Adding Flexibility to Hybrid Number Systems*

F. BARSI[1] AND M. C. PINOTTI[2]

[1] *Dipartimento di Matematica, Universitá di Perugia, Italy*

[2] *Istituto di Elaborazione dell'Informazione del CNR, Pisa, Italy*

Hybrid number systems (HNSs) represent a natural generalisation of weighted and residue number systems. In HNSs, an integer is represented by using both weighted and residue notations; their mathematical 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 apparent that varying the residue-to-weighted-range ratio should enable us to optimise the mathematical performances of these systems.

This paper shows that adding flexibility to hybrid systems is very simple. A general procedure is proposed whose complexity is the same as the well-known mixed radix converting algorithm. A VLSI architecture is presented and its area-time performances are evaluated.

Received October 1989, revised December 1991

1. INTRODUCTION

There are two basic approaches facing problems concerning fast numerical information processing. The first consists in devising methods which enhance time performances of binary, weighted mathematics; the second approach involves the application of different number systems such as residue systems.

When applied in computer mathematics, both weighted number systems (WNSs) and residue number systems (RNSs) exhibit several drawbacks. In fact, WNSs are very time consuming, owing to the mutual dependence of their digits in computations involving addition and multiplication. On the other hand, RNSs, which allow fast addition and multiplication, require lengthy and heavy procedures whenever the knowledge of the number magnitude is necessary to carry out operations such as comparison, sign detection or division.[12]

Since the early 1960s,[10,11,9,5] some different number systems have been proposed in attempts to retain the modular properties of RNSs without completely losing the explicit knowledge of number magnitude of WNSs. More recently,[1,3,6] starting from the premise that WNSs and RNSs can be considered opposite solutions to the problem of representing numbers, a more general definition of number systems, namely, hybrid number systems (HNSs), has been proposed. This includes RNSs and WNSs as particular cases. These systems represent integers by means of two separate parts (a residue and a weighted part) which may be operated in parallel. Their mathematical properties have been investigated[3] and have been found to be strongly dependent on the ratio of the residue range to the weighted range of the representation.

There are no serious difficulties in moving, according to particular applications, the residue-to-weighted-range ratio in HNSs. This additional *flexibility*, or, equivalently, the ability to perform a generalised number system conversion, may prove very useful in optimising time performances.

For the sake of completeness, the HNSs definition will be recalled in Section 2. Section 3 will analyse the problem of the generalised conversion and a unified procedure will be presented for expanding either the residue or the weighted part in a HNS notation. Finally, according to VLSI theory assumptions, Section 4 will propose and evaluate the logical design of a structure implementing both expansions.

2. HYBRID NUMBER SYSTEMS

In general, a number system can be defined as:

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

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

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

$$X = |X|_\mu + \mu\lfloor X/\mu\rfloor. \tag{1}$$

The number system which will be considered here[3] assumes that, for a given μ, $|X|_\mu$ and $\lfloor X/\mu\rfloor$ 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) $\{m_1, m_2, \dots, m_t\}$ and $\lfloor X/\mu\rfloor$ is represented in a weighted system of radices $\{m_{t+1}, m_{t+2}, \dots, m_n\}$. It is essential to realise that WNSs and RNSs become special cases of HNSs, with $t = 0$ and $t = n$, respectively.

HNSs exhibit features which are intermediate between that of residue and weighted systems,[3] namely:

- the ability, as in residue systems, of performing fast and partially carry-free addition and multiplication;
- the possibility, as in weighted systems, of performing easy magnitude comparison, sign and overflow detection and division.

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

Therefore, hybrid systems may represent a satisfactory trade-off allowing faster computation time compared with conventional number systems.[1,6] More specifically, addition and multiplication become faster by reducing the weighted part; further, operations such as sign detection, magnitude comparison or division are essentially based upon the weighted part of the representation. A very attractive goal should be the ability of moving (i.e. of making flexible) the representation without additional time costs. Of course, this is impossible; however, in this paper, it will be shown that adding *flexibility* to HNSs is very simple and the procedure which is required is very similar to a mixed radix conversion.[12]

3. ADDING FLEXIBILITY TO HYBRID NUMBER SYSTEMS

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

be the radices of a HNS. Without any substantial loss of generality, it will be assumed that *radices are pairwise prime integers and are ordered increasingly*, i.e. $m_i < m_j$ for $1 \leqslant i < j \leqslant n$. Moreover, recalling Identity (1)

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

will indicate the residue and the weighted ranges, respectively.

Any integer X will be given the representation:

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

where $R_X = |X|_\mu \equiv (x_1, x_2, \ldots, x_t\}$ is represented in the RNS of moduli $\{m_1, m_2, \ldots, m_t\}$ and $W_X = \lfloor X/\mu \rfloor \equiv \{x_{t+1}, x_{t+2}, \ldots, x_n\}$ is expressed in the WNS of radices $\{m_{t+1}, m_{t+2}, \ldots, m_n\}$.

Substituting in equation (1), obtains

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

3.1 Expanding the weighted part of the representation

Let $\{m_1, m_2, \ldots, m_t\}$ be the set of moduli of the residue part of the representation and suppose we wish to convert moduli $m_{t-s+1}, m_{t-s+2}, \ldots, m_{t-1}, m_t, s \leqslant t$, as additional radices of the weighted part. The residue and weighted ranges will become, respectively:

$$\mu^* = \prod_{i=1}^{t-s} m_i = \frac{\mu}{m_R} \quad P^* = \prod_{i=t-s+1}^{n} m_i = P m_R$$

$$\text{with } m_R = \prod_{i=t-s+1}^{t} m_i.$$

As a consequence, equation (1) will take the form

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

and the number X will be given the representation:

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

From equation (1′):

$$X = R_X^* + \frac{\mu}{m_R} W_X^*, \tag{2'}$$

$$W_X^* = \left\lfloor \frac{X}{\mu/m_R} \right\rfloor, \quad R_X^* = |X|_{\mu/m_R}. \tag{3}$$

The expansion is easily obtained supposing that the overall residue part R_X of the representation is to be converted in the associate mixed radix number system. In fact:

$$R_X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \ldots + a_{t-s} \prod_{i=1}^{t-s-1} m_i + a_{t-s+1} \frac{\mu}{m_R} + a_{t-s+2} \frac{\mu}{m_R} m_{t-s+1} + a_{t-s+3} \frac{\mu}{m_R} m_{t-s+1} m_{t-s+2} + \ldots + a_t \frac{\mu}{m_R} \prod_{i=1}^{s-1} m_{t-s+i}$$

and, from equation (2):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \ldots + a_{t-s} \prod_{i=1}^{t-s-1} m_i + a_{t-s+1} \frac{\mu}{m_R} + a_{t-s+2} \frac{\mu}{m_R} m_{t-s+1} + a_{t-s+3} \frac{\mu}{m_R} m_{t-s+1} m_{t-s+2} + \ldots + a_t \frac{\mu}{m_R} \prod_{i=1}^{s-1} m_{t-s+i} + \mu W_X.$$

Comparing with (3), it follows:

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

In conclusion:

$$\left.\begin{aligned} x_i^* &= x_i \quad \text{for} \quad i = 1, \ldots, t-s \\ x_i^* &= a_i \quad \text{for} \quad t-s+1, \ldots, t \\ x_i^* &= x_i \quad \text{for} \quad i = t+1, \ldots, n \end{aligned}\right\} \tag{5'}$$

Example 1

In the HNS 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 $m_2 = 7$ and $m_3 = 8$. To derive the new representation, the residue part is converted in the associate mixed radix system to find weighted digits x_2^* (of weight m_1) and x_3^* (of weight $m_1 . m_2$). It is found:

	$m_1 = 5$	$m_2 = 7$	$m_3 = 8$	
$R_X - \lvert R_X\rvert_5$	1	0	6	$a_1 = x_1 = x_1^* = 1$
	—	1	1	
$x\lvert\frac{1}{5}\rvert$		6	5	
		3	5	
$\left\lfloor\frac{R_X}{5}\right\rfloor - \left\lvert\left\lfloor\frac{R_X}{5}\right\rfloor\right\rvert_7$		4	1	$a_2 = x_2^* = 4$
		—	4	
$x\lvert\frac{1}{7}\rvert$			5	
			7	
			3	$a_3 = x_3^* = 3$

and the representation in the new HNS is: $\{1, \mathbf{4}, \mathbf{3}, 3, 2, 1\}$. In fact:

$$x_1^* + 5x\{4, 3, 3, 2, 1\} = 1 + 5\{4 + 3.7 + 3.7.8 + 2.7.8.9 + 1.7.8.9.11\} = 33726.$$

3.2 Expanding the residue part of the representation

In the hybrid system of radices $\{m_1, m_2, \ldots, m_t, m_{t+1}, \ldots, m_n\}$ where $\{m_1, m_2, \ldots, m_t\}$ are the moduli of the residue part, suppose that radices $m_{t+1}, \ldots, m_{t+s}, s \leqslant n-t$, are to be considered as additional moduli. The residue and weighted ranges will become, respectively:

$$\mu^{**} = \mu m_W \quad \text{and} \quad P^{**} = \frac{P}{m_W} \quad \text{where} \quad m_W = \prod_{i=1}^{s} m_{t+i}$$

and equation (1) will take the form:

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

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

$$X \equiv \{R_X^{**}, I_X^{**}\} = \{x_1^{**}, x_2^{**}, \ldots, x_t^{**}, x_{t+1}^{**}, \ldots, x_{t+s}^{**}, \ldots, x_n^{**}\}$$

where
$$X = R_X^{**} + \mu m_W\, W_X^{**} \tag{2''}$$

and
$$W_X^{**} = \left\lfloor \frac{X}{\mu m_W} \right\rfloor \quad R_X^{**} = |X|_{\mu m_W}. \tag{3'}$$

From equation (3′), it is shown that the *new* weighted part is expressed by the most significant $(n-t-s)$ digits of the original representation:

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

Moreover, from equations (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:

$$|X|_{\mu m_W} = |X|_{\mu} + \mu \left| \left\lfloor \frac{X}{\mu} \right\rfloor \right|_{m_W}. \tag{4'}$$

Computing R_X^{**} from equation (4′) requires a base extension of R_X from the range μ to the new residue range μm_W. In addition, the residue representation of $\mu|\lfloor X/\mu \rfloor|_{m_W}$ in the range μm_W is required; as the last term is a multiple of μ, it is sufficient to compute residue digits $\bmod\, m_{t+1}, \ldots, m_{t+s}$, i.e. recalling that

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

it follows that $\mu|\lfloor X/\mu \rfloor|_{m_W}$ is represented as

$$\begin{array}{ll} x_{t+1}\,\mu & \bmod m_{t+1}, \\ x_{t+1}\,\mu + x_{t+2}\,\mu m_{t+1} & \bmod m_{t+2}, \\ x_{t+1}\,\mu + x_{t+2}\,\mu m_{t+1} + x_{t+3}\,\mu m_{t+1}\, m_{t+2} & \bmod m_{t+3}. \\ \cdots\cdots\cdots & \\ \cdots\cdots\cdots & \end{array}$$

Again, the expanding procedure is equivalent to a mixed radix conversion in a residue system with $t+s$ moduli. To prove this, 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 this purpose, residue digits $\{x_1, x_2, \ldots, x_t\}$ are to be extended to m_{t+1} and, after t standard base extension steps, it will be obtained, $\bmod\, m_{t+1}$:

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

and, recalling that:[12]

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

the extended digit, $\bmod\, m_{t+1}$, will become, from equation (4′):

$$x_{t+1}^{**} = -\mu k_{t+1}^{(t)} + x_{t+1}\,\mu \quad \bmod m_{t+1}. \tag{5''}$$

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

$$k_{t+1}^{(t)}, k_{t+2}^{(t)}, k_{t+3}^{(t)}, \ldots, k_{t+s}^{(t)}. \tag{6}$$

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

$$\frac{1}{\mu}(-\mu k_{t+1}^{(t)} + x_{t+1}\,\mu) + k_{t+1}^{(t)} = x_{t+1}, \quad k_{t+2}^{(t)}, k_{t+3}^{(t)}, \ldots, k_{t+s}^{(t)}. \tag{6'}$$

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

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

and, in general, modulo m_{t+i}:

$$k_{t+i}^{(t+1)} = \frac{k_{t+i}^{(t)} - x_{t+1}}{m_{t+1}} \quad i = 2, 3, \ldots, s. \tag{7}$$

The $(t+2)$th residue digit will be easily found observing that equation (4′) holds in the form:

$$|X|_{(\mu m_{t+1})\, m_W/m_{t+1}} = |X|_{(\mu m_{t+1})} + \mu m_{t+1} \left| \left\lfloor \frac{X}{\mu m_{t+1}} \right\rfloor \right|_{m_W/m_{t+1}} \tag{4''}$$

and

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

with

$$\mu m_{t+1} \left| \left\lfloor \frac{X}{\mu m_{t+1}} \right\rfloor \right|_{m_W/m_{t+1}} \equiv \mu m_{t+1}\, x_{t+2} \qquad \bmod m_{t+2},$$

$$\mu m_{t+1} \left| \left\lfloor \frac{X}{\mu m_{t+1}} \right\rfloor \right|_{m_W/m_{t+1}} \equiv \mu m_{t+1}\, x_{t+2} + \mu m_{t+1}\, m_{t+2}\, x_{t+3} \qquad \bmod m_{t+3}.$$

....

....

As a conclusion:

$$x_{t+2}^{**} = -\mu m_{t+1}\, k_{t+2}^{(t+1)} + \mu m_{t+1}\, x_{t+2} = x_{t+1}\,\mu + x_{t+2}\,\mu m_{t+1} - \mu k_{t+2}^{(t)}.$$

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

$$\frac{1}{\mu m_{t+1}}(-\mu m_{t+1}\, k_{t+2}^{(t+1)} + \mu m_{t+1}\, x_{t+2}) + k_{t+2}^{(t+1)} = x_{t+2}, k_{t+3}^{(t+1)}, \ldots, k_{t+s}^{(t+1)} \tag{6''}$$

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

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

and

$$\begin{aligned} x_{t+3}^{**} &= -\mu m_{t+1} m_{t+2} k_{t+3}^{(t+2)} + \mu m_{t+1} m_{t+2} x_{t+3} \\ &= \mu x_{t+1} + \mu m_{t+1} x_{t+2} + \mu m_{t+1} m_{t+2} x_{t+3} - \mu k_{t+3}^{(t)}. \end{aligned}$$

In general, it can be concluded:

$$x_{t+i}^{**} = \mu \prod_{u=1}^{i-1} m_{t+u} (x_{t+i} - k_{t+i}^{(t+i-1)}) \bmod m_{t+i} \qquad (5'')$$

where $\quad \prod_{u=1}^{i-1} m_{t+u} = 1 \quad \text{for} \quad i = 1$

and

$$k_{t+i}^{(t+j)} = \frac{k_{t+i}^{(t+j-1)} - x_{t+j}}{m_{t+j}} \bmod m_{t+i}$$

$$j = 1, \ldots, s-1 \quad i = j+1, \ldots, s. \qquad (7'')$$

Above relations show that obtaining residue digits x_{t+i}^{**}, $i = 1, \ldots, s$ corresponds to performing a mixed radix conversion where, after the tth step, the values which are to be subtracted to $k_{t+i}^{(t+j-1)}$ are the weighted digits x_{t+j} of the original representation. This enables the same structure to be employed for both residue and weighted part expansions.

Example 2

In the same HNS of Example 1, consider $X = \{1, 0, 6, 3, 2, 1\} = 33726$ and assume that radices $m_{t+1} = m_4 = 9$, $m_{t+2} = m_5 = 11$ are to be used as additional moduli of the residue part. This assumption defines a new HNS 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 equation (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 extended to the new residue range μm_w.

$$\begin{array}{r|ccccc} & m_1 = 5 & m_2 = 7 & m_3 = 8 & m_4 = 9 & m_5 = 11 \\ & 1 & 0 & 6 & 0 & 0 \\ R_X - |R_X|_5 & - & 1 & 1 & 1 & 1 \\ \hline & & 6 & 5 & 8 & 10 \\ x\left|\frac{1}{5}\right| & & 3 & 5 & 2 & 9 \\ \hline & & 4 & 1 & 7 & 2 \\ \left\lfloor\frac{X}{5}\right\rfloor - \left|\left\lfloor\frac{X}{5}\right\rfloor\right|_7 & & - & 4 & 4 & 4 \\ \hline & & & 5 & 3 & 9 \\ x\left|\frac{1}{7}\right| & & & 7 & 4 & 8 \\ \hline & & & 3 & 3 & 6 \\ \left\lfloor\frac{X}{5*7}\right\rfloor - \left|\left\lfloor\frac{X}{5*7}\right\rfloor\right|_8 & & & - & 3 & 3 \\ \hline & & & & 0 & 3 \\ x\left|\frac{1}{8}\right| & & & & 8 & 7 \\ \hline & & & & k_4^{(3)} = 0 & k_5^{(3)} = 10 \end{array}$$

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

$$\begin{array}{r|cc} -x_4 & - & 3 \\ \hline & & 7 \\ x\left|\frac{1}{9}\right| & & 5 \\ \hline & & k_5^{(4)} = 2 \end{array}$$

and, according to equation (5″), it obtains:

$$x_4^{**} = x_{t+1}^{**} = \mu(x_4 - 0) = 3 \qquad \bmod 9$$
$$x_5^{**} = x_{t+2}^{**} = \mu m_{t+1}(x_5 - 2) = 0 \quad \bmod 11$$

and the representation of X in the new hybrid system becomes: $\{1, 0, 6, \mathbf{3}, \mathbf{0}, 1\}$. In fact:

$$\begin{aligned} \{1, 0, 6, 3, 0, 1\} &= \{1, 0, 6, 3, 0\} \\ &+ 1.(5.7.8.9.11) = 6006 + 27720 = 33726. \end{aligned}$$

In conclusion, it is worth noting for both Cases 3.1 and 3.2 that the only digits which are altered when expanding an HNS representation are those corresponding to the radices of the system which change their original meaning.

4. THE PROPOSED ARCHITECTURE

In the previous sections, it has been shown that the expansion of the weighted and the residue part of a HNS representation can be carried out by means of two procedures which essentially perform the same sequence of operations of a mixed radix conversion. Namely, given a HNS of radices $\{m_1, \ldots, m_t, m_{t+1}, \ldots, m_n\}$, the weighted expansion executes $(t-1)$ adding-multiplying steps starting from residue digits $x_1, \ldots, x_t$ whereas expanding the residue part requires a 0-input in the weighted digits to be converted and terminates after $(t+s)$ steps, where s is the number of residue digits to be added. As a consequence, a single device is able to carry out both extension processes and is very similar to a mixed radix converter.[6]

The proposed architecture (see Fig. 1) consists of $(n-1)$ processing units connected through a bus L. Two n-bit registers S and S' store the current and the expected state of the number system; the actual expansion is derived from S and S'. More precisely, the ith bit $b_S(i)\,(b_{S'}(i))$ of $S(S')$ is set to '1' if the ith radix is a residue modulus and '0' otherwise. A digit has to be converted if its current and expected states are different. Thus, the ith radix moves from the weighted to the residue part if $\bar{b}_S(i) \wedge b_{S'}(i) = 1$ and from the residue to the weighted part if $b_S(i) \wedge \bar{b}_{S'}(i) = 1$.

Unit $U_i (2 \leqslant i \leqslant n)$, corresponding to the ith radix, calculates mod m_i sums and products of two operands and has two associated combinational blocks I_i and T_i and 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 ith cell stores the mod m_i product of moduli $m_1, \ldots, m_{i-1}$. Note that the first digit coincides for both representations and cannot be altered in extension processes.[12] Thus U_1 reduces to a register RESULT$_1$ and to a gate GATE$_1$ sending x_1 to L.

Let us indicate with $h_i^{(w)}$ and $k_i^{(w)}$ the input from I_i and the output of unit U_i at the wth step, respectively.

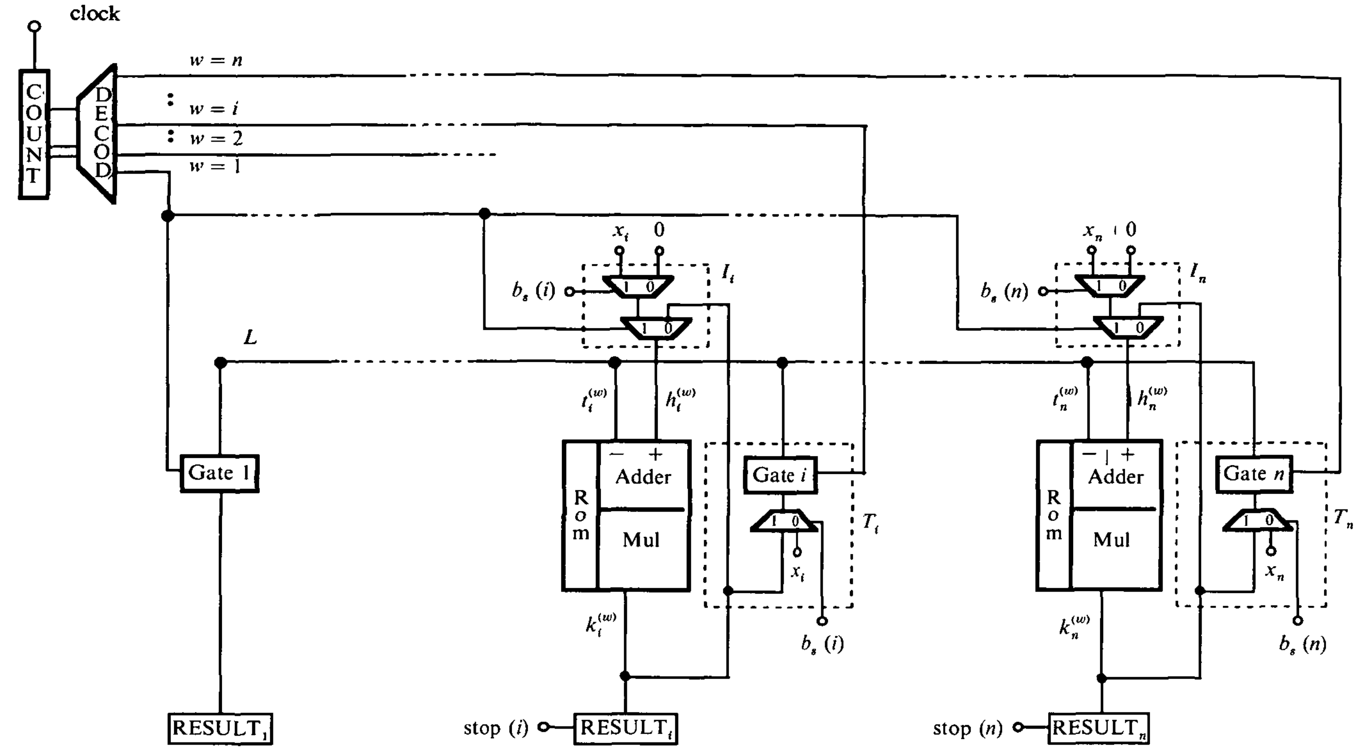

Figure 1.

Depending on $b_S(i)$ and on the wth step of the procedure, I_i selects:

$$h_i^{(1)} = x_i \quad \text{if } b_S(i) = 1 \text{ and } w = 1$$

$$h_i^{(1)} = 0 \quad \text{if } b_S(i) = 0 \text{ and } w = 1$$

$$h_i^{(w)} = k_i^{(w-1)} \quad \text{if } w \geqslant 2$$

and U_i calculates the value $k_i^{(w)} = (h_i^{(w)} - t_i^{(w)})(1/m_w) \bmod m_i$. The other operand $t_i^{(w)}$ is received from L, which carries the output of block T_w. In general, a block T_j send to L, at the jth cycle, the value $k_j^{(j-1)}$ if $b_S(j) = 1$ or x_j if $b_S(j) = 0$.

Whenever the ith digit is moving from one to another representation, it is seen, from (5′) and (5″), that unit U_i outputs the correct result after $(i-1)$ steps if weighted part is to be extended and after i steps in the other case. However, the proposed architecture assumes that U_i can perform the overall procedure provided that the proper output is stored in RESULT_i according to the following condition:

$$\text{stop}\,(i) = ((w = 1) \wedge \bar{b}_S(i) \wedge b_{S'}(i) \vee (w = (1-1)) \wedge b_S(i) \wedge \bar{b}_{S'}(i)).$$

Assuming that RESULT_i registers are initially loaded with the original representation $\{x_1, x_2, \ldots, x_t, x_{t+1}, x_{t+2}, \ldots, x_n\}$, the above condition guarantees that only the digits to be converted are altered. In fact, whenever a digit x_i is to be unchanged, $b_S(i) = b_{S'}(i)$ and stop $(i) = 0$ for any w.

4.1 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.[2,4,7,8] For the sake of simplicity, let us recall the major results which have been obtained for adders and multipliers in terms of upper bounds. All the proposed layouts are based upon a pipelined computation scheme, where operands are sliced into a number of strings of equal length.

Referring to adders, it has been shown[7] that binary addition can be performed by means of structures exhibiting the following area and time complexities:

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

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

Similarly, for binary multipliers,[8] it has been shown that:

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

The same results are obtained when mod m adders and multipliers are considered, i.e. whenever non-binary, weighted or residue mathematics is to be performed.[2,4]

Referring to adders, and denoting by $\log m$ the bit length of a digit, it has been shown that:[4]

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

$$T_M = O\left(T + \log \frac{\log m}{T}\right), \quad (8)$$

whereas, for multipliers[2,4]

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

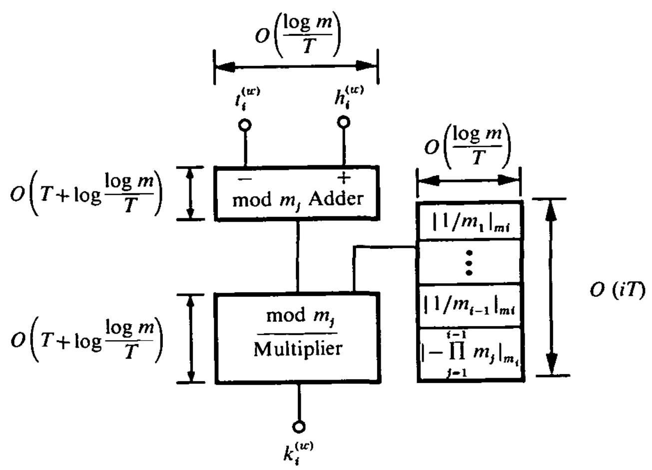

Figure 2.

From equations (8) and (9) and referring to Fig. 2, it is possible to derive the area occupancy of the ith unit U_i:

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

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

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

To conclude, the overall structure of Fig. 1 exhibits area

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

and has the following time requirements:

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

where

$j = t$ if the weighted part is to be extended

and $j = t + s$ in the other case.

REFERENCES

1. 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).
2. 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).
3. F. Barsi, Arithmetic properties of a class of Hybrid number systems. IEI Internal Report B4-01, Pisa, Italy (February 1989).
4. F. Barsi, Mod m arithmetic in binary systems. *Information Processing Letters* **40**, 303–309 (1991).
5. F. Barsi and P. Maestrini, Arithmetic codes in residue number systems with magnitude index. *IEEE Trans. Comput.* **C-27**, 1185–1188 (1978).
6. F. Barsi and E. Martinelli, A VLSI architecture for RNS with MI adders. *Integration, the VLSI journal* **11** (1), 67–83 (1991).
7. R. P. Brent and H. T. Kung, A regular layout for parallel adders. *IEEE Trans. Comput.* **C-31** (2), 260–264 (1982).
8. K. Mehlhorn and F. P. Preparata, Area-time optimal VLSI integer multiplier with minimum computation time. *Inform. Contr.* **58**, 137–156 (1983).
9. T. R. N. Rao and A. K. Trehan, Binary logic for residue arithmetic using magnitude index. *IEEE Trans. Comput.* **C-19**, 752–757 (1970).
10. A. Sasaki, Addition and subtraction in the residue number system. *IEEE Trans. Electronic Computers* **EC-16**, 157–164 (1967).
11. A. Sasaki, The basis for implementation of additive operations in residue number system. *IEEE Trans. Comput.* **C-17**, 1066–1073 (1968).
12. N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and its Applications to Computer Technology*. McGraw-Hill, New York (1967).

Announcement

18–19 June 1993

FEGIS '93: Far East Workshop on Geographic Information Systems, Singapore

The Workshop

FEGIS '93 focuses specifically on the development and application of geographic information systems (GISs). The workshop is timely, as many large-scale GIS projects have recently been initiated in this region and there is increased concern for more efficient and interoperable GISs. This workshop provides a rare and important occasion for applied researchers, GIS developers and users to share their experiences, their problems and results. In particular, sessions for large-scale GIS projects (by invitation) undertaken in this region are planned.

Tutorials and exhibitions

FEGIS '93 is to be held in conjunction with SSD '93, the 3rd International Symposium on Large Spatial Databases (23–25 June 1993). Tutorials are scheduled for SSD '93 and FEGIS '93 on 21–22 June 1993. Industrial exhibitions will be held from 17 to 25 June 1993. For further information on the symposium, please contact the secretariat.

Secretariat

Siew Foong, Ho, Department of Information Systems and Computer Science, National University of Singapore, Lower Kent Ridge, Singapore 0511. Email: hosf@nusvm.bitnet.