A PARALLEL RESIDUE-TO-BINARY CONVERTER

Wei Wang, M.N.S. Swamy, M.O. Ahmad and Yuke Wang

Centre for Signal Processing and communications

Department of Electrical and Computer Engineering, Concordia University

Montreal, Quebec, Canada H3G 1M8

ABSTRACT

In this paper, a high-speed parallel residue-to-binary converter is proposed for the 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 replaces the multiplications of the residue-to-binary conversion by simple cyclic shift and concatenation operations. For the purpose of comparison, the individual converters for the cases of k=0 and 1 are derived from the general architecture. The converter for S^0 is twice as fast as the previous converter using only one-half of the hardware, while that of S^1 is three times as fast, but requiring only 60% of the hardware.

1. INTRODUCTION

During the past decade, residue number system (RNS) arithmetic has received considerable attention in arithmetic computation and signal processing applications due to the inherent properties of the RNS such as parallelism, modularity, fault tolerance and carry-free operations [6],[8],[9]. 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 [1]-[5],[7],[10]-[13].

The moduli set $S^k=\{2^m-1,\ 2^{2^0 m}+1,\ 2^{2^1 m}+1,\ \ldots,2^{2^k m}+1\}$ for RNS applications was recently introduced in [5]; the R/B converters for the cases of k=0 and 1 were also proposed in the same article. These converters are simple and fast, since the multiplications in the R/B conversion have been replaced by simple shift operations of signed-digit numbers. However, no R/B converter for S^k has been designed for $k\geq 2$. Since more than two or three moduli must be considered for large dynamic ranges [2], an introduction of the converter for a general k is very 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, two individual converters for S^0 and S^1 are derived from the general architecture. The new converter for S^0 is twice as fast as the one in [5] requiring only one-half of the hardware, while that for S^1 is three times as fast as the corresponding one in [5], but requiring only 60% of the hardware.

2. BACKGROUND MATERIAL

A residue number system is defined in terms of a set of relatively prime moduli set $(P_1,P_2,\ldots,P_k)$, that is, (P_i,P_j)=1 for $i\neq j$. A binary number X can be represented as $X=(x_1,x_2,\ldots,x_k)$, where $x_i=X \bmod P_i$ and $0\leq x_i<P_i$. Such a representation is unique for any integer $X\in[0,M-1]$, where $M=P_1P_2\ldots P_k$ is the dynamic range of $(P_1,P_2,\ldots,P_k)$.

To convert $(x_1,x_2,\ldots,x_k)$ into the binary number X, the Chinese Remainder Theorem (CRT) is generally used. We define $X \bmod P_i$ by X_{P_i} and $\left|P_i^{-1}\right|_{P_j}$ to be the multiplicative inverse of $P_i \bmod P_j$, if $\left|P_i^{-1}\right|_{P_j} * P_i=1 \bmod P_j$.

Chinese Remainder Theorem The binary number X is computed by $X=\left|\sum_{i=1}^{k} N_i\left|N_i^{-1}\right|_{P_i} x_i\right|_M$, where $N_i=\frac{M}{P_i}$ and $\left|N_i^{-1}\right|_{P_i}$ is the multiplicative inverse of $N_i \bmod P_i$.

Assuming m and k to be integers, we define the moduli set S^k as $S^k=\{2^m-1,\ 2^{2^0 m}+1,\ 2^{2^1 m}+1,\ \ldots,2^{2^k m}+1\}=(P_{-1},P_0,P_1,\ldots,P_k)$ 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 and x_i and $\bar{x}_i$ are $(m2^i+1)$-bit binary numbers for i=0,1,…,k. The values of x_{-1}, x_i and $\bar{x}_i$ are given by,

$$x_{-1,(m-1)}x_{-1,(m-2)}\ldots x_{-1,1}x_{-1,0}=x_{-1}=X \bmod (2^m-1) \quad (1a)$$

$$x_{i,2^i m}x_{i,2^i m-1}\ldots x_{i,1}x_{i,0}=x_i=X \bmod (2^{2^i m}+1) \quad (1b)$$

$$\bar{x}_{i,2^i m}\bar{x}_{i,2^i m-1}\ldots \bar{x}_{i,1}\bar{x}_{i,0}=\bar{x}_i=(-X) \bmod (2^{2^i m+1}-1) \quad (1c)$$

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 of [5], which has been derived from the CRT.

Proposition 1 [5] For $i=0,1,\ldots,k$,

$$X=\left|N_{-1}\left|N_{-1}^{-1}\right|_{P_{-1}} x_{-1}+\sum_{i=0}^{k} N_i\left|N_i^{-1}\right|_{P_i} x_i\right|_M \quad (2)$$

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

$$\left|\sum_{i=0}^{k} N_i \left|N_i^{-1}\right|_{P_i} x_i\right|_M = \left|(2^{2^i m}-1)(2^{2^{i+1}m}+1)\cdots(2^{2^k m}+1)(2^{2^i m-(k-i+1)})x_i\right|_M .$$

The R/B converters based on Proposition 1 for the moduli sets S^0 and S^1 have been proposed in [5] using signed-digit numbers without multipliers. The following example illustrates this method.

Example 1 Given $m=2$, $k=1$, $S^1=\{3,5,17\}$ and $M=255$, we find the binary number $X=(2,1,11)$ by $X=\left|2^7 x_1+2^6(x_{-1}+x_0)+2^4(x_{-1}-x_0)-2^3 x_1+2^2(x_{-1}+x_0)+(x_{-1}-x_0)\right|_{2^8-1}$

The computation of X is now carried out in three steps as suggested in [5]. In Step 1, the multiplications are performed by shifting 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.

Section 1: 1+0+0-1=0 Section 2: 0+1-0=1
Section 3: 1+0+1-0=2 Section 4: 0-1+1+0=0
Section 5: 0-1-1+0=-2 Section 6: 1-0-0=1
Section 7: 0+1-0-1=0 Section 8: 1+1+0-0=2

In Step 2, the eight signed-digit outputs are converted into binary numbers by redundant adders/subtracters. $0+2^1*1+2^2*2+0+2^4*(-2)+2^5*1+0+2^7*2=100001010$ Thus, the sum is 00001010 and the carry-out bit is 1.

In Step 3, X is generated by adding 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 need the following definitions.

Definition 1 We define the variables T_1, T_{2i+2} and T_{2i+3} for $i=0,1,\cdots,k$ by

$$T_1=\left|(2^{2^0 m}+1)(2^{2^1 m}+1)\cdots(2^{2^k m}+1)(2^{m-(k+1)})x_{-1}\right|_M \quad (3a)$$

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

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

It is easy to see that $T_1=\left|N_{-1}\left|N_{-1}^{-1}\right|_{P_{-1}} x_{-1}\right|_M$, $T_{2i+2}+T_{2i+3}=\left|N_i\left|N_i^{-1}\right|_{p_i} x_i\right|_M$, and $X=\left|T_1+\sum_{i=0}^{k}(T_{2i+2}+T_{2i+3})\right|_M$ for $i=0,1,\cdots,k$. Since $M=2^{2^{k+1}m}-1$, all the T_i's are $(m2^{k+1})$-bit binary numbers. In Section 3, 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 $n \ge n_0$. For any n-bit binary number $x=x_{n-1}\cdots x_{n-n_0}x_{n-n_0-1}\cdots x_0$, the modulo multiplication $\left|x*2^{n_0}\right|_{2^n-1}$ is accomplished by moving the highest significant n_0 bits to the lowest significant position, i.e., $\left|x*2^{n_0}\right|_{2^n-1}= x_{n-n_0-1}\cdots x_0 x_{n-1}\cdots x_{n-n_0}$. We call this operation "cyclic shift".

Definition 3 We define the operation of "concatenation" of two numbers x_1 and x_2 to be $<x_1><x_2>=x_1 2^{m_2}+x_2$, 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 $[0]^a=\underbrace{<0><0>\cdots<0>}_{a}$ and $[1]^a=\underbrace{<1><1>\cdots<1>}_{a}$. Assuming that n, n_0 $(n \ge n_0)$ and k_0 are integers, and x is an n_0-bit binary number, we denote

$$[x]=\begin{cases}[0]^{n-n_0}<x> & \text{for } n>n_0\\ x & \text{for } n=n_0\end{cases}$$

and $$\overline{[x]}=\begin{cases}[1]^{n-n_0}<\bar{x}> & \text{for } n>n_0\\ \bar{x} & \text{for } n=n_0\end{cases}$$

Therefore, from Definitions 3 and 4, we get

$$[x]^{k_0+1}=\underbrace{[x][x]\cdots[x]}_{k_0+1}=\sum_{j=0}^{k_0}2^{jn}x = x+2^n x+\cdots+2^{k_0 n}x$$

$$\overline{[x]}^{k_0+1}=(2^{(k_0+1)n}-1)-[x]^{k_0+1}$$

3. R/B CONVERTER FOR S^k

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

$$T_1=<x_{-1,k}\cdots x_{-1,0}>\left[x_{-1}\right]^{2^{k+1}-1}<x_{-1,m-1}\cdots x_{-1,k+1}>$$

$$T_{2i+2}=<x_{i,k-i}\cdots x_{i,0}>\left[x_i\right]^{2^{k-i}-1}[0]^{m2^i-1}<x_{i,m2^i}\cdots x_{i,k-i+1}>$$

$$T_{2i+3}=[1]^{k-i}<\bar{x}_i>\left[\bar{x}_i\right]^{2^{k-i}-1}[1]^{m2^i-(k-i+1)}$$

where $[x_{-1}]=x_{-1}$, $[x_i]=[0]^{m2^i-1}<x_i>$,

$$\overline{[x_i]}=[1]^{m2^i-1}<\bar{x}_i>.$$

The proof is omitted for lack of space.

Example 2 We apply Theorem 1 to Example 1. Then

$$T_1=10|10|10|10,\ T_2=01|0001|0|0,\ T_3=1|110|1110,$$

$$T_4=1|000|0101,\ T_5=10100|111$$

$$X=\left|T_1+T_2+T_3+T_4+T_5\right|_M=1011$$

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.

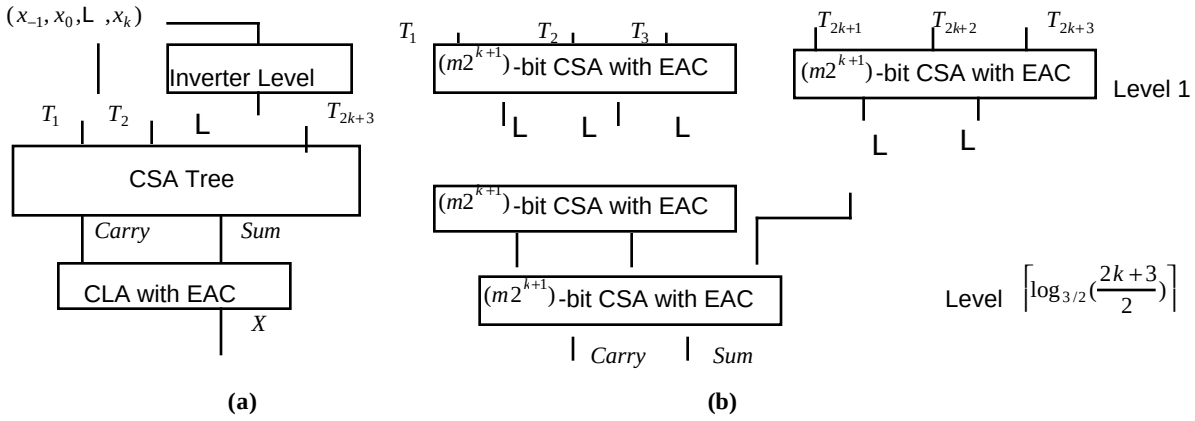


Figure 1. (a) Proposed R/B converter (b) CSA tree

The proposed parallel R/B converter for S^k is shown in Fig. 1 (a). It has three parts: an inverter level, a CSA tree and a carry look-ahead adder (CLA) with an end-around carry (EAC). The inverter level generates $\bar{x}_i$ from x_i for $i = 0, 1, \ldots, k$. The CSA tree consists of $2k+1$ CSAs with EAC in $\left\lceil \log_{3/2}(\frac{2k+3}{2}) \right\rceil$ levels [7] as shown in Fig. 1 (b); each CSA with EAC is an $(m2^{k+1})$-bit adder. This tree reduces the modulo addition of the $(2k+3)$ T_i's to a sum and a carry. Then the CLA with EAC adds the sum and carry together to generate the binary number X.

The inverter level has $\sum_{i=0}^{k}(m2^i+1) = m(2^{k+1}-1)+(k+1)$ inverters. Each of the CSAs consists of $m2^{k+1}$ full adders or half adders (FAs/HAs). The CLA with EAC used is the one proposed in [3] and approximately has the complexity of $m2^{k+1}$ FAs. Thus, the converter has $(2k+2)m2^{k+1}$ FAs/HAs and $m(2^{k+1}-1)+(k+1)$ inverters.

The delay of the inverter level is that of one inverter, t_{INV}. The CSA tree has $\left\lceil \log_{3/2}(\frac{2k+3}{2}) \right\rceil$ levels, each of which has a delay of an FA, t_{FA}. The delay of the CLA with EAC is approximately $m2^{k+1}t_{FA}$ [3]. Thus, the total delay of the converter is $t_{INV} + (\left\lceil \log_{3/2}(\frac{2k+3}{2}) \right\rceil + m2^{k+1})\, t_{FA}$.

4. TWO SPECIAL CASES

4.1 The R/B Converter for S^0

By Theorem 1, a binary number X based on the moduli set $S^0 = \{2^m - 1, 2^{2^0 m} + 1\}$ is computed as follows.

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

$$T_1 = x_{-1,0}x_{-1,(m-1)} \ldots x_{-1,1}x_{-1,0}x_{-1,(m-1)} \ldots x_{-1,1}$$

$$T_2 = < x_{0,0} > [0]^{m-1} < x_{0,m} \ldots x_{0,1} >$$

$$T_3 = < \bar{x}_{0,m} \bar{x}_{0,m-1} \ldots \bar{x}_{0,0} > [1]^{m-1}$$

For the moduli set S^0, we obtain the converter from the general architecture of Fig. 1. This converter consists of a *2m*-bit CSA with EAC and a *2m*-bit CLA with EAC. This is shown in Fig. 2 (a). Fig. 2 (b) shows the block diagram of the corresponding converter developed in [5] and is included here for the sake of comparison. It is easy to see that the proposed converter saves one 2m-bit carry propagation adder (CPA).

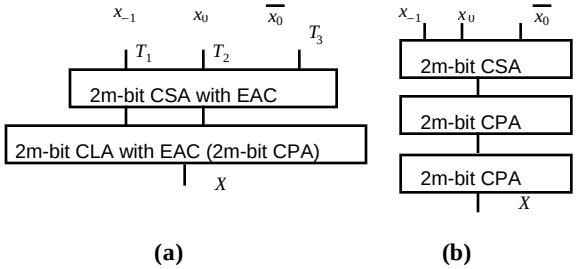


Figure 2. (a) Proposed converter (b) The converter in [5]

The 2m-bit CLA needs 2m FAs [3], while the *2m*-bit CSA with EAC needs 2m adders. Since T_2 has $(m-1)$ bits of "0" and T_3 has $(m-1)$ bits of "1", $(2m-2)$ of the FAs are reduced to half adders [11]. Hence, the CSA has 2 FAs and $(2m-2)$ HAs. As in [5], the inverter level is not considered in the performance evaluation, since its contribution is negligible. Using Table 1 of [5], the number of transistors used is calculated to be

$$2 * 20 + (2m-2) * 10 + 2m * 20 = 60m + 20$$

The delay calculation is carried out in a similar manner. The performance of the proposed converter and the one in [5] is compared in Table 1.

Table 1 Comparison of the converters for S^0

Δ = Delay of an HA

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

Thus, the proposed converter is twice as fast as the converter in [5] using only one-half of the hardware.

4.2 The R/B Converter for S^1

Using Theorem 1, a binary number X based on the moduli set $S^1 = \{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1\}$ is computed by

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

$$T_1 = < x_{-1,1} x_{-1,0} > \left[x_{-1} \right]^3 < x_{-1,(m-1)} \perp x_{-1,3} x_{-1,2} >$$

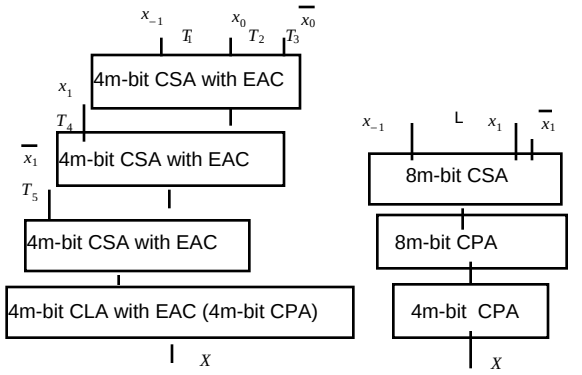
$$T_2 = < x_{0,1} x_{0,0} > [0]^{m-1} < x_{0,m} .. x_{0,0} > [0]^{m-1} < x_{0,m} .. x_{0,2} >$$

$$T_3 = < 1 \bar{x}_{0,m} \perp \bar{x}_{0,0} > [1]^{m-1} < \bar{x}_{0,m} \perp \bar{x}_{0,0} > [1]^{m-2}$$

$$T_4 = < x_{1,0} > [0]^{2m-1} < x_{1,2m} \perp x_{1,2} x_{1,1} >$$

$$T_5 = < \bar{x}_{1,2m} \perp \bar{x}_{1,1} \bar{x}_{1,0} > [1]^{2m-1}$$

For the moduli set S^1, we obtain the converter from the general architecture of Fig. 1. This converter consists of three *4m*-bit CSAs with EAC and a *4m*-bit CLA with EAC. This is shown in Fig. 3 (a). Fig. 3 (b) shows the block diagram of the corresponding converter developed in [5] and is included here for the sake of comparison. It is easy to see that the proposed converter saves one 8m-bit CPA while using one more 4m-bit CSA.



(a) **(b)**

Figure 3. (a) Proposed converter (b) The converter in [5]

The performance of the proposed converter and that of the corresponding one given in [5] is compared in Table 2.

Table 2 Comparison of the converters for S^1

	Transistors	Delay
Proposed Converter	240m+60	(8m+6) Δ
Converter in [5]	400m+20	***(28m+5)*** Δ ***[5]***

Thus, the proposed converter is three times as fast, but requiring only 60% of the hardware.

5. CONCLUSION

A high-speed parallel R/B converter for the general moduli set S^k 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 proposed R/B converter for S^0 is twice as fast as the existing one in [5] using only one-half of the hardware, while that for S^1 is three times faster, but requiring only 60% of the hardware.

ACKNOWLEDGMENT

This work was supported by NSERC (Canada), MICRONET and FCAR (Quebec).

6. REFERENCES

[1] A. Dhurkadas. "comments on "An efficient residue to binary converter design"". *IEEE Transactions on Circuits and Systems*, Vol. 37, pages 849-850. June 1990.

[2] Alexander Skavantzos. "An Efficient Residue to Weighted Converter for a New Residue Number System". *Proceedings of the 8th Great Lakes Symposium on VLSI*, pages 185-191, February, 1998.

[3] C. Efstathiou., D. Nikolos., and J. Kalamatianos. "Area-time efficient modulo $2^k - 1$adder design", *IEEE Trans. Circuits System. II.* Vol. 41, pages 463-466, July 1994.

[4] E.V. Jones. "Fast Conversion Between Binary And Residue Numbers". *Electronics Letters*. pages 1195-1198, 1988.

[5] F. Pourbigharaz., and H. M. Yassine., "A Signed-Digit Architecture for Residue to Binary Transformation". *IEEE Transactions on Computers*. Vol. 46 No. 10. pages 1146-1150, October 1997.

[6] H. L. Garner. " The Residue Number System". *IEEE Trans. Electronic Computers*. Vol. 8, pages 140-147, June 1959.

[7] Israel Koern. *Computer Arithmetic Algorithms*. Prentice Hall, 1993.

[8] M. A. Soderstrand., et al. Eds. *Residue number system arithmetic: modern applications in digital signal processing.* New York, IEEE Press, 1986.

[9] N. Szabo., and R. Tanaka. *Residue arithmetic and its applications to computer technology.* New York, McGraw-Hill, 1967.

[10] Stanislaw J. Piestrak. "Design of Residue Generators and Multi-operand Modular Adders Using Carry-Save Adders". *IEEE Transactions on Computers*. Vol. 423, No. 1, pages 68-77, January, 1994.

[11] Stanislaw J. Piestrak. "A High-speed Realization of a Residue to Binary Number System Converter". *IEEE Transactions on Circuits and Systems- II: analog and Digital Signal Processing.* Vol. 42, No. 10, pages 661-663, October, 1995.

[12] Sweidan Andraos. and Hiasat Ahmad. "A New Efficient Memoryless Residue to Binary Converter". *IEEE Transactions on Circuits and System.* Vol. 35, No. 11, pages 1441-1444, November, 1988.

[13] Yuke Wang. and M. Abd-el-Barr. "A New Algorithm for RNS Decoding", *IEEE Transactions on Circuits and Systems-I.* Vol. 43, No. 12, December, 1996.