IEE Proceedings - Computers and Digital Techniques, July 2000

RNS based logarithmic adder

A.P. Preethy and D. Radhakrishnan
Division of Computer Engineering, School of Applied Science
Nanyang Technological University, Nanyang Ave., Singapore 639798
Email: asdrkrishnan@ntu.edu.sg
Fax: (65) 792-6559

Abstract: The operation of logarithmic addition was always eclipsed with heavy volumes of look-up tables. So with a goal to reduce the ROM size, an elegant and novel technique for logarithmic addition using RNS (Residue Number System) is demonstrated in this paper. To formulate this technique, the properties of Finite Fields and Finite Rings are exploited. A multiple base logarithm has been defined first, which was then successfully used for the formulation of our proposed technique for logarithmic addition. With our approach, the ROM requirement has been reduced to a bare minimum, thereby reducing the complexity of logarithmic addition, enabling an elegant and efficient implementation.

1 Introduction

Computing techniques based on logarithmic principles can simplify multiplication, division, roots and powers. When logarithms are used, multiplication and division are reduced to addition and subtraction respectively, and powers and roots are reduced to multiplication and division respectively. In contrast, addition and subtraction involve complicated operations and require prohibitively large ROM size [1]. Hence, it is of great interest to probe into the issues related to the reduction of ROM size. It is observed that RNS is the most appropriate choice to achieve the same.

An RNS is defined by a set of relatively prime integers (moduli) $m_1, m_2, \ldots, m_r$. Each integer X in the range 0 to M-1 for $M = \Pi_{i=1}^{r} m_i$ is uniquely represented by an r tuple ($x_1, x_2, \ldots,\ x_r$), where each residue x_i = X mod m_i is defined as the least remainder when X is divided by the modulus m_i. In RNS, arithmetic operations on large integers are done by converting them into smaller residues and performing the operations independently and all in parallel, thereby speeding up the whole operation [2,3].

2 Index Representations in Finite Fields and Finite Rings

It follows from Number Theory that an algebraic field is formed with a set of elements together with two operations, + and *, satisfying certain properties [4,5]. Finite fields (Galois fields) are classified into two types: prime fields GF(p) with p integer elements, and polynomial fields GF(p^m) with p^m algebraic polynomials, where p is any prime and m is a positive integer.

In GF(p), all the integers in the range 0 to p-1 are closed under modulo addition and all the nonzero integers are closed under modulo multiplication. By virtue of the latter, each and every integer in the multiplicative group G(p) can be generated as successive powers of a certain primitive root *g*. $\forall$ *a*$\in$G(p), $\exists$ *g* and *i* such that $\left|g^{\phi(p)}\right|_p \equiv 1$ and $\left|g^i\right|_p \equiv a$, where ϕ(p) = p-1. $\phi(p)$ represents the number of non-negative integers less than *p* which are relatively prime to *p* and can be calculated using the general expression: $\phi(n) = n\Pi_{q|n}\left(1 - \frac{1}{q}\right)$, where the symbol $\Pi_{q|n}$ means the product extended over all primes *q* that divide into *n*.

In finite fields GF(p^m), the elements themselves are algebraic polynomials. When these elements are treated as integers, they do not form a field, but form a quotient ring called the ring of integers modulo p^m, denoted by Z/(p^m). These finite rings modulo p^m form two categories, namely Z/(2^m) and Z/(p^m) for even and odd values of p, and the elements in these are strictly integers.

Any integer $x\in$ [1, 2^m -1] can be uniquely coded in Z/(2^m) as a triple index code $<\alpha,\beta,\gamma>$ using the relationship $x = 2^{\alpha}\left|5^{\beta}(-1)^{\gamma}\right|_{2^m}$, where $\alpha\in\{0,1,\ldots,m-1\}$, $\beta\in\{0,1,\ldots,(2^{m-2}-1)\}$ and $\gamma\in\{0,1\}$ [6,7]. But it may be noted that the triplet codes formed for the even integers are not unique. In a similar manner, in Z/(p^m) for odd prime p, it can be seen that, $\exists$ $g\in$ Z/(p^m) such that $\left|g^{\phi(p^m)}\right|_{p^m} \equiv 1$, where ‘g’ is a primitive root of the quotient ring. So all nonzero elements of Z/(p^m) can be generated using the relationship $\left|g^{\alpha}p^{\beta}\right|_{p^m}$, where, $\alpha\in$ {0,1,…,$\phi(p^m)$-1} and $\beta\in$ {0,1,…,m-1} [8].

3 Logarithms - Definitions and Properties

Recalling from elementary algebra, the logarithm of a number *x* is defined as the exponent *y* to which a base *a* must be raised to obtain *x*. i.e., for $x = a^y$, $y = \log_a x$. The above definition can be directly applied to numerical calculations in the case of elements under GF(p), since these elements are expressed as powers of a single base *g* (primitive root), reduced to mod *p*. On the other hand, elements of the quotient ring Z/(p^m) for both odd and even values of *p*, can only be expressed as the product of the powers of two and three elements respectively, reduced to modulo p^m. Hence a single base logarithm is not sufficient in the above cases of Z/(p^m). This insufficiency brings us to the definition of a multiple base logarithm.

Definition 1: The multiple base logarithm of a number X, denoted here as $\mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X)$, is an m-tuple of exponents ($\alpha_1,\alpha_2,\dots,\alpha_m$) to which m bases $b_1,b_2,\dots,b_m$ must be raised respectively to satisfy the equality $X = b_1^{\alpha_1} b_2^{\alpha_2} \dots b_m^{\alpha_m}$, where m>1.

i.e. if $X = b_1^{\alpha_1} b_2^{\alpha_2} \dots b_m^{\alpha_m}$, then ($\alpha_1,\alpha_2,\dots,\alpha_m$) = $\mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X)$

3.1 Algebraic properties of Multiple base logarithms

All algebraic properties satisfied by normal logarithms apply to multiple base logarithms also. If any two integers $X_i\big|_{i=1,2}$ satisfy the equality $(\alpha_{i1},\alpha_{i2},\dots,\alpha_{im}) = \mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X_i)$, then the following relations hold:

(a) $\mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X_1X_2) = \mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X_1) + \mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X_2)$

(b) $\mathrm{lm}_{(b_1,b_2,\dots,b_m)}\left(\frac{X_1}{X_2}\right) = \mathrm{lm}_{(b_1,b_2,\dots.b_m)}(X_1) - \mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X_2)$

(c) $\mathrm{lm}_{(b_1,b_2,\dots,b_m)}\left(X^r\right) = r\ \mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X)$

(d) $\mathrm{lm}_{(b_1,b_2,\dots,b_m)}\left(\sqrt[r]{X}\right) = \frac{1}{r}\mathrm{lm}_{(b_1,b_2,\dots,b_m)}(X)$

The definition of multiple base logarithm can now be easily adopted for defining dual base logarithm for $Z/(p^m)$ and triple base logarithm for $Z/(2^m)$.

Definition 2: For any nonzero integer $X \in Z/(p^m)$ such that $X = \left|g^{\alpha} p^{\beta}\right|_{p^m}$, the dual base logarithm of X with respect to the bases (g, p) can be expressed as (α,β) = $\mathrm{lm}_{(g,p)}$ (X).

Definition 3: For any nonzero integer $X \in Z/(2^m)$ such that $X = 2^{\alpha}\left|5^{\beta}(-1)^{\gamma}\right|_{2^m}$, the triple base logarithm of X with respect to the bases (2,5,-1) can be expressed as (α,β,γ) = $\mathrm{lm}_{(2,5,-1)}$ (X).

4 Logarithmic Computations

In all the three cases of GF(p), Z/(p^m) and Z/(2^m) described above, multiplication of two integers is performed by adding their corresponding indices and then finding the inverse index value [7,8,9]. Thus multiplication is easily transformed into mere addition. But addition and subtraction in logarithmic domain are considered more complicated and they need heavy volumes of look up tables. One brute-force way is to perform these operations with the use of complete look-up tables [1]. However, the size $2^{2n} \times$ n bits of such a table is prohibitive for any reasonable value of n, n being the number of bits in an operand. As an alternative, the following method is commonly used in logarithmic number processors, which reduces the look-up table size to no larger than $2^n \times$ n [1].

Let $C = A \pm B$ such that $A = b^{e_a}, B = b^{e_b}$, and $C = b^{e_c}$. By taking *A* as common factor, *C* can be expressed as $C = A\left(1 \pm \frac{B}{A}\right)$. Taking logarithms on both sides,

$$e_c = \log_b \left| A\left(1 \pm \frac{B}{A}\right) \right| = \log_b A + \log_b \left| 1 \pm \frac{B}{A} \right| = \log_b A + \log_b \left| 1 \pm b^{(e_b - e_a)} \right|$$

$= e_a + e_f$, where $e_f = \log_b \left| 1 \pm b^{(e_b - e_a)} \right|$

The value of e_f should be precalculated and stored in a look-up table. Although the above method reduces the look-up table size, still, for large operand sizes the ROM size becomes prohibitively large. Hence the necessity arises to explore other techniques to reduce the size of look-up tables needed to perform logarithmic addition/subtraction.

5 Logarithmic Addition using RNS

It is observed that, by exploiting the properties of RNS, together with those of finite fields and finite rings, the look-up table size can be successfully brought down to a bare minimum. Based on this observation, a novel technique of logarithmic addition using residue number systems is proposed in this paper. As mentioned earlier, it follows from Number Theory that, in a finite field GF(p) addition is a closed operation in mod p. Based on the above property, we propose the following theorem which shows how logarithmic addition can be carried out in a finite field.

Theorem1: For any two nonzero integers $X, Y \in GF(p) \ni X = \left|g^{\alpha_x}\right|_p$, and $Y = \left|g^{\alpha_y}\right|_p$, where p is a prime, the index α of their sum reduced mod p is $\left|\alpha_x + \alpha_f\right|_{p-1}$, where $\alpha_f = \log_g \left|1 + g^{\left|\alpha_y - \alpha_x\right|_{p-1}}\right|_p$.

Proof: By applying logarithmic principles, the indices of *X* and *Y* can be written as $\alpha_x = \log_g X,$ and $\alpha_y = \log_g Y$. By the additive closure property of *GF(p),* $\left|X + Y\right|_p = \left|g^{\alpha}\right|_p$, for some value of $\alpha \in GF(p)$. Hence, $\alpha = \log_g \left|X + Y\right|_p = \log_g \left|X\left(1 + \frac{Y}{X}\right)\right|_p$ $= \log_g X + \log_g \left|1 + g^{\left|\alpha_y - \alpha_x\right|_{p-1}}\right|_p = \alpha_x + \alpha_f$, where $\alpha_f = \log_g \left|1 + g^{\left|\alpha_y - \alpha_x\right|_{p-1}}\right|_p$. But in GF(p), since $\left|g^{p-1}\right|_p \equiv 1,\ \alpha = \left|\alpha_x + \alpha_f\right|_{p-1}$. QED

Now let us consider addition under the finite rings in a similar manner. It becomes apparent that, a single index addition is not sufficient, rather an index vector addition is required in these cases. This necessitates the proposition of the following Lemma.

Lemma 1: If $(\alpha_1, \alpha_2, \ldots, \alpha_m) = \mathrm{lm}_{(b_1, b_2, \ldots, b_m)}(X)$ and $(\beta_1, \beta_2, \ldots, \beta_m) = \mathrm{lm}_{(b_1, b_2, \ldots, b_m)}(Y)$, then

$(\alpha_1, \alpha_2, \ldots, \alpha_m) + (\beta_1, \beta_2, \ldots, \beta_m) = (\alpha_1 + \beta_1,\ \alpha_2 + \beta_2,\ \ldots,\ \alpha_m + \beta_m).$

Proof:

$(\alpha_1, \alpha_2, \ldots, \alpha_m) + (\beta_1, \beta_2, \ldots, \beta_m) = \mathrm{lm}_{(b_1, b_2, \ldots, b_m)}(X) + \mathrm{lm}_{(b_1, b_2, \ldots, b_m)}(Y)$

$$=\mathrm{lm}_{(b_1,b_2,\dots,b_m)}(XY)=\mathrm{lm}_{(b_1,b_2,\dots,b_m)}\left(b_1^{\alpha_1+\beta_1}b_2^{\alpha_2+\beta_2}\dots b_m^{\alpha_m+\beta_m}\right)$$

$$=\left(\alpha_1+\beta_1,\ \ \alpha_2+\beta_2,\ \ \dots,\ \ \alpha_m+\beta_m\right) \qquad \text{QED}$$

By using the definitions given in Section 3 and also applying Lemma 1, addition in finite rings can be performed. The following Theorems 2 and 3 state the expressions for performing logarithmic addition in finite rings Z/(p^{m}) and Z/(2^{m}).

Theorem 2: For any two nonzero integers $X,Y\in Z/(p^m)\ni X=\left|g^{\alpha_x}p^{\beta_x}\right|_{p^m}$ and $Y=\left|g^{\alpha_y}p^{\beta_y}\right|_{p^m}$, the index pair (α,β) of their sum is given by

$$(\alpha,\beta)=\begin{cases}\left(\left|\alpha_x+\alpha_f\right|_{\Phi(p^m)},\left(\beta_x+\beta_f\right)\right) & \text{for } \beta_y\geq\beta_x\\ \left(\left|\alpha_y+\alpha_f\right|_{\Phi(p^m)},\left(\beta_y+\beta_f\right)\right) & \text{otherwise}\end{cases}$$

$$\text{where,}\quad (\alpha_f,\beta_f)=\mathrm{lm}_{(g,p)}\left|1+g^{\left|(-1)^s(\alpha_y-\alpha_x)\right|_{\Phi(p^m)}}p^{(-1)^s(\beta_y-\beta_x)}\right|_{p^m}\text{, with } s=0,\qquad \text{for}\,\beta_y\geq\beta_x$$
$$\text{and } s=1,\qquad \text{otherwise}$$

Proof : By the additive closure property of *Z/(p^{m})*, $\left|X+Y\right|_{p^m}\in Z/(p^m)$.

So $\left|X+Y\right|_{p^m}=\left|g^{\alpha}p^{\beta}\right|_{p^m}$ for some value of α and β.

Hence $(\alpha,\beta)=\mathrm{lm}_{(g,p)}\left|X+Y\right|_{p^m}=\mathrm{lm}_{(g,p)}\left|X\left(1+\frac{Y}{X}\right)\right|_{p^m}$

$$=\mathrm{lm}_{(g,p)}(X)+\mathrm{lm}_{(g,p)}\left|1+\frac{Y}{X}\right|_{p^m} \tag{1}$$

$$\left|\frac{Y}{X}\right|_{p^m}=\left|g^{\left|\alpha_y-\alpha_x\right|_{\Phi(p^m)}}p^{(\beta_y-\beta_x)}\right|_{p^m} \tag{2}$$

But when $\beta_y<\beta_x$, $p^{\beta_y-\beta_x}=p^{-a}$, where $a>0$. Then, $\left|\frac{Y}{X}\right|_{p^m}=\left|\frac{g^{\left|\alpha_y-\alpha_x\right|_{\Phi(p^m)}}}{p^a}\right|_{p^m}$.

Since p^a and p^m are non-relatively prime, $\left|\frac{Y}{X}\right|_{p^m}$ cannot be evaluated. So Eqn. 1 is rewritten as: $(\alpha,\beta)=\mathrm{lm}_{(g,p)}\left|Y+X\right|_{p^m}=\mathrm{lm}_{(g,p)}(Y)+\mathrm{lm}_{(g,p)}\left|1+\frac{X}{Y}\right|_{p^m}$

$$=\mathrm{lm}_{(g,p)}(Y)+\mathrm{lm}_{(g,p)}\left|1+g^{\left|\alpha_x-\alpha_y\right|_{\Phi(p^m)}}p^{(\beta_x-\beta_y)}\right|_{p^m} \qquad (3)$$

However, when $\beta_y \geq \beta_x$, Eqn. 1 becomes

$$(\alpha,\beta)\ =\mathrm{lm}_{(g,p)}(X)+\mathrm{lm}_{(g,p)}\left|1+g^{\left|\alpha_y-\alpha_x\right|_{\Phi(p^m)}}p^{(\beta_y-\beta_x)}\right|_{p^m} \qquad (4)$$

Combining Eqns. 3 and 4,

$$(\alpha,\beta)=\begin{cases}\mathrm{lm}_{(g,p)}(X)+\mathrm{lm}_{(g,p)}\left|1+g^{\left|\alpha_y-\alpha_x\right|_{\Phi(p^m)}}p^{(\beta_y-\beta_x)}\right|_{p^m}, & \text{for } \beta_y \geq \beta_x\\ \mathrm{lm}_{(g,p)}(Y)+\mathrm{lm}_{(g,p)}\left|1+g^{\left|\alpha_x-\alpha_y\right|_{\Phi(p^m)}}p^{(\beta_x-\beta_y)}\right|_{p^m}, & \text{otherwise}\end{cases}$$

Hence $(\alpha,\beta)=\begin{cases}(\alpha_x,\beta_x)+(\alpha_f,\beta_f) & \text{for } \beta_y \geq \beta_x\\ (\alpha_y,\beta_y)+(\alpha_f,\beta_f) & \text{otherwise}\end{cases}$ (5)

where, $(\alpha_f,\beta_f)=\mathrm{lm}_{(g,p)}\left|1+g^{\left|(-1)^s(\alpha_y-\alpha_x)\right|_{\Phi(p^m)}}p^{(-1)^s(\beta_y-\beta_x)}\right|_{p^m}$, with $s=0$, for $\beta_y \geq \beta_x$

and $s=1$, otherwise

Using Lemma 1, Eqn. 5 results in

$$(\alpha,\beta)=\begin{cases}\left(\left|\alpha_x+\alpha_f\right|_{\Phi(p^m)},(\beta_x+\beta_f)\right) \text{for } \beta_y \geq \beta_x\\ \left(\left|\alpha_y+\alpha_f\right|_{\Phi(p^m)},(\beta_y+\beta_f)\right) \quad \text{otherwise}\end{cases}$$

QED

In the above case, whenever β≥m, the sum is made zero [8].

Theorem3: For any two nonzero integers $X,Y\in Z/(2^m)\ni X=2^{\alpha_x}\left|5^{\beta_x}(-1)^{\gamma_x}\right|_{2^m}$ and $Y=2^{\alpha_y}\left|5^{\beta_y}(-1)^{\gamma_y}\right|_{2^m}$, the index triplet (α,β,γ) of their sum is given by,

$$(\alpha,\beta,\gamma) = \begin{cases} \left((\alpha_x + \alpha_f), \left|\beta_x + \beta_f\right|_{2^{m-2}} + \left|\gamma_x + \gamma_f\right|_2\right) & \text{for } \alpha_y \geq \alpha_x \\ \left((\alpha_y + \alpha_f), \left|\beta_y + \beta_f\right|_{2^{m-2}} + \left|\gamma_y + \gamma_f\right|_2\right) & \text{otherwise} \end{cases}$$

where ,

$$(\alpha_f, \beta_f, \gamma_f) = \mathrm{lm}_{(2,5,-1)}\left|1 + 2^{(-1)^s(\alpha_y - \alpha_x)}\left|5^{\left|(-1)^s(\beta_y - \beta_x)\right|_{2^{m-2}}}(-1)^{\left|(-1)^s(\gamma_y - \gamma_x)\right|_2}\right|_{2^m}\right|_{2^m}, \text{ with } s = 0 \text{ for } \alpha_y \geq \alpha_x$$

and $s = 1$ otherwise

Proof: By the additive closure property of $Z/(2^m)$, $\left|X + Y\right|_{2^m} \in Z/(2^m)$.

So, $\left|X + Y\right|_{2^m} = 2^\alpha\left|5^\beta(-1)^\gamma\right|_{2^m}$ for some value of α, β and γ.

Hence $(\alpha,\beta,\gamma) = \mathrm{lm}_{(2,5,-1)}\left|X + Y\right|_{2^m} = \mathrm{lm}_{(2,5,-1)}\left|X\left(1 + \frac{Y}{X}\right)\right|_{2^m}$

$$= \mathrm{lm}_{(2,5,-1)}(X) + \mathrm{lm}_{(2,5,-1)}\left|1 + \frac{Y}{X}\right|_{2^m} \qquad (6)$$

But $$\left|\frac{Y}{X}\right|_{2^m} = \left|2^{(\alpha_y - \alpha_x)}\left|5^{\left|\beta_y - \beta_x\right|_{2^{m-2}}}(-1)^{\left|\gamma_y - \gamma_x\right|_2}\right|_{2^m}\right|_{2^m} \qquad (7)$$

When $\alpha_y < \alpha_x$, $2^{\alpha_y - \alpha_x} = 2^{-b}$, where $b > 0$.

Then $\left|\frac{Y}{X}\right|_{2^m} = \left|\frac{\left|5^{\left|\beta_y - \beta_x\right|_{2^{m-2}}}(-1)^{\left|\gamma_y - \gamma_x\right|_2}\right|_{2^m}}{2^b}\right|_{2^m}$. Since 2^b and 2^m are non-relatively prime,

$\left|\frac{Y}{X}\right|_{2^m}$ cannot be evaluated. So Eqn. 6 is rewritten as:

$$(\alpha,\beta,\gamma) = \mathrm{lm}_{(2,5,-1)}\left|Y + X\right|_{2^m} = \mathrm{lm}_{(2,5,-1)}(Y) + \mathrm{lm}_{(2,5,-1)}\left|1 + \frac{X}{Y}\right|_{2^m}$$

$$= \mathrm{lm}_{(2,5,-1)}(Y) + \mathrm{lm}_{(2,5,-1)}\left|1 + 2^{(\alpha_x - \alpha_y)}\left|5^{\left|\beta_x - \beta_y\right|_{2^{m-2}}}(-1)^{\left|\gamma_x - \gamma_y\right|_2}\right|_{2^m}\right|_{2^m} \qquad (8)$$

However, when $\alpha_y \geq \alpha_x$, Eqn. 6 becomes,

$$(\alpha,\beta,\gamma) = \mathrm{lm}_{(2,5,-1)}(X) + \mathrm{lm}_{(2,5,-1)}\left|1 + 2^{(\alpha_y - \alpha_x)}\left|5^{\left|\beta_y - \beta_x\right|_{2^{m-2}}}(-1)^{\left|\gamma_y - \gamma_x\right|_2}\right|_{2^m}\right|_{2^m} \qquad (9)$$

Combining Eqns. 8 and 9,

$$(\alpha,\beta,\gamma)=\begin{cases} \mathrm{lm}_{(2,5,-1)}(X)+\mathrm{lm}_{(2,5,-1)}\left|1+2^{(\alpha_y-\alpha_x)}\left|5^{\left|\beta_y-\beta_x\right|_{2^{m-2}}}(-1)^{(\gamma_y-\gamma_x)}\right|_{2^m}\right|_{2^m}, & \text{for } \alpha_y \geq \alpha_x \\ \mathrm{lm}_{(2,5,-1)}(Y)+\mathrm{lm}_{(2,5,-1)}\left|1+2^{(\alpha_x-\alpha_y)}\left|5^{\left|\beta_x-\beta_y\right|_{2^{m-2}}}(-1)^{(\gamma_x-\gamma_y)}\right|_{2^m}\right|_{2^m}, & \text{otherwise} \end{cases}$$

Hence $$(\alpha,\beta,\gamma)=\begin{cases} (\alpha_x,\beta_x,\gamma_x)+(\alpha_f,\beta_f,\gamma_f) & \text{for } \alpha_y \geq \alpha_x \\ (\alpha_y,\beta_y,\gamma_y)+(\alpha_f,\beta_f,\gamma_f) & \text{otherwise,} \end{cases} \tag{10}$$

where,

$$(\alpha_f,\beta_f,\gamma_f)=\mathrm{lm}_{(2,5,-1)}\left|1+2^{(-1)^s(\alpha_y-\alpha_x)}\left|5^{\left|(-1)^s(\beta_y-\beta_x)\right|_{2^{m-2}}}(-1)^{\left|(-1)^s(\gamma_y-\gamma_x)\right|_2}\right|_{2^m}\right|_{2^m}, \text{ with } s=0 \text{ for } \alpha_y \geq \alpha_x$$

and $s=1$ otherwise

Using Lemma 1, Eqn. 10 results in

$$(\alpha,\beta,\gamma)=\begin{cases} \Big((\alpha_x+\alpha_f), \left|\beta_x+\beta_f\right|_{2^{m-2}}+\left|\gamma_x+\gamma_f\right|_2\Big) & \text{for } \alpha_y \geq \alpha_x \\ \Big((\alpha_y+\alpha_f), \left|\beta_y+\beta_f\right|_{2^{m-2}}+\left|\gamma_y+\gamma_f\right|_2\Big) & \text{otherwise} \end{cases}$$

QED

Whenever α = m-1, β and γ are made zero, and when α > m-1, the sum is made zero [8]. The following example illustrates how logarithmic addition is performed in finite fields and finite rings.

Example 1: This example shows the procedure for performing logarithmic addition in the three different cases of GF(p), Z/(p^m), and Z/(2^m).

Case 1: Modulus is 31 with primitive root g = 3. The index coding for the nonzero elements of GF(31) is given in Table 1(a). Let 21 and 18 be the indices of two operands X=15 and Y=4 respectively. Using Theorem 1, the index of their sum is $\alpha=\left|21+\log_3\left|1+3^{\left|18-21\right|_{30}}\right|_{31}\right|_{30}$ = 4, which corresponds to a sum of 19.

Case 2: Modulus is 3^3 with primitive root g = 2. Also, $\phi(3^3)$ = 18. The index coding for the nonzero elements of Z/(3^3) is given in Table 1(b). Let (5,1) and (2,0) be the index pairs of the operands X=15 and Y=4 respectively. Since $\beta_y < \beta_x$, s = 1 in

Theorem 2. Hence $(\alpha_f, \beta_f) = \mathrm{lm}_{(2,3)} \left| 1 + 2^{|5-2|_{18}} 3^{(1-0)} \right|_{3^3} = \mathrm{lm}_{(2,3)} 25 = (10,0)$. Using Theorem 2, the index pair corresponding to the sum of the operands is given by $(\alpha,\beta) = \left(|2+10|_{18},\ 0+0 \right) = (12,0)$. This corresponds to the number 19, thus verifying the result.

Case 3: Modulus is 2^5. Also, $2^{m-2} = 8$. An index coding for the nonzero elements of $Z/(2^5)$ is given in Table 1(c). Let (0,4,1) and (2,0,0) be the index triplets of the operands X=15 and Y=4 respectively. In Theorem 3, since $\alpha_y \geq \alpha_x$, $s = 0$. Hence, $(\alpha_f, \beta_f, \gamma_f) = \mathrm{lm}_{(2,5,-1)} \left| 1 + 2^{(2-0)} \left| 5^{|0-4|_8} (-1)^{|0-1|_2} \right|_{2^5} \right|_{2^5} = \mathrm{lm}_{(2,5,-1)} 29 = (0,3,0)$. So the sum $(\alpha,\beta,\gamma) = (0+0, |4+3|_8, |1+0|_2) = (0,7,1)$. This corresponds to the number 19.

The look-up tables showing the index coding in the cases of GF(31), $Z/(3^3)$, and $z/(2^5)$ are given in Tables 1(a), 1(b) and 1(c).

Table 1(a). Index coding for elements x of GF(31) with a primitive root of 3

x	1	2	3	4	5	6	7	8	9	10
α	0	24	1	18	20	25	28	12	2	14

x	11	12	13	14	15	16	17	18	19	20
α	23	19	11	22	21	6	7	26	4	8

x	21	22	23	24	25	26	27	28	29	30
α	29	17	27	13	10	5	3	16	9	15

Table 1(b). Index coding for elements x of $Z/(3^3)$ with a primitive root of 2

x	1	2	3	4	5	6	7	8	9	10	11	12	13
α, β	0,0	1,0	0,1	2,0	5,0	1,1	16,0	3,0	0,2	6,0	13,0	2,1	8,0

x	14	15	16	17	18	19	20	21	22	23	24	25	26
α, β	17,0	5,1	4,0	15,0	1,2	12,0	7,0	4,1	14,0	11,0	3,1	10,0	9,0

Table 1(c). Index coding for elements x of $Z/(2^5)$

x	1	2	3	4	5	6	7	8	9	10
α, β, γ	0,0,0	1,0,0	0,3,1	2,0,0	0,1,0	1,3,1	0,2,1	3,0,0	0,6,0	1,1,0

x	11	12	13	14	15	16	17	18	19	20
α, β, γ	0,5,1	2,3,1	0,7,0	1,2,1	0,4,1	4,0,0	0,4,0	1,6,0	0,7,1	2,1,0

x	21	22	23	24	25	26	27	28	29	30	31
α, β, γ	0,5,0	1,5,1	0,6,1	3,3,1	0,2,0	1,7,0	0,1,1	2,2,1	0,3,0	1,4,1	0,0,1

6 ROM Requirements

It has been shown in [10] that a 36-bit processor can be implemented using a balanced 5-bit moduli set {17,19,23,25,27,29,31,32}. Small look-up tables are used to generate the relevant logarithms. For generating α_f for each prime modulus, look-up tables of size up to 32×5 only are needed. For moduli 25 and 27, the look-up table sizes needed for generating (α_f , β_f) are 64×6 and 128×7 respectively. For the modulus 32, the corresponding look-up table size is 128×7, for generating $(\alpha_f , \beta_f , \gamma_f)$, thus requiring only a total of less than 500 bytes for the entire system.

7 Conclusions

A novel technique for logarithmic addition that finds wide applications in many fields of scientific computing, is proposed in this paper. A new multiple base logarithm has been defined which was used to formulate an algorithm for logarithmic addition in integer rings. Furthermore, by exploiting the properties of RNS and those of finite fields and finite rings, we have succeeded in reducing the ROM requirements for logarithmic addition to a bare minimum of less than 500 bytes, for a 36-bit RNS based processor using a 5-bit balanced moduli set.

8 References

1 KOREN, I.: 'Computer Arithmetic Algorithms′ (Prentice-Hall, New Jersey, 1993)

2 SZABO, N. S., and TANAKA, R. I.: 'Residue Arithmetic and its Applications to Computer Technology′ (McGraw-Hill, New York, 1967)

3 SODERSTRAND, M. A., JENKINS, W. K., JULLIEN, G. A., and TAYLOR, F. J.: 'Residue Number System Arithmetic: Modern Applications in Digital Signal Processing′ (IEEE Press, New York, 1986)

4 MCELIECE, R. J.: 'Finite Fields for Computer Scientists and Engineers′ (Kluwer Academic Publishers, Boston, 1987)

5 NIVEN, I., and ZUCKERMAN, H. S.: 'An Introduction to the Theory of Numbers' (John Wiley & Sons, New York, 1980)

6 VINOGRADOV, I. M.: 'Elements of Number Theory' (Dover Publications, New York, 1954)

7 CARDARILLI, G. C., LOJACONO, R., MARTINELLI, G., and SALERNO, M.: ‘Structurally passive digital filters in residue number systems', *IEEE Trans. Circuits Syst.*, 1988, **35**, **2**, pp. 149-158

8 RADHAKRISHNAN, D.: 'Modulo multipliers using polynomial rings', *IEE Proc. Circuits, Devices, Syst.*, 1998, **145, 6**, pp. 443-445

9 NAKL, A.: ‘Decimal Arithmetic Unit,' Stroje Na Zprocovani, Prague, CSAU, Czechoslovakia, 1962.

10 RADHAKRISHNAN, D., and PREETHY, A. P.: 'A Novel 36 bit Single Fault Tolerant Multiplier using 5 bit Moduli', Proceedings of *IEEE TENCON 98*, New Delhi, India, 1998, vol. 1, pp. 128-130