REPRESENTATION AND PROCESSING OF FRACTIONS IN A RESIDUE SYSTEM

Dilip K. Banerji* and Saroj Kaushik**

* School of Computer & Systems Sciences, JNU, N. Delhi-110067

**Centre for Computer Science & Engineering, IIT, N. Delhi-110016

ABSTRACT

This paper proposes a scheme for the representation and processing of fractions in a residue system. The scheme is based on a mixed radix representation of a fraction in a residue system. The algorithms for basic arithmetic operations of addition, subtraction, and multiplication involving fractions are developed and are shown to provide some improvement over an existing method. Application of these algorithms to division of two integers in the residue system has been shown.

INTRODUCTION

Residue Number Systems are not new to mathematicians as envidenced by the fact that one of the earliest known results in this area, the Chinese Remainder Theorem was stated as far back as the first century A.D. The Residue Number System is well-suited for the representation and processing of integers as far as the operations of addition, subtraction and multiplication are concerned, due to its carry free characteristic. But it might be desirable for the machine to be able to handle fractions as well. Operations with fractions have been described before by Svoboda [5]. In this paper, we propose a scheme for the representation of fractions in a residue system and, based on this, we define algorithms for basic arithmetic operations involving fractions. The advantages of the proposed algorithms over Svoboda's algorithms have been discussed.

RESIDUE CODES

Consider a residue system consisting of an ordered set of moduli which are pairwise relatively prime as

$\mathcal{M}=\{m_1, m_2, \ldots, m_n\}$, $m_i \geq 2$, for $1 \leq i \leq n$.

Let $M = \prod_{i=1}^{n} m_i$.

The residue representation of an integer X, $0 \leq X < M$, is defined as $X \leftrightarrow (x_1, x_2, \ldots, x_n)$, where x_i is the least non-negative remainder obtained after dividing X by m_i. The least non-negative remainder is denoted as $|X|_{m_i}=x_i$. This representation is unique [6]

We represent a fraction f in the Residue System as $(. r_{-1}, r_{-2}, \ldots, r_{-n})$, where

$$f = \frac{r_{-1}}{m_1} + \frac{r_{-2}}{m_1 m_2} + \ldots + \frac{r_{-n}}{m_1 m_2 \ldots m_n} \qquad (1)$$

and $0 \leq r_{-i} < m_i$, $1 \leq i \leq n$.

This, in fact, is a mixed-radix representation of a fraction which is a generalization of the representation of a fraction in a fixed-radix system.

CONVERSION SCHEMES

In this section, we propose schemes which will convert a given fraction in base b to its equivalent representation in a residue system and vice-versa.

INPUT CONVERSION

The problem of converting a base b fraction f to the representation $(. r_1, r_2, \ldots, r_{-n})$ in the residue system is called 'Input Conversion'. From relation (1), we obtain

$$fm_1 = r_{-1} + \frac{r_{-2}}{m_2} + \text{---} + \frac{r_{-n}}{m_2 m_3 \text{---} m_n} \qquad (2)$$

Let $fm_1 = I_1 + f_1$, where I_1 and f_1 denote the integral and fractional parts of fm_1, respectively. We now show that

$$I_1 = r_{-1} \text{ and } f_1 = \frac{r_{-2}}{m_2} + \text{---} + \frac{r_{-n}}{m_2 \text{---} m_n}.$$

Consider $F = \frac{r_{-2}}{m_2} + \text{---} + \frac{r}{m_2 m_3 \text{---} m_n}$,

$$= \frac{r_{-2} \prod_{i=3}^{n} m_i + r_{-3} \prod_{i=4}^{n} m_i + \text{---} + r_{-n}}{\prod_{i=2}^{n} m_i}$$

or

$$F \leq \frac{(m_2-1) \prod_{i=3}^{n} m_i + (m_3-1) \prod_{i=4}^{n} m_i + \ldots + (m_n-1)}{\prod_{i=2}^{n} m_i},$$

or $$F \leq \frac{\prod_{i=2}^{n} m_i - 1}{\prod_{i=2}^{n} m_i} < 1.$$

Hence $F < 1$. This means that F represents the fractional part and r_{-1} represents the integral part of fm_1, i.e., $I_1 = r_{-1}$ and $f_1 = F$. Therefore, from relation (2), r_{-1} can be obtained by taking the integral part of fm_1. Similarly r_{-2} can be computed by taking the integral part of the product $f_1 m_2$, and so on. In general, the algorithm, which we refer to as the 'Input Conversion Algorithm' is derived as follows:

Step 1 : Initialize $R_o = f$, $r_o = 0$ and $i = 1$.

Step 2 : Compute $R_{-i} = (R_{-i+1} - r_{-i+1})\, m_i$

Step 3 : Obtain $r_{-i} = \lfloor R_{-i} \rfloor$, where $\lfloor R \rfloor$ denotes the integral part of R.

Step 4 : $i \leftarrow i+1$. If $i \leq n$ then, go to Step 2, else Exit.

Therefore, a given base b fraction denoted by $\langle f \rangle_b$ is converted by the 'Input Conversion Algorithm' into its fractional representation in the residue system. The latter representation is denoted by $\langle f \rangle_{\mathcal{M}}$ i.e.,

$$\langle f \rangle_{\mathcal{M}} \longrightarrow (.r_{-1}, r_{-2}, \cdots, r_{-n})$$

such that $\langle f \rangle_{\mathcal{M}} \leq \langle f \rangle_b$.

Lemma : For a given $\langle f \rangle_b$, the fractional representation computed by the Input Conversion Algorithm is unique. The proof of this is fairly straightforward.

Implementation of Input Conversion Algorithm:

The implementation of the proposed algorithm is fairly straightforward using residue arithmetic operations. Consider $\langle f \rangle_b = .a_{-1}\, a_{-2} \cdots a_{-j}$, where a_{-i}, $1 \leq i \leq j$, are the base b digits of f. Let us ignore the radix point and treat $\langle a_{-1} a_{-2} \cdots a_{-j} \rangle_b$ as an integer. Let $I_f = \langle a_{-1}\, a_{-2} \cdots a_{-j} \rangle_b$.
If we assume the condition that $b^j m_k < M$, where m_k is the largest modulus, then the product $I_f m_1$ can be obtained by residue multiplication. If this product is converted back to its base b representation by using the method suggested in [1,2], then the rightmost j digits correspond to the fractional part f_1 of the product fm_1, and the rest correspond to the integral part I_1 (Recall $I_1 = r_{-1}$). Similarly, other r_{-i}, $2 \leq i \leq n$, are also computed. We now express this in algorithmic form.

Step 1 : Initialize $f_o = f$ and $i = 0$,

Step 2 : Convert m_{i+1} and I_{f_i} into its residue representation.

Step 3 : Multiply I_{f_i} by m_{i+1} in the residue number system.

Step 4 : Convert the residue representation of $I_{f_i} m_{i+1}$ into its base b representation.

Step 5 : The rightmost j digits correspond to the fractional part of $f_i m_{i+1}$ denoted by f_{i+1}, and the integral part is assigned to r_{-i+1}.

Step 6 : $i \leftarrow i+1$. If $i < n$, then go to step 2, else Exit.

We now consider an example to illustrate the working of the Input Conversion Algorithm.

Example 1 : Let $\mathcal{M} = \{3, 5, 7, 11\}$, b=10, and $\langle f \rangle_b = 0.51$. Then $j = 2$ satisfies the relation $b^j m_k < M$. Here $I_f = 51$. Now $I_f m_1 \leftrightarrow (10, 2, 4, 3)$ is be computed by using residue multiplication. Using the output translation method [1,2], we obtain $\langle I_f m_1 \rangle_b = 102$. Hence $\langle f_1 \rangle_b = 0.02$ and $r_{-1} = 1$. The residue code for I_{f_1} (for computing r_{-2}) is easily obtained as follows:

$I_{f_1} = I_f m_1 - 10^j r_{-1} \leftrightarrow (2, 2, 2, 2)$ in the residue system. Then $I_{f_1} m_2$ is computed by using residue multiplication, and r_{-2} is calculated in a manner similar to that for r_{-1}. Similarly, the successive r_{-i} can also be computed.

OUTPUT CONVERSION

The problem of obtaining the base b estimate $\langle f \rangle_b$ of a given residue fraction $\langle f \rangle_{\mathcal{M}}$ is referred to as Output Conversion. Consider the relation,

$$\langle f \rangle_{\mathcal{M}} = \frac{r_{-1}}{m_1} + \frac{r_{-2}}{m_1 m_2} + \cdots + \frac{r_{-n}}{m_1 m_2 \cdots m_n},$$

It can be expressed as

$$\langle f \rangle_{\mathcal{M}} = \frac{1}{m_1}\left(r_{-1} + \cdots + \frac{1}{m_{n-1}}\left(r_{-n+1} + \frac{r_{-n}}{m_n}\right)\right)\cdots) \quad \text{---(3)}$$

Hence $$\langle f \rangle_{\mathcal{M}} = \frac{\frac{1}{m_1}\left(r_{-1} b^j + \cdots + \frac{1}{m_{n-1}}\left(r_{-n+1} b^j + \frac{r_{-n} b^j}{m_n}\right)\right)\cdots)}{b^j},$$

where j is the number of base b digits in the fraction f. Clearly, the value of j is limited by the requirement $b^j m_k < M$, where m_k is the largest modulus. We approximate $\langle f \rangle_b$ as

$$\langle f \rangle_b = \frac{\left[\frac{1}{m_1}\left(r_{-1} b^j + \cdots + \left[\frac{1}{m_{n-1}}\left(r_{-n+1} b^j + \left[\frac{r_{-n} b^j}{m_n}\right]\right)\right]\cdots\right)\right]}{b^j} \quad \text{---(4)}$$

where $[N]$ denotes the rounded value of N.

We now represent a fraction f as $\langle f \rangle_{\mathcal{M}} \leftrightarrow (.r_{-1}, r_{-2}, \text{---}. r_{-n})$.

Implementation of Output Conversion Method

The output conversion method can be implemented by using residue operations only. Initially, it is assumed that the residue representations of $\left\lfloor \frac{m_i}{2} \right\rfloor$*, $1 \leq i \leq n$ are available in the machine. The value of $\frac{X}{m}$ is rounded to the nearest integer in the following way:

$$\left[\frac{X}{m}\right] = \left\lfloor \frac{X + \left\lfloor \frac{m}{2} \right\rfloor}{m} \right\rfloor = \left\lfloor \frac{Y}{m} \right\rfloor.$$

That is, we add X and $\left\lfloor \frac{m}{2} \right\rfloor$, and then scale the sum Y by m. (Scaling of Y involves computing $\frac{Y - |Y|_m}{m}$, which is the largest integer $\leq \frac{Y}{m}$).

We now suggest the algorithm which can be implemented using residue arithmetic operations only. The value of j has been decided by the relation $b^j m_k < M$, where m_k is the largest modulus.

Step 1 : Form the residue representations of b^j and r_{-i}, $1 \leq i \leq n$.

Step 2: Set i = -n and T = 0.

Step 3 : Multiply r_i and b^j in residue form and form $X = r_i b^j + T$

Step 4 : Compute $T = \left[\frac{X}{m_{-i}}\right]$.

Step 5 : i ← i+1. If $i \leq -1$, then go to step 3.

Step 6 : $\langle f \rangle_b = .T$ and exit.

The following example illustrates the use of this algorithm.

Example 2 : Let $\mathcal{M} = \{10, 11, 13, 17\}$, b = 10 and $\langle f \rangle_{\mathcal{M}} \leftrightarrow (.8, 2, 9, 12)$.

In this case, the minimum value of j is j = 3. We now compute $r_{-4}b^3$ in the residue system.

Moduli		10	11	13	17
$r_{-4} = 12$	↔	(2	1	12	12)
10^3	↔	(0	10	12	14)
$X_1 = r_{-4}10^3$	↔	(0	10	1	15)

It is assumed that the residue representations of $\left\lfloor \frac{m_i}{2} \right\rfloor$, $1 \leq i \leq n$ are available. Therefore,

$\left\lfloor \frac{m_1}{2} \right\rfloor = 5 \leftrightarrow (5, 5, 5, 5)$,

$\left\lfloor \frac{m_2}{2} \right\rfloor = 5 \leftrightarrow (5, 5, 5, 5)$,

$\left\lfloor \frac{m_3}{2} \right\rfloor = 6 \leftrightarrow (6, 6, 6, 6)$,

and

$\left\lfloor \frac{m_4}{2} \right\rfloor = 8 \leftrightarrow (8, 8, 8, 8)$.

Now

$$\left[\frac{X_1}{m_4}\right] = \left\lfloor \frac{X_1 + \left\lfloor \frac{m_4}{2} \right\rfloor}{m_4} \right\rfloor = \left\lfloor \frac{Z_1}{m_4} \right\rfloor$$

$Z_1 \leftrightarrow (8, 7, 9, 6)$ is obtained by residue addition. Further, Z_1 is scaled by m_4. Scaling of any X by m has been shown in [6] Let

$$T = \left[\frac{X_1}{m_4}\right] = \left\lfloor \frac{Z_1}{m_4} \right\rfloor .$$

Therefore,

$T \leftrightarrow (5, 1, 3, 8)$.

Compute $r_{-3}10^3 \leftrightarrow (0, 2, 4, 7)$ by using residue multiplication. Add $r_{-3}10^3$ and T to form X_2 as

$X_2 = r_{-3}10^3 + T \leftrightarrow (5, 3, 7, 15)$,

Compute $T = \left[\frac{X_2}{m3}\right] \leftrightarrow (7, 10, 6, 16)$.

Now

$r_{-2}10^3 \leftrightarrow (0, 9, 11, 11)$.

Then

$X_3 = r_{-2}10^3 + T \leftrightarrow (7, 8, 4, 10)$.

Compute

$T = \left[\frac{X_3}{m_2}\right] \leftrightarrow (0, 8, 3, 12)$.

Finally,

$X_4 = r_{-1}10^3 + T \leftrightarrow (0, 0, 8, 5)$

and

$T = \left[\frac{X_4}{m_1}\right] \leftrightarrow (5, 0, 6, 9)$.

Therefore,

$\langle f \rangle_b = .T$.

Using the output translation algorithm, we obtain $(5, 0, 6, 9) \leftrightarrow \langle 825 \rangle_{10}$.

Therefore,

$\langle f \rangle_b = 0.825$

(Actual value $\simeq .825$).

ARITHMETIC OPERATIONS

We now define algorithms for the basic arithmetic operations involving fractions. Let f_1 and f_2 be two fractions represented as

$\langle f_1 \rangle_{\mathcal{M}} \leftrightarrow (.r'_{-1}, r'_{-2}, \text{---}, r'_{-n})$

and $\langle f_2 \rangle_{\mathcal{M}} \leftrightarrow (.r''_{-1}, r''_{-2}, \text{---}, r''_{-n})$.

Denote

$\langle f_1 \rangle_{\mathcal{M}}$ by f_1 and $\langle f_2 \rangle_{\mathcal{M}}$ by f_2.

ADDITION

Let $f = f_1 + f_2$.

*$\lfloor I \rfloor$ denotes the largest integer $\leq$ I.

Then $f \longleftrightarrow (.r_{-1}, r_{-2}, \text{---}, r_{-n})$ is defined as follows:

$$r_{-j} = |r'_{-j} + r''_{-j} + c_{-j}|_{m_j}, \quad 1 \leq j \leq n,$$

where $$c_{-j+1} = \begin{cases} 1 & \text{iff } r'_{-j} + r''_{-j} + c_{-j} \geq m_j \\ 0 & \text{otherwise} \end{cases} \quad \text{----(5)}$$

and $c_{-n} = 0$. Here c_{-i} is the carry into the ith position. It is clear that if $c_o = 1$, then the sum exceeds 1.

We now suggest an implementation of fractional addition by using residue addition. Let

$$f'_1 \longleftrightarrow (r'_{-1}, r'_{-2}, \text{---}, r'_{-n})$$

and

$$f'_2 \longleftrightarrow (r''_{-1}, r''_{-2}, \text{---}, r''_{-n})$$

denote two integers obtained by ignoring the radix points from the representations of f_1 and f_2. Then,

$$f' = f'_1 + f'_2 \longleftrightarrow (r^*_{-1}, r^*_{-2}, \text{---}, r^*_{-n}),$$

where

$$r^*_{-i} = |r'_{-i} + r''_{-i}|_{m_i}, \quad 1 \leq i \leq n,$$

is obtained by simple residue addition. Let

$$c \longleftrightarrow (c_{-1}, c_{-2}, \text{---}, c_{-n})$$

be a residue number obtained by using condition (5). Here $c_{-n} = 0$. Let

$$f'' = f' + c \longleftrightarrow (r_{-1}, r_{-2}, \text{---}, r_{-n}),$$

where

$$r_{-i} = |r^*_{-i} + c_{-i}|_{m_i}, \quad 1 \leq i \leq n.$$

Then

$$f \longleftrightarrow (.r_{-1}, r_{-2}, \text{---}, r_{-n})$$

is the fractional part of the sum of f_1 and f_2. If $c_o = 1$, then the sum exceeds 1.

SUBTRACTION

The following two steps are needed for computing $f = f_1 - f_2$.

(1) Compute the complement (additive inverse) $\bar{f}_2$ of f_2.

(2) Compute $f_1 + \bar{f}_2$ using the addition algorithm suggested earlier.

We first discuss the computation of the complement $\bar{f}$ of a given fraction f. The complement is defined as 1 - f and can be computed as follows:

$$\frac{m_1-1}{m_1} + \frac{m_2-1}{m_1 m_2} + \text{---} + \frac{m_{n-1}-1}{m_1 m_2 \cdots m_{n-1}} + \frac{m_n}{m_1 m_2 \cdots m_n} = 1.$$

Therefore, for

$$f = \frac{r_{-1}}{m_1} + \frac{r_{-2}}{m_1 m_2} + \text{---} + \frac{r_{-n}}{m_1 m_2 \text{---} m_n},$$

we get

$$\bar{f} = 1 - f \quad \frac{m_1-1-r_{-1}}{m_1} + \frac{m_2-1-r_{-2}}{m_1 m_2} + \text{---} + \frac{m_n - r_{-n}}{m_1 m_2 \text{--} m_n}.$$

Hence, the fractional representation for the complement of f is

$$\bar{f} \longleftrightarrow (.m_1-1-r_{-1}, m_2-1-r_{-2}, \text{---}, m_n - r_{-n}).$$

Now,

$$f_1 + \bar{f}_2 = f_1 + (1-f_2) = 1 + (f_1 - f_2).$$

There arise two possibilities:

(1) If $f_1 > f_2$, then $f_1 - f_2$ is obtained by subtracting 1 from $f_1 + f_2$, which is equivalent to ignoring the carry from the leftmost position, when f_1 and f_2 are added. This carry is used to indicate that the difference is positive.

(2) The absence of any carry from the leftmost fractional position in computing $f_1 + f_2$ indicates that the difference is obtained as follows:

$f_1 + \bar{f}_2 = 1 + (f_1 - f_2) = 1 - \|f_1 - f_2\|$**, since $f_1 < f_2$. Now $1 - \|f_1 - f_2\|$ is the correct complement of $\|f_1 - f_2\|$. Therefore, we take the complement of $1 - \|f_1 - f_2\|$ to get the absolute value of $f_1 - f_2$.

MULTIPLICATION

For simplicity we first consider the multiplication of fractions in a system of two moduli and then extend the method to n moduli system.

The Two Moduli Residue System:

Consider $\mathcal{M} = \{m_1, m_2\}$.

Let

$$f_1 = \frac{r'_{-1}}{m_1} + \frac{r'_{-2}}{m_1 m_2}$$

and

$$f_2 = \frac{r''_{-1}}{m_1} + \frac{r''_{-2}}{m_1 m_2}$$

$$f = \frac{r_{-1}}{m_1} + \frac{r_{-2}}{m_1 m_2} \approx f_1 f_2$$ is obtained as follows:

$$f_1 f_2 = \left(\frac{r'_{-1}}{m_1} + \frac{r'_{-2}}{m_1 m_2}\right) \times \left(\frac{r''_{-1}}{m_1} + \frac{r''_{-2}}{m_1 m_2}\right),$$

$$= \frac{r'_{-1}\, r''_{-1}}{m_1^2} + \frac{r'_{-1}\, r''_{-2} + r'_{-2}\, r''_{-1}}{m_1^2 m_2} + \frac{r'_{-2}\, r''_{-2}}{m_1^2\, m_2^2}.$$

Since $\frac{r'_{-1}\, r''_{-2}}{m_1^2\, m_2^2}$ is very small, we neglect it from the product. Therefore,

**$\|X\|$ represents the absolute value of X.

$$f_1 f_2 \approx \frac{r'_{-1} r''_{-1}}{m_1^2} + \frac{r'_{-1} r''_{-2} + r'_{-2} r''_{-1}}{m_1^2 m_2},$$

$$= \frac{\frac{r'_{-1} r''_{-1}}{m_1}}{m_1} + \frac{\frac{r'_{-1} r''_{-2}}{m_1} + \frac{r'_{-2} r''_{-1}}{m_1}}{m_1 m_2}.$$

Now r_{-1} and r_{-2} of f are computed by the following algorithm:

Step 1 Express

$$Z_1 = r'_{-1} r''_{-1} \text{ as } Z_1 = k_1 m_1 + y_1,$$

$$Z_2 = r'_{-1} r''_{-2} \text{ as } Z_2 = k_2 m_1 + y_2$$

and

$$Z_3 = r'_{-2} r''_{-1} \text{ as } Z_3 = k_3 m_1 + y_3,$$

for

$$0 \leq y_1, y_2, y_3 < m_1.$$

Step 2 Set $r_{-2} = |k_2 + k_3 + y_1|_{m_2}$.

Step 3 Compute $\varepsilon = \dfrac{k_2 + k_3 + y_1 - r_{-2}}{m_2}$

Step 4 Set $r_{-1} = k_1 + \varepsilon$ and Exit.

Analysis of the algorithm shows that the maximum absolute error in computing f is given by

$$e_{max} \leq \left(\frac{1}{m_1} - \frac{1}{m_2}\right) + \frac{2}{m_1 m_2} + \frac{1}{m_1^2}.$$

The amount of error can be reduced by slightly modifying the multiplication algorithm. If we use y_2 and y_3 from Step 1 to obtain

$$y_2 + y_3 = k_4 m_1 + y_4, \quad 0 \leq y_4 < m_1,$$

then set

$$r_{-2} = |k_2 + k_3 + k_4 + y_1|_{m_2}$$

and

$$\varepsilon = \frac{k_2 + k_3 + k_4 + y_1 - r_{-2}}{m_2}.$$

The maximum absolute error is now given by

$$e_{max} \leq \left(\frac{1}{m_1} - \frac{1}{m_2}\right) + \frac{1}{m_1 m_2} + \frac{1}{m_1^2}.$$

Therefore, for a given system $\{m_1, m_2\}$, the upper bound for the error is fixed. Hence the maximum relative error decreases as the product $f_1 f_2$ approaches 1.

Multiplication in the n Moduli System:

It becomes fairly difficult to define a general algorithm for multiplication in the n moduli system. Therefore, we first map the n moduli system to an equivalent two moduli system. The fractions are then multiplied in this two moduli system using the previous algorithm and the result is converted back to the n moduli system. The conversion relations are obtained as follows:

Let the equivalent two moduli system consists of m'_1 and m'_2, where

$$m'_1 = \prod_{j=1}^{i} m_j \quad \text{and} \quad m'_2 = \prod_{j=i+1}^{n} m_j.$$

Given a fraction f, we can express it in the n moduli system as

$$f = \frac{r_{-1}}{m_1} + \frac{r_{-2}}{m_1 m_2} + \text{---} + \frac{r_{-n}}{m_1 m_2 \text{---} m_n} \qquad (6)$$

A fraction f can be expressed in two moduli systems as

$$f = \frac{r'_{-1}}{m'_1} + \frac{r'_{-2}}{m'_1 m'_2}.$$

From relation (6), we obtain

$$\frac{r'_{-1}}{m'_1} = \frac{r_{-1}}{m_1} + \frac{r_{-2}}{m_1 m_2} + \text{---} + \frac{r_{-i}}{m_1 m_2 \cdots m_i},$$

or

$$r'_{-1} = r_{-1} \prod_{j=2}^{i} m_j + \text{---} + r_{-i} \qquad (7)$$

Similarly,

$$r'_{-2} = r_{-(i+1)} \prod_{j=i+2}^{n} m_j + \text{---} + r_{-n} \qquad (8)$$

Therefore, both r'_{-1} and r'_{-2} can be computed from r_{-i}, $1 \leq i \leq n$. After multiplication, the result must be converted back to the original system. From r'_{-1} and r'_{-2}, r_{-i}, $1 \leq i \leq n$ can be computed iteratively. The following example will show the addition, subtraction, and multiplication of two fractions.

Example 3: Let $\mathcal{M} = \{10, 11, 13, 17\}$, b=10. Consider

$$f_1 \longleftrightarrow (.8, 2, 9, 10) \approx .825$$

and

$$f_2 \longleftrightarrow (6, 5, 8, 15) \approx .625$$

Now,

$$f' = f'_1 + f'_2 \longleftrightarrow (4, 7, 4, 8).$$

Compute $c \longleftrightarrow (0, 1, 1, 0)$ by using condition (5).

Here $c_o = 1$. Therefore, the sum exceeds 1.

Now $f'' = f' + c \longleftrightarrow (4, 8, 5, 8)$,

Therefore, $f \longleftrightarrow (.4, 8, 5, 8)$ is the fractional part of the sum of f_1 and f_2. By using the output conversion method,

$$\langle f \rangle_b = .477$$

Since $c_o = 1$, therefore, the final value of the sum in base b is 1.477.

Subtraction

In order to compute $f = f_1 - f_2$, we first compute $\bar{f}_2$ as

$\bar{f}_2 \longleftrightarrow (.3, 5, \bar{4}, 2)$.

We now add f_1 and $\bar{f}_2$ as

$$f_1 + \bar{f}_2 \longleftrightarrow (.1, 8, 0, 12)$$

and there is a carry from the leftmost position. Hence the difference is positive and is equivalent to .173 in decimal system.

Multiplication

Let $m_1' = m_1 m_2$, $m_2' = m_3 m_4$ and

$$f_1' = \frac{r'_{-1}}{m_1'} + \frac{r'_{-2}}{m_1' m_2'},$$

$$f_2' = \frac{r''_{-1}}{m_1'} + \frac{r''_{-2}}{m_1' m_2'}.$$

Compute the values of r'_{-1}, r'_{-2}, r''_{-1} and r''_{-2} by using relations (6) and (7) as

$$r'_{-1} = 90,$$
$$r'_{-2} = 163,$$
$$r''_{-1} = 71,$$
$$r''_{-2} = 151.$$

Therefore, the values of f_1 and f_2 have been mapped to the 2 moduli system as

$$f_1' \longleftrightarrow (90, 163)$$

and

$$f_2' \longleftrightarrow (71, 151).$$

$f = f_1' \times f_2'$ is computed by the method suggested in the previous Section. Therefore,

$$f_1' \times f_2' \approx f = \frac{r^*_{-1}}{m_1'} + \frac{r^*_{-2}}{m_1' m_2'} = \frac{59}{m_1'} + \frac{17}{m_1' m_2'}.$$

Thus, $f \longleftrightarrow (.59, 17)$ in the 2 moduli system. From the values of r^*_{-1} and r^*_{-2}, the residues r_{-1}, r_{-2}, r_{-3}, r_{-4} of f in the 4 moduli system {10, 11, 13, 17} can be computed as follows:

$$r^*_{-1} = r_{-1} m_2 + r_{-2}$$

i.e.,

$$59 = r_{-1} m_2 + r_{-2}$$

Then

$$r_{-2} = |59|_{m_2} = 4 \text{ and } r_{-1} = 5.$$

Similarly,

$$r_{-4} = |17|_{m_4} = 0 \text{ and } r_{-3} = 1.$$

Therefore, $f \longleftrightarrow (.5, 4, 1, 0) \approx \langle .5370 \rangle_{10}$. The actual result is approximately $\langle .5379 \rangle_{10}$. Hence, the relative error in computing the product is about .18 percent. The maximum absolute error in this system is .0047.

APPLICATION TO DIVISION

The algorithms suggested earlier can be used for carrying out integer division in the Residue Number Systems. The process of integer division, where the value of the quotient is computed to the nearest integer has been discussed before in [3,6]. With the division method proposed in this section, the quotient is obtained up to fractional accuracy.

Let us consider the evaluation of $d = \frac{b}{a}$, where both a and b are integers, and $a \neq 0$. For simplicity, let us assume that both a and b are positive. Let $\frac{b}{a} = b \times y$, where y denotes the reciprocal of a. Newton's method is employed to compute y and this is multiplied by b to compute d. By definition, $y = 1/a$.

Hence $\frac{1}{y} - a = 0$.

Let $f(y) = \frac{1}{y} - a$.

Substituting this in the iterative formula [4],

$$y_{i+1} = y_i - \frac{f(y_i)}{f'(y_i)},$$

we get

$$y_{i+1} = y_i - \frac{(\frac{1}{y_i} - a)}{(-\frac{1}{y_i^2})} = y_i \times (2 - a x y_i).$$

It is known [4] that the iterations will converge if and only if the starting value y_o satisfies the condition $0 < y_o < \frac{2}{a}$. This condition also ensures that $0 < a \times y_i < 2$.

Hence $2 - a \times y_i = I_i + f_i$, where $I_i = 0$ or 1 and $0 \leq f_i < 1$. Therefore, $y_i \times (2 - a \times y_i) = I_i \times y_i + f_i \times y_i$. Now $I_i \times y_i$ either equals 0 or y_i, and $f_i x y_i$ is obtained by using the fractional multiplication algorithm suggested earlier.

We now consider the selection of the initial value y_o. The following scheme yields a suitable value of y_o.

1. If $a = m_i$, $1 \leq i \leq n$ or $a = \prod m_j$, where $\prod m_j$ denotes the product of some moduli, then choose $y_o = \frac{1}{m_i}$ or $\frac{1}{\prod m_j}$, respectively. No further iterations are required and $\frac{1}{a} = y_o$.

2. If $a \neq m_i$ or $a \neq \prod m_j$, then compute the mixed radix digits of a [6]. If r_i is the most significant non-zero mixed-radix digit of a, then

choose $y_o = \frac{2}{\alpha}$, where $\alpha = m_1 m_2 \text{ --- } m_i$. Clearly, $\alpha > a$. Hence $\frac{1}{\alpha} < \frac{1}{a}$. Therefore, $a \times y_o = a \times \frac{2}{\alpha} < 2$, and this ensures convergence.

One possible method for the implementation of this scheme is to have two tables of starting values available. For a given set of moduli $\{m_1, m_2, \text{---}, m_n\}$, the first table would be an associative store. This store contains entries for all m_i, $1 \leq i \leq n$, and for all products $\prod m_j$. Associated with an entry m_i or $\prod m_j$ is the fractional representation of $\frac{1}{m_i}$ or $\frac{1}{\prod m_j}$, respectively. Such a table would have a total of $\sum_{i=1}^{n} {}^{n}C_i$ entries, where ${}^{n}C_i = \frac{n!}{i!\,(n-i)!}$.

The second table would contain the fractional representations of

$$\frac{2}{\prod_{i=1}^{j} m_i}, \quad 1 \leq j \leq n.$$

If r_k is found to be the most significant non-zero mixed radix digit of a, then k is used to address the second table. The kth location in the table would contain $\frac{2}{m_1 m_2 \text{---} m_k}$.

A simple example in a two moduli system is used to illustrate the division process.

Example 4. Let $\mathcal{M} = \{10, 11\}$, a=2, and b = 1. Then

$$d = \frac{b}{a} = \frac{1}{2}.$$

Choose y_o such that $0 < y_o < \frac{2}{a}$ i.e., $0 < y_o < 1$. Let $y_o = .2 \longleftrightarrow (.2, 0)$. Therefore, $a \times y_o = .4 \longleftrightarrow (.4, 0) < 2$, and this ensures convergence. The iterations are listed in Table 1.

Table 1
Iterations for Example 4

No. i	a x y_i	f_i	y_i x f_i	y_{i+1}
0	(.4, 0)	(.6, 0)	(.1, 2)	(.3, 2)
1	(.6, 4)	(.3, 7)	(.1, 0)	(.4, 2)
2	(.8, 4)	(.1, 7)	(.0, 6)	(.4, 8)
3	(.9, 5)	(.0, 6)	(.0, 2)	(.4, 10)
4	(.9, 9)	(.0, 2)	(.0, 0)	(.4, 10)

We see from Table 1 that $y_5 = y_4$. Hence we stop further iteration. Therefore, $\frac{1}{a} \approx y_4 \longleftrightarrow (.4, 10) \approx \langle .49 \rangle_{10}$. Hence,

$$d = \frac{1}{2} \approx \langle .49 \rangle_{10}.$$

COMPARISON

Operations with fractions have been described before by Svoboda [5]. Using his scheme, a fraction x is defined as $x = \frac{X}{M}$, where $0 \leq X < M$ and $M = \prod_{i=1}^{n} m_i$. If $(x_1, x_2, \ldots, x_n)$ is the residue code for X, then x is represented by $/x_1, x_2, \ldots, x_n/$. Svoboda's representation might be considered a 'pure residue' representation for a fraction, since the residue digits of X are used to represent x. Although the algorithms for addition, subtraction, and multiplication of fractions are fairly straightforward using Svododa's representation, his method suffers from one major disadvantage.

His multiplication algorithm requires extension of the range M to avoid the possibility of multiplicative overflow. The extended range M_{ext} must be such that $M_{ext} = MM' \geq (M-1)^2$, where M' is the factor by which the original range must be extended. Therefore, for any practical sized M, the range must be squared in order to use the multiplication algorithm. Alternatively, the computations must be carried out in relation to a reduced range M_{red} such that $M_{red} \leq \sqrt{M}$. This would, of course, limit the accuracy of the results. In comparison, our fractional multiplication algorithm does not require the range to be extended or reduced.

Another disadvantage of Svoboda's representation becomes obvious when we consider the addition of x and y.

$$x + y = \frac{X}{M} + \frac{Y}{M} = \frac{X + Y}{M}.$$

To determine whether $x + y \geq 1$ or not, we must determine if $X + Y \geq M$. This process requires at least 2n operations for n moduli, whereas, in our method $x + y \geq 1$ is indicated by the carry from the left-most fractional position after addition.

CONCLUSION

In this paper, we have proposed a scheme for representing fractions in a residue system and defined arithmetic algorithms involving fractions. Some of the algorithms are long but otherwise, straightforward. Application of these algorithms to integer division in the residue systems has been proposed. A method for choosing a trial solution which always ensures convergence in division has also been suggested. Some of advantages of our scheme over Svoboda's scheme have been discussed.

REFERENCES

1. Banerji, D.K., "Residue Arithmetic in Computer Design," Ph.D. Dissertation, Deptt. of Applied Analysis and Computer Science, University of Waterloo, Ontario, Canada, March 1971.

2. Banerji, D.K. and Brzozowski, J.A., "Translation Algorithms for Residue Number Systems", IEEE Trans. on Computers, Vol. C-21, 1281-1285, December 1972.

3. Keir, Y.A., Cheney, P.W., and Tannenbaum, M., "Division and Overflow Detection in Residue Number Systems", IRE Transactions on Electronic Computers, Vol. EC-11, pp. 501-507, August 1962.

4. Ralston, A., "A First Course in Numerical Analysis", McGraw-Hill, 1965.

5. Svoboda, A., "The Numerical System of Residual Classes (SRC)", Digital Information Processors, Walter Hoffman (Ed.), John Wiley, 1962.

6. Szabo, N.S., and Tanaka, R.I., "Residue Arithmetic and its Application to Computer Technology", McGraw-Hill, 1967.