RNS Reverse Converters based on the New Chinese Remainder Theorem I

Hector Pettenghi
Departamento de Engenharia Elétrica e Eletrônica, CTC
Universidade Federal de Santa Catarina (UFSC)
Florianópolis, SC, Brazil 88040-900
Email: hector@eel.ufsc.br

Leonel Sousa
INESC-ID, IST
Universidade de Lisboa
Lisboa, Portugal 1000-029
Email: las@sips.inesc-id.pt

***Abstract*—In the last years, research on residue number systems (RNS) has targeted larger dynamic ranges in order to further explore the inherent parallelism of these systems. In this paper, a performance analysis is presented for RNS-to-binary architectures based on New Chinese Remainder Theorem I (New CRT-I). Four different approaches are explored, each of them focused on the area or delay reduction of one specific stage of the converter. In addition, a selection of the constants associated to these algorithm approaches is proposed, which results into significant area reductions. Experimental results show that the use of an appropriate parameter selection can achieve a reduction of area and delay around 21% in comparison with the best solutions existing in the state-of-the-art using the New CRT-I technique and conventional parameter selection.**

I. Introduction

Residue arithmetics, based on Residue Number Systems (RNS), have been in use in Digital System Processing (DSP), for many years [1]. RNS is a carry-free arithmetic system with modular characteristics offering the potential for high-speed and parallel computation. Arithmetic operations, such as addition, subtraction, and multiplication, can be carried out independently and concurrently in several residue channels more efficiently than in the conventional binary systems [1]. The adoption of RNS has provided significant efficiency improvements for different types of DSP applications [1].

The choice of the moduli set is of key importance in order to obtain balanced moduli sets. Moduli sets with large number of channels can improve the arithmetic computation at the cost of reverse conversion performance. With efficient reverse converters capable of supporting large moduli sets, it is possible to compensate this extra cost, especially when several arithmetic operations have to be performed, such as in cryptographic or signal processing systems. The algorithms for reverse conversion are mainly based on the Chinese Remainder Theorem (CRT), on the mixed-radix conversion (MRC), and on New CRT-I [2].

Traditional RNS, with moduli set $\{\boldsymbol{2^{n+\beta}}, 2^n \pm 1\}$, provide a Dynamic Range (DR), of around $3n$-bit [3], which is not enough for many DSP applications. Other RNS with larger DR such as $4n$ [4], $5n$ [5], [6], $6n$ [7], [8], or $8n+1$ [9] still have the limitation of the number of channels, which translates into inefficient arithmetic RNS units for DSP applications with large operands. The use of New CRT-I algorithm has demonstrated to be one of the most efficient solutions for large DRs [2]. A method based on New CRT I algorithm for designing RNS based on generic moduli sets of the form $\{\boldsymbol{2^{n+\beta}}, 2^n \pm 1, \boldsymbol{2^n \pm k_1}, \boldsymbol{2^n \pm k_2}, ..., \boldsymbol{2^n \pm k_f}\}$ is presented in [10], where k_j are odd values and $0 \leq \beta \leq n$. However, reverse conversion requires a complex modular precomputation of the inputs in return of a simplification of the final modular conversion that requires only one comparison. Other generic solutions based on the New CRT-I algorithm can be implemented with a complex final converter [2]. Thus, there is not a methodology that provide to the user the best solution for the desired performance target, which is presented in this work.

The remaining of this paper is organized as follows. Two novel approaches derived from the existing approaches based on New CRT-I are presented in Section II for RNS reverse conversion. Two different ways of parameter selection are applied to our proposals in order to obtain a performance estimation in Section III. The experimental results obtained with the proposed architectures are presented in Section IV. Section V concludes this paper, with also some final remarks.

II. New CRT-I for generic moduli set

A value represented in RNS can be converted back to binary (X) using the CRT [1]:

$$X = \left| \sum_{i=1}^{N} \hat{m}_i \left| \hat{m}_i^{-1} \right|_{m_i} R_i \right|_M = \left\lfloor \frac{X}{m_1} \right\rfloor m_1 + R_1, \tag{1}$$

where R_i denotes the residue for m_i, N number of channels, $\hat{m}_i = M/m_i$, and $\left|\hat{m}_i^{-1}\right|_{m_i}$ represents the multiplicative inverse of $\hat{m}_i$ with respect to modulus m_i. From this Eq. (1) four possible solutions are explored denoted by $X^{(1)}$, $X^{(2)}$, $X^{(3)}$ and $X^{(4)}$. The value of X can be computed by New CRT-I [2] as follows:

$$X^{(1)} = \Bigg| \overbrace{\phi_1}^{V_{11}}(R_2 - R_1) + \overbrace{\phi_2 m_2}^{V_{12}}(R_3 - R_2) + \overbrace{\phi_{N-1} m_2 m_3 \dots m_{N-1}}^{V_{1(N-1)}}(R_N - R_{N-1}) \Bigg|_{\hat{m}_1} m_1 + R_1, \tag{2}$$

where $\left|\phi_j \prod_{i=1}^{j} m_i\right|_{m_N} = 1$ and $1 \leq j \leq N-1$. The subtraction of the residues in Eq. (2) can be avoided as follows:

$$X^{(2)} = \Big| \overbrace{-\phi_1}^{V_{21}} R_1 + \overbrace{(\phi_1 - \phi_2 m_2)}^{V_{22}} R_2 + \ldots + \overbrace{(\phi_{N-2} m_2 m_3 \ldots m_{N-2} - \phi_{N-1} m_2 m_3 \ldots m_{N-1})}^{V_{2(N-1)}} R_{N-1} + + \overbrace{\phi_{N-1} m_2 m_3 \ldots m_{N-1}}^{V_{2N}} R_N \Big|_{\hat{m}_1} \quad m_1 + R_1. \tag{3}$$

However, this approach introduces in return a penalty of one more term in the modular $\hat{m}_1$ addition. It is important to note that the subtraction of the residues in Eq. (2) and V_{2i} values in Eq. (3) can correspond to negative values in the $\hat{m}_1$ modular addition that must be fitted into a residual positive range. This adjustment can be performed by adding a correction term COR_1 for Eq. (2) and COR_2 for Eq. (3): if the inputs with associated negative weights are complemented it is possible to chose a correction term which sets the range to $0 \leq ABS(V_{ji}) < m_i - 1$, with $1 \leq j \leq 2$ and $1 \leq i \leq N-1$ [11], where ABS defines the absolute value. Therefore, the correction terms COR_1 and COR_2, are chosen as the minimum value that satisfies:

$$\left| \sum_{i=1}^{N-1} (V_{1i} R_{(i+1)} + ABS(V_{1i} \bar{R}_i)) + COR_1 \right|_{\hat{m}_1} = 0, \tag{4}$$

$$\left| \sum_{i \in V_{2i} > 0} V_{2i} R_i + \sum_{i \in V_{2i} < 0} ABS(V_{2i} \bar{R}_i) + COR_2 \right|_{\hat{m}_1} = 0, \tag{5}$$

where $\bar{R}_i$ is the complement of R_i. The value of the correction term is computed by setting the inputs $\{R_N, ..., R_1, R_0\}$ to 0 and choosing the minimum correction value that sets Eq. (4) and Eq. (5) for the computation of $X^{(1)}$, and $X^{(2)}$, respectively. In both cases the selection of the V_{ji} can be done in order to derive terms $V_{ji} R_i > 0$ with $1 \leq j \leq 2$ and $1 \leq i \leq N-1$, to avoid the use of the correction term. However, the use of negative terms $V_{ji} R_i$ and COR_j can be useful to reduce the complexity of the Final Converter (FC), as it is will be demonstrated in the next section.

Fig. 1(a) and (b) present the reverse converter of $DR = 6n$ using Eq. (2) and Eq. (3) for the moduli set $\{m_1 = 2^{2n}, m_2 = 2^n - k_1, m_3 = 2^n + k_2, m_4 = 2^n - k_3, m_5 = 2^n + k_4\}$, where k_j are odd values $-2^{n-1} + 1 < k_j < 2^n - 1$. The values of k_j are chosen in order to derive a set of co-prime numbers. The implementation of $X^{(1)}$ includes a first level of CPAs to calculate $(R_{(i+1)} + \bar{R}_i)$, a second level of multipliers to compute $V_{ji}(R_{(i+1)} - R_i)$, another level of CPA to derive $T = \sum_{j=1}^{N-1} V_{ji}(R_{(i+1)} + \bar{R}_i) + COR_j$, and a FC based on comparisons [12], where the number of comparisons is $\varepsilon = \lceil \frac{T_{max}}{\hat{m}_1} \rceil$, and T_{max} is the maximum value at FC input. The output of the FC is concatenated with R_1 as depicted in Fig. 1 (symbol $*$ denotes the concatenation of two binary numbers). $X^{(2)}$ can be implemented in the same way simply avoiding the first level of CPAs.

Moreover, internal $\hat{m}_1$ modular operations can be carried out in Eq. (2) and Eq. (3) in order to reduce the complexity of the FC at cost of area and delay in the modular precomputation of the inputs as presented in $X^{(3)}$ and $X^{(4)}$.

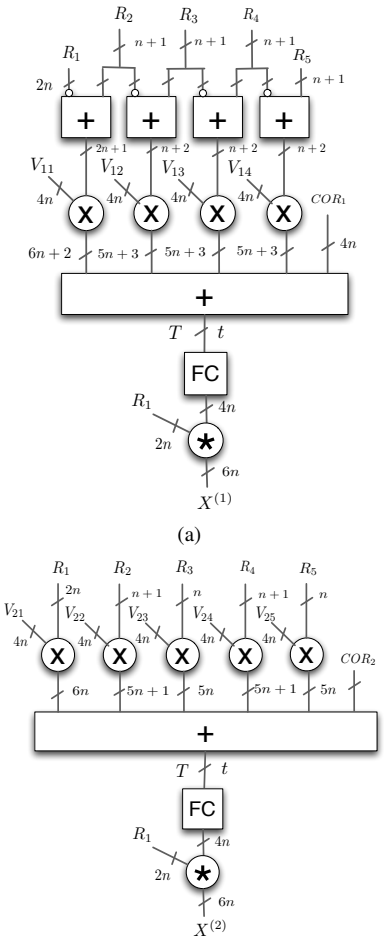

Fig. 1. Block diagram of the reverse converter of $DR = 6n$ (a) using Eq. (2), and (b) using Eq. (3).

$$X^{(3)} = \left| \left| \overbrace{|V_{11}|_{\hat{m}_1}}^{V_{31}} (R_2 - R_1) \right|_{\hat{m}_1} + \left| \overbrace{|V_{12}|_{\hat{m}_1}}^{V_{32}} (R_3 - R_2) \right|_{\hat{m}_1} + \ldots + \left| \overbrace{|V_{1(N-1)}|_{\hat{m}_1}}^{V_{3(N-1)}} (R_N - R_{N-1}) \right|_{\hat{m}_1} \right|_{\hat{m}_1} m_1 + R_1. \tag{6}$$

$$X^{(4)} = \left| \left| \overbrace{|V_{21}|_{\hat{m}_1}}^{V_{41}} R_1 \right|_{\hat{m}_1} + \left| \overbrace{|V_{22}|_{\hat{m}_1}}^{V_{42}} R_2 \right|_{\hat{m}_1} + \ldots + \left| \overbrace{|V_{2N}|_{\hat{m}_1}}^{V_{4(N)}} R_N \right|_{\hat{m}_1} \right|_{\hat{m}_1} m_1 + R_1. \tag{7}$$

It is important to note that the solution $X^{(4)}$ was already presented in [10]. If necessary, the correction terms COR_3 and COR_4 for $X^{(3)}$ and $X^{(4)}$ are obtained by using the same approach as for Eq. (4) and Eq. (5). The architectures of $DR = 6n$ with a moduli set $\{2^{2n}, 2^n + k_1, 2^n + k_2, 2^n + k_3, 2^n + k_4\}$, where k_j are odd values $-2^{n-1} + 1 < k_j < 2^n - 1$, are presented in Fig. 2(a) and Fig. 2(b) by using Eq. (6) and Eq. (7), respectively. The main difference in comparison to

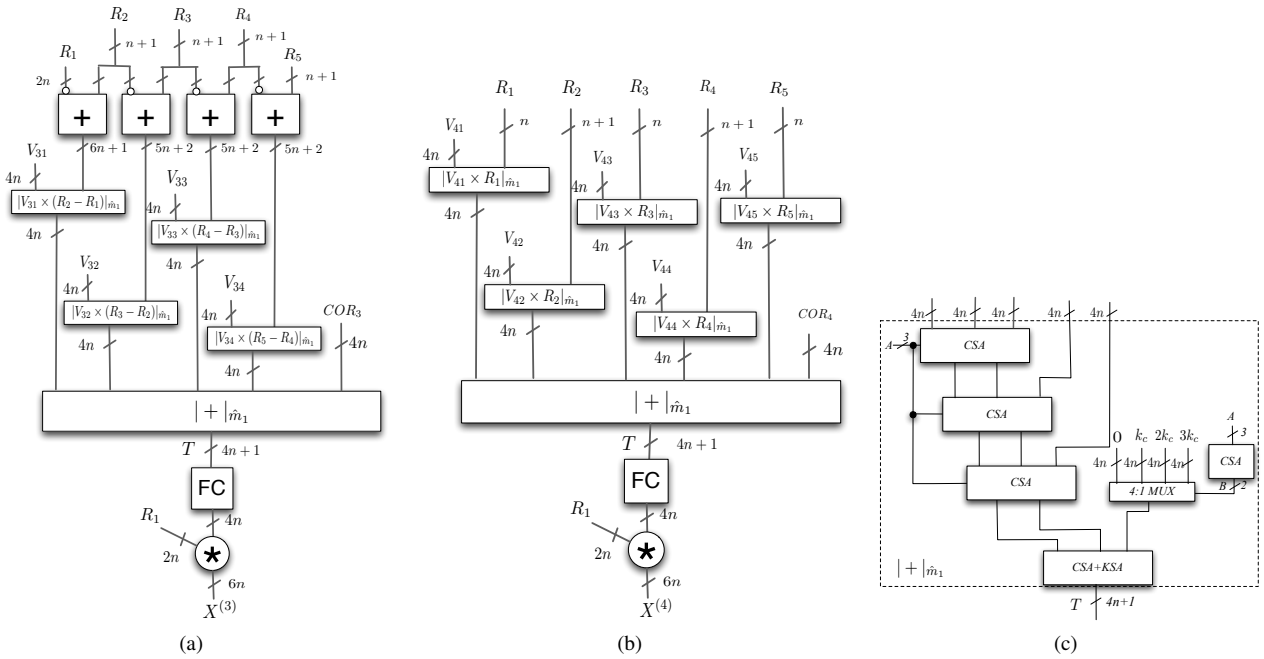

Fig. 2. Block diagram of the reverse converter of $DR = 6n$ using (a) Eq. (6), (b) using Eq. (7), and (c) Modular $\hat{m}_1$ addition block.

the previous architectures analyzed are the modular $\hat{m}_1$ multiplications and modulo addition at the FC input. The modulo $|V_{ji}R_i|_{\hat{m}_1}$, terms can be efficiently computed using custom-designed look-up tables, for less than 10 input bits [11], or with arithmetic through modular multiplications for larger number of input bits. However, in order to obtain a fair comparison with other proposals, all the architectures herein presented are memoryless implementations.

The addition of the $|V_{ji}R_i|_{\hat{m}_1}$ terms is performed by a binary Carry Save Adder (CSA) tree as shown for the case of five inputs in Fig. 2(c), resulting two outputs, the Carry and Save vectors. The output of each CSA is used to define the correction value for reducing modulo $\hat{m}_1$ the final value, taking into account that the modulo reduced MSB of the carry vector of each CSA has value $k_c = 2^{4n} - \hat{m}_1$ [10]. The addition of the Carry and Save vectors with the correction value by a final CSA and a Kogge and Stone Adder (KSA), results in the T value with $4n+1$ bits, as depicted in Fig. 2(c). The final $\left\lfloor \frac{X}{m_1} \right\rfloor$ value is obtained by a FC realized with a single comparator [12] that is concatenated with R_1.

III. Parameter selection

The parameters V_{ji}, $1 \leq j \leq 2$ and $1 \leq i \leq N$, are comprised into $[0, \hat{m}_1)$. Therefore, $V_{ji} = |V_{ji}|_{\hat{m}_1}$. In [12] an alternative solution for input weight selection is presented. This weight selection is performed by choosing the smallest of the two possible values associated to the inputs, V_{ji} or $V_{ji} - \hat{m}_1$. Essentially, if $|V_{ji}|_{\hat{m}_1} > \frac{\hat{m}_1 - 1}{2}$, the chosen weight is $V_{ji} = V_{ji} - \hat{m}_1$, fitting the weights into the range $\frac{-\hat{m}_1+1}{2} \leq V_{ji} \leq \frac{\hat{m}_1-1}{2}$. Applying the correction term, a positive range of the weights $ABS(V_{ji})$ can be obtained, which are comprised in the reduced range $0 \leq ABS(V_{ji}) \leq \frac{\hat{m}_1-1}{2}$. It is important to note that if $V_{ji} < 0$ in $X_{(1)}$ then $V_{ji}(R_{i+1} + \bar{R}_i)$ can be rewritten as $ABS(V_{ji})(\bar{R}_{i+1} + R_i)$ after applying the correction factor COR_1. Thus, by using this technique of weight selection, the weights of the residue values are reduced, obtaining the minimal word length at the FC input. The number of comparisons required, ε, for the architectures extracted from Fig. 1(a) and (b) are reduced to the minimum. The traditional weight selection and the one that minimize the FC will be denoted by $h = A$ and $h = B$, respectively, when applied to Eq. (2), and Eq. (3), as $X^{(jh)}$, where $1 \leq j \leq 2$. In the case of the architectures derived from Eq. (6) and Eq. (7) they require only one single comparison in the FC, and the previous $\hat{m}_1$ modular addition is transparent to the weight selection chosen. However, if the $\hat{m}_1$ modular multiplications are not implemented using ROMS, they can be reduced by using the values of $ABS(V_{ji})$ with less number of logic ones in their representation to simplify the multiplications. The traditional weight selection and the one that minimize the number of logic ones will be denoted by $h = A$ and $h = B$, respectively, when applied to Eq. (6) and Eq. (7), as $X^{(jh)}$, where $3 \leq j \leq 4$.

TABLE I. Parameters $ABS(V_{ji})$ and COR_j for structures $X^{(jh)}$ analyzed

j	h	V_{j1}	V_{j2}	V_{j3}	V_{j4}	V_{j5}	COR_j
1	A	48961	35701	55081	58786	–	53715
	B	14024	27284	7904	4199	–	9269
2	A	48961	13260	19380	3705	58786	53925
	B	14024	13260	19380	3705	4199	35701
3	A	48961	35701	55081	58786	–	53715
	B	14024	27284	7904	4199	–	9269
4	A	14024	13260	43605	59280	58786	0
	B	14024	13260	19380	3705	4199	35701

For example, for the five modulo set $\{2^{2n}, 2^n \pm 1, 2^n \pm 3\}$ and $n = 4$, $\{m_1 = 256, m_2 = 19, m_3 = 13, m_4 = 17, m_5 = 15\}$, the parameters $ABS(V_{ji})$ and COR_j are shown in Table I for the eight cases herein considered. Table II shows the number of comparisons at the FC, ε, of the converters derived from Eq. (2) and Eq. (3), and the number of logic "1"s required at the modular multiplications of the converters derived from Eq. (6) and Eq. (7). For the cases $X^{(1h)}$ and $X^{(2h)}$, the use of the proposed parameter selection $h = B$ provides an average reduction of 3.5 times in the number of comparisons

TABLE II. FEATURES OF THE REVERSE CONVERTERS PRESENTED

	$X^{(1A)}$	$X^{(1B)}$	$X^{(2A)}$	$X^{(2B)}$
ε	228	64	219	66
	$X^{(3A)}$	$X^{(3B)}$	$X^{(4A)}$	$X^{(4B)}$
$\#''1''$s	35	27	41	37

required in the FC when compared with the parameter selection used in the state-of-the-art. For the cases $X^{(3h)}$ and $X^{(4h)}$, the reduction of the number of logic "1"s in the $ABS(V_{ji})$ parameters is not so pronounced when the parameter selection $h = B$ is chosen. However, it is important to note that the bit length of the multiplicand $ABS(V_{ji})$ are reduced too.

IV. EXPERIMENTAL RESULTS

In order to assess the performance and cost of the generic reverse conversion approaches herein presented, the compared structures were described in a synthesizable VHDL and implemented on a 90nm Standard Cell ASIC technology from UMC [13], using the Design Vision synthesis tool (version E-2010.12-SP4).

The generic moduli set $\{m_1, m_2, m_3, \ldots, m_{DR-1}\} = \{2^{2n}, 2^n + k_1, 2^n + k_2, \ldots, 2^n + k_{DR-2}\}$ is used, where k_j are odd values $-2^{n-1} + 1 < k_j < 2^n - 1$. The values of k_j are chosen in order to derive the most balanced moduli set. Experimental results for moduli set $\{256, 13, 19, 15, 17\}$ and $\{256, 11, 23, 13, 19, 15, 17\}$, associated to $DR = 6n$ and $DR = 8n$ for $n = 4$, respectively, are presented in Table III, for the four ways of expressing the New CRT-I, $X^{(jh)}$, $1 \leq j \leq 4$, and two ways for parameter selection $h = A, B$.

The experimental results suggest that the use of $X^{(2h)}$ and $X^{(3h)}$ show a reduction of the Area Delay Product (ADP) in comparison with $X^{(1h)}$ and $X^{(4h)}$, respectively. In all the cases analyzed, the use of parameter selection type B is more efficient in terms of area, without significant differences in delay. It is important to note that this parameter selection has not been used so far in the state-of-the-art ($X^{(1A)}$ [2] and $X^{(4A)}$ [10]). Between them, $X^{(2B)}$ seems to be the best solution in terms of speed, and $X^{(3B)}$ in terms of area (in bold in Table III). In comparison with the state-of-the-art for $DR = 6n$ and $n = 4$, $X^{(2B)}$ compared with the New CRT-I algorithm implementation, $X^{(1A)}$ presented in [2], shows an 63.31%, 21.40%, and 2.5 times reduction of area, delay and ADP, respectively, whereas $X^{(3B)}$ is 20.97% smaller than $X^{(4A)}$ implemented in [10].

For $DR = 8n$ and $n = 4$ the area reduction by using the new parameter selection in $X^{(1B)}$ and $X^{(2B)}$ is not so deep as for $DR = 6n$. This is due to the higher complexity for larger DR is balanced with the reduction of the number of comparisons at the FC. These results can be extended for larger DRs demonstrating the benefits of our proposals in comparison with the state-of-the-art for generic DRs [10].

V. CONCLUSIONS

In this paper are presented four reverse converters approaches with generic moduli sets using the New CRT-I algorithm. Two of them are already presented in the state-of-the-art and the other ones are proposals that reduce the required area and the imposed delay, respectively. Moreover, a new parameter selection is proposed and applied to each presented architecture. This parameter selection allows reduce the area of

TABLE III. EXPERIMENTAL RESULTS FOR STRUCTURES ANALYZED FOR $DR = 6n$ AND $DR = 8n$ FOR $n = 4$

	$DR = 6n$			$DR = 8n$		
	Area (μm^2)	Delay (ns)	ADP (pm^2s)	Area (μm^2)	Delay (ns)	ADP (pm^2s)
$X^{(1A)}$	140038	2.71	379.50	57377	3.53	202.543
$X^{(1B)}$	45275	2.67	120.88	50552	3.48	**175.92**
$X^{(2A)}$	137142	2.17	297.60	82082	2.98	244.60
$X^{(2B)}$	51379	**2.13**	109.44	68712	**2.90**	199.26
$X^{(3A)}$	23026	4.98	114.67	46350	5.95	275.78
$X^{(3B)}$	**21157**	5.09	**107.69**	**45838**	5.95	272.74
$X^{(4A)}$	26772	4.68	125.29	54636	5.27	287.93
$X^{(4B)}$	25196	4.68	117.92	52564	5.23	274.91

the four converters in comparison with the parameter selection used up to now, without penalty on speed. Experimental results show that the best solutions of our proposals can achieve a reduction of area and delay of around 21% in comparison with the smallest and fastest solutions known to date that use the New CRT-I technique. In terms of ADP, the use of the new parameter selection provides an average reduction of 50.28%, and 8.95% for $DR = 6n$ and $DR = 8n$, respectively, demonstrating to be an efficient solution to improve the performance of the converters.

REFERENCES

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

[2] Y. Wang,"Residue-to-binary converters based on new chinese remainder theorems," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 3, pp. 197–205, Mar. 2000.

[3] Y. -Y. Wang, X. Song, M. Aboulhamid, and H. Shen, "Adder based residue to binary converters for $(2^n - 1, 2^n, 2^n + 1)$," *IEEE Trans. Signal Processing*, vol. 50, no. 7, pp. 1772–1779, Jul. 2002.

[4] P. Mohan and A. Premkumar, "RNS-to-binary converters for two four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1245–1254, Jun. 2007.

[5] B. Cao, C.-H. Chang, and T. Srikanthan, "A residue-to-binary converter for a new five-moduli set," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 54, no. 5, pp. 1041–1049, May 2007.

[6] A. Hiasat, "VLSI implementation of new arithmetic residue to binary decoders," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 1, pp. 153–158, Jan. 2005.

[7] L. Sousa, and S. Antão, "MRC-based RNS reverse converters for the four-moduli sets $\{2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ and $\{2^n + 1, 2^n - 1, 2^{2n}, 2^{2n+1} - 1\}$," *IEEE Trans. Circuits and Systems II*, vol. 59, no. 4, pp. 244–248, Apr. 2012.

[8] A. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, "Efficient reverse converter designs for the new 4-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ based on new CRTs," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 823–835, Apr. 2010.

[9] H. Pettenghi, R. Chaves, and L. Sousa, "RNS reverse converters for moduli sets with dynamic ranges up to $(8n+1)$-bit," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 60, no. 6, pp. 1487–1500, Jun. 2013.

[10] H. Pettenghi, R. Chaves, and L. Sousa, "Method to Design General RNS Reverse Converters for Extended Moduli Sets," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, accepted for publication, 2014.

[11] S. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," *IEEE Trans. Computers*, vol. 43, no. 1, pp. 68–77, Jan. 1994.

[12] H. Pettenghi, R. Chaves, and L. Sousa, "Method for designing modulo $2^n \pm k$ Binary-to-RNS converters," *in Proc. of Int. Conference on Design of Circuits and Integrated Systems (DCIS)*, Nov. 2013.

[13] Virtual Silicon Technology, Inc.: UMC high density standards cells library -0.13μm CMOS process v2.3, 2010.