VLSI Design, 2002 Vol. 14 (2), pp. 183–191

Taylor & Francis
Taylor & Francis Group

A Parallel Residue-to-binary Converter for the Moduli Set $\{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1, \ldots, 2^{2^k m} + 1\}$*

WEI WANG[a], M. N. S. SWAMY[a,†], M. O. AHMAD[a] and YUKE WANG[b]

[a]*Department of Electrical and Computer Engineering, Centre for Signal Processing and Communications, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, Que., Canada H3G 1M8;* [b]*Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083-0688, USA*

(Received 13 January 2000; Revised 9 March 2000)

In this paper, a high-speed parallel residue-to-binary converter is proposed for a recently introduced moduli set $S^k = \{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1, \ldots, 2^{2^k m} + 1\}$ for a general value of k. The proposed converter uses simple cyclic shift and concatenation operations and does not require any multiplier. Individual converters for the cases of $k = 0$ and $k = 1$ are derived from the general architecture and compared with those existing in the literature. The converter for S^0 is twice as fast requiring only one-half of the hardware, while that of S^1 is three times as fast, but requiring only 60% of the hardware, as compared to the corresponding ones existing in the literature. Furthermore, the proposed converters are implemented using 0.5-micron CMOS VLSI technology. Based on S^0, the layouts for 8-bit, 16-bit, 32-bit and 64-bit converters are generated, and the corresponding simulation results obtained.

Keywords: VLSI implementation; Computer arithmetic; Parallel algorithm; Residue number system; Chinese remainder theorem; Digital signal processing

INTRODUCTION

During the past decade, the residue number system (RNS) arithmetic has received considerable attention in arithmetic computation and signal processing applications, such as the fast Fourier transform, digital filtering and image processing. The main reasons for this attention are the inherent properties enjoyed by the RNS such as parallelism, modularity, fault tolerance and carry free operations [2–4]. The crucial step for any successful RNS application is the residue-to-binary (R/B) conversion. In recent years, the conversion process has been studied very extensively [5–20].

In order to use the RNS to represent binary numbers, a moduli set has to be chosen. Recently, several new moduli sets have been proposed [5–9]. One such set is $S^k = \{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1, \ldots, 2^{2^k m} + 1\}$, for which the R/B converters for the cases of $k = 0$ and $k = 1$ have been also proposed [5]. According to Ref [5], this moduli set is expected to play an important role in the RNS, since the multiplications in the R/B conversion of this moduli set have been replaced by simple shift operations of signed-digit numbers. It has been shown in Ref. [5] that the R/B converter for S^k is much faster and simpler compared to the existing converters. For 8-bit dynamic range, the FA-based R/B converter of Ref. [10] requires 837 transistors with 51 gate delays while the converter based on S^k of Ref. [5] requires only 510 transistors with 38 gate delays. However, no R/B converter for S^k has been designed so far for $k \geq 2$. Since more than two or three moduli must be considered for large dynamic ranges [11], an introduction of the converter for a general k is essential.

In this paper, we propose a high-speed parallel R/B converter for the general moduli set S^k; this converter also uses no multipliers. Instead of shifting the signed-digit numbers, we use simple cyclic shift and concatenation operations. For the purpose of comparison, the individual converters for the cases of $k = 0$ and $k = 1$ are derived from the general architecture. The new converter for S^0 is twice as fast as the one in Ref. [5] requiring only one-half of the hardware, while that for S^1 is three times as fast as

*Preliminary results contained in this work were presented at the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing [1].

†Corresponding author. Tel.: +1-514-848-3091. Fax: +1-514-848-2802. E-mail: swamy@ece.concordia.ca

ISSN 1065-514X print/ISSN 1563-5171 online
DOI: 10.1080/10655140290010097

the corresponding one in Ref. [5], but requiring only 60% of the hardware. For the same 8-bit dynamic range mentioned above, the proposed converter requires only 220 transistors with 18 gate delays. Furthermore, we implement the proposed converters using 0.5-micron CMOS VLSI technology. Based on the moduli set S^0, layouts of the 8-bit, 16-bit, 32-bit and 64-bit R/B converters are generated and simulation results obtained.

The paper is organized as follows. In the second section, we introduce the necessary background material. In the third section, we propose a parallel converter for the general moduli set S^k. Using these results, we derive new converters for S^0 and S^1 in the fourth section, while in the fifth section, we present VLSI implementation for these converters. In the sixth section, we present the conclusion.

BACKGROUND MATERIAL

For any two numbers X and P_i, $x_i = X \bmod P_i$ is defined as $X = x_i + bP_i$ for some integer b such that $0 \le x_i < P_i$. For convenience, we denote $X \bmod P_i$ by $|X|_{P_i}$.

Residue Number System. A residue number system is defined in terms of a set of relatively prime moduli set $\{P_1, P_2, \ldots, P_n\}$, that is, $\mathrm{GCD}(P_i, P_j) = 1$ for $i \neq j$. A binary number X can be represented as $X = (x_1, x_2, \ldots, x_n)$, where $x_i = |X|_{P_i}$, $0 \le x_i < P_i$. Such a representation is unique for any integer $X \in [0, M-1]$, where $M = P_1 P_2 \ldots P_n$ is the dynamic range of the moduli set $\{P_1, P_2, \ldots, P_n\}$.

To convert $(x_1, x_2, \ldots, x_n)$ into the binary number X, the Chinese remainder theorem (CRT) and mixed radix conversion (MRC) are generally used.

Chinese Remainder Theorem. The binary number X is computed by

$$X = \left| \sum_{i=1}^{n} N_i |N_i^{-1}|_{P_i} x_i \right|_M, \tag{1}$$

where $N_i = M/P_i$ and $|N_i^{-1}|_{P_i}$ is the multiplicative inverse of $|N_i|_{P_i}$ [4].

Mixed Radix Conversion. The number X can be computed by the formula

$$X = \sum_{i=1}^{n} v_i a_i, \tag{2}$$

where $v_i = \prod_{j=1}^{i-1} P_j$ for $2 \le i \le n$ and $v_1 = 1$; the a_i, called the *mixed radix digits*, are computed by the formulas: $Y_1 = X$, $Y_i = (Y_{i-1} - a_{i-1})|P_{i-1}^{-1}|_{P_i}$, $a_i = |Y_i|_{P_i}$.

The MRC approach is a sequential algorithm and is not as "parallel" as the CRT method. Thus, to solve the residue-to-binary conversion problem, the CRT schemes are considered for efficient VLSI implementations [21].

Assuming m and k to be integers, we define the moduli set S^k as $S^k = \{P_1, P_2, \ldots, P_n\} = \{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1, \ldots, 2^{2^k m} + 1\}$ and $M = P_{-1} P_0 \ldots P_k = 2^{2^{k+1} m} - 1$. A binary number X in the dynamic range $[0, M-1]$ is represented as $(x_{-1}, x_0, x_1, \ldots, x_k)$, where x_{-1} is an m-bit binary number, x_i is an $(m2^i + 1)$-bit binary number for $i = 0, 1, \ldots, k$, and $\bar{x}_i$ is the one's complement of x_i. The values of x_{-1}, x_i and $\bar{x}_i$ are given by

$$x_{-1,(m-1)} x_{-1,(m-2)} \ldots x_{-1,1} x_{-1,0} = x_{-1} = |X|_{2^m - 1}, \tag{3a}$$

$$x_{i,2^i m} x_{i,2^i m - 1} \ldots x_{i,1} x_{i,0} = x_i = |X|_{2^{2^i m} + 1}, \tag{3b}$$

$$\overline{x_{i,2^i m}}\ \overline{x_{i,2^i m - 1}} \ldots \overline{x_{i,1} x_{i,0}} = \bar{x}_i. \tag{3c}$$

For the moduli set S^k, the binary number $X = (x_{-1}, x_0, x_1, \ldots, x_k)$ can be computed by the following proposition [5], which has been derived from the CRT.

PROPOSITION 1 [5]

$$X = \left| N_{-1} |N_{-1}^{-1}|_{P_{-1}} x_{-1} + \sum_{i=0}^{k} N_i |N_i^{-1}|_{P_i} x_i \right|_M, \tag{4}$$

where

$$\left| N_{-1} |N_{-1}^{-1}|_{P_{-1}} x_{-1} \right|_M = \left| (2^{2^0 m} + 1)(2^{2^1 m} + 1) \ldots (2^{2^k m} + 1)(2^{m-(k+1)}) x_{-1} \right|_M$$

and

$$\left| N_i |N_i^{-1}|_{P_i} x_i \right|_M = \left| (2^{2^i m} - 1)(2^{2^{i+1} m} + 1) \ldots (2^{2^k m} + 1)(2^{2^i}) x_i \right|_M,$$

$$i = 0, 1, \ldots, k.$$

Based on Proposition 1, the R/B converters for the moduli sets S^0 and S^1 have been proposed in Ref. [5] employing no multipliers, but using signed-digit number operations. The following example illustrates the method.

Example 1 Given $m = 2$, $k = 1$, $S^1 = \{2^2 - 1, 2^{2^0 \times 2} + 1, 2^{2^1 \times 2} + 1\} = \{3, 5, 17\}$ and $M = 255$, we find the binary number $X = (2, 1, 11)$ for S^1 as

$$\begin{aligned} X &= \left| (2^m + 1)(2^{2m} + 1) 2^{m-2} x_{-1} + (2^m - 1)(2^{2m} + 1) 2^{m-2} x_0 + (2^{2m} - 1) 2^{2m-1} x_1 \right|_M \\ &= \left| 2^{4m-1} x_1 + 2^{4m-2}(x_{-1} + x_0) + 2^{3m-2}(x_{-1} - x_0) - 2^{2m-1} x_1 + 2^{2m-2}(x_{-1} + x_0) + 2^{m-2}(x_{-1} - x_0) \right|_{2^{4m-1}} \\ &= \left| 2^7 x_1 + 2^6(x_{-1} + x_0) + 2^4(x_{-1} - x_0) - 2^3 x_1 + 2^2(x_{-1} + x_0) + (x_{-1} - x_0) \right|_{2^8 - 1}. \end{aligned} \tag{5}$$

The computation of X given by Eq. (5) is now carried out in three steps as suggested in Ref. [5]. In Step 1, the multiplications in Eq. (5) are performed by partitioning x_{-1}, x_0 and x_1 into eight sections, while the additions and subtractions are performed by redundant adders/subtractors to output one signed-digit number for each section. These are detailed below.

$$x_{-1} = 10 = x_{-1,1}x_{-1,0}, \quad x_0 = 001 = x_{0,2}x_{0,1}x_{0,0},$$

$$x_1 = 01011 = x_{1,4}x_{1,3}x_{1,2}x_{1,1}x_{1,0}.$$

Section 1: $x_{1,1} + x_{0,2} + x_{-1,0} - x_{0,0} = 1 + 0 + 0 - 1 = 0$,

Section 2: $x_{1,2} + x_{-1,1} - x_{0,1} = 0 + 1 - 0 = 1$,

Section 3: $x_{1,3} + x_{-1,0} + x_{0,0} - x_{0,2} = 1 + 0 + 1 - 0 = 2$,

Section 4: $x_{1,4} - x_{1,0} + x_{-1,1} + x_{0,1} = 0 - 1 + 1 + 0 = 0$,

Section 5: $x_{-1,0} - x_{0,0} - x_{1,1} + x_{0,2} = 0 - 1 - 1 + 0 = -2$,

Section 6: $x_{-1,1} - x_{0,1} - x_{1,2} = 1 - 0 - 0 = 1$,

Section 7: $x_{-1,0} + x_{0,0} - x_{0,2} - x_{1,3} = 0 + 1 - 0 - 1 = 0$,

Section 8: $x_{1,0} + x_{-1,1} + x_{0,1} - x_{1,4} = 1 + 1 + 0 - 0 = 2$.

In Step 2, the above eight signed-digit outputs are converted to binary numbers. Redundant adders/subtractors are used to produce a sum and a carry bit. The carry bit is a signed-digit number in the range $[-2, 2]$ and controls the operation of Step 3. For the case under consideration, $0 + 2^1 \times 1 + 2^2 \times 2 + 0 + 2^4 \times (-2) + 2^5 \times 1 + 0 + 2^7 \times 2 = 100001010$ represents the sum and the carry bit; the sum is 00001010 and the carry bit is 1.

In Step 3, X is generated by adding the carry bit 1 to the sum. Thus, $X = 00001010 + 1 = 1011$.

In order to develop the R/B converter for the general moduli set S^k, we introduce the following definitions.

DEFINITION 1 We define the variables T_1, T_{2i+2} and T_{2i+3} as

$$T_1 = \left|N_{-1}|N_{-1}^{-1}|_{P_{-1}}x_{-1}\right|_M = \left|(2^{2^0m}+1)(2^{2^1m}+1)\ldots(2^{2^km}+1)(2^{m-(k+1)})x_{-1}\right|_M, \tag{6a}$$

$$T_{2i+2} = \left|(2^{2^im})(2^{2^{i+1}m}+1)\ldots(2^{2^km}+1)(2^{2^im-(k-i+1)})x_i\right|_M, \quad i = 0, 1, \ldots, k, \tag{6b}$$

$$T_{2i+3} = \left|(-1)(2^{2^{i+1}m}+1)\ldots(2^{2^km}+1)(2^{2^im-(k-i+1)})x_i\right|_M, \quad i = 0, 1, \ldots, k. \tag{6c}$$

It is easy to see that $T_{2i+2} + T_{2i+3} = |N_i|N_i^{-1}|_{P_i}x_i|_M$ for $i = 0, 1, \ldots, k$ and $X = |T_1 + \sum_{i=0}^{k}(T_{2i+2} + T_{2i+3})|_M$. Since $M = 2^{2^{k+1}m} - 1$, all the T_i's are $(m2^{k+1})$-bit binary numbers. In "R/B converter for S^k section", we will show that each of the T_i's can be generated by concatenation and cyclic shift operations, which are defined below.

DEFINITION 2 Assume n and n_0 to be integers such that $n \geq n_0$. For any n-bit binary number $x = x_{n-1}\ldots x_{n-n_0}x_{n-n_0-1}\ldots x_0$, the modulo multiplication $|x \times 2^{n_0}|_{2^n-1}$ is carried out by moving the highest significant n_0 bits to the lowest significant position, that is, $|x \times 2^{n_0}|_{2^n-1} = x_{n-n_0-1}\ldots x_0x_{n-1}\ldots x_{n-n_0}$. We call this operation, which also has been used in Refs. [3,14,15], a cyclic shift.

DEFINITION 3 We define concatenation of two numbers x_1 and x_2 to be $\langle x_1\rangle\langle x_2\rangle = x_1 2^{m_2} + x_2$, where x_1 is an m_1-bit number and x_2 an m_2-bit number.

DEFINITION 4 For any integer $a > 0$, we denote $[0]^a = \langle 0\rangle\underset{a}{\cdot}\langle 0\rangle$ and $[1]^a = \langle 1\rangle\underset{a}{\cdot}\langle 1\rangle$. Assuming that n, n_0 ($n \geq n_0$) and k_0 are integers, and x is an n_0-bit binary number, we denote

$$[x] = \begin{cases} [0]^{n-n_0}\langle x\rangle & \text{for } n > n_0 \\ x & \text{for } n = n_0 \end{cases}, \tag{7a}$$

$$[\bar{x}] = \begin{cases} [1]^{n-n_0}\langle \bar{x}\rangle & \text{for } n > n_0 \\ \bar{x} & \text{for } n = n_0 \end{cases} \tag{7b}$$

Therefore, from Definitions 3 and 4, we get

$$[x]^{k_0+1} = \underset{k_{0+1}}{[x]\ldots[x]} = \sum_{j=0}^{k_0} 2^{jn}x = x + 2^n x + \ldots + 2^{k_0 n}x, \tag{7c}$$

$$[\bar{x}]^{k_0+1} = (2^{(k_0+1)n} - 1) - [x]^{k_0+1}. \tag{7d}$$

In addition, we need the following well-known results:

$$(2^{2^0n}+1)(2^{2^1n}+1)\ldots(2^{2^an}+1) = \frac{2^{2^{a+1}n}-1}{2^n - 1}, \tag{8}$$

$$\frac{2^{an}-1}{2^n-1} = 1 + 2^n + 2^{2n}\ldots + 2^{(a-1)n} = \sum_{j=0}^{a-1} 2^{jn}. \tag{9}$$

R/B CONVERTER FOR S^K

An R/B converter for the moduli set S^k computes $X = |T_1 + \sum_{i=0}^{k}(T_{2i+2} + T_{2i+3})|_M$. The parallel R/B converter

to be proposed later in this section is based on the following theorem, which presents a method of generating T_1, T_{2i+2} and T_{2i+3} by the concatenation and cyclic shift operations on x_{-1}, x_i and $\bar{x}_i$, respectively.

THEOREM 1

$$T_1 = \langle x_{-1,k}\ldots x_{-1,0}\rangle [x_{-1}]^{2^{k+1}-1}\langle x_{-1,m-1}\ldots x_{-1,k+1}\rangle, \tag{10a}$$

$$T_{2i+2} = \langle x_{i,k-i}\ldots x_{i,0}\rangle [x_i]^{2^{k-i}-1} \times [0]^{m2^i-1}\langle x_{i,m2^i}\ldots x_{i,k-i+1}\rangle, \tag{10b}$$

$$T_{2i+3} = [1]^{k-i}\langle \bar{x}_i\rangle [\bar{x}_i]^{2^{k-i}-1}[1]^{m2^i-(k-i+1)}. \tag{10c}$$

Proof

$$T_1 = \left|(2^{2^0 m}+1)(2^{2^1 m}+1)\ldots(2^{2^k m}+1)(2^{m-(k+1)})x_{-1}\right|_M,$$

$$\text{(by Definition 1, 6a)} = \left|\frac{2^{2^{k+1}m}-1}{2^m-1}(2^{m-(k+1)}x_{-1})\right|_M,$$

$$\text{(using 8)} = \left|\left(\sum_{j=0}^{2^{k+1}-1} 2^{jm}x_{-1}\right)2^{m-(k+1)}\right|_M,$$

$$\text{(using 9)} = \left|[x_{-1}]^{2^{k+1}}2^{m-(k-1)}\right|_{2^{2^{k+1}m}-1}, \text{ (by Definition 4)}$$

$$= \langle x_{-1,k}\ldots x_{-1,0}\rangle [x_{-1}]^{2^{k+1}-1}\langle x_{-1,m-1}\ldots x_{-1,k+1}\rangle,$$

(by Definition 2, 3a)

$$T_{2i+2} = \left|(2^{2^i m})(2^{2^{i+1}m}+1)\ldots(2^{2^k m}+1)(2^{2^i m-(k-i+1)})x_i\right|_M,$$

$$\text{(by Definition 1, 6b)} = \left|\frac{2^{2^{k+1}m}-1}{2^{2^{i+1}m}-1}(2^{2^{i+1}m-(k-i+1)}x_i)\right|_M,$$

$$\text{(using 8)} = \left|\frac{2^{(2^{k-i})2^{i+1}m}-1}{2^{2^{i+1}m}-1}(2^{2^{i+1}m-(k-i+1)}x_i)\right|_M$$

$$= \left|\left(\sum_{j=0}^{2^{k-i}-1} 2^{j(2^{j+1}m)}x_i\right)2^{2^{i+1}m-(k-i+1)}\right|_M,$$

$$\text{(using 9)} = \left|[x_i]^{2^{k-i}}2^{2^{i+1}m-(k-i+1)}\right|_{2^{2^{k+1}m}-1}, \text{(by Definition 4)}$$

$$= \langle x_{i,k-i}\ldots x_{i,0}\rangle [x_i]^{2^{k-i}-1}[0]^{m2^i-1}\langle x_{i,m2^i}\ldots x_{i,k-i+1}\rangle,$$

(by Definition 2, 3b),

$$T_{2i+3} = \left|(-1)(2^{2^{i+1}m}+1)\ldots(2^{2^k m}+1)(2^{2^i m-(k-i+1)})x_i\right|_M,$$

$$\text{(by Definition 1, 6c)} = \left|(-1)[x_i]^{2^{k-i}}2^{2^i m-(k-i+1)}\right|_{2^{2^{k+1}m}-1}$$

$$= \left|(2^{2^{k+1}m}-1-[x_i]^{2^{k-i}})2^{2^i m-(k-i+1)}\right|_{2^{2^{k+1}m}-1},$$

(using the modulo operation)

$$= \left|\left([x_{\bar{i}}]^{2^{k-i}}\right)2^{2^i m-(k-i+1)}\right|_{2^{2^{k+1}m}-1}$$

$$= [1]^{k-i}\langle \bar{x}_i\rangle [\overline{x_i}]^{2^{k-i}-1}[1]^{m2^i-(k-i+1)},$$

(by Definition 2, 3c)

Hence the theorem. □

Note:

1. For T_1, $n = n_0 = m$. Hence, from Definition 4, we have $[x_{-1}] = x_{-1}$.
2. For T_{2i+2}, $n = m2^{i+1}$ and $n_0 = m2^i + 1$. Therefore, from Definition 4, we have $[x_i] = [0]^{m2^i-1}\langle x_i\rangle$.
3. For T_{2i+3}, $n = m2^{i+1}$ and $n_0 = m2^i + 1$. Thus, from Definition 4, $[\overline{x_i}] = [1]^{m2^i-1}\langle \bar{x}_i\rangle$.

Example 2 We now apply Theorem 1 to Example 1 already considered. Recalling that $m = 2$, $k = 1$, $S^1 = (3, 5, 17)$ and $M = 255$, we find the binary number $X = (2, 1, 11)$ for the moduli set S^1 as follows.

$$x_{-1} = 10, \quad x_0 = 001, \quad [x_0] = 0001, \quad \bar{x}_0 = 110,$$

$$[\overline{x_0}] = 1110, \quad x_1 = 01011, \quad \bar{x}_1 = 10100$$

It is observed that it is not necessary to calculate $[x_1]$ and $[x_{\bar{1}}]$, since the term $[x_i]^{2^{k-i}-1}$ is eliminated when $k = i = 1$. Now,

$$T_1 = \langle x_{-1,1}\ldots x_{-1,0}\rangle [x_{-1}]^{2^{1+1}-1}\langle x_{-1,2-1}\ldots x_{-1,1+1}\rangle$$

$$= 10|10|10|10,$$

$$T_2 = \langle x_{0,1-0}\ldots x_{0,0}\rangle [x_0]^{2^{1-0}-1}[0]^{2\times 2^0-1}\langle x_{0,2\times 2^0}\ldots x_{0,1-0+1}\rangle$$

$$= 01|0001|0|0,$$

$$T_3 = [1]^{1-0}\langle \bar{x}_0\rangle [\bar{x}_0]^{2^{1-0}-1}[1]^{2\times 2^0-(1-0+1)} = 1|110|1110,$$

$$T_4 = \langle x_{1,1-1}\ldots x_{1,0}\rangle [x_1]^{2^{1-1}-1}[0]^{2\times 2^1-1}\langle x_{1,2\times 2^1}\ldots x_{1,1-1+1}\rangle$$

$$= 1|000|0101,$$

$$T_5 = [1]^{1-1}\langle \bar{x}_1\rangle [\bar{x}_1]^{2^{1-1}-1}[1]^{2\times 2^1-(1-1+1)} = 10100|111,$$

$$X = |T_1 + T_2 + T_3 + T_4 + T_5|_M$$

$$= |10101010 + 01000100 + 11101110 + 10000101 + 10100111|_{2^8-1} = 1011.$$

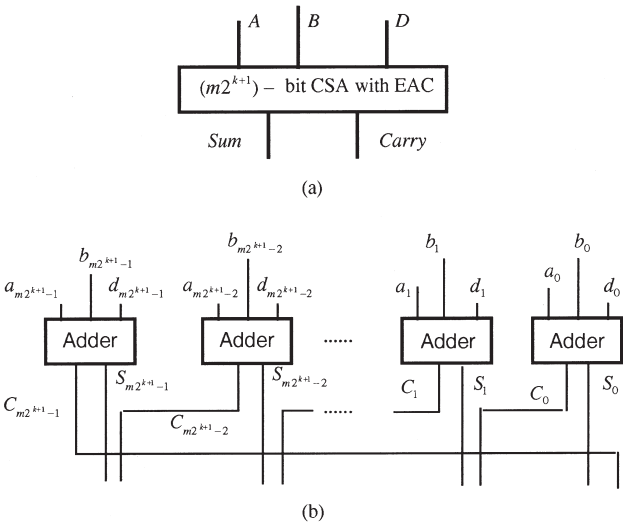

FIGURE 1 The $m2^{k+1}$-bit CSA with EAC: (a) block diagram, (b) detailed architecture.

It is easy to see that the above calculations based on Theorem 1 are very much simpler than those in Example 1. Theorem 1 also enables us to implement a parallel R/B converter for the general moduli set S^k without using multipliers.

The converter to be presented later in this section uses an $m2^{k+1}$-bit carry save adder (CSA) with an end-around carry (EAC) as a fundamental building block. A CSA with EAC consists of a series of adders as shown in Fig. 1. Since each adder accepts a 3-bit input and produces a 2-bit output, the $m2^{k+1}$-bit CSA accepts three $m2^{k+1}$-bit numbers, $A = a_{m2^{k+1}-1}\ldots a_0$, $B = b_{m2^{k+1}-1}\ldots b_0$, and $D = d_{m2^{k+1}-1}\ldots d_0$, and produces two $m2^{k+1}$-bit numbers, namely, Sum and Carry. The highest significant bit in Carry, $C_{m2^{k+1}-1}$, is moved to the lowest significant bit, which is called the end-around carry. Assuming Carry $= C_{m2^{k+1}-2}\ldots C_0 C_{m2^{k+1}-1}$ and Sum $= S_{m2^{k+1}-1}\ldots S_0$, we have $|A+B+D|_{2^{2^{k+1}m}-1} = |\text{Sum} + \text{Carry}|_{2^{2^{k+1}m}-1}$.

For the $(2k+3)$ input numbers $T_1, T_2, \ldots, T_{2k+3}$, we need a CSA tree to reduce these input numbers to a pair of Sum and Carry. This CSA tree consists of $(2k+1)$ CSA's each with an EAC arranged in $\lceil \log_{3/2}(2k+3/2) \rceil$ levels, where $\lceil x \rceil$ stands for the least integer $i \geq x$. Each CSA is of $m2^{k+1}$ bits [22]. This CSA tree is shown in Fig. 2.

The proposed parallel R/B converter for S^k is shown in Fig. 3. It consists of two parts: a CSA tree and a modulo adder. The CSA tree reduces the modulo addition of the $(2k+3)$ T_i's to a pair of Sum and Carry, i.e.

$$\left| T_1 + \sum_{i=0}^{k} (T_{2i+2} + T_{2i+3}) \right|_M = |\text{Carry} + \text{Sum}|_M.$$

Then, the modulo adder adds Sum and Carry together to generate the binary number X.

The CSA tree consists of $(2k+1)$ CSA's, each of which consists of $m2^{k+1}$ full adders or half adders

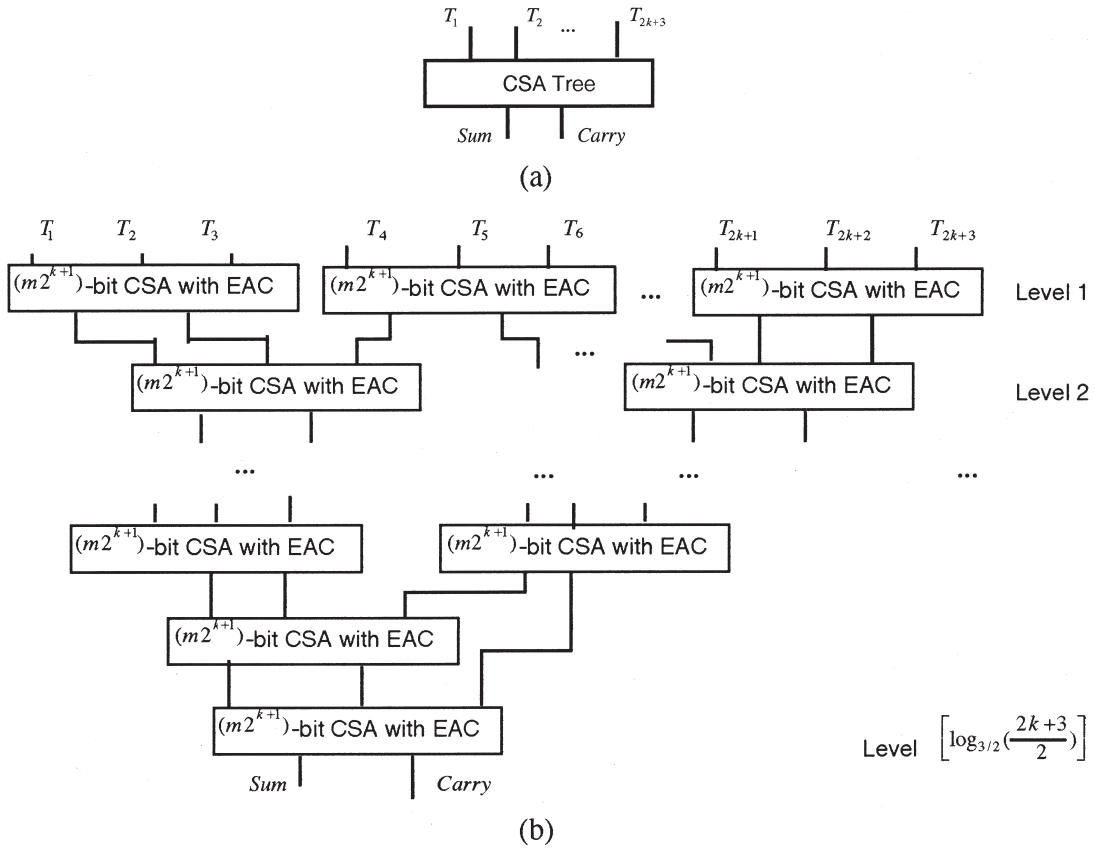

FIGURE 2 The CSA tree: (a) block diagram, (b) detailed architecture.

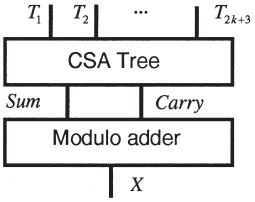

FIGURE 3 The proposed R/B converter.

(FAs/HAs). The modulo adder used in Fig. 3 is the one proposed in Ref. [23] and has an approximate complexity of $m2^{k+1}$ full adders. Thus, the converter has a total of $(2k+2)m2^{k+1}$ FAs/HAs.

The CSA tree has $\lceil \log_{3/2}(2k+3/2) \rceil$ levels, each having a delay, t_{FA}, of a full adder [24]. The delay of the modulo adder is approximately $m2^{k+1}t_{\mathrm{FA}}$ [23]. Thus, the total delay of the converter is $(\lceil \log_{3/2}(2k+3/2) \rceil + m2^{k+1})t_{\mathrm{FA}}$.

TWO SPECIAL CASES

R/B Converter For S^0

The moduli set $S^0 = \{2^m - 1, 2^{2^0 m} + 1\}$ with $M = 2^{2m} - 1$ is a special case of the moduli set S^k ($k = 0$). A binary number X in the dynamic range $[0, M-1]$ is represented by (x_{-1}, x_0). By Theorem 1, X is computed as follows:

$$X = |T_1 + T_2 + T_3|_{2^{2m}-1}$$

$$T_1 = \langle x_{-1,0} \ldots x_{-1,0} \rangle [x_{-1}]^{2^{0+1}-1} \langle x_{-1,m-1} \ldots x_{-1,0+1} \rangle$$

$$= x_{-1,0}x_{-1,(m-1)} \ldots x_{-1,1}x_{-1,0}x_{-1,(m-1)} \ldots x_{-1,1}$$

$$T_2 = \langle x_{0,0-0} \ldots x_{0,0} \rangle [x_0]^{2^{0-0}-1} [0]^{m2^0-1} \langle x_{0,m2^0} \ldots x_{0,0-0+1} \rangle$$

$$= \langle x_{0,0} \rangle [0]^{m-1} \langle x_{0,m} \ldots x_{0,1} \rangle$$

$$T_3 = [1]^{0-0} \langle x_\theta \rangle [x_\theta]^{2^{0-0}-1} [1]^{m2^0-(0-0+1)}$$

$$= \langle \overline{x_{0,m}x_{0,m-1}} \ldots \overline{x_{0,0}} \rangle [1]^{m-1}$$

For the moduli set S^0, we obtain the converter from the general architecture of Fig. 3. This converter consists of a $2m$-bit CSA with EAC and a $2m$-bit modulo adder. This is shown in Fig. 4.

Out of the $2m$ adders needed for the $2m$-bit CSA with EAC, $(2m-2)$ adders are half adders, since T_2 has

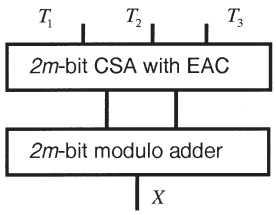

FIGURE 4 Proposed converter for S^0.

TABLE I Performance comparison of the converters for S^0

	Transistors	Delay
Proposed converter	$60m + 20$	$(4m+2)\Delta$
Converter in Ref. [5]	$120m + 30$	$(9m+2)\Delta$

$(m-1)$ bits of 0's and T_3 has $(m-1)$ bits of 1's [15]. Hence, the CSA has 2 full adders and $(2m-2)$ half adders. As mentioned in the previous section, the $2m$-bit modulo adder is equivalent to $2m$ full adders in terms of complexity. Thus, the complexity measured by the number of transistors of the proposed converter for S^0 is equivalent to $2 \times 20 + (2m-2) \times 10 + 2m \times 20 = 60m + 20$ transistors, using Table I of Ref. [5].

The delay calculation is carried out in a similar manner. Table I gives a performance comparison of the proposed converter with the one in Ref. [5], where Δ refers to the delay of a logic gate. It is clear from Table I that the proposed converter is twice as fast as the converter in Ref. [5], but requiring only one-half of the hardware. For 8-bit dynamic range ($m = 4$), the FA-based R/B converter of Ref. [10] requires 837 transistors with 51 gate delays, the converter based on S^k of Ref. [5] requires 510 transistors with 38 gate delays, while the proposed converter requires only 220 transistors with 18 gate delays.

R/B Converter For S^1

The moduli set $S^1 = \{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1\}$ with $M = 2^{4m} - 1$ is another special case of the moduli set S^k ($k = 1$). A binary number X in the dynamic range $[0, M-1]$ is represented by (x_{-1}, x_0, x_1). By Theorem 1, X is computed as follows:

$$X = |T_1 + T_2 + T_3 + T_4 + T_5|_{2^{4m}-1}$$

$$T_1 = \langle x_{-1,1} \ldots x_{-1,0} \rangle [x_{-1}]^{2^{1+1}-1} \langle x_{-1,m-1} \ldots x_{-1,1+1} \rangle$$

$$= \langle x_{-1,1}x_{-1,0} \rangle [x_{-1}]^3 \langle x_{-1,(m-1)} \ldots x_{-1,3}x_{-1,2} \rangle$$

$$T_2 = \langle x_{0,1-0} \ldots x_{0,0} \rangle [x_0]^{2^{1-0}-1} [0]^{m2^0-1} \langle x_{0,m2^0} \ldots x_{0,1-0+1} \rangle$$

$$= \langle x_{0,1}x_{0,0} \rangle [0]^{m-1} \langle x_{0,m} \ldots x_{0,0} \rangle [0]^{m-1} \langle x_{0,m} \ldots x_{0,2} \rangle$$

$$T_3 = [1]^{1-0} \langle x_\theta \rangle [\overline{x_0}]^{2^{1-0}-1} [1]^{m2^0-(1-0+1)}$$

$$= \langle 1\overline{x_{0,m}} \ldots \overline{x_{0,0}} \rangle [1]^{m-1} \langle \overline{x_{0,m}} \ldots \overline{x_{0,0}} \rangle [1]^{m-2}$$

$$T_4 = \langle x_{1,1-1} \ldots x_{1,0} \rangle [x_1]^{2^{1-1}-1} [0]^{m2^1-1} \langle x_{1,m2^1} \ldots x_{1,1-1+1} \rangle$$

$$= \langle x_{1,0} \rangle [0]^{2m-1} \langle x_{1,2m} \ldots x_{1,2}x_{1,1} \rangle$$

$$T_5 = [1]^{1-1} \langle x_\dagger \rangle [x_\dagger]^{2^{1-1}-1} [1]^{m2^1-(1-1+1)}$$

$$= \langle \overline{x_{1,2m}} \ldots \overline{x_{1,1}x_{1,0}} \rangle [1]^{2m-1}$$

For the moduli set S^1, we obtain the converter again from the general architecture of Fig. 3. This converter

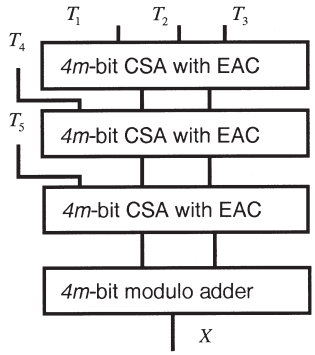

FIGURE 5 Proposed converter for S^1.

TABLE II Performance comparison of the converters for S^1

	Transistors	Delay
Proposed converter	$240m + 60$	$(8m + 6)\Delta$
Converter in Ref. [5]	$400m + 20$	$(28m + 5)\Delta$

consists of three $4m$-bit CSA's with EAC and a $4m$-bit modulo adder. This is shown in Fig. 5.

The performance of the proposed converter is compared with that of the converter in Ref. [5] using the same measures (namely, the number of transistors and delays) as used in "R/B converter S^0 section". These results are given in Table II. It is evident from Table II that the proposed converter is three times as fast, while requiring only 60% of the hardware.

VLSI IMPLEMENTATION

The VLSI implementation of the proposed converters is carried out using 0.5-micron CMOS technology. The design flow is as follows [25, 26]. First, we construct a new library to include all the building blocks mentioned in Sections "R/B converter for S^k" and "Two special cases". Then, the architectures of the converters are coded in VHDL language using this library. The codes are simulated at the RTL level to verify the correctness of the design. The logic synthesis is carried out to optimize the design, and the gate-level simulation performed. Next, the placement and routing are carried out automatically to generate the layout. Finally, the performance analysis at the layout level is carried out. The software packages used include Cadence V4.4.1 (Release 9504), Synopsys V3.4b, CMC Generic Environment for Cadence V1.7, CMC CMOSIS5 Design Kit V2.2 for Cadence, and CMC CMOSIS5 Design Kit V1.0 for Synopsys.

Following the design flow, the R/B converters based on S^k can be implemented for any given k, for the 8-bit, 16-bit, 32-bit and 64-bit dynamic ranges. Here, we only show the implementation of the converters based on S^0. For this implementation, the $2^8 - 1$, $2^{16} - 1$, $2^{32} - 1$, and $2^{64} - 1$ ranges are chosen instead of the four entire binary ranges. The reason is that these ranges are efficient for S^0 and the entire binary ranges can be accommodated by the proposed architectures with an additional combinational circuit or ROM for the overflow cases; this will introduce additional complexity. The implementation of the 8-bit, 16-bit, 32-bit and 64-bit converters of S^0 for the $2^8 - 1$, $2^{16} - 1$, $2^{32} - 1$, and $2^{64} - 1$ ranges based on the special sets are shown in Table III.

Results concerning the area and time performance of these four converters are obtained using Cadence V4.4.1 (Release 9504) tools and summarized in Table IV. The core area refers to the area of the circuit layout, the chip area refers to the area of the circuit and input/output/power pads, while the time indicates the latency of the chip. No comparison of the performance has been carried out with the existing converter for S^0 in Ref. [5], since no implementation data is available in Ref. [5].

CONCLUSION

A high-speed parallel R/B converter for the general moduli set $S^k = \{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1, \ldots, 2^{2^k m} + 1\}$ has been proposed. The new converter uses no multipliers. The individual converters for the moduli sets S^0 and S^1 have been derived from the general architecture. The R/B converter for S^0 is twice as fast requiring only one-half of the hardware, while that for S^1 is three times faster requiring only 60% of the hardware,

TABLE III Specific sets for 8-bit, 16-bit, 32-bit and 64-bit converter of S^0

Dynamic range	$[0, 2^8 - 2]$	$[0, 2^{16} - 2]$	$[0, 2^{32} - 2]$	$[0, 2^{64} - 2]$
Moduli set $\{2^m - 1, 2^m + 1\}$	$\{15, 17\}$	$\{255, 257\}$	$\{2^{16} - 1, 2^{16} + 1\}$	$\{2^{32} - 1, 2^{32} + 1\}$

TABLE IV Layout performance of the four R/B converters for S^0

	Moduli set	Core area ($\mu m \times \mu m$)	Chip area ($\mu m \times \mu m$)	Time (ns)
8-bit converter	$\{15, 17\}$	164×352.8	1383.9×1540.9	52.74
16-bit converter	$\{255, 257\}$	311.3×771.3	2265.8×3451.1	60.1
32-bit converter	$\{2^{16} - 1, 2^{16} + 1\}$	339.2×857.7	3302.2×4735.7	79.1
64-bit converter	$\{2^{32} - 1, 2^{32} + 1\}$	454.9×1091.7	4217.7×5704.3	105.04

compared to the corresponding ones in Ref. [5]. VLSI implementation of the proposed converter has been carried out in 0.5-micron CMOS technology. Based on S^0, layouts of the 8-bit, 16-bit, 32-bit and 64-bit R/B converters have been generated and the results on the performance of these converters obtained through simulation.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada, the MICRONET National Network of Centers of Excellence, and Fonds pour la Formation de Chercheurs et l'Aide a Recherche of Quebec.

References

[1] Wang, W., Swamy, M.N.S., Ahmad, M.O. and Wang, Y. (1999) "A parallel residue-to-binary converter", in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arisona.
[2] Garner, H.L. (1959) "The residue number system", *IRE Trans. Electronic Computers* **8**, 140–147.
[3] Szabo, N. and Tanaka, R. (1967) *Residue Arithmetic and its Applications to Computer Technology* (McGraw-Hill, New York).
[4] Soderstrand, M.A., Ed. (1986) *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing* (IEEE Press, New York).
[5] Pourbigharaz, F. and Yassine, H.M. (1997) "A signed-digit architecture for residue to binary transformation", *IEEE Trans. Computers* **46**(10), 1146–1150.
[6] Hiasat, A.A. and Abdel-Aty-Zohdy, Hoda S. (1998) "Residue-to-Binary Arithmetic Converter for the Moduli Set (2^k, $2^k - 1$, $2^{k-1} - 1$)", *IEEE Trans. Circuits Syst.* **45**, 204–208.
[7] Premkumar, A.B. (1992) "An RNS to binary converter in ($2n + 1$, $2n$, $2n - 1$) moduli set", *IEEE Trans. Circuits Syst. II* **39**, 480–482.
[8] Premkumar, A.B. (1995) "An RNS to binary converter in a three moduli set with common factors", *IEEE Trans. Circuits Syst. II* **42**, 298–301.
[9] Cardarilli, G.C., Re, M. and Lojacona, R. (1998) "RNS-to-binary conversion for efficient VLSI implementation", *IEEE Trans. Circuits Syst. II* **45**(6).
[10] Stouraitis, T. (1992) "Efficient converters for residue and quadratic-residue number systems", *IEE Proc. Circuits, Devices, Syst.* **139**(6), Part G.
[11] Skavantzos, Alexander (1998) "An efficient residue to weighted converter for a new residue number system", *Proc. 8th Great Lakes Symp. on VLSI*, 185–191.
[12] Dhurkadas, A. (1990) "Comments on an efficient residue to binary converter design", *IEEE Trans. Circuits Syst.* **37**, 849–850.
[13] Jones, E.V. (1988) "Fast conversion between binary and residue numbers", *Electronics Lett.*, 1195–1198.
[14] Ibrahim, K. and Saloum, S. (1988) "An efficient residue to binary converter design", *IEEE Trans. Circuits. Syst.* **35**, 1156–1158.
[15] Piestrak, Stanislaw J. (1995) "High-speed realization of a residue to binary number system converter", *IEEE Trans. Circuits Syst. II* **42**(10), 661–663.
[16] Andraos, Sweidan and Ahmad, Hiasat (1988) "A new efficient memoryless residue to binary converter", *IEEE Trans. Circuits Syst.* **35**(11), 1441–1444.
[17] Wang, Yuke and Abd-el-Barr, M. (1996) "A new algorithm for RNS decoding", *IEEE Trans. Circuits Syst. I* **43**(12).
[18] Wang, Yuke, Song, Xiaoyu and Aboulhamid, M. (1998) "Near-optimal residue to binary converter for the moduli", *8th Great Lakes Symp. on VLSI*.
[19] Wang, Yuke, Swamy, M.N.S. and Omair Ahmad, M. (1999) "Residue to binary number converters for three moduli sets", *IEEE Trans. Circuits Syst. II*, **46**(2), 180–183.
[20] Yuke Wang. (1998) "New Chinese Remainder Theorems", Asilomar Conference, USA, 165–171.
[21] Srikanthan, T., Bhardwaj, M. and Clarke, C.T. (1998) "Area-time-efficient VLSI residue-to-binary converters", *IEE Proc. -Comput. Digit. Tech.* **145**(3), 229–235.
[22] Koern, Israel (1993) *Computer Arithmetic Algorithms* (Prentice Hall, Englewood Cliffs, NJ).
[23] Efstathiou, C., Nikolos, D. and Kalamatianos, J. (1994) "Area-time efficient modulo 2k − 1 adder design", *IEEE Trans. Circuits Syst. II* **41**, 463–466.
[24] Piestrak, Stanislaw J. (1994) "Design of residue generators and multi-operand modular adders using carry-save adders", *IEEE Trans. Comput.* **423**(1), 68–77.
[25] Canadian Microelectronics Corporation(1997), *Basic Digital IC Design Flow Instruction*.
[26] SYNOPSYS, Ref. Man. 3.0. SYNOPSYS Inc., 1992.

Authors' Biographies

Wei Wang received his B.Sc. (Electrical and Computer Engineering) degree in 1992 from the Beijing University of Aeronautics, China. From Sept. 1992 to Feb. 1997, he worked as an electronics and computer engineer in the China Precision Engineering Institute, China. Feb. 1997 to Feb. 1998, he was the manager of the technical department at American Amerihua Company. Since Feb. 1998, Wei Wang has been a graduate student pursuing his Ph.D. degree in the department of Electrical and Computer Engineering at Concordia University, Canada. In 1996, he was awarded the Science Progress Award by China Aviation Ministry. His research interests include computer arithmetic, VLSI and digital signal processing.

M. N. S. Swamy received the B.Sc. (Hons.) degree in mathematics from Mysore University, India, in 1954, the Diploma in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1957, and the M.Sc. and Ph.D. Degrees in electrical engineering from the University of Saskatchewan, Saskatoon, Sask., Canada, in 1960 and 1963, respectively. He is presently a Research professor and the Director of the Centre for Signal Processing and Communications, Department of Electrical and Computer Engineering, Concordia University, Montreal, P.Q., Canada, where he served as the Chairman of the Department of Electrical Engineering from 1970 to 1977 and Dean of Engineering and Computer Science from 1977 to 1993. He has also taught in the Electrical Engineering Departments at the Technical University of Nova Scotia, Halifax, N.S., and the University of Calgary, Calgary, Alta., Canada, as well as in the Department of Mathematics at the University of Saskatchewan. He has published extensively in the areas of circuits, systems, and signal processing. He is co-author of two book chapters and three books: Graphs, Networks, and algorithms (New York: Wiley, 1981); Graphs: Theory and Algorithms (New York: Wiley, 1992) and Switched Capacitor Filters: Theory, Analysis and Design (Englewood Cliffs, NJ: Prentice-Hall, 1995). A Russian translation of the first book was published by Mir Publishers, Moscow, in 1984, while a Chinese version was published by the Education Press, Beijing, in 1987. He

holds four patents. Dr Swamy is a Fellow of the Institute of Electrical and Electronics Engineers, the Engineering Institute of Canada, the Institution of Engineers (India), and the Institution of Electronics and Telecommunications Engineering (India). Presently, he is the Editor-in-Chief of Circuits, Systems, and Signal processing, and an Associate Editor of the Fibonacci Quarterly. He was Vice President of the IEEE Circuits and Systems (CAS) Society in 1976, Program Chairman for the 1973 IEEE CAS Symposium, and General Chairman for the 1984 IEEE CAS Symposium. He was Editor-in-Chief of the IEEE Transactions on Circuits and Systems I from 1999 to 2001. He is co-recipient (with Dr L.M. Roytman and Dr E.I. Plotkin) of the IEEE CAS 1986 Guillemin-Cauer Best Paper Award. He is a member of the Micronet National Network of Centres of Excellence. He was recently awarded a DSc (Honoris causa) by Ansted University.

M. Omair Ahmad received the B.Eng. degree from Sir George William University, Montreal, P.Q., Canada. and the Ph.D. degree from Concordia University, Montreal, both in electrical engineering. From 1978 to 1979, he was a member of the faculty of the New York State University College, Buffalo. In September 1979, he joined the faculty of Concordia University, where he was an Assistant Professor of Computer science. Subsequently, he joined the university's Department of Electrical and Computer Engineering, where he is presently a Professor. He has published extensively in the area of signal processing. His current research interests include the areas of multidimensional filter design, image and video signal processing, nonlinear signal processing, DSP algorithms, and VLSI circuits for DSP applications. He holds three patents.Dr Ahmad was the Local Arrangements Chairman of the 1984 IEEE International Symposium on Circuits and Systems. During 1988, he was a member of the Admission and Advancement Committee of the IEEE. He was an examiner in the Order of Engineers of Quebec. From 1999 to 2001, he was an Associate Editor of the IEEE Transactions on Circuits and Systems I and is presently Chairman of the Circuits and Systems Chapter of the Montreal Section of the IEEE. He is a member of the Micronet National Network of Centers of Excellence. He is a recipient of the Wighton Fellowship awarded by the Sandford Fleming Foundation. He is a Fellow of the Institute of Electrical and Electronics Engineers.

Yuke Wang received a B.Sc. (Mathematics, 1989) from the University of Science and Technology of China, an M.Sc. (Mathematics, 1992) and a Ph.D. (computer science, 1996) from the University of Saskatchewan, Saskatoon, Canada. From 1997 to 1999, he was an assistant professor in the Department of Electrical and Computer Engineering of Concordia University in Montreal, Canada. He is currently with the Yuke Wang is with the Department of Computer Science, University of Texas at Dallas, Richardson, Texas, USA. His main interests are VLSI (logic optimization, data structures, and algorithms) an related areas.