An Improved RNS Reverse Converter for the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ Moduli Set

K.A. Gbolagade[1,2], R. Chaves[3], L. Sousa[3], S.D. Cotofana[1]
1. Computer Engineering Lab., Delft University of Technology, Delft, The Netherlands,
2. University for Development Studies, Navrongo, Ghana,
and
3. Instituto Superior Tecnico, TuLisbon/INESC-ID, Portugal.

***Abstract*—In this paper, we propose a novel high speed memoryless reverse converter for the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$. First, we simplify the traditional Chinese Remainder Theorem in order to obtain a reverse converter that only requires arithmetic mod-$(2^{2n+1}-1)$. Second, we further improve the resulting architecture to obtain a purely adder based reverse converter. The proposed converter has a critical path delay of $(7n+7)$ Full Adders (FA) while the best state of the art converter for this moduli set requires $(10n+5)$ FA on the critical path. To validate these results, the converters are implemented in a Standard Cell 0.18-μm CMOS technology and the results assert that, on average, the proposed converter achieves about 19% delay reduction at the expense of less than 3% area increase.**

I. Introduction

Residue Number Systems (RNS) have significant advantages over conventional binary number systems. This is due to their inherent features, such as carry free operations, parallelism, modularity, and fault tolerance. RNS have been widely applied in Digital Signal Processing (DSP) applications [5]. However, despite all these advantages, RNS have not found a widespread usage in general purpose processors since sign detection, magnitude comparison, overflow detection, and division are rather difficult to perform. Several solutions for these problems, which rely heavily on RNS to binary conversion, have been proposed [5]. This is one of the major reasons why building efficient RNS to binary converters has become an important research topic.

For a successful RNS utilization, moduli set choice and data conversion are critical, in particular the RNS to binary conversion (reverse conversion). Moduli set choice is an important issue since the complexity and the speed of the resulting conversion algorithm depend on the chosen moduli set. Several structures have been proposed to perform the reverse conversion for different moduli sets, e.g., $\{2^n, 2^n-1, 2^n+1\}$ [1], $\{2^n, 2^{n+1}-1, 2^n-1\}$ [3], [2].In [3], the moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$ was proposed, by the elimination of the modulus (2^n+1) from the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ proposed in [6]. The motivation for this is related to the fact that the modulo (2^n+1)-type arithmetic is more complex and degrades the entire RNS performance, both in terms of area cost and conversion delay. However, the moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$, which is able to utilize fast modulo operations, is insufficient for applications requiring larger dynamic ranges. Consequently, the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$ was proposed in [4] together with a reverse converter based on Mixed Radix Conversion (MRC).

In this paper, a novel and more efficient reverse converter for the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set is proposed. First, we simplify the traditional Chinese Remainder Theorem (CRT) and obtain a new converter that only requires mod-$(2^{2n+1}-1)$ operations. Further simplifications result in a simple and more efficient hardware structure, composed of only Carry Save Adders (CSAs) with end-around carries (EACs) and two Carry Propagate Adders (CPAs).

II. Proposed Algorithm

The proposed algorithm is described using the following theorems:

Theorem 1: Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^{2n+1}-1, m_2 = 2^n, m_3 = 2^n-1$, the following holds true:

$$|(m_1m_2)^{-1}|_{m_3} = 1, \quad (1)$$

$$|(m_1m_3)^{-1}|_{m_2} = 1, \quad (2)$$

$$|(m_2m_3)^{-1}|_{m_1} = 2^{2n+1}-2^{n+2}-3. \quad (3)$$

Proof: It can be easily shown that $|1 \times (m_1m_2)|_{m_3} = 1$, $|1 \times (m_1m_3)|_{m_2} = 1$, and $\left|(2^{2n+1}-2^{n+2}-3) \times (m_2m_3)\right|_{m_1} = 1$. ■

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

The following important relations are used in the subsequent theorem: Given the moduli set $\{m_1,m_2,m_3\}$ with $m_1 = 2^{2n+1}-1, m_2 = 2^n, m_3 = 2^n-1$, the following holds true:

$$m_1 = 2m_2m_2 - 1, \tag{4}$$

$$m_2 = m_3 + 1. \tag{5}$$

Theorem 2: The decimal equivalent of the residues (x_1,x_2,x_3) with respect to the moduli set $\{m_1,m_2,m_3\}$ in the form $\{2^{2n+1}-1, 2^n, 2^n-1\}$, assuming $X \in [0, \prod_{i=1}^{3} m_i - m_3^2)$, can be computed as follows:

$$X = m_2 \left\lfloor \frac{X}{m_2} \right\rfloor + x_2, \tag{6}$$

$$\begin{aligned} \left\lfloor \frac{X}{m_2} \right\rfloor = {} & x_3 - x_2 + m_3 \big|(-2^{n+2}-2)x_1 + \\ & 2m_2x_2 + 2m_2x_3 + 2x_3\big|_{m_1} \end{aligned} \tag{7}$$

Proof: Since (6) follows the basic integer division definition in RNS, which is always true, we only need to show the correctness of (7).

The traditional CRT [8] is given by:

$$X = \left| \sum_{i=1}^{k} M_i \left|M_i^{-1} x_i\right|_{m_i} \right|_M , \tag{8}$$

where $M = \prod_{i=1}^{k} m_i$, $M_i = \frac{M}{m_i}$, and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i. For $k=3$ in (8) and by substituting (1), (2), and (3) into (8) we obtain the following:

$$\begin{aligned} X = {} & \big|m_2m_3(2^{2n+1}-2^{n+2}-3)x_1 + \\ & m_1m_3x_2 + m_1m_2x_3\big|_M , \end{aligned} \tag{9}$$

and by substituting (4) and (5) in the above equation, we obtain:

$$\begin{aligned} X = {} & \big|(m_2m_3)(2^{2n+1}-2^{n+2}-3)x_1 + \\ & 2m_2m_2m_3x_2 - m_3x_2 + \\ & 2m_2m_2(m_3+1)x_3 - m_2x_3\big|_M . \end{aligned} \tag{10}$$

(10) can be further simplified by using the following lemma, presented in [8]:

$$|am_1|_{m_1m_2} = m_1 |a|_{m_2} \tag{11}$$

Applying (11) and (5), (10) becomes:

$$\begin{aligned} X = {} & \Big|m_2x_3 - m_3x_2 + m_2m_3 \big|(2^{2n+1}-2^{n+2}-3)x_1 + \\ & 2m_2x_2 + 2m_2x_3 + 2x_3\big|_{m_1}\Big|_M \end{aligned} \tag{12}$$

If (5) is applied to simplify even further in (12), we have:

$$\begin{aligned} X = {} & \big|m_2x_3 - m_2x_2 + x_2 + \\ & m_2m_3 \big|(2^{2n+1}-2^{n+2}-3)x_1 + \\ & 2m_2x_2 + 2m_2x_3 + 2x_3\big|_{m_1}\Big|_M \end{aligned} \tag{13}$$

Dividing both sides of the above equation by m_2 and taking the floor, we have:

$$\begin{aligned} \left\lfloor \frac{X}{m_2} \right\rfloor = {} & \Big|x_3 - x_2 + m_3 \big|-2^{n+2}x_1 - 2x_1 + \\ & 2m_2x_2 + 2m_2x_3 + 2x_3\big|_{m_1}\Big|_{m_1m_3} \end{aligned} \tag{14}$$

From (7), it can be seen easily that (14) is the same as $\left|\left\lfloor \frac{X}{m_2} \right\rfloor\right|_{m_1m_3}$. The next stage of the proof is to demonstrate that the corrective addition required for the calculation of the mod-m_1m_3 can be avoided in most of the cases. We demonstrate that by considering the two extreme cases, i.e., the most positive and most negative value one may get in (14).

- *Most positive value*: in order to get the most positive value in (14), the mod-m_1m_3 operation is maximized. To achieve this, x_3 is maximized while x_2 is minimized. Thus, $x_3 = 2^n - 2$, $x_2 = 0$ and because of this, mod-m_1 operand becomes $2^{2n+1} - 2^{n+1} - 4$. Therefore, the mod-m_1 operation result cannot assume the maximum value of $2^{2n+1} - 2$. Substituting the values $\left|2^{n+1}(2^n-2) + 2(2^n-2)\right|_{m_1} = 2^{2n+1} - 2^{n+1} - 4$, $x_2 = 0$, and $x_3 = 2^n - 2$ in (14), we obtain: $\left|2^{3n+1} - 2^{2n+2} - 2^n + 2\right|_{m_1m_3}$. However, m_1m_3 equals $2^{3n+1} - 2^{2n+1} - 2^n + 1$, which is always greater than the value in $\left|2^{3n+1} - 2^{2n+1} - 2^n\right|_{m_1m_3}$. No corrective addition of m_1m_3 is required in order to obtain the desired result and therefore (7) holds true.
- *Most negative value*: in order to get the most negative value in (14), the following must hold true: $x_3 = 0$, $x_2 = 2^n - 1$, and $\left|-2^{n+2}x_1 - 2x_1 + 2m_2x_2 + 2m_2x_3 + 2x_3\right|_{m_1} = 0$. Substituting these values in (14), we obtain $|-2^n + 1|_{m_1m_3}$. The value $-2^n + 1$ is negative and its absolute value is always less than m_1m_3, thus only one corrective addition is needed.

From (6), the minimum X value that needs a corrective addition of m_1m_3 occurs when $\left\lfloor \frac{X}{m_2} \right\rfloor$ has the lowest value, since $m_2 \left\lfloor \frac{X}{m_2} \right\rfloor$ grows faster than x_2. By using the minimum values in (14), specifically $\left\lfloor \frac{X}{m_2} \right\rfloor = (-m_3 + m_1m_3)$, and $x_2 = m_3$, the minimum value of X can be computed as:

$$\begin{aligned} X_{min} &= m_2(-m_3 + m_1m_3) + m_3 \\ &= M - m_3(m_2 - 1) = M - (m_3)^2 \end{aligned} \tag{15}$$

On the other hand, the maximum value of X that needs a corrective addition is given by:

$$\begin{aligned} X_{max} &= m_2(-1 + m_1m_3) + m_3 \\ &= M - (m_3 + 1) + m_3 = M - 1 \end{aligned} \tag{16}$$

Therefore, the numbers within the interval $[0, M-(m_3)^2)$ require no corrective addition and thus, (7) holds true. ■

We can further reduce the hardware complexity of the reverse converter by simplifying (7) using the following two properties:

Property 1: Modulo (2^s-1) multiplication of a residue number by 2^t, where s and t are positive integers, is equivalent to t bit circular left shifting.

Property 2: Modulo (2^s-1) of a negative number is equivalent to the one's complement of the number, which is obtained by subtracting the number from (2^s-1).

Suppose that (7) is written as:

$$\left\lfloor \frac{X}{m_2} \right\rfloor = x_3 - x_2 + 2^n A - A, \quad (17)$$

$$A = |u_0 + u_1 + u_2 + u_3 + u_4|_{2^{2n+1}-1}. \quad (18)$$

For simplicity sake, let us represent (17) as:

$$\left\lfloor \frac{X}{m_2} \right\rfloor = B_1 + B_2 + B_3, \quad (19)$$

$$B_1 = -x_2, B_2 = 2^n A + x_3, B_3 = -A. \quad (20)$$

Let the binary representations of the residues be:

$$x_1 = (x_{1,2n}x_{1,2n-1}...x_{1,1}x_{1,0}), \quad (21)$$

$$x_2 = (x_{2,n-1}x_{2,n-2}...x_{2,1}x_{2,0}), \quad (22)$$

$$x_3 = (x_{3,n-1}x_{3,n-2}...x_{3,1}x_{3,0}). \quad (23)$$

In (18), u_0, u_1, u_2, u_3, and u_4 are represented as follows:

$$u_0 = \left|-2^{n+2}x_1\right|_{2^{2n+1}-1}$$

$$= (\underbrace{\overline{x}_{1,n-2}\overline{x}_{1,n-3}...\overline{x}_{1,0}}_{n-1}\underbrace{\overline{x}_{1,2n}\overline{x}_{1,2n-1}...\overline{x}_{1,n-1}})_{n+2}(24)$$

$$u_1 = |-2x_1|_{2^{2n+1}-1}$$

$$= (\underbrace{\overline{x}_{1,2n-1}\cdots\overline{x}_{1,0}\overline{x}_{1,2n}}_{2n+1}), \quad (25)$$

$$u_{i,i=2,3} = \left|2^{n+1}x_i\right|_{2^{2n+1}-1}$$

$$= (\underbrace{x_{i,n}x_{i,n-1}\cdots x_{i,0}}_{n}\underbrace{00\cdots0}_{n+1}), \quad (26)$$

$$u_4 = |2x_3|_{2^{2n+1}-1}$$

$$= (\underbrace{00\cdots0}_{n}\underbrace{x_{2,n-1}x_{2,n-2}\cdots x_{2,0}}_{n}\underbrace{0}_{1}). \quad (27)$$

Given the binary representation:

$$A = (\underbrace{a_{2n}a_{2n-1}\cdots a_1 a_0}_{2n+1}), \quad (28)$$

B_2 can be written as:

$$B_2 = (\underbrace{a_{2n}a_{2n-1}\cdots a_0 x_{3,n-1}x_{3,n-2}\cdots x_{3,0}}_{3n+1}). \quad (29)$$

In (19), in order to carry out the summation, B_1 and B_3 must have equal number of bits, i.e., $(3n+1)$-bits, as B_2. They are represented as:

$$B_1 = -x_2 = (\underbrace{111\cdots11}_{2n+1}\underbrace{\overline{x}_{2,n-1}\overline{x}_{2,n-2}\cdots\overline{x}_{2,0}}_{n}), \quad (30)$$

$$B_3 = -A = (\underbrace{111\cdots11}_{n}\underbrace{\overline{a}_n\overline{a}_{n-1}\cdots\overline{a}_0}_{2n+1}). \quad (31)$$

III. Hardware Realization

The hardware structure of the proposed reverse converter is based on (18) and (19). In Figure 1, u_0, u_1, u_2, u_3, and u_4 are added by CSAs with end-around carries (EACs) producing the values s_3 and c_3. These values must be added modulo $2^{2n+1}-1$ in order to obtain A, i.e., with a one's complement adder, namely a CPA with EAC. B_2 is easily obtained by concatenating the operand x_3 with the n-bit left shift of A. The three operands B_1, B_2, and B_3 are added using a CSA with EAC. It should be noted that in order to make B_1 and B_3 $(3n+1)$-bit numbers, 1's are appended to the result of complementations, as given in (30) and (31). Thus, the addition of the most significant $(2n+1)$-bits performed in this CSA can be performed by Half Adders (HA). In addition, since these HA have two inputs equal to 1, the final one's complement adder will always generate an EAC. Taking this into consideration the one's complement adder can be reduced to a normal CPA with a constant carry-in of equals to 1. The final result, computed from (6) is obtained simply by a shift and a concatenation operation not requiring any additional hardware.

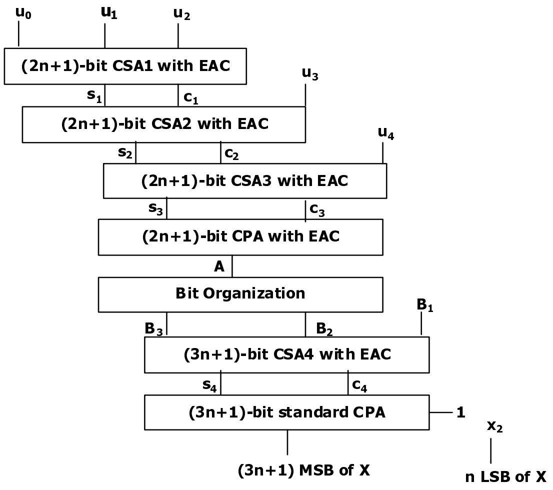

Figure 1. Proposed Reverse Converter

IV. Performance Evaluation

The performance of the proposed converter is evaluated by both performing a theoretical analysis and experimentally by implementing it on an Application Specific Integrated Circuit (ASIC). The results of theoretical analysis is presented in Table I. This table suggests that, in terms of area, the reverse converter in [4] is slightly better than the herein proposed one since our proposal requires $(5n+4)$HA against 2HA, $2n$ XNOR and $2n$ OR gates required by the one in [4]. However, in terms of delay, the proposed converter is $(3n-2)t_{FA}$ faster than the reverse converter in [4].

Table I
Area and Delay Comparisons

Components	Converter in [4]	Proposed Converter
FA	$9n+2$	$9n+2$
HA	2	$5n+4$
XNOR	$2n$	–
OR	$2n$	–
Delay	$(10n+5)t_{FA}$	$(7n+7)t_{FA}$

For the experimental assessment, the converters were described in VHDL and implemented on a 0.18μm Standard Cell technology from UMC [7]. The experimental results, presented in Table II, suggest that, on average, the proposed structure is capable of performing the reverse conversion 19% faster, with an extra area cost of 3%.To compare both conversion structures, the Area-Time (AT) efficiency metric was used. This metric suggests that the proposed reverse converter is 16% more efficient than the one in [4].

V. Conclusions

In this paper, a novel high speed memoryless residue to binary converter for $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set is proposed. First, we simplified the traditional CRT to obtain a reverse converter that requires mod-$(2^{n+1}-1)$ instead of both of mod-$(2^{2n+1}-1)$ and mod-(2^n-1) required by the reverse converter in [4]. We further simplified the resulting architecture in order to obtain a pure adder-based memoryless converter, which is made up of only CSAs and CPAs. This leads to a structure that is amenable to efficient VLSI chip realization.

The performance of the proposed reverse converter is evaluated both theoretically and experimentally. Experimental results suggest that the proposed structure is, on average, 19% faster, with an additional area cost of 3%.Moreover, the AT metric indicates that the proposed reverse converter is 16% more efficient than the one in [4].

Table II
Implementation Results

n	Our Area	MRC Area	Our Delay (ns)	MRC Delay (ns)
4	1301	1270	7.34	9.73
5	1633	1609	8.31	10.47
6	1994	1852	9.42	11.25
8	2623	2555	11.05	12.47
16	5261	5071	13.07	16.01
24	7794	7962	18.84	20.54
32	10517	10038	15.15	22.23

VI. Acknowledgment

The authors wish to acknowledge HIPEAC Network of Excellence.

References

[1] R. Chaves and L. Sousa. Improving residue number system multiplication with more balanced moduli sets and enhanced modular arithmetic structures. *IET Comp. Digital Tech., Vol. 5, No.1, pp.472-480, Sept.*, 2007.

[2] S. Lin, M. Sheu, and C. Wang. Efficient vlsi design of residue to binary converter for the moduli set $\{2^n, 2^{n+1}-1, 2^n-1\}$. *IEICE Trans. INF. and SYST., Vol. E91-D, No.7, pp. 2058-2060, July*, 2008.

[3] P.V.A. Mohan. RNS-to-binary converter for a new three-moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$. *IEEE Trans. on Circuits and Systems-II: Express briefs, Vol. 54, No.9, pp. 775-779, September*, 2007.

[4] A.S. Molahosseini, K. Navi, and M.K. Rafsanjani. A new residue to binary converter based on mixed-radix conversion. *3rd International Conference On Information and Communication Technologies: From Theory to Applications (ICTTA 2008), pp. 1-6, April*, 2008.

[5] L. Sousa. Efficient method for magnitude comparison in rns based on two pairs of conjugate moduli. *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, 2007.

[6] A.P. Vinod and A.B. Premkumar. A memoryless residue to binary converter for the 4-superset $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$. *Journal of Circuits, Syst. and Computers, Vol. 10, pp. 85-99,*, 2000.

[7] Virtual Silicon Technology Inc. *UMC High Density Standard Cells Library - 0.13µm CMOS process*, v2.3 edition, December 1999.

[8] Y. Wang. Residue-to-binary converters based on new chinese remainder theorems. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 47, No.3, pp. 197-205, March*, 2000.