRD}_4(x_1, x_3, x_4) = 13 + 84 = 97$$
$$\hat{x}_5 = \langle 97 \rangle_{11} = 9 = x_5$$
$$\therefore TRD_5(x_1, x_3, x_4, x_5)$$
$$= 97 < 140(\text{within legitimate range}).$$

Thus, the error is located at module $m_2$ and must be corrected to $x_2 = \langle 97 \rangle_5 = 2$.This algorithm can also be used for multiple error corrections. However, at least three redundant moduli are required. The procedures are similar.

## 4. Scaling with Error Correction

The above proposed algorithm used for error detection and correction has the advantage of not requiring lookup tables. No CRT (Chinese residue theorem) decoding processes are required. However, it is still time consuming and requires extensive hardware complexity for each module having multiple-value inputs to the match unit and selecting a correct one as a output. To improve this drawback, an optimal matching algorithm is proposed here for the error correction. The following two theorems will be used and an example follows.

Theorem 5. Let $m_1$ and $m_2$ be two relative prime numbers in RNS for module 1 and module 2 respectively. Then there must exist the relation represented by the equation $\langle m_1 x_1 \rangle_{m_2} - \langle m_2 x_2 \rangle_{m_1} = |k|$, where $0 \le \langle m_1 x_1 \rangle_{m_2} \le m_2$, $0 \le \langle m_2 x_2 \rangle_{m_1} \le m_1$ so that $0 \le |k| \le m_2$, assuming $m_2 > m_1$. The $x_{1,} x_2$ and $k$ are restricted to integers.

Proof: As a first step, let $k = 0$. It is easily seen that $x_1 = m_2$ and $x_2 = m_1$ will be satisfied. Next consider $k \ne 0$. Since there are two different pair combination $\langle m_1 x_1 \rangle_{m_2} \le m_2$ and $\langle m_2 x_2 \rangle_{m_1} < m_1$, thus the difference between $m_1 x_1$ and $m_2 x_2$ of $k$ will always be satisfied for $0 \le |k| \le m_2$, where k is restricted in integers.

Theorem 6. If the values of $m_1$ and $m_2$ and $k$ in the equation $\langle m_1 p_1 \rangle_{m_2} - \langle m_2 p_2 \rangle_{m_1} = |k|$ are known, then $p_1$ and $p_2$ can always be determined from equation $\langle (m_2 - m_1) p_1 \rangle_{m_1} = k$ or $\langle (m_2 - m_1) p_2 \rangle_{m_2} = k$, where $p_1$, $p_2$ and $k$ are within the range: $0 \le p_{1k} \le (m_1 \text{ or } m_2)$

Proof: Let the difference value of $|m_2 - m_1|$ be equal to $d$, then $d$ will be the integers within the range between 0 and $m_2$, *i.e.*, $p_1 = 0,1,2,\cdots,(m_1 - 1)$, or $p_2 = 0,1,2,\cdots,(m_2 - 1)$. These two expressions show that we can always select an integer value p, within the interval between 0 and $(m_1 - 1)$ or $(m_2 - 1)$ to satisfy the conditions $\langle dp_1 \rangle_{m_1} = k$ or $\langle dp_2 \rangle_{m_2} = k$

Example 4-1 Let $m_1 = 5$, and $m_2 = 7$. Find the minimum values of $p_1$ and $p_2$ respectively from the following equation :

$$7p_1 - 5p_2 = 3$$

Since $m_1 = 5$ and $m_2 = 7$, we have
$d = m_2 - m_1 = 7 - 5 = 2$,
and

$$\langle 2p_1 \rangle_5 = 3 \quad (4\text{-}1),$$

or

$$\langle 2p_2 \rangle_7 = 3 \quad (4\text{-}2),$$

from Equation (4-1)
$\langle 2p_1 \rangle_5 = 3$ so $p_1 = 4,\ (2 \cdot 4) - 5 = 3$,
from Equation (4-2)
$\langle 2p_2 \rangle_7 = 3$, so $p_2 = 5$ for $(2 \cdot 5) - 7 = 3$.

This result can be verified by substituting $7 \cdot 4 - 5 \cdot 5 = 3$ into the above equation. Theorem 6 is very useful as shown in the following example.

In Theorem 3 of Section III, the number of cycles on track $L_{ij}$ from the starting point "0" to its target position "$d_{ij}$" can be expressed by setting $\hat{d}_{ij} = 0$, *i.e.*

$$\langle s_{ij} k_{ij} \rangle_{m_i} = d_{ij}, \text{ or } \langle s_{ij} k_{ij} + p_i m_i \rangle_{m_i} = d_{ij} \quad (4\text{-}3),$$

where $s_{ij}$ is the module i stride distance referring to module j. Similarly, the number of cycles on track $l_{jk}$ from the starting point "0" to its target position "$x_k$" can be expressed by setting $\hat{d}_{ij} = 0$, *i.e.*;

$$\langle s_{jk} k_{jk} \rangle_{m_2} = d_{jk} \text{ or } \langle s_{jk} k_{jk} + p_k m_j \rangle_{m_j} = d_{jk} \quad (4\text{-}4)$$

Since, from theorem 3, the cyclic length of the residue digits differences reference to module $m_j$ is constant (uniform), then there must exist a condition,
$c_{ij} \cdot s_{ij} \cdot k_{ij} = c_{jk} \cdot s_{jk} \cdot k_{jk}$ Eliminating the above terms from Equations (4-3) and (4-4),

$$c_{ij} \cdot p_i \cdot m_i - c_{jk} \cdot p_{jk} \cdot m_k = c_{ij} \cdot d_{ij} - c_{jk} \cdot d_{jk} = D'_{ik}$$

$$p'_i m_i - p'_i m_k = D'_{ik}$$

where $p'_i = c_{ij} p_i$, $p'_k = c_{jk} p_k$ and
$D'_{ik} = c_{ij} \cdot d_{ij} - c_{jk} \cdot d_{jk}$

Example 4-2

Let the moduli set $(m_1, m_2, m_3, m_4, m_5) = (4,5,7,9,11)$
$x = (x_1, x_2, x_3, x_4, x_5) = (1,2,6,7,9)$, and the error
$x' = (x'_1, x'_2, x'_3, x'_4, x'_5) = (1,2,5,7,9)$, the error occurs at $m_3$.

Follow the same procedures of the Example 4-1 to use this algorithm.

$$\langle S_{14} k_{14} \rangle_4 = \langle 5k_{14} \rangle_4 = 2, \text{ or } \langle 5k_{14} + 4p_1 \rangle_4 = 2 \quad (4\text{-}5)$$

$$\langle S_{24} k_{24} \rangle_5 = \langle 4k_{24} \rangle_5 = 0, \text{ or } \langle 4k_{24} + 5p_2 \rangle_5 = 0 \quad (4\text{-}6)$$

$$\langle S_{34} k_{34} \rangle_7 = \langle 2k_{34} \rangle_7 = 5, \text{ or } \langle 2k_{34} + 7p_3 \rangle_7 = 5 \quad (4\text{-}7)$$

$$\langle S_{45} k_{45} \rangle_{11} = \langle 2k_{45} \rangle_{11} = 9, \text{ or } \langle 2k_{45} + 11p_5 \rangle_{11} = 9 \quad (4\text{-}8)$$

Eliminating $5k_{14}$ and $4k_{24}$ from Equation's (4-5) and (4-6)
$16p_1 - 25p_2 = 8$,
$p_1 = 13$, and $p_2 = 8$,
solve for $k_{14}$ from (4-5),

$$\langle 5 * k_{14} + 4 * 13 \rangle_4 = 2,$$

$$\therefore \quad k_{14} = 10,$$

or $4 \cdot k_{24} = 5 \cdot 8 = 40$,
$\therefore \quad k_{24} = 10,$.

Check from Equation (4-5),

$$\langle 2 \times 10 \rangle_{11} = 9, \quad \langle 2 \times 10 \rangle_7 = 6 \neq 5.$$

This shows that the error occurs at module $m_3$. From this result, we can immediately obtain $\langle 2 \times 10 \rangle_7 = 6$. Noting that it may happen that the assigned referenced memory moduli falls coincidentally with error memory module $m_3$. In this occurrence, we cannot find the correct (integers) values of $P_1$ and $P_2$ within the legitimate range. It seems that this algorithm can only detect error. To complete the error correction procedure, we can simply change the referenced module to any other and follow the same procedure as before. This guarantees that the proposed algorithm in Theorem 4 will also work well in this case. The hardware structure for illustrating this algorithm is shown in **Figure 3**.

The proposed TRD (target Race Distance) scheme used for error correction can be used for scaling and assigning numbers in a residue number system. A redundant residue number system (RRNS) is defined as before in an RNS with *r* additional moduli. The moduli $\{m_1, m_2, \cdots, m_i, \cdots, m_k\}$, are called the nonredundant moduli, while the extra r moduli, $\{m_{k+1}, m_{k+2}, \cdots, m_{k+r}\}$ are the redundant moduli. The interval, $[0, M_k - 1]$, is called the legitimate range where $M_k = \prod_{i=1}^{k} m_i$ and the interval, $[M_k, M_{kr} - 1]$, is the illegitimate range, where $M_{kr} = M_k M_r = M_k \prod_{i=1}^{r} m_{k+i}$ is the total range. In the RRNS, the negative numbers within the dynamic range are represented as states at the upper extreme of the total range, which is part of the illegitimate range. The positive members are mapped to the interval $\left[0, \frac{(M_k - 1)}{2}\right]$, if $M_k$ is odd, or $\left[0, \frac{M_k}{2}\right]$, if $M_k$ is even. The negative numbers are mapped to the interval

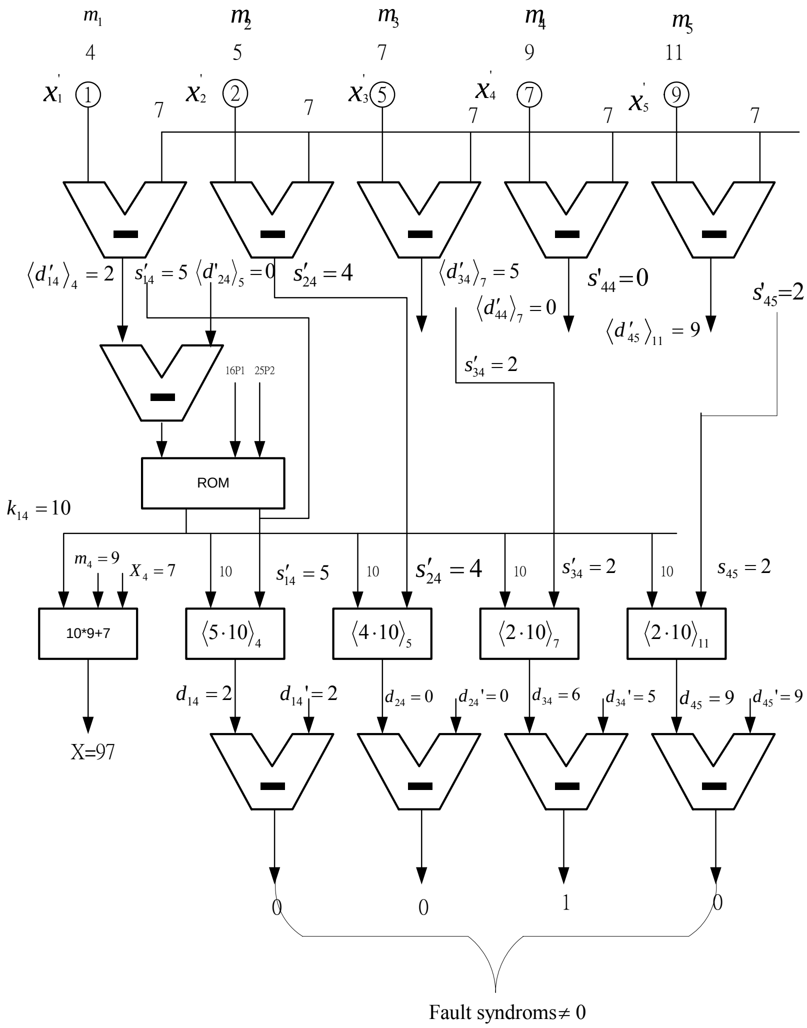


**Figure 3. In the block diagram using optimal matching between multiples $P_i$ m and $P_k m_k$, the residue digits are corrected by $x_i$ - $x_4$ = $d_{i4}$.**

$\left[M_{kr} - \frac{(M_k - 1)}{2}, M_{kr} - 1\right]$ if $M_k$ is odd or

$\left[M_{kr} - \frac{M_k}{2}, M_{kr} - 1\right]$ if $M_k$ is even [14].

The one-to-one correspondence between the integers of the dynamic range and the states of the legitimate range in the RRNS can be established using a polarity shift. [11], The polarity shift is defined as below.

$$X_p = X + \frac{M_k}{2} \text{ for } M_k \text{ even} = X + \frac{M_k - 1}{2} \text{ for } M_k \text{ odd.}$$

where $X_p$ denotes the value $X$ after a polarity shift and $X \in \left[-\frac{M_k}{2}, \frac{M_k}{2}\right]$ if $M_k$ is odd, so that $X_p \in [0, M_k]$, a polarity shift needs to be performed prior to correcting or scaling since $X_p$ belongs to the legitimate range. If a single residue digit error $e_j = \{0, 0, \cdots, e_j, 0, \cdots, 0\}$ is introduced and corresponds to modules $m_j$, then, after a polarity shift.

$$x'_p = \langle X_p + E_j \rangle_{Mkr} = \left\langle X_p + \frac{M_{kr}}{m_j} \langle w_j e_j \rangle_{m_j} \right\rangle M_{kr}$$

$$= \left\langle X_p + \frac{M_{kr}}{m_j} e'_j \right\rangle_{Mkr},$$

where $w_j$ is the multiplicative inverse of $\frac{M_{kr}}{m_j}$ moduli $m_j$ *i.e.* $\left\langle \left( \frac{M_{kr}}{m_j} \right)_{wj} \right\rangle_{m_j} = 1$ and $e'_j = \langle w_j e_j \rangle_{m_j}$ The $x'_p$ denotes a single residue digit error and must fall within the illegitimate range , $m_k \le x'_p < m_{kr}$ [11].

Since $\left[\frac{x'_p}{M_k}\right] < M_r \left(x'_p < m_k \cdot m_r\right)$, and can be represented uniquely by $\{a_{k+1}, a_{k+2}, \cdots, a_{k+r}\}$ ,where $a_{k+r}$'s are the coefficient from the Chinese Remainder Theorem (CRT), *i.e*, $x'_p = \sum_{i=1}^{k+r} a_i \prod_{j=i-1}^{k+r-1} m_j$ , where $m_0 = 1, 0 \le a_i \le m_i$. Note that the redundant digits $a_{k+1}, a_{k+2}, \cdots, a_{k+r}$ are zeros if no error is introduced, while at least one redundant digit is not equal to zero if a single error is introduced. Therefore, it has the same meaning that $\left[\frac{x'_p}{M_k}\right]_r$ or $\{a_{k+1}, a_{k+2}, \cdots, a_{k+r}\}$ is used to be the entries of the error correction.

1) $M_r > m_i m_{k+5}$, $1 \le i \le k$, $1 \le s \le r$ , and ,

2) $M_r > 2m_i m_j - m_i - m_j$, $1 \le i$, $j \le k$, and $i \ne j$

Although the errors detection and correction described in section II have been simplified the processes due to no need of CRT conversion. It is still hardware complex and time consuming for the residue scaling operation. To improve this drawback, a direct residue-scaling algorithm can be used. It is flexible and direct to detect and prevent the errors. The flexibility means that the scaling factor can be arbitrary chosen any single module such as $m_i$, *i.e.* not necessarily beginning from $m_1, m_2, \cdots$ to $m_k$. in order. The direct capability means no requirement for CRT extension processes for decoding or lookup tables. The following theorem (theorem 7) and example are clarified.

Theorem 7. If the scaling factor K is one of the module set $= \{m_1, m_2, \cdots, m_i, \cdots, m_k, \cdots, m_{k+r}\}$ and the residue digits are $\{x_1, x_2, \cdots, x_i, \cdots, x_k, \cdots, x_{k+r}\}$ , respectively, then the residue digit $x_i$ scaled by a factor $m_{j,i\ne j,} = \left(\frac{x_i}{m_i}\right) = y_i$ can be obtained using the equation

$$\langle m_i y_i \rangle_{m_j} = x_i \quad (4\text{-}9).$$

Proof: It is easy to show that when $m_1 \ne m_2$ , and Equation (4-9) is divided by $m_i$ on both side, we have

$$\left\langle \frac{m_i y_i}{m_i} \right\rangle_{m_j} = \frac{x_i}{m_i} = y_i \quad (4\text{-}10).$$

Example 4-3. For convenient comparison of the proposed TRD algorithm to other schemes such as appeared in [14], we take the same numerical example in [11]. Let the moduli set $\{m_1, m_2, m_3, m_4, m_5, m_6\} = \{2,5,7,9,11,13\}$, where $\{m_1, m_2, m_3, m_4\}$ are regular moduli and $\{m_5, m_6\}$ are redundant moduli. Then $M_k = 2\cdot5\cdot7\cdot9 = 630$, $M_r = 11*13 = 143$, $M_{kr} = M_k M_r = 630\cdot143 = 90090$ , and $X_p \in \left[-\frac{M_k}{2}, \frac{M_k}{2}\right] = [-315, 315]$. The sufficient conditions for correcting single residue digits errors are

1) $M_r > \{m_i m_{k+s}\}_{\max}$ $i = 1,2,3$ , or 4, $s = 1$ , or 2, $k = 4$ , The maximum $m_i m_{k+s} = m_4 \cdot m_6 = 9\cdot13 = 117 < 143(M_r)$ , and

2) $M_r > \{2m_i m_j - m_i - m_j\}_{\max} i, j = 1,2,3,\text{or } 4,$

$$\begin{aligned} \text{The max } \{2m_i m_j - m_i - m_j\} &= 2m_3 m_4 - m_3 - m_4 \\ &= 2\cdot7\cdot9 - 7 - 9 \\ &= 110 < 143(= M_r). \end{aligned}$$

Thus the moduli set satisfies the necessary and sufficient conditions for correcting single errors digit. Assume $X = -311 = \{1,4,4,4,8,1\}$ and a single digit error $e_2 = 4$ is introduced, then $X' = \{1,3,4,4,8,1\} = 53743$ . After a polarity shift, $X'_p = X' + \frac{M_k}{2} = \{0,3,4,4,4,4\}$.

Follow the same procedures as shown in Example 4-2. CPRDD is applied for correction without the need for using a table.

1) Assign the moduli $m_4 = 9$ as the reference moduli, the following residue digit references and its corresponding CPRDD equations: $\langle sk_{ij} \rangle_{m_i} = \langle d_{ij} \rangle_{m_i}$ are obtained

$$\langle d_{14} \rangle_2 = 1, \langle 7k_{14} \rangle_2 = 1;$$

$$\langle d_{24} \rangle_5 = 4, \langle 4k_{24} \rangle_5 = 4;$$

$$\langle d_{34} \rangle_7 = 0, \langle 2k_{34} \rangle_7 = 0;$$

$$\langle d_{45} \rangle_{11} = 7, \langle 2k_{45} \rangle_{11} = 7;$$

$$\langle d_{46} \rangle_{13} = 3, \langle 4k_{46} \rangle_{13} = 3.$$

2) Choose two highest digit difference as one pair for equal target race distance e.g. $\langle 2k_{45} \rangle_{11} = 7$ and $\langle 4k_{46} \rangle_{13} = 3$ . Then the true primary RDD equations are

$$\langle 2k_{45} + 11p_1 \rangle_{11} = 7 \quad (4\text{-}11),$$

And $\langle 4k_{46} + 13p_2 \rangle_{13} = 3$ (4-12),

where $p_1$ and $p_2$ are selected so that the two RDD are equal distances.

3) Eliminating k terms in Equation's (4-11) and (4-12) by putting $k_{45} = k_{46}$

$(22p_1) - (13p_2) = 11$, $p_1 = \frac{11 + (13p_2)}{22}$ where

$p_2 = 11$ , then $p_1 = \frac{11 + 143}{22} = 7$ .

4) Substituting $p_1$ and $p_2$ into equations (4-9) and (4-10) respectively, we have $2k_{45} = -11\cdot7 + 7 = -70$ , then $k_{45} = -35$ , and $4k_{46} + 13\cdot11 = 3$ , also, $k_{46} = \frac{-143 + 3}{4} = -35$ .

5) Checking other three RDD's

$$\langle 4k_{24}\rangle_5 = 4 - \rangle\langle 4\times(-35)\rangle_5 = 4,$$

$$\langle 2k_{34}\rangle_7 = 0 - \rangle\langle 2\times(-35)\rangle_7 = 0,$$

$$\langle 7k_{14}\rangle_2 = 1 - \rangle\langle 7\times(-35)\rangle_2 = 1.$$

The only different module residue occurs on module number at $m_2 = 5$, *i.e.*, $x'_{24} - x_{24} = 4 - 0 = 4$. The three target distances, can be from any module residue, say, (except $m_2 = 5$), $m = 7$. $x = -9\cdot 35 + 4 = -311$.

The residue representation of $X$ is therefore, $X = \{1,4,4,4,8,1\}$. If a single digit error $e_2 = 4$ is introduced, then, $X' = 53743 = \{1,3,4,4,8,1\}$.The corresponding error is therefore

$$e_2 = \{(x'_1 - x_1),(x'_2 - x_2),(x'_3 - x_3),\cdots,(x'_6 - x_6)\}.$$

$$\therefore\quad X' - X = \{0,-1,0,0,0,0\} = \{0,4,0,0,0,0\}$$

After a polarity shift,

$$x'_p = x' + \frac{M}{2} = 54058 = \{1,3,4,4,8,1\} + \{0,3,4,4,4,4\}$$

and the scaling factor $\frac{1}{K}$ to $x'_p$ is $\left[\frac{x'_p}{K}\right] = 85\left(M_k = 2\cdot 5\cdot 7\cdot 9 = 630\right)$. The final step must use a lookup table to obtain the result, $\left[\frac{x_p}{K}\right]$ [13].

For verifying our proposed algorithm, the table of the corresponding $\left[\frac{x'_p}{K}\right]$ is not required as in [13]. The processes for finding and correcting a single error based on our method are described below.

1) Find the residue digit difference to a selected module, say $m_4$ as before $x' = 53743 = \{1,3,4,4,8,1\}$. For verifying that our proposed algorithm detects and corrects single error without using a table, the same numerical example is used to describe the procedure as follows:

$$d_{14} = \langle -3\rangle_2 = 1,$$

$$d_{24} = \langle -1\rangle_5 = 4,$$

$$\mathrm{d}_{34} = \langle 0\rangle_7 = 0,$$

$$d_{45} = \langle -4\rangle_{11} = 7,$$

$$d_{46} = \langle 3\rangle_{13} = 3.$$

Then

$$\langle 7k_{14}\rangle_2 = 1,$$

$$\langle 4k_{24}\rangle_5 = 4,$$

$$\langle 2k_{34}\rangle_7 = 0,$$

$$\langle 2k_{45}\rangle_{11} = 7,$$

$$\langle 4k_{46}\rangle_{13} = 3.$$

2) Choose two highest digit differences as one pair for equal target race distances. e.g. $\langle 2k_{45}\rangle_{11} = 7$ and $\langle 4k_{46}\rangle_{13} = 3$, the following two equations can be obtained:

$$\langle 2k_{45} + 11p_1\rangle_{11} = 7 \qquad (4\text{-}13a)$$

$$\langle 4k_{46} + 13p_2\rangle_{13} = 3 \qquad (4\text{-}13b).$$

3) Eliminating k terms in (4-13a) and (4-13b) by putting $k_{45} = k_{46}$

$22p_1 - 13p_2 = 0$ then $p_1 = 13$ and $p_2 = 22$.

4) Substituting $p_1$ and $p_2$ into Equation's (4-13a) and (4-13b) respectively, we have $2k_{45} + 11\cdot 13 = 0$, then $k_{45} = \frac{-143}{2}$, and $4k_{46} + 13\cdot 22 = 0$, also, $k_{46} = \frac{-143}{2}$

| $X'_p = X' + \frac{M}{2}$ | $X_p = X + \frac{M}{2} = -311 + 315 = 4$ |
|---|---|
| $\langle 1+1\rangle_2 = 0$ | $\langle 4\rangle_2 = 0$ |
| $\langle 0+3\rangle_5 = 3$ | $\langle 4\rangle_5 = 4$ |
| $\langle 0+4\rangle_7 = 4$ | $\langle 4\rangle_7 = 4$ |
| $\langle 0+4\rangle_9 = 4$ | $\langle 4\rangle_9 = 4$ |
| $\langle 7+8\rangle_{11} = 4$ | $\langle 4\rangle_{11} = 4$ |
| $\langle 3+1\rangle_{13} = 4$ | $\langle 4\rangle_{13} = 4$ |

Obviously, the error is located at $m_2 = 5$ thus $e_2 = X'_p - X_p = 3 - 4 = -1$.

Furthermore, the CPRDD algorithm can be used directly and in parallel for residue scaling and error correction. Thus the process is greatly speeded up.

Example 4-4 For convenient comparison, the same numeric example as in [13] is illustrated here. Consider $\{m_1, m_2, m_3, m_4, m_5, m_6\} = \{2,5,7,9,11,13\}$, and scaling factor $K = m_1\cdot m_2 = 2\cdot 5 = 10$. If an input $X = -205 = \{1,0,5,2,4,3\}$ and a single residue digit error $e_3 = 1$, corresponding to $m_3 = 7$,

Then $X' = 25535 = \{1,0,6,2,4,3\}$.

After a polarity shift,

$$x'_p = x' + \frac{M}{2} = 25850 = \{0,0,6,2,0,6\}$$

1) Dividing by $m_1 = 2$ after subtracting $x'_{p_1} = 0$ from $x_2, x_3, \cdots, x_6$

$\langle 2p_2\rangle_5 = 0$, this leads $p_2 = 0$,

$\langle 2p_3\rangle_7 = 6, p_3 = 3$,

$\langle 2p_4\rangle_9 = 2, p_4 = 1$,

$\langle 2p_5\rangle_{11} = 0, p_5 = 0$,

$\langle 2p_6\rangle_{13} = 6, p_6 = 3$.

2) Dividing by $m_2 = 5$ after subtracting $x'_{p_2} = 0$ from $x_3, x_4, \cdots, x_6$

$\langle 5k_3\rangle_7 = 3, k_3 = 2, 9, 16, \cdots;$

$\langle 5k_4\rangle_9 = 1, k_4 = 2, 11, 20, \cdots;$

$\langle 5k_5\rangle_{11} = 0, k_5 = 0, 11, 22, \cdots;$

$\langle 5k_6\rangle_{13} = 3, k_6 = 11, 24, 37, \cdots.$

Since from above only $k_3$ does not match with all other's $k_i$, *i.e.* $k_3 \cap k_{4,5,6} = 0$ and $k_4 \cap_5 \cap k_6 = 11$. Therefore, there occurs an error at $m_3 = 7$. Once this error is detected, it is easily found and corrected from the above equations, $\langle 5k_3\rangle_7 = \langle 5\cdot 11\rangle_7 = 6$, which in turn $p'_3 = k_3 = 6$ and $X = \langle 2p'_3\rangle_7 = \langle 2*6\rangle_7 = 5$.

$$m_i = \begin{pmatrix}2 & 5 & 7 & 9 & 11 & 13\end{pmatrix}$$

$$x = -205, x_i = \begin{pmatrix}1 & 0 & 5 & 2 & 4 & 3\end{pmatrix}$$

$$630/2 = 315 = \frac{M}{2} = \begin{pmatrix}1 & 0 & 0 & 0 & 7 & 3\end{pmatrix}$$

$$-205 + 315, x + \frac{M}{2} = \begin{pmatrix}0 & 0 & 5 & 2 & 0 & 6\end{pmatrix} = 110$$

that $\left(p_1 p_2 p_3 p_4 p_5 p_6\right) = \begin{pmatrix}0 & 0 & 5 & 2 & 0 & 6\end{pmatrix}$

Divided by "2",

$\langle 2p'_2\rangle_5 = p_2 = 0, p'_2 = 0;$

$\langle 2p'_3\rangle_7 = p_3 = 5, p'_3 = 6;$

$\langle 2p'_4\rangle_9 = p_4 = 2, p'_4 = 1;$

$\langle 2p'_5\rangle_{11} = p_5 = 0, p'_5 = 0;$

$\langle 2p'_6\rangle_{13} = p_6 = 6, p'_6 = 3;$

$\left(p'_2 p'_3 p'_4 p'_5 p'_6\right) = \begin{pmatrix}0 & 6 & 1 & 0 & 3\end{pmatrix} = 55(=110/2)$.

Divided by "5"

$\langle 5p''_3\rangle_7 = p'_3 = 6, p''_3 = 4$;

$\langle 5p''_4\rangle_9 = p'_4 = 1, p''_4 = 2;$

$\langle 5p''_5\rangle_{11} = p'_5 = 0, p''_5 = 0;$

$\langle 5p''_6\rangle_{13} = p'_6 = 3, p''_6 = 11$

$\left(p''_3 p''_4 p''_5 p''_6\right) = (42011) = 11(=55/5)$.

The hardware structure of this example for the residue scaling is shown in **Figure 4**.

Actually this algorithm can be divided by any arbitrary moduli.

Example 4-5

Divided by any arbitrary moduli, say $m_4 = 9$, it must subtract $x'_{p4} = 2$ from $X$

$\langle x'_{p1} - x'_{p4}\rangle_2 = \langle 0-2\rangle_2 = 0,$

$\langle x'_{p2} - x'_{p4}\rangle_5 = \langle 0-2\rangle_5 = \langle -2\rangle_5 = 3,$

$\langle x'_{p3} - x'_{p4}\rangle_7 = \langle 6-2\rangle_7 = 4,$

$\langle x'_{p5} - x'_{p4}\rangle_{11} = \langle 0-2\rangle_{11} = 9,$

$\langle x'_{p6} - x'_{p4}\rangle_{13} = \langle 6-2\rangle_{13} = 4,$

Then

$\langle 9k_1\rangle_2 = 0, k_1 = 0;$

$\langle 9k_2\rangle_5 = 3, k_2 = 2;$

$\langle 9k_3\rangle_7 = 4, k_3 = 2;$

$\langle 9k_5\rangle_{11} = 9, k_5 = 1;$

$\langle 9k_6\rangle_{13} = 4, k_6 = 12;$

check $(25850-2)/9 = 2872$.

This results

$$x'_1 = \langle 2872\rangle_2 = 0, x'_2 = \langle 2872\rangle_5 = 2, x'_3 = \langle 2872\rangle_7 = 2,$$

$$x'_5 = \langle 2872\rangle_{11} = 1, \text{ and } x'_6 = \langle 2872\rangle_{13} = 12.$$

It can be seen from above that $x'_1 = k_1, x'_2 = k_2, x'_3 = k_3, x'_4 = k_4 = 0, x'_5 = k_5$, and $x'_6 = k_6$, which are equal each other as expected.

Example 4-6

For processing two residue scalings and error corrections in parallel, we take Example 4-4 as an illustration. Let scaling factor $K = 2*5 = 10$, *i.e.*, the first residue scaling factor is 2 and the second one is 5 or verse versa. It is easily shown that the extended CPRDD algorithm is used and can be completed in one cycle. That is

$\langle 10p'_2\rangle_5 = p_2 = 0, p'_2 = 0, 1, \cdots, 11, 12, \cdots;$

$\langle 10p'_3\rangle_7 = p_3 = 5, p'_3 = 4, 11, 18, \cdots;$

$\langle 10p'_4\rangle_9 = p_4 = 2, p'_4 = 2, 11, 20, \cdots;$

$\langle 10p'_5\rangle_{11} = p_5 = 0, p'_5 = 0, 11, 22, \cdots;$

$\langle 10p'_6\rangle_{13} = p_6 = 6, p'_6 = 11, 24, 37, \cdots.$

The result is identical

$$X_p = \frac{x + \frac{M}{2}}{10} = \frac{-205+315}{10} = \frac{110}{10} = 11, \textit{ i.e.},$$

$\langle 11\rangle_7 = 4, \langle 11\rangle_9 = 2, \langle 11\rangle_{11} = 0$, and $\langle 11\rangle_{13} = 11$, which are identical results as shown in Example 4-4.

Example 4-7 For error correction

$$m_2\ m_3\ m_4\ m_5\ m_6 \quad \begin{pmatrix}5 & 7 & 9 & 11 & 13\end{pmatrix}$$

$$\frac{x + \frac{M}{2}}{2} = \begin{pmatrix}0 & 3^* & 1 & 0 & 3\end{pmatrix} = 55,\ k_{ij} = \frac{55}{11}$$

 

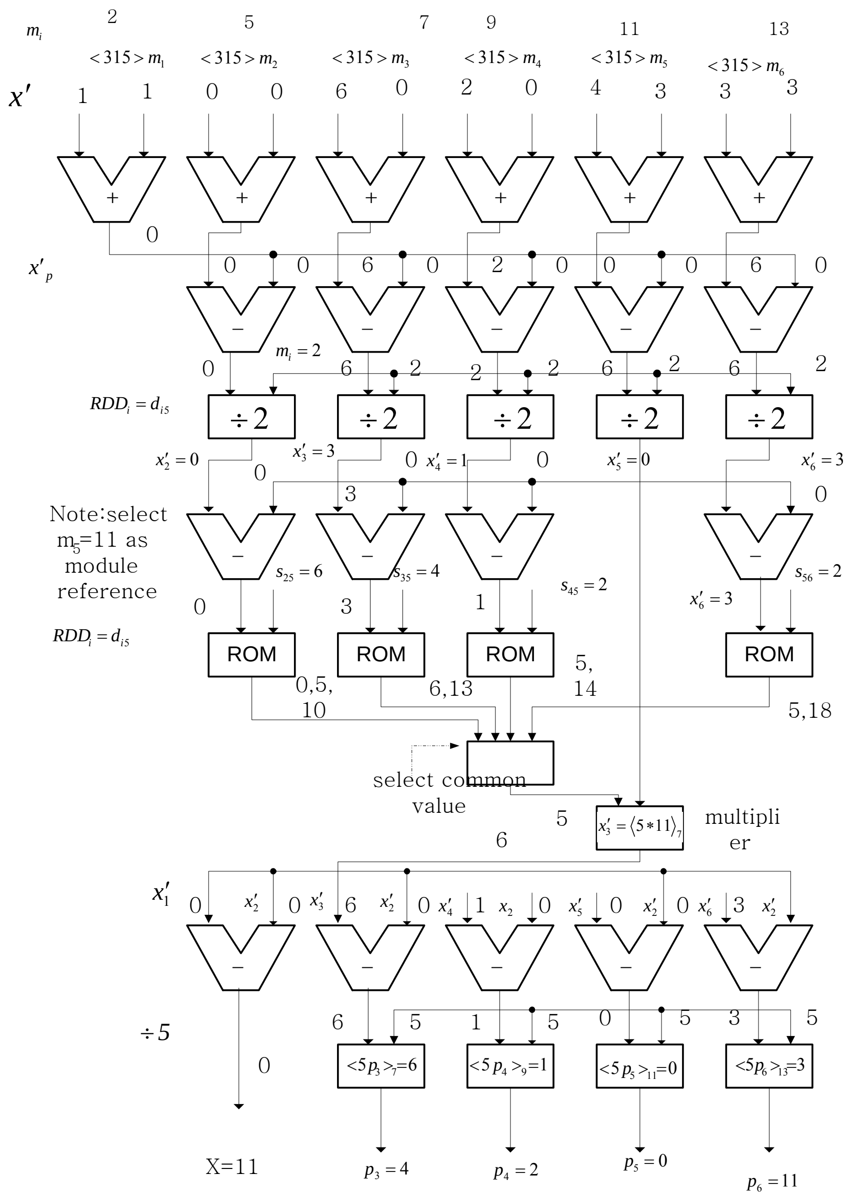


**Figure 4. Hardware structure of the residue scaling number for Example 4-4.**

$d_{25}=0,|S_{25}|=6,\langle 6\cdot k_{25}\rangle_5=0,k_{25}=0,5;$

$d_{35}=3,|S_{35}|=4,\langle 4\cdot k_{35}\rangle_7=3,k_{35}=6,13;$

$d_{45}=1,|S_{45}|=2,\langle 2\cdot k_{45}\rangle_9=1,k_{45}=5,14;$

$\langle d_{56}\rangle_{13}=\langle -3\rangle_{13}=10,|S_{56}|=2,\langle 2\cdot k_{56}\rangle_{13}=10,k_{56}=5,18;$

the correct $k_{i,5}=5$ $\therefore$ $\mathrm{RDD}=11\cdot 5=55$, and $\langle k'_{35}\rangle_7=\langle 55\rangle_7=6.$

This shows $k_{25}=k_{45}=k_{56}(=5)k_{35}$. Therefore the error correction is made by $d_{35}=\langle 4\cdot 5\rangle_7=6,$ and $X_3=d_{35}+X_5=6+0=6$, which corresponds to the value in Example 4-4, in scaling factor $k=10$, (dividing by "5" part).

From above results, this checks that scaling $\frac{x}{k}=11-\frac{315}{10}=-20.5$ which is within the accuracy of the residue scaling factor.

In a general case, $x_i \neq x_j \neq 0$, this time we must modify the subtraction of $x_i$ and $x_j$ from the $X$, before the process of the scaling. If $k=m_i\cdot m_j$ is the scaling factor, then the subtraction must change to $X^{\Delta}=X-\left(x_i+x_j{}^{\Delta}\right)$, where $x_j{}^{\Delta}=m_i k_i$ so that $x_j=x_i+\langle m_i k_i\rangle_{m_j}$ or $\langle m_i k_i\rangle_{m_j}=x_j-x_i$. Let us consider the following example:

Example 4-8

$X=\{1,0,2,0,3,5\}=135$ of moduli set

$\{2,5,7,9,11,13\}$. The scaling factor $K = m_1 \cdot m_3 = 2 \cdot 7 = 14$. is assumed. Then, residue $x_j^{\Delta} = m_1 k_i = 2k_i$, and $k_i$ can be found from $\langle 2k_i \rangle_7 = x_1 - x_3 = (2-1) = 1$. $\left(k_i < \dfrac{M_r}{m_i m_j}\right) k_i = 4$. Thus $x^{\Delta}_3 = 2 \cdot 4 = 8$, and $\left(x_i + x^{\Delta}_j\right) = 1 + 8 = 9$.

Alternatively, it could be from other module $m_3$, $x^{\Delta}_j = \langle 7k_j \rangle_2$, where $\langle 7k_j \rangle_2 = x_1 - x_3 = \langle 1-2 \rangle_2 = \langle -1 \rangle_2 = 1$, and $7k_j = 7 \therefore \left(x_i + x_j^{\Delta}\right) = (2+7) = 9$ which has the same number to be subtracted.

$m_1 m_2 m_3 m_4 m_5 m_6$

$m_i \quad 2 \quad 5 \quad 7 \quad 9 \quad 11 \quad 13$

$x_i \quad 1 \quad 0 \quad 2 \quad 0 \quad 35 = 135$

$x_i - 9 = \langle -8 \rangle_2 \langle -9 \rangle_5 \langle -7 \rangle_7 \langle -9 \rangle_9 \langle -6 \rangle_{11} \langle -4 \rangle_{13}$

$0 \quad 1 \quad 0 \quad 0 \quad 5 \quad 9 \quad 135 - 9 = 126$

From CPRDD algorithm, the scaling processes are performed as before, we then have the following results by scaling factor $K = m_1 \times m_3 = 2 \times 7 = 14$;

$\langle 14P_2 \rangle_5 = 1, P_2 = 4, 9, \cdots;$

$\langle 14P_3 \rangle_7 = 0, P_3 = 0, 1, 2, \cdots, 9, \cdots;$

$\langle 14P_4 \rangle_9 = 0, P_4 = 9, 18, \cdots;$

$\langle 14P_5 \rangle_{11} = 5, P_5 = 9, \cdots;$

$\langle 14P_6 \rangle_{13} = 9, P_6 = 9, \cdots$

Thus $X = 9$, which is exactly the value $\dfrac{126}{14} = 9$ and is the most closed to $\left\lfloor \dfrac{135}{14} \right\rfloor = 9$.

This result can be checked using sequential steps as follows:

For $x_i - 1, i = 1, 2, 3, \cdots, 6;$

$m_i = (2 \quad 5 \quad 7 \quad 9 \quad 11 \quad 13)$

$x_i = (1 \quad 0 \quad 2 \quad 0 \quad 3 \quad 5)$

$x_i - 1 = (0 \quad -1 \quad 1 \quad -1 \quad 2 \quad 4)$

Divided by 2:

$\langle 2k_2 \rangle_5 = -1, k_2 = -3;$

$\langle 2k_3 \rangle_7 = 1, k_3 = 4;$

$\langle 2k_4 \rangle_9 = -1, k_4 = -5;$

$\langle 2k_5 \rangle_{11} = 2, k_5 = 1;$

$\langle 2k_6 \rangle_{13} = 4, k_6 = 2.$

$m_i (5 \quad 7 \quad 9 \quad 11 \quad 13)$

$(k_2 k_3 k_4 k_5 k_6)$

$k_i = (-3 \quad 4 \quad -5 \quad 1 \quad 2)$

$k_i - 4 = \left(\langle -7 \rangle_5 \langle 0 \rangle_7 \langle -9 \rangle_9 \langle -3 \rangle_{11} \langle -2 \rangle_{13}\right)$

$30 \quad 0 \quad 811$

Divided by 7:

$\langle 7q_2 \rangle_5 = \langle -2 \rangle_5 = 3, q_2 = 9;$

$\langle 7q_3 \rangle_7 = 0, q_3 = 9;$

$\langle 7q_4 \rangle_9 = 0, q_4 = 9;$

$\langle 7q_5 \rangle_{11} = \langle -3 \rangle_{11} = 8, q_5 = 9;$

$\langle 7q_6 \rangle_{13} = 11, q_6 = 9.$

This result of $q = \left\lfloor \dfrac{135}{14} \right\rfloor = 9$ shows that the CPRDD algorithm has the capability of parallel processing operations in residue scaling and error corrections, *i.e.*, any combination moduli scaling factors for Ks of moduli set $\{m_1, m_2, \cdots, m_k\}$ can be performed simultaneously.

## 5. Conclusions

The arithmetic operations in the residue number system for addition, subtraction, and multiplication can be speeded up by using its parallel processing properties. However, some difficult operations, such as error detection and correction, must go through conversion or decoding processes from the residue representation to the regional binary number x. This is because the decoding technique is usually accomplished using the mixed-radix digit (MRD) or Chinese Remained Theorem (CRT), which are time consuming processes requiring hardware complexity. We proposed two algorithms for scaling and error correction without the need for lookup tables or increasing the encoding process.

The Cyclic property of the Residue-Digit Difference (CPRDD) algorithm can detect and correct errors from the RNS cyclic property. Any residue moduli set has a specific cycle length, which can be obtained from the individual residue number, difference, each pair, to a reference memory module $m_i$. Once the cyclic length is known, then the original value x is easily found, and in turn, the errors can be detected and corrected.

The TRD (Target Race Distance) algorithm combined with CPRDD is used for scaling and for error detection and correction. The scaling results and error correction can be directly performed by these two algorithms without using MRD or CRT. Thus, the decoding process is significantly reduced, and the hardware structure is greatly simplified. Several examples are illustrated and verified for these two algorithms.

## REFERENCES

[1] R. W. Watson, "Error Detection and Correction and Other Residue-Interacting Operations in a Redundant Residue Number System," University of California, Berkeley,

1965.

[2] R. W. Watson and C. W. Hastings, "Self-Checked Computation Using Residue Arithmetic," *Proceedings of the IEEE*, Vol. 54, No. 12, 1966, pp. 1920-1931. http://dx.doi.org/10.1109/PROC.1966.5275

[3] S. S. S. Yau and Y. C. Liu, "Error Correction in Redundant Residue Number Systems," *IEEE Transactions on Computers*, Vol. C-22, No. 1, 1973, pp. 5-11. http://dx.doi.org/10.1109/T-C.1973.223594

[4] D. Mandelbaum, "Error Correction in Residue Arithmetic," *IEEE Transactions on Computers*, Vol. C-21, No. 6, 1972, pp. 538-545.

[5] F. Barsi and P. Maestrini, "Error Correcting Properties of Redundant Residue Number Systems," *IEEE Transactions on Computers*, Vol. 22, No. 3, 1973, pp. 307-315. http://dx.doi.org/10.1109/T-C.1973.223711

[6] F. Barsi and P. Maestrini, "Error Detection and Correction by Product Codes in Residue Number Systems," *IEEE Transactions on Computers*, Vol. 23, No, 9, 1974, pp. 915-924. http://dx.doi.org/10.1109/T-C.1974.224055

[7] V. Ramachandran, "Single Residue Error Correction in Residue Number Systems," *IEEE Transactions on Computers*, Vol. C-32, No. 5, 1983, pp. 504-507. http://dx.doi.org/10.1109/TC.1983.1676264

[8] W. K. Lenkins and E. J. Altman, "Self-Checking Properties of Residue Number Error Checkers Based on Mixed Radix Conversion," *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 2, 1988, pp. 159-167. http://dx.doi.org/10.1109/31.1717

[9] W. K. Lenkins, "Residue Number System Error Checking Using Expanded Projection," *Electronics Letters*, Vol. 18, No. 21, 1982, pp. 927-928. http://dx.doi.org/10.1049/el:19820632

[10] W. K. Lenkins, "The Design of Error Checkers for Self-Checking Residue Number Arithmetic," *IEEE Transactions on Computers*, Vol. C-32, No. 4, 1983, pp. 388-396. http://dx.doi.org/10.1109/TC.1983.1676240

[11] M. H. Etzel and W. K. Jenkins, "Redundant Residue Number Systems for Error Detection and Correction in Digital Filters," *IEEE Transactions on Acoustics Speech and Signal Processing*, Vol. 28, No. 10, 1980, pp. 588-544.

[12] C. C. Su and H. Y. Lo, "An Algorithm for Scaling and Single Residue Error Correction in Residue Number Systems," *IEEE Transactions on Computers*, Vol. 39, No. 8, 1990, pp.1053-1064. http://dx.doi.org/10.1109/12.57044

[13] H. Y. Lo, "An Optimal Matched and Parallel Mixed-Radix Converter," *Journal of Information Science and Engineering*, Vol. 10, 1994, pp. 411-421.

[14] A. P. Shenoy and R. Kumaresan, "Fast Base Extension Using a Redundant Modus in RNS," *IEEE Transactions on Computers*, Vol. 38, No. 2, 1989, pp. 152-161. http://dx.doi.org/10.1109/12.16508

[15] E. D. Diclaudis, G. Orlandi and F, Piazza, "A Systolic Redundant Residue Arithmetic Error Correction Circuit," *IEEE Transactions on Computers*, Vol. 42, No. 4, 1993, pp. 427-433. http://dx.doi.org/10.1109/12.214689

[16] S. S. Wang and M. Y. Shau, "Single Residue Error correction Based on K-Term $M_j$-Projection," *IEEE Transactions on Computers*, Vol. 44, No. 1, 1995, pp. 129-131. http://dx.doi.org/10.1109/12.368003

[17] R. S. Katti, "A New Residue Arithmetic Error Correction Scheme," *IEEE Transactions on Computers*, Vol. 45, No. 1, 1996, pp. 13-19. http://dx.doi.org/10.1109/12.481482

 