Short Note

VLSI Binary-Residue Converters for Pipelined Processing

Residue Number Systems (RNS) are suited for high-speed applications and VLSI implementations, because of the modular and parallel nature of their arithmetic. In this paper a new solution is provided to the problem of designing VLSI structures for converting integers to and from residue number systems in the area of pipelined applications, with the constraint that the layout width is comparable with the data stream width. The proposed structure is suited for both direct and reverse conversion and has complexity figures better than previously known results, evaluated under several hypotheses on the RNS parameters.

Received March 1989, revised October 1989

1. Introduction

Residue Number Systems (RNS) appear particularly suited for special-purpose hardware implementations, such as signal processing, because of the parallel nature of their arithmetic. The VLSI technology increased the bias towards RNS-based systems, since regular and modular layouts can be designed. However, the problem of converting data from the weighted system to the residue system and vice versa must be dealt with accurately in order to keep a high processing rate for the overall structure. The problem of designing VLSI systems for such conversions has been considered by some authors, in an attempt to optimise the area-time complexity of the devices[1,2,3,4] or, more generally, of VLSI implementations of RNS architectures.[5,6]

In this paper a new solution is provided, which exploits the same structure for the two conversion problems and which exhibits VLSI complexity figures better than previously proposed ones in the area of pipelined applications. In particular, the asymptotic value of the area of the VLSI pipeline structure proposed in Ref. 3 is reduced, while the same pipeline interval is asymptotically kept, together with the constraint that the layout width is comparable with the data stream width. Such a constraint allows the structure to be easily embedded in layouts of RNS-based systems along data paths. Complexity figures are provided under several hypotheses on the relation between the number of moduli and the range of integers represented in residue notation.

Given two functions $f(n)$ and $g(n)$ in this paper we write $f(n) = \vartheta(g(n))$ to mean $k_1 g(n) \leqslant f(n) \leqslant k_2 g(n)$, for k_1, $k_2 > 0$ and all sufficiently large n. Similarly $f(n) = \Omega(g(n))$ means $f(n) \geqslant kg(n)$, for $k > 0$ and all sufficiently large n.

Moreover, to derive complexity figures, the generally accepted VLSI model of computation introduced in Refs 7–9 has been adopted in this paper.

2. Residue number systems

We recall that any integer X in the range

$$0 \leqslant X < \prod_{i=1}^{p} m_1 = M < 2^n, \quad n = \lceil \log M \rceil$$

that is the smallest integer larger than or equal to $\log M$, can be represented in a residue number system by the expression $X = \{x_1, x_2, \ldots, x_p\}$, where $x_i = |X|_{m_i} = X - \lfloor X/m_i \rfloor m_i$ is the ith residue digit, $\lfloor X/m_i \rfloor$ denotes the largest integer not exceeding X/m_i, and $\{m_i\}$ is a set of p integers, called moduli. It can also be shown that if the moduli are relatively prime numbers, this is a unique representation.[10] In this paper all moduli are assumed of the same order, i.e. $m_i = \vartheta(m)$, $1 \leqslant i \leqslant p$, and consequently

$$M = \prod_{i=1}^{p} m_i = \vartheta(m^p) \cdot \vartheta(\text{const}^p),$$
$$\log M = \vartheta(n) = \vartheta(p \log m),$$
$$p = \vartheta(n/\log m).$$

3. The conversion from the weighted to the residue representation

The algorithm used for converting an integer X from the weighted to the residue number system is a generalisation of the algorithm explained in Ref. 1, and is based on the following considerations.

Let r_j^i be the powers of 2 modulo m_i, that is $r_j^i = |2^j|_{m_i}$, $0 \leqslant j \leqslant n-1$, $1 \leqslant i \leqslant p$, and let b_j be the value of the jth bit of the binary representation of X. Then the residue representation of X is given by $X = \{x_1, x_2, \ldots, x_i, \ldots, x_p\}$, where

$$x_i = \left| \sum_{j=0}^{n-1} r_j^i \cdot b_j \right|_{m_i}, \quad 1 \leqslant i \leqslant p.$$

In order to perform this summation, the ordered set $\{b_j\}$ can be partitioned into n/q groups of q adjacent bits; n/q can be assumed to be an integer, without loss of generality:

$$x_i = \left| \sum_{h=1}^{n/q} \left| \sum_{j=(h-1)q}^{hq-1} r_j^i \cdot b_j \right|_{m_i} \right|_{m_i}.$$

All possible values resulting from the inner summation, each depending on the values of the hth group of q bits, can be stored in ROMs having 2^q words of $\lceil \log m_i \rceil$ bits. Then the n/q values read from ROMs are added by means of n/q processing elements (PE's); to implement a modulo m_i addition, each PE is supplied with an accumulator register and is able to perform binary addition and subtraction and to test the sign of the result. In Fig. 1 the layout of the proposed VLSI structure is drawn.

4. The conversion from the residue to the weighted representation

The weighted representation of X can be obtained as

$$X = \left| \sum_{i=1}^{p} X_i \right|_M$$

where X_i is the $n$bit weighted representation of $\{0, 0, \ldots, x_i, \ldots, 0\}$.

Moreover, X_i can be obtained as

$$X_i = \sum_{k=1}^{\lceil \log m_i \rceil} b_k^{(i)} \cdot Q^{(i,k)},$$

$b_k^{(i)}$ being the value of the kth bit of x_i and $Q^{(i,k)}$ the $n$bit number whose residue representation consists of all residue digits equal to zero, but the ith digit, $q_i^{(i,k)}$, which equals 2^{k-1}, that is $Q^{(i,k)} = \{0, 0, \ldots, q_i^{(i,k)} = 2^{k-1}, \ldots, 0, 0\}$.

Then

$$X = \left| \sum_{i=1}^{p} \sum_{k=1}^{\lceil \log m_i \rceil} b_k^{(i)} \cdot Q^{(i,k)} \right|_M.$$

This computation of $\vartheta(n)$ sums can be performed by means of a structure very similar to the portion dedicated to a single module of the structure used for the direct conversion and shown in Fig. 1, the only difference being the dimensions.

5. Design complexity evaluation

The layout area for direct conversion can be evaluated by considering dimension values shown in Fig. 1. Total vertical and horizontal sizes respectively result in

$$L_v = \vartheta\left(\sum_{i=1}^{p} \log m_i \right) = \vartheta(p \log m) = \vartheta(n)$$
$$L_h = \vartheta((2^q + \log\log m) n/q).$$

According to the time performance of parallel adders, that is $T = \vartheta(\log\log m)$,[11] and assuming that the time needed to read a ROM of 2^q words is $\vartheta(q)$, the pipeline interval is $\vartheta(q + \log\log m)$.

As the choice of p and consequently of m, depends on the particular application which requires the use of RNS arithmetic, the complexity figures $A(n)$ and $T_p(n)$ for the area and the pipeline interval of the conversion circuits and, in particular for our case, the choice of the optimal value of q are conveniently evaluated for several possible hypotheses, as shown in Tables 1, 2. It is worth noting that the proposed structure exhibits the same pipeline interval as in Ref. 3, which, however, was evaluated for log $m = \vartheta(\log n)$ and $p = \vartheta(n/\log n)$. Nevertheless the area complexity is better for a factor $\vartheta(1/\log\log\log n)$, and it can be verified that an improvement in the area complexity is reached for any given hypothesis, but for the asymptotic hypothesis $p = \vartheta(n)$ and $\log m = \vartheta(\text{const})$, for which the two structures result of the same complexity.

As far as the inverse conversion structure is concerned, referring again to Fig. 1 and considering only the structure related to a single module, with n bits of representation, it is easy to obtain the following complexity figures

$$A = \vartheta((2^q + \log n) n^{2/q}),$$
$$T_p = \vartheta(q + \log n).$$

In this case the area-time performance does not depend on the RNS parameters p and $\log m$, but only on the partition parameter q. Choosing $q = \vartheta(\log\log n)$ yields

$$A = \vartheta(n^2 \log n / \log\log n),$$
$$T_p = \vartheta(\log n).$$

The pipeline interval is the same as in Ref. 3, that is $\vartheta(\log n)$, but, also in the inverse conversion, the layout complexity is reduced, namely for a factor $\vartheta(1/\log\log n)$.

We note that the total vertical size of the structure is independent of the value of

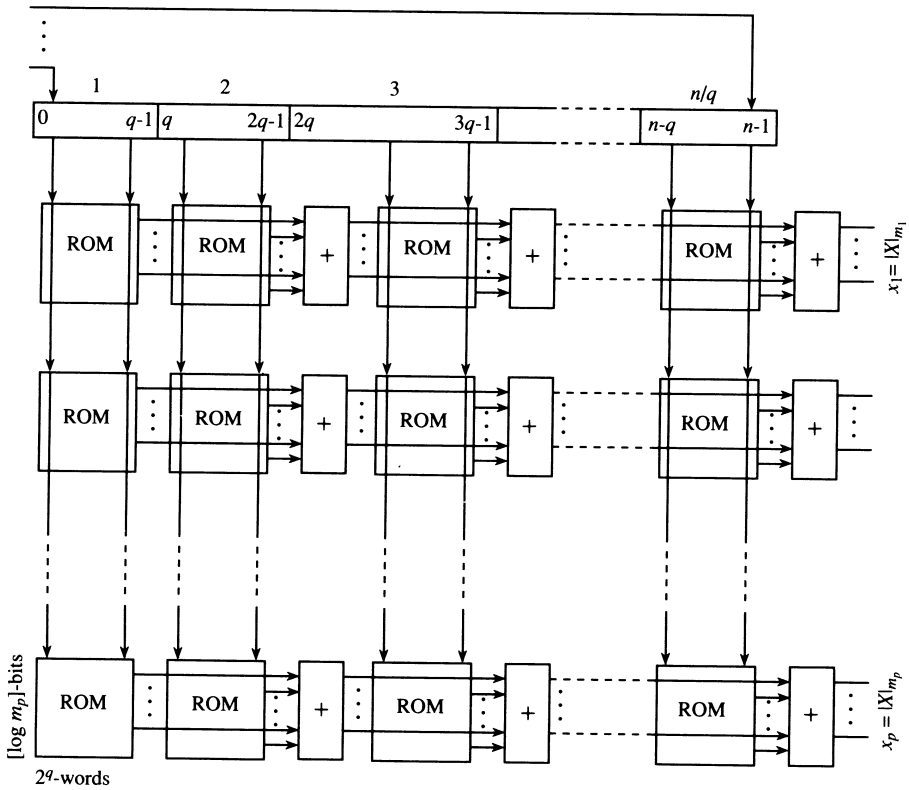

Fig. 1. The layout of the weighted-to-residue convertor.

Table 1

p	$\log m$	A	T_p	q_{opt}
ϑ(const)	$\vartheta(n)$	$\vartheta((2^q+\log n)n^2/q)$	$\vartheta(q+\log n)$	$\vartheta(\log\log n)$
$\vartheta(\log n)$	$\vartheta(n/\log n)$	$\vartheta((2^q+\log n)n^2/q)$	$\vartheta(q+\log n)$	$\vartheta(\log\log n)$
$\vartheta(n/\log n)$	$\vartheta(\log n)$	$\vartheta((2^q+\log\log n)n^2/q)$	$\vartheta(q+\log\log n)$	$\vartheta(\log\log\log n)$
$\vartheta(n/\log\log n)$	$\vartheta(\log\log n)$	$\vartheta((2^q+\log\log\log n)n^2/q)$	$\vartheta(q+\log\log\log n)$	$\vartheta(\log\log\log n)$
$\vartheta(n)$	ϑ(const)	$\vartheta(2^q n^2/q)$	$\vartheta(q)$	ϑ(const)

Table 2

p	$\log m$	A_{opt}	$T_{p_{opt}}$
ϑ(const)	$\vartheta(n)$	$\vartheta\left(\frac{n^2\log n}{\log\log n}\right)$	$\vartheta(\log n)$
$\vartheta(\log n)$	$\vartheta(n/\log n)$	$\vartheta\left(\frac{n^2\log n}{\log\log n}\right)$	$\vartheta(\log n)$
$\vartheta(n/\log n)$	$\vartheta(\log n)$	$\vartheta\left(\frac{n^2\log\log n}{\log\log\log n}\right)$	$\vartheta(\log\log n)$
$\vartheta(n/\log\log n)$	$\vartheta(\log\log n)$	$\vartheta\left(\frac{n^2\log\log\log n}{\log\log\log\log n}\right)$	$\vartheta(\log\log\log n)$
$\vartheta(n)$	ϑ(const)	$\vartheta(n^2)$	ϑ(const)

parameters m, p, q, and is of the same order as data stream width. As a consequence, any structure defined according to our approach can interface data represented in positional notation and RNS-based processors without increasing the complexity of data stream width.

As a concluding remark, in Ref. 6 it was stated that complexity figures better than those we obtained in 1984 are achieved,[1] for the same problem discussed here, when a table look-up approach derived from Ref. 5 is used; at the same time some lower bounds on complexity were claimed. Actually, present work extends the approach in Ref. 1, gaining better results, whereas it cannot be affirmed that the look-up table technique is superior to ours, because, unfortunately, results in Ref. 6 are affected by some faults, as is shown in the Appendix.

Acknowledgement

This work described in this paper has been supported by the National Programme on Solid-State Electronics and Devices of the Italian National Research Council.

G. ALIA AND E. MARTINELLI*†

Istituto di Elettronica e Telecomunicazioni – Facoltà di Ingegneria dell'Università di Pisa, Via Diotisalvi, 56100 Pisa, Italy.

* Istituto di Elaborazione dell'Informazione del C.N.R., Via S. Maria, 46, 56100 Pisa, Italy.

† To whom correspondence should be addressed.

References

1. G. Alia, F. Barsi and E. Martinelli, A fast VLSI conversion between binary and residue systems. *Information Processing Letters* **18**, 141–145 (1984).
2. G. Alia and E. Martinelli, A VLSI algorithm for direct and reverse conversion from weighted binary number system to residue number system. *IEEE Trans. Circuits and Systems CAS-31*, **12**, 1033–1039 (1984).
3. C. D. Thompson, VLSI design with multiple active layers. *Information Processing Letters* **21**, 109–111 (1985).
4. R. M. Capocelli and R. Giancarlo, Efficient VLSI networks for converting an integer from binary systems to residue number system and vice versa. *IEEE Trans. Circuits and Systems CAS-35*, **11**, 1425–1430 (1988).
5. M. A. Bayoumi, G. A. Jullien and W. C. Miller, A VLSI model for residue number system architectures. *Integration* **2**, 191–211 (1984).
6. M. A. Bayoumi, Lower bounds for VLSI implementation of residue number system architectures. *Integration* **4**, 263–269 (1986).
7. C. D. Thompson, A complexity theory for VLSI, *Ph.D. Thesis*, Carnegie-Mellon University, Computer Science Department (1980).
8. R. P. Brent and H. T. Kung, The area-time complexity of binary multiplication. *JACM* **28**, 521–534 (1981).
9. C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, Reading, MA (1980).
10. N. S. Szabo and R. J. Tanaka, *Residue Arithmetic and its Application to Computer Technology*. McGraw-Hill, New York (1967).
11. R. P. Brent and H. T. Kung, A regular layout for parallel adders. *IEEE Trans. Computers C-31*, **3**, 260–264 (1982).

Appendix

The expression derived for area complexity in Ref. 6 is wrong because, using the same notation, the term $M\log M$ has been rewritten as $Lm\log(Lm)$ instead of $m^L L\log m$; in fact

$$M = \prod_{i=1}^{L} m_i = \vartheta(m^L).$$

Consequently, the final correct expression of

the area complexity for the computational graph should be

$$A = \Omega(L \log m(L \log m + m + m^L)) = \Omega(Lm^L \log m).$$

Similarly, for time complexity it should be $T = \Omega(L \log m)$.

Moreover, the lower bounds for VLSI implementations of RNS architecture discussed in Ref. 6 do not refer to the global problem (as is generally intended when lower bounds are investigated), but only to the table look-up approach; in this case the term 'constructive upper bounds' seems more appropriate.

Finally, in the RNS-based computational graph defined in Ref. 6, the scaling and conversion section, which is implemented by means of a table whose area is $\vartheta(M \log M)$, is so large that it is needless to use RNS-based systems; in fact such a table can be directly used to perform the whole computation in the dynamic range M.

Correspondence

Dear Sir,

The nature of information, which has been the subject of much recent correspondence, is readily understood if it is accepted that the probability terms in Shannon and Weaver's expression for negentropy can validly be replaced by frequency terms.[1] Information is then released from its conventional constraint of being applicable to only a 'message string' in a single dimension (e.g. as propagated over a wired link), to being a geometrical property of the three dimensions of space. The justification for this change is to be found in *Concepts of Modern Mathematics*, in which book Ian Stewart reaches the conclusion that probability is best defined as *average frequency of occurrence*.[2]

The expression of negentropy in terms of frequency then allows the introduction of means of dealing with time. That is to say, as a further dimension leading on to the concept of spatiotemporal, or four-dimensional, information as required, for instance, in theories of relativity.

However, the question of the effects of relative phase then arises in systems of complex signals containing a mixture of frequencies, e.g. speech. While the technology of relative phase between continuous, fixed-frequency signals is well understood, that for apparently random discrete pulses is not recognised as being of such importance in information technology as it is in radar, say. Yet the electrical activity of nerves displays just such characteristics.

Further consideration of the problem then shows that single, otherwise indistinguishable pulses can stand in representation of either of the half cycles of a single cycle of oscillation. Since the half cycles of a single oscillation of electromagnetic radiation are in antiphase to each other, it follows that phase must also be quantised at the half-cycle level, thus not only providing a means of resolving all the components of a complex signal, but ultimately of reconciling quantum mechanics with relativity in the field of theoretical physics.

Yours sincerely

B. E. P. CLEMENT
Clement Neuronic Systems Limited,
15 Everest Drive,
Crickhowell,
Powys NP8 1DH

References

1. C. E. Shannon and W. Weaver, In *The Mathematical Theory of Communication*, p. 50. Illinois, University of Illinois Press (1949).
2. I. Stewart, In *Concepts of Modern Mathematics*, p. 253. Harmondsworth, Penguin Books (1975).