Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE

PISA

ADDING FLEXIBILITY TO HYBRID NUMBER SYSTEMS

Ferruccio BARSI and Maria Cristina Pinotti

Progetto finalizzato
"Materiali e Dispositivi per l'Elettronica a Stato Solido"

Nota interna B4-48

Settembre 1989

ADDING FLEXIBILITY TO HYBRID NUMBER SYSTEMS

Ferruccio BARSI and Maria Cristina PINOTTI

Istituto di Elaborazione dell'Informazione del CNR, Pisa, Italy

Abstract

Hybrid Number Systems (HNS's) represent a natural generalization of weighted and residue number systems. In HNS's, an integer is represented by using both weighted and residue notations; their arithmetic properties, which have been investigated in depth, are strongly dependent on the ratio of the residue to the weighted range of the representation. It is immediate that the ability of varying the residue-to-weighted-range ratio should enable to optimize the arithmetic performances of these systems.

This paper shows that adding flexibility to hybrid systems is very simple and is equivalent to perform a number system conversion. A general procedure is proposed whose complexity is the same of the well known mixed radix converting algorithm.

A VLSI architecture is presented and its area-time performances are evaluated according to VLSI theory assumptions.

Index Terms

Computer Arithmetic, Hybrid Number Systems, Number System Conversion, Residue Number Systems, VLSI architecture, Weighted Number Systems

---

This research has been supported by the National Program on Solid-State Electronics and Devices of the Italian National Research Council

1. INTRODUCTION

There are two basic approaches to face problems concerning fast numerical information processing. The first one consists in devising methods which enhance time performances of binary, weighted arithmetic, whereas the alternative approach involves the application of different number systems such as residue systems.

When applied in computer arithmetic, both Weighted Number Systems (WNS's) and Residue Number Systems (RNS's) exhibit several drawbacks. In fact, WNS's may result very time consuming owing to the mutual dependence of their digits in all those computations involving additions and multiplications. On the contrary, RNS's, which allow fast addition and multiplication, require lenghty and heavy procedures whenever the knowledge of the number magnitude is necessary to carry out operations such as comparison, sign detection or division [1].

Since early 60's [2, 3, 4, 5], some formulations of different number systems have been proposed in attempts to retain the modular properties of RNS's without completely releasing the explicit knowledge of number magnitude of WNS's. More recently [6, 7, 8], starting from the observation that both WNS's and RNS's may be considered as opposite solutions to the problem of representing numbers, a more general definition of number systems, namely, Hybrid Number Systems (HNS's), has been proposed which includes RNS's and WNS's as particular cases. These systems represent integers by means of two separate parts (a residue and a weighted part) which may be operated concurrently. Their arithmetic properties have been investigated [7] and it has been found that they are strongly dependent on the ratio of the residue range to the weighted range of the representation.

However, as it will be shown in this paper, there are no serious difficulties in moving, according to a particular application, the residue-to-weighted-range ratio in HNS's. This additional *flexibility* , or, equivalently, the ability of doing a generalized number system conversion, may prove very useful to optimize time performances or, as in division, to reduce the approximation of the results.

For the sake of completeness, the HNS's definition will be recalled in Section 2. Section 3 will analyze the problem of number system conversion and a unified procedure will be presented for expanding either the residue or weighted part in a hybrid notation.

Finally, according to VLSI theory assumptions, Section 4 will propose and evaluate the logical design of a structure implementing a general number system converter.

2. HYBRID NUMBER SYSTEMS

In the most general formulation, defining a number system is equivalent to:

i) choosing an ordered set of positive integers $\{m_1, m_2, ..., m_n\}$, which are referred to as *the radices of the system* ;

ii) specifying a law relating any integer X to a set of digits $\{x_1, x_2, ..., x_n\}$, where, in general, $0 \leq x_i < m_i$. In addition, a one-to-one correspondence is to be guaranteed between integers belonging to a given interval and the corresponding representations.

In practice, any number system, which enables integers to be represented by finite sequences of symbols, is based upon the following identity, holding for arbitrary X and arbitrary positive integer μ:

$$X = |X|_\mu + \mu \left\lfloor \frac{X}{\mu} \right\rfloor \tag{1}$$

Weighted Number Systems and Residue Number Systems represent the most common examples of the application of Identity (1).

The number system which will be considered here [7] assumes that, for a given μ in Identity (1), $|X|_\mu$ and $\left\lfloor \frac{X}{\mu} \right\rfloor$ are given different representations. More specifically, it is supposed that $|X|_\mu$ is represented in a Residue System of pairwise prime *moduli* (the radices of the residue system) $\{m_1, m_2, ..., m_t\}$ and $\left\lfloor \frac{X}{\mu} \right\rfloor$ is represented in a weighted system of radices $\{m_{t+1}, m_{t+2}, ..., m_n\}$.

Following this approach, it is immediate to realize that Weighted Number System and Residue Number System can be considered as special cases, with $t=0$ and $t=n$, respectively.

The above defined *Hybrid Number System* exhibits features which are intermediate between that of residue and weighted systems [7], namely:

- the possibility, as in residue systems, of performing modular, carry-free addition and multiplication

- the ability, as in weighted systems, of performing fast magnitude comparison, sign and overflow detection and division.

Therefore, in performing arithmetic, hybrid systems may represent a satisfactory tradeoff allowing faster computation time as compared with conventional number systems [6] [8]. Regardless to special applications where the residue or weighted rate of numerical information is imposed by external constraints, it can be observed, in general, that fast addition and multiplication are made easier whenever the residue part overcomes the weighted notation. On the contrary, operations such as sign detection, magnitude comparison or division are essentially based upon the weighted part of hybrid notation. A very attractive goal should be the ability of moving (i.e., of making flexible) the representation without any additional time cost. Of course, this is an impossible objective as moving representation requires numerical information processing; however, in this paper, it will be shown that adding *flexibility* to hybrid systems is very simple and the procedure which is required to this purpose is based on the well known mixed radix conversion procedure of the residue part of the representation [1].

3. ADDING FLEXIBILITY TO HYBRID NUMBER SYSTEMS

Let :

$$\{m_1, m_2, \ldots\ldots, m_t, m_{t+1}, m_{t+2}, \ldots\ldots, m_n\}$$

be the radices of a Hybrid Number System and, recalling Identity (1), let's assume:

$$\mu = \Pi_{i=1}^{t} m_i \quad \text{and} \quad P = \Pi_{i=t+1}^{n} m_i$$

as the residue and weighted ranges, respectively.

Any integer X will be given the representation:

$$X \equiv \{ R_X, W_X \} = \{x_1, x_2, \ldots\ldots, x_t, x_{t+1}, x_{t+2}, \ldots\ldots, x_n\}$$

where $R_X = |X|_\mu \equiv \{x_1, x_2, ..., x_t\}$ is represented in the residue system of pairwise prime moduli $\{m_1, m_2, ..., m_t\}$ and $W_X = \left\lfloor \frac{X}{\mu} \right\rfloor \equiv \{x_{t+1}, x_{t+2}, ..., x_n\}$ is expressed in the mixed-radix, weighted system of radices $\{m_{t+1}, m_{t+2}, ..., m_n\}$.

Substituting in Identity (1), it will be obtained :

$$X = R_X + \mu W_X \tag{2}$$

3.1 Expanding the weighted part of the representation

Let $\{m_1, m_2, ..., m_t\}$ be the set of moduli of the residue part of the representation and suppose we wish to convert moduli $m_{r,1}, m_{r,2}, ..., m_{r,s}$, $m_{r,i} \in \{m_1, m_2, ..., m_t\}$, $i= 1, ..., s$, $s \leq t$, as additional radices of the weighted part. As a consequence Identity (1) will become:

$$X = |X|_{\mu^*} + \mu^* \left\lfloor \frac{X}{\mu^*} \right\rfloor \tag{1'}$$

where:

$$\mu^* = \frac{\Pi_{i=1}^{t} m_i}{\Pi_{i=1}^{s} m_{r,i}} = \frac{\mu}{m_R} \qquad P^* = (\Pi_{i=t+1}^{n} m_i)\,(\Pi_{i=1}^{s} m_{r,i}) = P\, m_R$$

with $m_R = \Pi_{i=1}^{s} m_{r,i}$

and the number X will be given the representation:

$$X \equiv \{ R^*_X, W^*_X \} = \{x^*_1, x^*_2, ..., x^*_t, x^*_{t+1}, x^*_{t+2}, ..., x^*_n\}$$

where, from Identity (1'):

$$X = R^*_X + \frac{\mu}{m_R} W^*_X \tag{2'}$$

$$W^*_X = \left\lfloor \frac{X}{\frac{\mu}{m_R}} \right\rfloor \qquad R^*_X = |X|_{\frac{\mu}{m_R}} \tag{3}$$

From Equality (3) it is immediately seen that the residue part of the *new* hybrid system has, for identical moduli, the same residue digits of the original representation, whereas, comparing Equalities (2) and (2'), it is obtained:

$$W^*_X = m_R W_X + \frac{R_X - R^*_X}{\frac{\mu}{m_R}} = m_R W_X + \frac{|X|_\mu - |X|_{\frac{\mu}{m_R}}}{\frac{\mu}{m_R}} = m_R W_X + \xi_R \qquad (4)$$

$$\xi_R = \frac{|X|_\mu - |X|_{\frac{\mu}{m_R}}}{\frac{\mu}{m_R}} \qquad (5)$$

i.e., $m_R W_X$ is expressed by the original weighted digits $\{x_{t+1}, x_{t+2}, \ldots\ldots, x_n\} = \{x^*_{t+1}, x^*_{t+2}, \ldots\ldots, x^*_n\}$; observing that ξ_R is representable in the residue range m_R as $0 \leq \xi_R < m_R$ and has weight "1", it can be equivalently expressed in the weighted system of radices $m_{r,1}, m_{r,2}, \ldots\ldots, m_{r,s}$.

To summarize, the overall procedure of repartitioning the radices of the system reduces to computing ξ_R and converting it from the residue to the associate mixed radix system. More specifically, the procedure will consist of two steps:

i) *Step 1.* Extend $|X|_{\frac{\mu}{m_R}}$ to moduli $m_{r,1}, m_{r,2}, \ldots\ldots, m_{r,s}$; this is equivalent to perform, in parallel, s base extensions, whose final steps [ ] consist in solving:

$$\frac{1}{\frac{\mu}{m_R}} x^*_{r,i} + k_{r,i} \equiv 0 \qquad \text{mod } m_{r,i} \qquad\qquad i = 1, 2, \ldots, s$$

where $k_{r,i}$ is the mod $m_{r,i}$ value which is obtained after the (t-s).th standard step of the procedure. It can be observed that Equality (5) can be given the equivalent formulation:

$$\left|\xi_R\right|_{m_{r,i}} \equiv \xi_{r,i} \equiv \frac{1}{\frac{\mu}{m_R}} x_{r,i} - \frac{1}{\frac{\mu}{m_R}} x^*_{r,i} \equiv \frac{1}{\frac{\mu}{m_R}} x_{r,i} + k_{r,i} \quad \text{mod } m_{r,i} \qquad i = 1, 2, \ldots, s$$

from which it is immmediate to derive the residue representation of ξ_R.

ii) *Step 2.* Convert ξ_R in the associate mixed radix system of radices $m_{r,1}, m_{r,2}, \ldots\ldots, m_{r,s}$;

The same results, leading to the same computational steps, can be obtained starting from a different viewpoint. In fact, suppose that the overall residue part R_X of the representation is to be converted in the associate mixed radix number system. As the ordering of residue moduli can be arbitrarily altered, it is possible to start the converting procedure [1] from moduli $m_{k,1}, m_{k,2}, \ldots\ldots, m_{k,t-s}$, where $m_{k,j} \notin \{m_{r,1}, m_{r,2}, \ldots\ldots, m_{r,s}\}$, $j = 1, 2, \ldots, t-s$. It is obtained:

$$R_X = a_{k,1} + a_{k,2} m_{k,1} + a_{k,3} m_{k,1} m_{k,2} + \ldots\ldots\ldots + a_{k,t-s} \frac{\mu}{\prod_{i=1}^{t-s-1} m_{k,i}} +$$

$$+ a_{r,1} \frac{\mu}{m_R} + a_{r,2} \frac{\mu}{m_R} m_{r,1} + a_{r,3} \frac{\mu}{m_R} m_{r,1} m_{r,2} + \ldots\ldots + a_{r,s} \frac{\mu}{m_R} \prod_{i=1}^{s-1} m_{r,i}$$

From Equality (2):

$$X = a_{k,1} + a_{k,2} m_{k,1} + a_{k,3} m_{k,1} m_{k,2} + \ldots\ldots\ldots + a_{k,t-s} \frac{\mu}{\prod_{i=1}^{t-s-1} m_{k,i}} +$$

$$+ a_{r,1} \frac{\mu}{m_R} + a_{r,2} \frac{\mu}{m_R} m_{r,1} + a_{r,3} \frac{\mu}{m_R} m_{r,1} m_{r,2} + \ldots\ldots + a_{r,s} \frac{\mu}{m_R} \prod_{i=1}^{s-1} m_{r,i} + \mu W_X$$

Comparing with (3), it is obtained:

$$W_X^* = a_{r,1} + a_{r,2} m_{r,1} + a_{r,3} m_{r,1} m_{r,2} + \ldots\ldots + a_{r,s} \prod_{i=1}^{s-1} m_{r,i} + m_R W_X$$

i.e.,

$$x_{r,i}^* = a_{r,i} \qquad i = 1, 2, \ldots\ldots, s \qquad (5')$$

Example 1

In the Hybrid Number System of radices $m_1 = 5, m_2 = 7, m_3 = m_t = 8, m_4 = m_{t+1} = 9, m_5 = 11, m_6 = m_n = 13$, *consider integer* $X = \{1, 0, 6, 3, 2, 1\} = 33726$ *and suppose we wish to extend the weighted part of the representation to include* *moduli* $m_{r,1} = m_1 = 5$ *and* $m_{r,2} = m_{r,s} = m_3 = 8$. *To this purpose, observe first that, from Equality (3):*

$$x^*_2 = x_2$$

whereas , starting from residue digit mod m_2 *, the overall residue part of the representation is to be converted in the associate mixed radix representation in order to find weighted digits* x_1 *(of weight* m_2*) and* x_3 *(of weight* $m_1 \cdot m_2 = 5$ *) .*

	$m_1 = 5$	$m_2 = 7$	$m_3 = 8$	
R_X -	1	0	6	$a_{k,1} = x_2 = x^*_2 = 0$
$\lvert R_X \rvert_7$ -	0	0	0	
	1	=	6	
x $\left\lvert \frac{1}{7} \right\rvert_m$	3		7	
	3		2	$a_{k,2} = x^*_1 = 3$
- $\left\lvert \left[\frac{R_X}{7} \right] \right\rvert_5$	3		3	
	=		7	
x $\left\lvert \frac{1}{5} \right\rvert_8$			5	
			3	$a_{k,3} = x^*_3 = 3$

As a conclusion, the representation in the new hybrid system becomes:
{3,0, 3, 3, 2, 1}.
In fact:

$$x_2^* + \{3, 3, 3, 2, 1\} = 0 + 7\{3 + 3 \cdot 5 + 3 \cdot 5 \cdot 8 + 2 \cdot 5 \cdot 8 \cdot 9 + 1 \cdot 5 \cdot 8 \cdot 9 \cdot 11\} = 33726$$

3.2 Expanding the residue part of the representation

In the hybrid system of radices $\{m_1, m_2, ..., m_t, m_{t+1}, ..., m_n\}$ where $\{m_1, m_2, ..., m_t\}$ are the moduli of the residue part, suppose that radices $m_{t+1}, ..., m_{t+s}$, $s \leq n-t$, are to be converted as additional moduli of the residue part . To this purpose, it is necessary to ensure that $m_1, m_2, ..., m_t, m_{t+1}, ..., m_{t+s}$ are pairwise prime numbers to avoid redundancy and the reduction of the arithmetic range. Now, Identity (1) will take the form:

$$X = |X|_{\mu^{**}} + \mu^{**} \left\lfloor \frac{X}{\mu^{**}} \right\rfloor \qquad (1'')$$

where:

$$\mu^{**} = \mu\, m_W \qquad \text{and} \qquad P^{**} = \frac{P}{m_W}$$

with

$$m_W = \prod_{i=1}^{s} m_{t+i}$$

The *new* number system, as re-defined by preceding assumptions, will represent an integer X as

$$X \equiv \{\ R^{**}_X,\ I^{**}_X\ \} = \{x^{**}_1, x^{**}_2, ..., x^{**}_t, x^{**}_{t+1}, . ., x^{**}_{t+s}, ..., x^{**}_n\}$$

where:

$$X = R_X^{**} + \mu\, m_W W_X^{**} \qquad (2'')$$

and:

$$W_X^{**} = \left\lfloor \frac{X}{\mu m_w} \right\rfloor \qquad R_X^{**} = |X|_{\mu m_w} \qquad (3')$$

From Equality (3'), it is deduced that the *new* weighted part is expressed by the leftmost (n-t-s) digits of the original representation

$$\{x^{**}_{t+s+1}, x^{**}_{t+s+2}, ..., x^{**}_n\} = = \{x_{t+s+1}, x_{t+s+2}, ..., x_n\}$$

Moreover, from Equalities (2) and (2"):

$$R_X + \mu W_X = R_X^{**} + \mu\, m_W W_X^{**}$$

which is equivalent to:

$$R_X{}^{**} = R_X + \mu W_X - \mu\, m_W\, W_X{}^{**} =$$

$$= R_X + \mu \left\lfloor \frac{X}{\mu} \right\rfloor - \mu\, m_W \left\lfloor \frac{X}{\mu m_W} \right\rfloor = R_X + \mu \left\| \left\lfloor \frac{X}{\mu} \right\rfloor \right\|_{m_w}$$

or, in other terms:

$$|X|_{\mu m_w} = |X|_\mu + \mu \left\| \left\lfloor \frac{X}{\mu} \right\rfloor \right\|_{m_v} \qquad (4')$$

Computing $R_X{}^{**}$ from Equality (4') requires a base extension of R_X from the range μ to the new residue range $\mu\ m_W$. In addition, the residue representation of $\mu \left\| \left\lfloor \frac{X}{\mu} \right\rfloor \right\|_{m_v}$ in the range $\mu\ m_W$ is required; as the last term is a multiple of μ, it is sufficient to compute residue digits mod m_{t+1}, ..., m_{t+s} , i.e., recalling that:

$$\left\| \left\lfloor \frac{X}{\mu} \right\rfloor \right\|_{m_w} = x_{t+1} + x_{t+2} m_{t+1} + \ldots\ldots + x_{t+s} \Pi_{i=1}^{s-1} m_{t+i}$$

it follows that $\mu \left\| \left\lfloor \frac{X}{\mu} \right\rfloor \right\|_{m_w}$ is represented as:

$$x_{t+1}\,\mu \qquad \text{mod } m_{t+1}$$

$$x_{t+1}\,\mu + x_{t+2}\,\mu\, m_{t+1} \qquad \text{mod } m_{t+2}$$

$$x_{t+1}\,\mu + x_{t+2}\,\mu\, m_{t+1} + x_{t+3}\,\mu\, m_{t+1}\, m_{t+2} \qquad \text{mod } m_{t+3}$$

....

....

Again, the overall system converting procedure is equivalent to a mixed radix conversion in a residue system with t+s moduli. To prove this assertion, suppose that the procedure of expanding the residue representation is performed step by step, i.e., suppose first that radix m_{t+1} is to be assumed as an additional modulus of the residue part. To do this, the residue digits $\{x_1, x_2, \ldots\ldots, x_t\}$ are to be extended to m_{t+1} and, after t standard base extension steps, it will be obtained, mod m_{t+1} :

$$k_{t+1}^{(t)}$$

and, recalling that [1] :

$$|R_X|_{m_{t+1}} = |-\mu k^{(t)}_{t+1}|_{m_{t+1}}$$

the extended digit , mod m_{t+1} , will become, from Equality (4'):

$$x^{**}_{t+1} = -\mu\, k^{(t)}_{t+1} + x_{t+1}\,\mu \quad \text{mod } m_{t+1} \qquad (5'')$$

Before iterating the above procedure, assume that the first t standard base extension steps have been concurrently performed mod m_{t+1}, ..., m_{t+s} , thus obtaining:

$$k^{(t)}_{t+1},\ \ k^{(t)}_{t+2},\ \ k^{(t)}_{t+3}, \ldots\ldots, k^{(t)}_{t+s} \qquad (6)$$

Now, suppose that radix m_{t+2} is also to be converted as a residue modulus. Again, a base extension is necessary starting from residue digits $\{x_1, x_2, \ldots\ldots, x_t, x^{**}_{t+1}\}$ and, after t steps, the following terms are obtained modulo m_{t+1}, ..., m_{t+s} :

$$\frac{1}{\mu}(-\mu\, k^{(t)}_{t+1} + x_{t+1}\,\mu) + k^{(t)}_{t+1} = x_{t+1}\ , \qquad k^{(t)}_{t+2}, \quad k^{(t)}_{t+3}, \ldots\ldots, k^{(t)}_{t+s} \qquad (6')$$

Next, (t+1).th step will give, modulo m_{t+2} :

$$k^{(t+1)}_{t+2} = \frac{k^{(t)}_{t+2} - x_{t+1}}{m_{t+1}}$$

and, more generally, modulo m_{t+i}:

$$k^{(t+1)}_{t+i} = \frac{k^{(t)}_{t+i} - x_{t+1}}{m_{t+1}} \qquad i = 2, 3, \ldots.., s \qquad (7)$$

The (t+2).th residue digit will be easily found observing that Equality (4') holds in the form:

$$|X|_{(\mu m_{t+1})\frac{m_w}{m_{t+1}}} = |X|_{(\mu m_{t+1})} + \mu m_{t+1}\left|\left\lfloor\frac{X}{\mu m_{t+1}}\right\rfloor\right|_{\frac{m_w}{m_{t+1}}} \qquad (4'')$$

and

$$\left|\left\lfloor\frac{X}{\mu m_{t+1}}\right\rfloor\right|_{\frac{m_w}{m_{t+1}}} = x_{t+2} + x_{t+3}m_{t+2} + \ldots + x_{t+s}\prod_{i=2}^{s-1} m_{t+i}$$

with

$$\mu m_{t+1} \left\|\left\lfloor \frac{X}{\mu m_{t+1}} \right\rfloor\right\|_{\frac{m_w}{m_{t+1}}} \equiv \mu m_{t+1}\, x_{t+2} \qquad \text{mod } m_{t+2}$$

$$\mu m_{t+1} \left\|\left\lfloor \frac{X}{\mu m_{t+1}} \right\rfloor\right\|_{\frac{m_w}{m_{t+1}}} \equiv \mu m_{t+1}\, x_{t+2} + \mu m_{t+1}\, m_{t+2}\, x_{t+3} \qquad \text{mod } m_{t+3}$$

……

……

As a conclusion:

$$x^{**}_{t+2} = -\,\mu\, m_{t+1}\, k^{(t+1)}_{t+2} + \mu\, m_{t+1}\, x_{t+2} = x_{t+1}\,\mu + x_{t+2}\,\mu\, m_{t+1} - \mu\, k^{(t)}_{t+2}$$

Similarly, to find x^{**}_{t+3}, it will be obtained, after (t+1) steps:

$$\frac{1}{\mu m_{t+1}}\left(-\,\mu m_{t+1} k^{(t+1)}_{t+2} + \mu m_{t+1} x_{t+2}\right) + k^{(t+1)}_{t+2} = x_{t+2},\ k^{(t+1)}_{t+3},\ \ldots\ldots,\ k^{(t+1)}_{t+s} \qquad (6'')$$

and next, (t+2).th step will produce, mod m_{t+i} :

$$k^{(t+2)}_{t+i} = \frac{k^{(t+1)}_{t+i} - x_{t+2}}{m_{t+2}} \qquad i = 3, \ldots\ldots, s \qquad (7')$$

and

$$x^{**}_{t+3} = -\mu m_{t+1} m_{t+2} k^{(t+2)}_{t+3} + \mu m_{t+1} m_{t+2} x_{t+3} =$$

$$= \mu\, x_{t+1} + \mu m_{t+1}\, x_{t+2} + \mu m_{t+1} m_{t+2}\, x_{t+3} - \mu\, k^{(t)}_{t+3}$$

In general:

$$k^{(t+j)}_{t+i} = \frac{k^{(t+j-1)}_{t+i} - x_{t+j}}{m_{t+j}} \quad \text{mod } m_{t+i} \quad j = 1, \ldots, s-1 \quad i = j+1, \ldots, s \qquad (7``)$$

and

$$x^{**}_{t+i} = \mu \Pi_{u=1}^{i-1} m_{t+u} \left(x_{t+i} - k^{(t+i-1)}_{t+i}\right) \quad \text{mod } m_{t+i} \qquad (5'')$$

where

$\Pi_{u=1}^{i-1} m_{t+u} = 1$ for $i = 1$

The above relations show that there is no difference between the standard mixed radix conversion or base extension procedure and the re-definition of mixed radix digits as residue digits. This enables the same structure be employed for both residue or weighted part expansion.

Example 2

In the hybrid number system of Example 1, consider the same integer $X = \{1, 0, 6, 3, 2, 1\} = 33726$ *and assume that radices* $m_{t+1} = m_4 = 9$, $m_{t+2} = m_5 = 11$ *are to be considered as additional moduli of the residue part. This assumption defines a new hybrid system where the residue part has moduli* $\{m_1, m_2, m_3, m_4, m_5\}$ *whereas the weighted part has the single radix* m_6. *From Equality (3'), it follows:*

$|X|_{m_1} = x_1^{**} = x_1 = 1$

$|X|_{m_2} = x_2^{**} = x_2 = 0$

$|X|_{m_3} = x_3^{**} = x_3 = 6$

$W_X^{**} = x_6^{**} = x_6 = 1$

To obtain the remaining digits of R_X^{**}, $|X|_\mu$ *is first extended to the new residue range* μm_w.

	$m_1=5$	$m_2=7$	$m_3=8$	$m_4=9$	$m_5=11$
R_X	1	0	6	0	0
$- \lvert R_X \rvert_5$	1	1	1	1	1
	=	6	5	8	10
$x \left\lvert \frac{1}{5} \right\rvert_m$		3	5	2	9
		4	1	7	2

$-\left\|\left\lfloor\frac{X}{5}\right\rfloor\right\|_7$	4	4	4	4
	=	5	3	9
$x\left\|\frac{1}{7}\right\|_m$		7	4	8
		3	3	6
$-\left\|\left\lfloor\frac{X}{5*7}\right\rfloor\right\|_8$		3	3	3
		=	0	3
$x\left\|\frac{1}{8}\right\|_m$			8	7
			$k_4^{(3)} = 0$	$k_5^{(3)} = 10$

The procedure goes on by considering the original weighted digit x_4. = 3.

$-x_4$	3	3
	=	7
$x\left\|\frac{1}{9}\right\|_{11}$		5
		$k_5^{(4)} = 2$

and, according to Equality (5"), it is obtained:

$x_4^{**} = x^{**}_{t+1} = \mu\,(x_4 - 0) = 3 \mod 9$

$x_5^{**} = x^{**}_{t+2} = \mu\ m_{t+1}(x_5 - 2) = 0 \mod 11$

The representation of X in the new hybrid system becomes:

{1, 0, 6, 3, 0, 1}.

In fact:

{1, 0, 6, 3, 0, 1} = {1, 0, 6, 3, 0} + 1·(5·7·8·9·11) = 6006 + 27720 = 33,726.

4. THE PROPOSED ARCHITECTURE

In this section, an architecture will be proposed to carry out both extension procedures presented in Sections 3.1 and 3.2.

This structure, shown in Fig.1, is very similar to the one performing the mixed radix conversion in residue systems; it consists, for a given hybrid system H of radices $\{m_1, \ldots, m_t, m_{t+1}, \ldots, m_n\}$, of one processing unit for each radix of the system and some additional combinational blocks. The unit U_i, associated with the i-th radix, (Fig. 2), calculates mod m_i sums and products and contains a local ROM storing i constant values; the first (i-1) cells contain the mod m_i multiplicative inverses of the moduli associated with units $U_1, \ldots, U_{i-1}$ whereas the i-th cell stores the mod m_i product of moduli $m_1, \ldots, m_{i-1}$. Each unit is connected, through gates G_i, to a bus L: the i-th gate enables U_i writing its current output on the bus L at the i.th step of the procedure.

Two bit-registers S and S' store the present and the next state of the number system definition : the i-th bits $b_S(i)$ and $b_{S'}(i)$ of S and S', will be set to "1" if and only if the i-th radix is a modulus of the residue part of the original and final number system and "0" otherwise.

Registers $TEMP_i$, i = 1, ..., n , will contain the final value of the computations performed by units U_i and registers $RESULT_i$, which are first loaded with the input digits of the original representation, will contain the final results of the procedure.

Before describing in depth the proposed architecture, let's indicate with $k_i^{(w)}$ the value calculated by U_i after w steps from the beginning of the procedure. In particular:

$k_i^{(0)} = x_i$ if $b_S(i) = 1$

$k_i^{(0)} = 0$ if $b_S(i) = 0$

where $\{x_1, x_2, \ldots, x_t, \ldots, x_n\}$ represents the original notation.

Case 1: Expanding the weighted part of the representation.

For the sake of simplicity, we will limit our consideration to the case of converting consecutive moduli $m_{t-s+1}, \ldots, m_t$ as additional radices of the weighted part. In the more general case where moduli $m_{r,1}, \ldots, m_{r,s}$ are arbitrary selected in the set $\{m_1, \ldots, m_t\}$, a preliminary re-ordering is required to ensure that the first (t-s) radices coincide with the moduli of the residue part of the system.

In the system H of radices $\{m_1, \ldots, m_t, m_{t+1}, \ldots, m_n\}$, with $\mu = \Pi_{i=1}^{t} m_i$, and $P = {}_{i=t+1}^{n} m_i$, suppose, without loss of generality, that the moduli of the residue part are ordered increasingly, i. e.: $m_i < m_j$ for $1 \leq i < j \leq t$.

Given an integer $X = \{x_1, \ldots, x_t, x_{t+1}, \ldots, x_n\}$, suppose that an extension of the weighted part of the representation is requested to include s additional moduli. Register S' will contain::

$$b_{S'}(i) = \begin{cases} 1 & i \in [1, \ldots, t\text{-}s], \\ 0 & i \in [t\text{-}s+1, \ldots, n]; \end{cases}$$

According to Relation (5'), the new digits are obtained by converting $R_X = \{x_1, \ldots, x_t\}$ in the associated mixed radix representation $\{a_1, \ldots, a_t\}$. The original weighted digits are unchanged, hence we take no notice of the results of the last (n-t) units and their values will not be saved in the corresponding RESULT_i registers.

Let's describe the functioning of U_i.

```
Procedure A
        begin
           $j := 1;$
           $\text{RESULT}_i := x_i;$
           $\text{halt} := t;$
           while $j <= \text{halt}$ do
           begin
              if $j = i$ then
                        begin
                           open $G_i$ ;
                           send $a_i = k_i^{(j-1)}$ to L;
                           $\text{TEMP}_i := a_i;$
                        end;
              $k_i^{(j)} = \left| \frac{k_i^{(j-1)} - a_j}{m_j} \right|_{m_i}$ ;
```

```
        j := j+1;
    end;
    if j = halt+1 and (($b_S$(i) xor $b_{S'}$(i)) = 1) then
                                                    begin
                                                       RESULT$_i$ := TEMP$_i$;
                                                       stop := true;
                                                    end;
end.
```

We wait until all registers TEMP$_i$ of the first t units are loaded and the new representation of X is contained in registers RESULT$_i$.

Case 2: Expanding the residue part of the representation

Assume H, μ, P, X as defined in Case 1, and let

$$b_{S'}(i) = \begin{cases} 1 & i \in [1, \ldots, t+s], \\ 0 & i \in [t+s+1, \ldots, n]; \end{cases}$$

be the required next definition of the number system.

According to (3'), as $x^{**}_{t+s+j} = x_{t+s+j}$, with $j = 1, \ldots, n-t-s$, we disregard units $U_{t+s+1}, \ldots, U_n$ related to the weighted part of the representation.

Units $U_1, \ldots, U_t$, for which $b_{S'}(i) = b_S(i) = 1$, calculate the digits $a_1, a_2, \ldots, a_t$ of the associated mixed radix number system. The procedure they execute is the same of Case 1, except for the value of variable halt which is set to (t+s+1).

The description of the procedure, as performed by units $U_{t+1}, \ldots, U_{t+s}$ is the following:

Procedure B

```
begin
   j := 1;
   RESULT$_i$ := $x_i$;
   halt := t + s + 1;
   while j <= t do
   begin
```

```
    $k_i^{(j)} = \left|\frac{k_i^{(j-1)} - a_j}{m_j}\right|_{m_i}$;
    j := j+1;
  end;
  while j <= halt do
  begin
    if j = i + 1  then
                  begin
                     open $G_{i+1}$;
                     $TEMP_i := k_i^{(i)}$;
                  end;
    if j = i      then
                  begin
                     open $G_i$;
                     send $x_i$ ($b_S(i) = 1$) to the bus L;
                     $k_i^{(j)} := \left|\left(- \mu \Pi_{u=1}^{i-1} m_{t+u}\right)\left(k_i^{(j-1)} - x_i\right)\right|_{m_j}$;
                  end;
                  else
                     $k_i^{(j)} = \left|\frac{k_i^{(j-1)} - x_j}{m_j}\right|_{m_j}$;
                  end;
    j := j+1;
  end;
  if j = t+s+2 and (($b_S(i)$ xor $b_{S'}(i)$) = 1))  then
                                                   $RESULT_i := TEMP_i$;
                                                   stop := true;
                                                   end;
end
```

At the (t+s+2)-th step, all the registers $RESULT_i$ are correctly loaded and the procedure is completed.

VLSI complexity figures

Evaluating the VLSI complexity of the proposed structure is rather simple. In fact, the overall architecture consists of a single row of processing units performing modular additions and multiplications.

Binary and mod m adders and multipliers have been extensively traited by several authors [9, 10, 11, 12]. For the sake of simplicity, let's recall the major results which have been obtained for both adders and multipliers in terms of constructive upper bounds. All the proposed layouts are based upon a pipelined computation scheme, where operands are sliced in an appropriate number of strings of equal lenght.

Referring to adders, it has been shown [9] that binary addition can be performed by means of structures exhibiting the following area and time complexities:

$$A_B = O\left(\frac{b}{T} \times \log \frac{b}{T}\right) \qquad T_B = O\left(T + \log \frac{b}{T}\right)$$

where b is the total number of bits of the operands and T represents the number of strings into which the operand words are divided.

Similarly, for binary multipliers [11], it has been shown that:

$$A_B^* = O\left(\frac{b}{T} \times \frac{b}{T}\right) \qquad T_B^* = O\left(T + \log \frac{b}{T}\right)$$

Similar results are obtained when mod m adders and multipliers are considered, i.e., whenever non binary, weighted or residue arithmetic is to be performed [].

Referring to adders, and observing that log m indicates the bit lenght of the modular word, it has been shown that [10] :

$$A_M = O\left(\frac{\log m}{T} \times \left(\log \frac{\log m}{T} + T\right)\right) \qquad T_M = O\left(T + \log \frac{\log m}{T}\right) \tag{8}$$

whereas, for multipliers [10, 12]:

$$A_M^* = O\left(\frac{\log m}{T} \times \left(\frac{\log m}{T} + T\right)\right) \qquad T_M^* = O\left(T + \log \frac{\log m}{T}\right) \tag{9}$$

From Relations (8) and (9) and referring again to Fig.2, it is immediate to derive the area occupancy of i.th unit U_i:

$$A_{U_i} = O\left(\frac{\log m}{T} \times \left(\frac{\log m}{T} + i\,T\right)\right) \tag{10}$$

and the time which is required to perform a single mixed radix converting step:

$$T_{U_i} = O\left(T + \log \frac{\log m}{T}\right) \tag{11}$$

As a conclusion, the overall structure of Fig.1 exhibits an area:

$$A = O\left(n \frac{\log m}{T} \times \left(\frac{\log m}{T} + n\,T\right)\right) \qquad (12)$$

and has the following time requirements:

$$T = O\left(j \times \left(T + \log \frac{\log m}{T}\right)\right) \qquad (13)$$

where:

$j = t$ in Case 1 (expanding the weighted part of the representation) and

$j = t + s$ in Case 2 (extension of the residue part).

REFERENCES

[1] N.S. Szabo and R.I. Tanaka, *Residue Arithmetic and its Applications to Computer Technology*, Ney York, McGraw-Hill, 1967

[2] A. Sasaki, "Addition and subtraction in the residue number system", IEEE Trans. Electronic Computers, Vol. EC-16, pp. 157-164, April 1967

[3] A. Sasaki, "The basis for implementation of additive operations in residue number system", IEEE Trans. Comput., Vol. C-17, pp. 1066-1073, November 1968

[4] T.R.N. Rao and A.K. Trehan, "Binary logic for residue arithmetic using magnitude index", IEEE Trans. Comput., Vol. C-19, pp. 752-757, August 1970

[5] F. Barsi and P. Maestrini, "Arithmetic codes in residue number systems with magnitude index", IEEE Trans. Comput., Vol. C-27, pp. 1185-1188, December 1978

[6] G. Alia, F. Barsi and E.Martinelli, "Addition and overflow handling in a class of redundant RNS with magnitude index", IEI Internal Report B4-33, Pisa, Italy, December 1987

[7] F. Barsi, "Arithmetic properties of a class of Hybrid number systems", IEI Internal Report B4-01, Pisa, Italy, February 1989

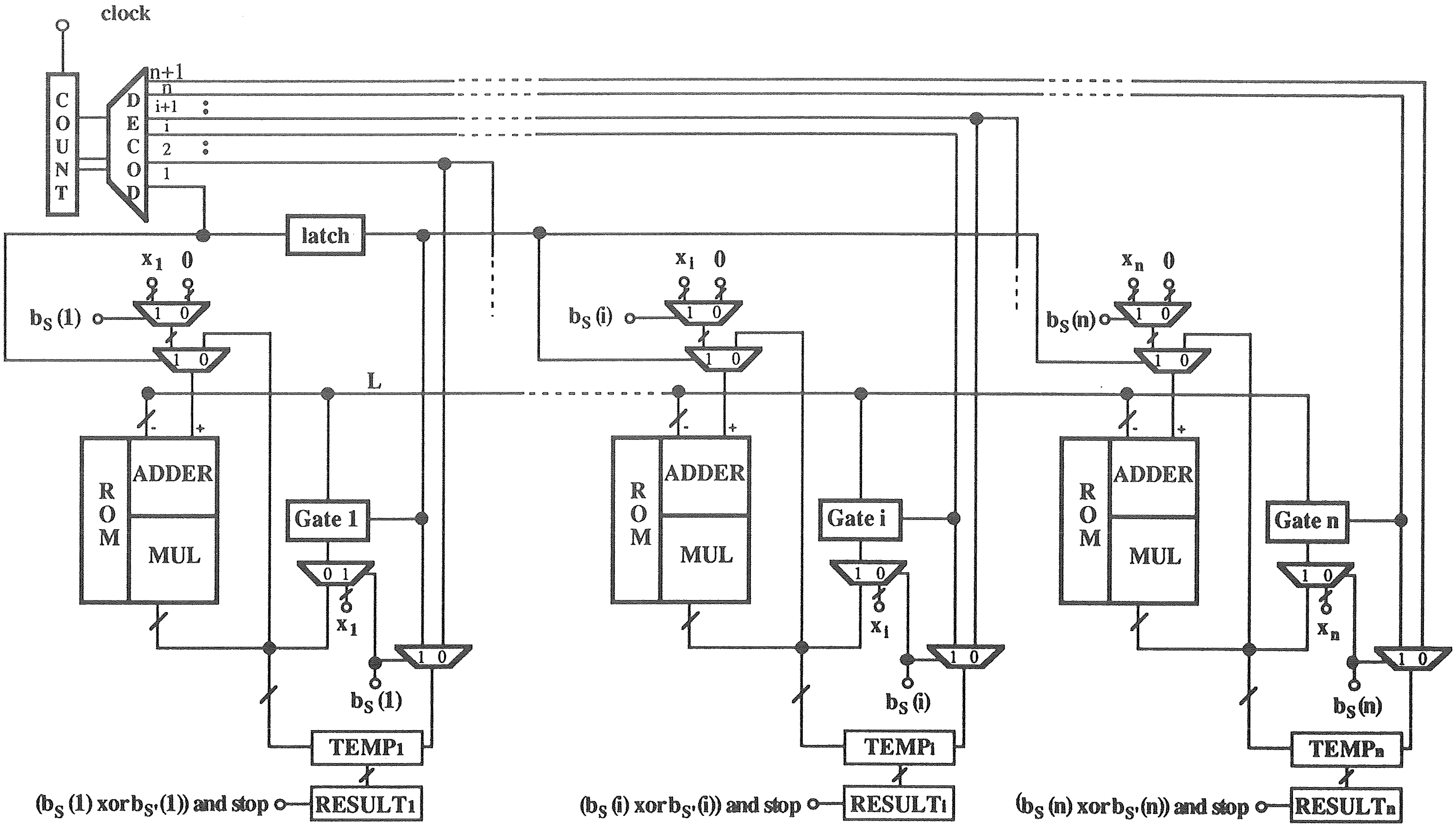
[8] F. Barsi and E. Martinelli, "A VLSI architecture for RNS with MI adders", IEI Internal Report B4-15, Pisa, Italy, April 1989

[9] R.P. Brent and H.T. Kung, "A regular layout for parallel adders", IEEE Trans. Comput. Vol. C-31, n.2, pp. 260-264, March 1982

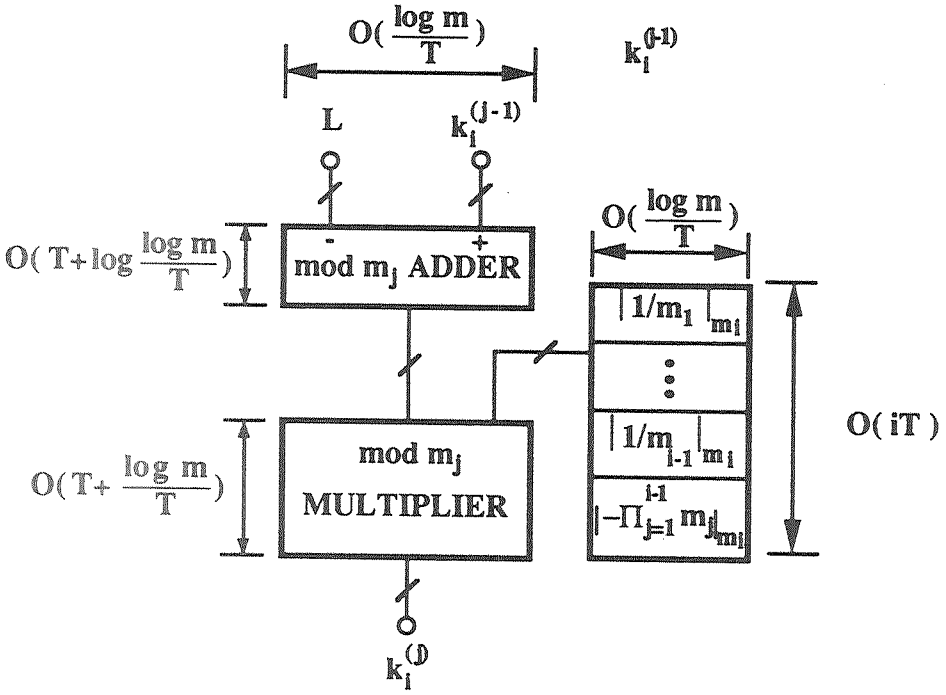
[10] F. Barsi, "Residue arithmetic in binary systems", IEI Internal Report B4-37, Pisa, Italy, August 1988

[11] K. Mehlhorn and F.P. Preparata, "Area-time optimal VLSI integer multiplier with minimum computation time", Inform. Contr., Vol. 58, pp. 137-156, 1983

[12] G. Alia and E. Martinelli, "The VLSI residue multiplication and its implication in the direct and reverse positional-to-residue conversion", IEI Internal Report B4-70, Pisa, Italy, December 1986



Barsi and Pinotti Fig. 1



$i = 1, \ldots, n$

$$k_i^{(0)} = \begin{cases} x_i & \text{if } b_S(i) = 1 \\ 0 & \text{if } b_S(i) = 0 \end{cases}$$

Barsi and Pinotti Fig. 2