A New Class of Monotone Functions of the Residue Number System

G. Pirlo, D. Impedovo

***Abstract* —** This paper presents a new class of monotone functions that can be computed from the Residue Number System (RNS) to the integers. On the basis of these functions new implementations are proposed for residue-to-binary conversion and magnitude comparison that are superior to traditional techniques, if a modulus of the kind 2^k (k integer) is included in the set of RNS moduli.

***Keywords* —** Chinese Remainder Theorem, Magnitude Comparison, Multi-operand Modular Adder, Residue Number System, Residue-to-Binary Conversion.

I. Introduction

IN the Residue Number System (RNS), the effective implementation of non-modular operations like residue-to-binary conversion and magnitude comparison is mandatory. Residue-to-binary conversion is necessary for the use of RNS arithmetic units into general purpose computers, which are based on the binary number system. Magnitude comparison supports other logic operations that are complex in the RNS due to the difficulty in defining an order relation on quotient sets [12].

The main techniques for the implementation of non-modular operations use the Mixed-Radix Conversion (MRC) - which is strictly sequential, or the Chinese Remainder Theorem (CRT) - which is more attractive since it provides a parallel conversion formula [13].

Other techniques have been proposed which use functions defined from the RNS to the integers. The 'diagonal function' exploits the observation that the integers in residue representation dispose themselves on diagonals when they are arranged in the multi-dimensional discrete space associated to the RNS [2, 3]. Unfortunately, although the 'diagonal function' is a powerful tool for magnitude comparison, it does not support residue-to-binary conversion [2,3].

This paper introduces a new class of monotone functions that support effectively both magnitude comparison in RNS and residue-to-binary conversion. Through the paper an effective scheme for the computation of the new functions is presented and the superiority of the new implementations of magnitude comparison and residue-to-binary conversion with respect to traditional techniques is demonstrated.

The organisation of the paper is the following: Section 2 introduces the new functions and presents the scheme to compute them directly from the RNS. Section 3 shows the new implementations of residue-to-binary conversion and magnitude comparison. The comparative analysis of the performance of the new techniques and the traditional approaches is reported in Section 4. Section 5 presents a case study.

Authors are with the Dipartimento di Informatica, Università degli Studi di Bari “Aldo Moro”, via Orabona, 4, 70125 Bari, Italy (corresponding author: giuseppe.pirlo@ uniba.it).

II. Monotone Functions of the RNS

In the RNS based on the pairwise relatively prime moduli $m_1,m_2,...,m_N$, an integer $X\in[0,M-1]$ $(M=m_1\cdot m_2\cdot\ldots\cdot m_N)$ is uniquely represented by the N-tuple $(x_1,x_2,..,x_N)$, where $x_i=\left|X\right|_{mi}$ is the residue of X modulo mi , i=1,2,..,N [12]. Let be $I\subset\{1,2,..,N\}$, $I\neq\emptyset$, the function F_I proposed in this paper is:

$$F_I(X)=\sum_{i\in I}\left[\frac{X}{m_i}\right] \qquad (1)$$

where $[a]$ denotes the largest integer not exceeding a.

Theorem 1 shows an effective scheme to compute $F_I(X)$ directly from the RNS representation of X.

Theorem 1 :

Let $m_1, m_2,.., m_N$ be the set of pairwise relatively prime moduli of the RNS, let $I\subset\{1,2,..,N\}$, $I\neq\emptyset$, and

$$M_i=\frac{M}{m_i}$$

$$M_I=\sum_{i\in I}M_i$$

$$S_{INV}=\sum_{i\in I}\left|\frac{1}{m_i}\right|_{M_I}$$

where

$$\left|\frac{1}{m_i}\right|_{M_I}$$

is the multiplicative inverse of m_i modulo M_I, $i\in I$ [12]. If $(x_1,x_2,..,x_N)$ is the RNS representation of $X\in[0,M-1]$, the value $F_I(X)$ can be computed as:

$$F_I(X)=\left|\sum_{i=1}^{N} b_i \cdot x_i\right|_{M_I} \qquad (2)$$

with:

$$b_i = -\left|\frac{1}{m_i}\right|_{M_I}, i\in I$$

$$b_j = \left| M_j \cdot S_{INV} \cdot \left|\frac{1}{M_j}\right|_{m_j}\right|_{M_I}, j\in J$$
,

with

J={1,2,…,N}-I (hence *I∪J={1,2,…,N}* and *I∩J=∅*).

Proof : See Figure 1.

Moreover, the coefficients b_k, $k\in\{1,2,\ldots,N\}$ are well defined if there exist the multiplicative inverses

$$\left|\frac{1}{M_j}\right|_{m_j}, j\in J$$

and

$$\left|\frac{1}{m_i}\right|_{M_I}, i\in I$$

This is true if and only if m_j,M_j and m_i,M_I are couples of relatively prime integers [1]. This is shown in Theorem 2.

Theorem 2:

Let $m_1, m_2,.., m_N$ be the set of pairwise relatively prime moduli of the RNS and $I\subset\{1,2,..,N\}$, $I\neq\emptyset$. Let

$$M=\prod_{i=1}^{N} m_i$$

$$M_i = \frac{M}{m_i}$$
,

and

$$M_I = \sum_{i\in I} M_i$$

the following conditions are true:

(a) m_j and M_j are relatively prime;
(b) m_i and M_I are relatively prime.

Proof: See Figure 2.

Example: For the RNS of moduli m_1=37, m_2=41, m_3=43, m_4=64, if I={2,4}, it results that $M_I=M_2+M_4$=101824+65231= 167055 and b(1)=9030, b(2)=8149, b(3)=27195, b(4)=122681. Now, let X=17435→(RNS)(12,23,19,7), we have:

- from eq. 1:

$$F_I(X)= \left\lfloor\frac{X}{m_2}\right\rfloor+\left\lfloor\frac{X}{m_4}\right\rfloor = 709$$

- from eq. 2:

$$F_I(X)= F_I(X)=\left|\sum_{i=1}^{N} b_i \cdot x_i\right|_{M_I} = 709 .$$

III. Monotone Functions for Non-Modular Operations in the RNS

Let $m_1,m_2,...,m_N$ be the set of relatively prime moduli of the RNS and let be

$$M=\prod_{i=1}^{N} m_i$$

The new implementations of magnitude comparison and residue-to-binary conversion are reported in the following [4].

1) Magnitude Comparison.

Let $X,Y\in[0,M-1]$ be two integers whose RNS representation is $X\rightarrow(x_1,x_2,\ldots,x_N)$ and $Y\rightarrow(y_1,y_2,\ldots,y_N)$, respectively. From (1) we have $X<Y \Rightarrow F_I(X)<F_I(Y)$ or ($F_I(X)=F_I(Y)$ and $x_i<y_i$, $i\in I$). In fact, since

$$X=\left\lfloor\frac{X}{m_i}\right\rfloor\cdot m_i + x_i$$

$$Y=\left\lfloor\frac{Y}{m_i}\right\rfloor\cdot m_i + y_i$$

Proof of Theorem 1
From

$$x_i = X - m_i \cdot \left[\frac{X}{m_i}\right]$$

it follows that:

$$\left|\sum_{i=1}^{N} b_i \cdot x_i\right|_{M_I} = \left|\sum_{i=1}^{N} b_i \cdot \left(X - m_i \cdot \left[\frac{X}{m_i}\right]\right)\right|_{M_I} = \left|\sum_{i=1}^{N} b_i \cdot X - \sum_{i=1}^{N} b_i \cdot m_i \cdot \left[\frac{X}{m_i}\right]\right|_{M_I}$$

$$\left|-S_{INV} \cdot X + S_{INV} \cdot X \cdot \left(\sum_{j\in J}\left|\frac{1}{M_j}\right|_{m_j}\right) + \sum_{i\in I}\left[\frac{X}{m_i}\right] - \sum_{j\in J} S_{INV} \cdot X - \sum_{j\in J}\left(M_j \cdot m_j\right)\cdot S_{INV} \cdot \left|\frac{1}{M_j}\right|_{m_j} \cdot \left[\frac{X}{m_j}\right]\right|_{M_I} =$$

$$\stackrel{\text{(Appendix A,B)}}{=} \left|X \cdot S_{INV} \cdot \left(-1 + \sum_{j\in J} M_j \cdot \left|\frac{1}{M_j}\right|_{m_j}\right) + \sum_{i\in I}\left[\frac{X}{m_i}\right] - \sum_{j\in J} M \cdot S_{INV} \cdot \left|\frac{1}{M_j}\right|_{m_j} \cdot \left[\frac{X}{m_j}\right]\right|_{M_I}$$

$$\left|0 + \sum_{i\in I}\left[\frac{X}{m_i}\right] - 0\right|_{M_I} = \left|\sum_{i\in I}\left[\frac{X}{m_i}\right]\right|_{M_I} = F_I(X).$$

Q.E.D.

Figure 1. Proof of Theorem 1

Proof of Theorem 2

(a) Assuming that m_j and M_j are not relatively prime, since

$$M_j = m_1 \cdot m_2 \cdot \ldots \cdot m_{j-1} \cdot m_{j+1} \cdot \ldots \cdot m_N,$$

a modulus m_i must exist, i=1,2,..,N, i≠j, which is not relatively prime with m_j. This contradicts the hypothesis.

(b) Assuming that m_i and M_I are not relatively prime, it follows that three integers $\overline{\alpha}, \overline{\beta}, \overline{\delta}$ exist so that $m_i = \overline{\alpha} \cdot \overline{\beta}$ and $M_I = \overline{\alpha} \cdot \overline{\delta}$ (with $\overline{\alpha} \neq 1$). Now, since

$$M_I = \sum_{k\in I} M_k = \sum_{k\in I}\frac{M}{m_k} = \left(\frac{M}{m_i} + m_i \cdot \sum_{\substack{k\in I\\ k\neq i}}\frac{M_i}{m_k}\right)$$

substituting $m_i = \overline{\alpha} \cdot \overline{\beta}$ and $M_I = \overline{\alpha} \cdot \overline{\delta}$ in (3) we obtain

$$\overline{\alpha} \cdot \overline{\delta} = \frac{M}{m_i} + \overline{\alpha} \cdot \overline{\beta} \cdot \sum_{\substack{k=I\\ k\neq i}}\frac{M_i}{m_k} \quad \Rightarrow \quad \overline{\alpha} \cdot \left(\overline{\delta} - \overline{\beta} \cdot \sum_{\substack{k=I\\ k\neq i}}\frac{M_i}{m_k}\right) = \frac{M}{m_i} = \prod_{\substack{k=1\\ k\neq i}}^{N} m_k \cdot$$

Thus, a modulus m_k exists, k≠i, so that $\overline{\alpha}$ divides m_k. This means that m_k and m_i are not relatively prime moduli. This contradicts the hypothesis.

Q.E.D.

Figure 2: Proof of Theorem 2

Table I. Performance Analysis

		Time Complexity	ROM	R-to-B Conversion	Magnitude comparison
Serial Technique	**MRC** (see [13])	O(N)	Ω(N)	Y	Y
Parallel Techniques	**CRT** (see [13])	O(logN)	$\Omega(N^2)$	Y	Y
	D (see [2])	O(logN)	$\Omega(N^2)$	Not supported	Y
	$\mathbf{D_k}$ (see [3])	O(logN)	$\Omega(N^2)$	Not supported	Y
	$\mathbf{F_I}$	O(logN)	$\Omega(N^2)$	Y	Y

it follows that if $X<Y$ and $F_I(X)=F_I(Y)$ it results that $x_i<y_i$, $i\in I$. Therefore, magnitude comparison can be performed as follows:

STEP 1. *Compute* $\boldsymbol{F_I}(X)$ *and* $\boldsymbol{F_I}(Y)$

STEP 2. *Compare* $\boldsymbol{F_I}(X)$ *and* $\boldsymbol{F_I}(Y)$*; if* $\boldsymbol{F_I}(X)=\boldsymbol{F_I}(Y)$ *then compare* x_i *and* y_i.

2) ***Residue-to-binary Conversion.***

Let be I={i}, from (1) we have

$$F_I(X)=\left\lfloor \frac{X}{m_i} \right\rfloor.$$

Now, since

$$X=m_i\cdot\left\lfloor \frac{X}{m_i} \right\rfloor+x_i$$

it results:

$$X=m_i\cdot F_I(X)+x_i \tag{3}$$

If the modulus m_i is a power of 2, i.e. $m_i=2^k$ (k integer), the implementation of eq. (3) implies shift-left operation rather than ordinary multiplication and the binary representation of X is obtained by concatenating the binary representations of FI(X) (most significant bits of X) and x_i (least significant bits of X). In this case X is obtained as follows:

STEP 1. *Compute* $\boldsymbol{F_I}(X)$ *, then do* $X=\boldsymbol{F_I}(X)\cup x_i$*;*
(where $\boldsymbol{F_I}(X)\cup x_i$ is the concatenation of the binary representations of $\boldsymbol{F_I}(X)$ and x_i).

Example: For the RNS of moduli $m_1=37$, $m_2=41$, $m_3=43$, $m_4=64$, if $I=\{4\}$, it results that $M_I=M_4=65231$ and b(1)=3526, b(2)=3182, b(3)=10619, b(4)=47904. Now, let X=1119797→(RNS)(29,5,34,53) and Y=432163→(RNS)(3,23,13,35) we have:

❖ ***Magnitude Comparison.***

Since $F_I(X)=17496$ and $F_I(Y)=4714$, from $F_I(X)>F_I(Y)$ it follows that X>Y.

❖ ***Residue-to-Binary Conversion.***

Since the binary representations of $F_I(X)=17496$ and $x_4=53$ are 100010001011000 and 110101, respectively, it results that the binary representation of X=1119797 is 100010001011000∪110101=100010001011000110101. Analogously, since the binary representations of $F_I(Y)=4714$ and $y_4=35$ are 1101001100000 and 100011, respectively, it results that the binary representation of Y=432163 is 1101001100000∪100011=1101001100000100011.

IV. Performance Analysis

Let $m_1,m_2,...,m_N$ be the set of relatively prime moduli of the RNS $M=m_1\cdot m_2\cdot\ldots\cdot m_N$ and let $X\in[0,M-1]$ be an integer whose RNS representation is $X\rightarrow(x_1,x_2,\ldots,x_N)$.

❑ **Mixed Radix Conversion (MRC).**

The MRC is based on the formula [13]:

$$X=a_1+m_1a_2+m_1m_2a_3+\ldots+m_1m_2\ldots m_{N-1}a_N \tag{4}$$

where $a_1,a_2,\ldots a_N$ are the Mixed Radix digits, which can be obtained recursively:

$$a_1=x_1,\ a_2=(X-a_1)/m_1,\ \ldots. \tag{5}$$

- **Chinese Remainder Theorem (CRT).**

The CRT is based on the conversion formula [13]:

$$X = \left| \sum_{i=1}^{N} N_i \cdot x_i \right|_M \qquad (6)$$

where

$$N_i = M_i \cdot \left| \frac{1}{M_i} \right|_{m_i}$$

$$M_i = \frac{M}{m_i}$$

- **Diagonal Function (D).**

The 'diagonal function' of the RNS of moduli $m_1,m_2,...,m_N$, is defined as [2]:

$$D(X) = \left| \sum_{i=1}^{N} k_i \cdot x_i \right|_{SQ} \qquad (7)$$

where:

$$SQ = \sum_{i=1}^{N} M_i ,$$

is the 'diagonal modulus' of the RNS ($M_i=M/m_i$, i=1,2,…,N) and k_i is the multiplicative inverse of m_i modulo SQ).

- **Diagonal Function by reduction of the RNS space dimensionality (D_k).**

A more effective implementation of the 'diagonal function' can be obtained when we consider the set of moduli

$$vm_1=m_1 * m_2 * \ldots * m_{i1},\ vm_2= =m_{i1+1} * m_{i1+2} * \ldots * m_{i2},\ \ldots,\ vm_j= =m_{ij-1+1} * m_{ij-1+2} * \ldots * m_{ij},\ \ldots,\ vm_k= =m_{ik-1+1} * m_{ik-1+2} * \ldots * m_{ik}$$

(where $\forall$p,q=1,2,...,k: i_p,i_q integers; p<q $\Rightarrow$ $i_p<i_q$ and i_k=N) [3]. For the set of moduli $vm_1,vm_2,...,vm_k$, the 'diagonal function' $\mathbf{D_k}(\cdot)$ is:

$$D_k(X) = \left| \sum_{i=1}^{k} vk_i \cdot vx_i \right|_{SQ_k} \qquad (8)$$

where $(vx_1, vx_2, \ldots, vx_k)$ is the representation of X in the RNS of moduli $vm_1,vm_2,...,vm_k$, that is :

$$vx_i = |X|_{vm_i}$$

$$SQ_k = \sum_{j=1}^{k} \frac{M}{vm_j}$$

and

$$vk_j = - \left| \frac{1}{vm_j} \right|_{SQ_k} .$$

Table I compares the different techniques. The MRC is based on a strictly sequential process (eqs.(4)-(5)). It has a time delay O(N) and its ROM requirement is O(N) [13]. The implementation of eqs.(2) (6),(7),(8) have a time complexity O(logN), since the addition of N values can be performed in parallel using a tree of adders. Since the RNS moduli are pairwise relatively prime, it follows that necessarily m_i must be greater or equal than i , $\forall$i=1,2,…,N (for instance we have: $m_1 \geq 2$, $m_2 \geq 3$, $m_3 \geq 5$, $m_4 \geq 7$, and so on). Therefore, the total storage ROM is greater than $(1+2+3+\ldots+N)=\Omega(N^2)$ [5]. Moreover, the 'diagonal function' (D) and its improved implementation (D_k) do not support residue-to-binary conversion, whereas the CRT and the F_I (for I={i} and $m_i=2^k$, k integer) support both magnitude comparison and residue-to-binary conversion.

V. A Case Study

Table II compares the parallel techniques for the RNS of N=4 moduli $m_1=37,m_2=41,m_3=43,m_4=64$ [9]:

- L(l, a) denotes a look-up table of 2^l locations with a-bit word length. It has a time delay equal to t_L;

- MOMA(N, a) denotes a multi-operand modular adder for N operands with a-bit word length. It uses a tree of *Carry Save Adders* (CSA) and a *Ripple Carry Adder* (RCA) for final summation [7,9]. It requires (N+1)· a full adders (FA) and its time delay is $t_{MOMA(N,A)} = \theta(N) \cdot t_{FA} + 2t_{RCA(a)}$, where $\theta(N)$ is the minimum number of levels in the CSA tree with N operands (for the case N=4 it results that $\theta(N)=2$) [7], t_{FA} is the time delay of a FA, $t_{RCA(a)}$ is the time delay of a RCA with a-bit word length.

- C(p) denotes a binary comparator with p-bit word length. It has a time delay equal to $t_{C(p)}$.

Moreover, let Δ be the delay of a NAND gate, the following delays are assumed: $t_L=\Delta$, $t_{FA}=2\Delta$, $t_{RCA(a)}=a\cdot\Delta$, $t_{C(p)}=4\Delta$ (for $8<p\leq 64$) [9]. In Table II, as suggested in [3], the MRC is used in the Preparatory Step (PS) for computing $D_k(X)$ (in order to obtain the values $vx_1=|X|_{vm_1}$ and $vx_2=|X|_{vm_2}$). In this case

Table II: A Case Study

			CRT	D	D_k (for $vm_1=m_1*m_4=2368$, $vm_2=m_2*m_3=1763$)	F_I (for I={4})
			M=4174784 ($a=\lceil \log_2 M\rceil=22$ bit)	SQ=376975 ($a=\lceil \log_2 SQ\rceil=19$ bit)	$SQ_2=4131$ ($a=\lceil \log_2 SQ_k\rceil=13$ bit)	$M_I=65231$ ($a=\lceil \log_2 M_I\rceil=16$ bit)
Function Implementation	*PS - Delay*		-	-	t_{MRC}	-
	ROM		4 L(6,22)	4 L(6,19)	4 L(6, 13)	4 L(6,16)
	MOMA	*FA*	$(N+1)\cdot 22=5*22=110$	$(N+1)\cdot 19=5*19=90$	$(N+1)\cdot 22=5*13=65$	$(N+1)\cdot 16=80$
		Delay	$\theta(N)\cdot t_{FA}+2t_{RCA(\lceil \log 2M\rceil)}=$ $=2\cdot 2\Delta+2\cdot 22\Delta=48\Delta$	$\theta(N)\cdot t_{FA}+2t_{RCA(\lceil \log 2 SQ\rceil)}=$ $=2\cdot 2\Delta+2\cdot 19\Delta=42\Delta$	$\theta(N)\cdot t_{FA}+2t_{RCA(\lceil \log 2SQk\rceil)}=$ $=2\cdot 2\Delta+2\cdot 13\Delta=30\Delta$	$\theta(N)\cdot t_{FA}+2t_{RCA(\lceil \log 2 MI\rceil)}=$ $=2\cdot 2\Delta+2\cdot 16\Delta=36\Delta$
R-to-B Conversion	*Extra hardware*		---	**Not Supported**	**Not Supported**	---
	Delay		$t_L+t_{MOMA(N,\lceil \log 2M\rceil)}=$ $=\Delta+48\Delta=49\Delta$			$t_L+t_{MOMA(N,\lceil \log 2MI\rceil)}=$ $=\Delta+36\Delta=37\Delta$
Magnitude Comparison	*Extra hardware*		C(22)	C(19+6)	C(13+6+6)	C(16+6)
	Delay		$2*(t_L+t_{MOMA(N,\lceil \log 2M\rceil)})+t_{c(\lceil \log 2M\rceil)}=$ $=2*(\Delta+48\Delta)+4\Delta=102\Delta$	$2*(t_L+t_{MOMA(N,\lceil \log 2SQ\rceil)})+t_{c(\lceil \log 2SQ\rceil)}=$ $=2*(\Delta+42\Delta)+4\Delta=90\Delta$	$2*(t_{MRC}+t_L+t_{MOMA(N,\lceil \log 2SQk\rceil)})+t_{c(\lceil \log 2SQk\rceil)}=$ $2*(2\Delta+\Delta+30\Delta)+4\Delta=70\Delta$	$2*(t_L+t_{MOMA(N,\lceil \log 2MI\rceil)})+t_{c(\lceil \log 2MI\rceil)}=$ $=2*(\Delta+36\Delta)+4\Delta=78\Delta$

the PS has a time delay of $t_{MRC}=2\Delta$ and it does not require extra hardware.

From Table II it results that the new approach is superior to the approaches based on the 'diagonal function' since they support magnitude comparison only. The new approach is also superior to the CRT in terms of time delay (a 24% save for residue-to-binary conversion, a 23% save for magnitude comparison) and waste of hardware (a 27% save for FA and ROM).

Finally, we remark that unlike other techniques that provide approximate methods for non-modular operation [6,8], the new functions support exact methods for residue-to-binary conversion and magnitude comparison without imposing severe constraints on the set of moduli, as other approaches [10,11,14].

VI. Conclusion

This paper presents a new class of monotone functions - defined from the RNS to the integers – that support parallel implementations of residue-to-binary conversion and magnitude comparison, if a modulus of the kind 2^k (k integer) is included in the set of RNS moduli. The new implementations are superior both to the recent techniques based on the 'diagonal functions', which support magnitude comparison only, and to the Chinese Remainder Theorem (CRT), in terms of time delay and waste of hardware.

Appendix

A) Since $I\cup J=\{1,2,\ldots,N\}$, let

$$P_I=\prod_{i\in I} m_i$$

$$P_J=\prod_{j\in J} m_j$$

then

$$P_I\cdot P_J=M.$$

Therefore, from the CRT it follows that an integer K exists so that [13]:

$$\equiv$$

$$\left|\left(-1+\sum_{j\in J} M_j \cdot \left|\frac{1}{M_j}\right|_{m_j}\right)\right|_{M_I}.$$

$$\left|K \cdot P_J\right|_{M_I} \equiv$$

$$\left|K \cdot M \cdot \left|\frac{1}{P_I}\right|_{M_I}\right|_{M_I}.\equiv$$

Hence

$$\left|\left|X\right|_M \cdot S_{INV} \cdot \left(-1+\sum_{j\in J} M_j \cdot \left|\frac{1}{M_j}\right|_{m_j}\right)\right|_{M_I} \equiv$$

$$\left|\left|X\right|_M \cdot K \cdot S_{INV} \cdot M \cdot \left|\frac{1}{P_I}\right|_{M_I}\right|_{M_I} \equiv$$

$$\left|\left|X\right|_M \cdot K \cdot \left|\frac{1}{P_I}\right|_{M_I} \cdot M_I\right|_{M_I} \equiv 0\ .$$

B) From

$$\left|M \cdot S_{INV}\right|_{M_I} \equiv \left|M_I\right|_{M_I} = 0$$

it results

$$\left|\sum_{j\in J} M \cdot S_{INV} \cdot \left|\frac{1}{M_j}\right|_{m_j} \cdot \left[\frac{\left|X\right|_M}{m_j}\right]\right|_{M_I} \equiv$$

$$\left|M_I \cdot \sum_{j\in J}\left|\frac{1}{M_j}\right|_{m_j} \cdot \left[\frac{\left|X\right|_M}{m_j}\right]\right|_{M_I} \equiv 0\ .$$

REFERENCES

[1] A.A. Albert, *Fundamental Concepts of Higher Algebra*, University of Chicago Press, Chicago, 1956.

[2] G. Dimauro, S. Impedovo and G. Pirlo, "A new technique for fast numbers comparison in the Residue Number System", *IEEE-Transaction on Computers*, Vol. 42, No. 5, May 1993, pp. 608-612.

[3] G.Dimauro, S. Impedovo, G. Pirlo, and A. Salzo, "RNS architectures for the implementation of the 'diagonal function', *Information Processing Letters*, vol. 73, 2000, pp.189-198.

[4] G. Dimauro, S. Impedovo, R. Modugno, G. Pirlo, R. Stefanelli, "Residue-to-Binary Conversion by the Quotient Function", *IEEE Trans. Circuits Syst – Part II*, Vol.50, No.8, Aug. 2003, pp. 488-493.

[5] K.M. Elleithy, M.A. Bayoumi, "Fast and flexible architectures for RNS arithmetic decoding", IEEE Trans. Circuits Syst. - II, Vol. 39, Apr. 1992, pp. 226-235.

[6] C.Y.Hung and B. Parhami, "An approximate sign detection method for residue number systems", Computers Math. Applic., Vol. 10, No. 4/5, , 1984, pp. 331-342.

[7] K.Hwang, *Computer Arithmetic: Principle,Architecture, Design*, NewYork: Wiley, 1979.

[8] J.Y.Kim, K. H. Park, H. S. Lee, "Efficient Residue-to-Binary Conversion Technique with Rounding Error Compensation" , *IEEE Trans. Circuits Syst.*, Vol. CAS-38, n. 3, March 1991, pp. 315-317.

[9] S. J. Piestrak, "Design of Residue Generators and Multi-operand Modular Adders Using Carry-Save Adders", *IEEE-Transaction on Computers*, Vol. 43, No. 1, Jan. 1994, pp. 68-77.

[10] F. Pourbigharaz, H.M. Yassine, "A Signed-Digit Architecture for residue to Binary Transformation", *IEEE-Transaction on Computer*, . Vol. 46, N. 10, Oct. 1997, pp. 1146-1150.

[11] A.B. Premkumar, "An RNS to Binary Converter in a Three Moduli Set with Common Factors", IEEE *Trans. Circuits Syst – Part II*, Vol. 42, N. 4, April 1995, pp. 298-301.

[12] S.Szabó and R. I. Tanaka, *Residue Arithmetic and its Applications to Computer Technology*, New York: McGraw-Hill, 1967.

[13] F.J.Taylor, "Residue arithmetic: a tutorial with examples", *Computer*, pp.50-62, vol.17, no. 5, May 1984.

[14] W.Wang, M.N.S.Swamy, M.O.Ahmad, Y.Wang, "A High-Speed Residue-to-Binary Converter for Three-Moduli (2^k, 2^{k-1}, 2^{k-1}-1) RNS and a Scheme for Its VLSI Implementation", *IEEE Trans. Circuits Syst – Part II*, Vol.47, No.1, Dec.2000, pp. 1576-1581.

Giuseppe Pirlo received the Computer Science degree cum laude in 1986 at the Department of Computer Science of the University of Bari, Italy. Since then he has been carrying out research in the field of pattern recognition and image analysis.
He received a fellowship from IBM in 1988. Since 1991 he has been Assistant Professor at the Department of Computer Science of the University of Bari, where he is currently Associate Professor. His interests cover the areas of pattern recognition and biometry, image analysis, intelligent systems, computer arithmetic, communication and multimedia technologies. He has developed several scientific projects and published over one-hundred fifty papers in the field of handwriting recognition, automatic signature verification, document analysis and processing, parallel architectures for computing, multimedia technologies for collaborative work and distance learning.
Prof. Pirlo is reviewer for many international journals including IEEE T-PAMI, IEEE T-SMC, Pattern Recognition, IJDAR, Information Processing Letters, etc. . He has been in the scientific committee of many International Conferences and has served as reviewer of ICPR, ICDAR, ICFHR, IWFHR, ICIAP, VECIMS, CISMA, etc. . He is general co-chair of the International Conference on Frontiers in Handwriting Recognition (ICFHR 2012).
He is IEEE member and member of the IAPR - Technical Committee on "Reading Systems" (TC-11). He serves as member of the SIe-L Head Committee and is member of the e-learning Committee of the University of Bari.

Donato Impedovo received the MEng degree cum laude in Computer Engineering in 2005 and the PhD degree in Computer Engineering in 2009 both from the Polytechnic of Bari (Italy). In 2011 he received the M.Sc. (II Level italian Master degree) on Remote Science Technologies from the University of Bari. He is, currently, with the Department of Computer Science

(University of Bari). His research interests are in the field of pattern recognition and biometrics. He is co-author of more than 20 articles on these fields in both international journals and conference proceedings. He received ‘The Distinction’ for the best young student presentation in May 2009 at the International Conference on Computer Recognition Systems (CORES – endorsed by IAPR). He serves as reviewer for the Elsevier Pattern Recognition journal, IET Journal on Signal Processing and IET Journal on Image Processing and for many International Conferences including ICPR and ICASSP. He is IAPR and IEEE member.
name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests ends the paragraph.