MRC-Based RNS Reverse Converters for the Four-Moduli Sets $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ and $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$

Leonel Sousa, *Senior Member, IEEE*, and Samuel Antão, *Student Member, IEEE*

***Abstract*—The moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ has been recently proposed for supporting residue number systems with dynamic ranges of $5n$ bits. In this brief, we suggest modifying this moduli set to $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$, in order to enlarge the dynamic range to $6n$ bits. We propose a method that unifies the design of efficient reverse converters for the original and the modified moduli sets. A unified architecture was derived to design individual reverse converters for each moduli set or to achieve a single configurable reverse converter. Experimental results show that the delay of the converters designed with the proposed method and implemented on a 65-nm CMOS integrated circuit is improved by 12% on average. Moreover, the product of the area with the square of the delay is improved up to 25% and 21%, when compared to the related state of the art and values of n between 6 and 32, for dynamic ranges of $5n$ and $6n$ bits, respectively.**

***Index Terms*—Application-specific integrated circuit (ASIC), computer arithmetic, residue number systems (RNSs), reverse converter.**

I. Introduction and Background

RESIDUE NUMBER systems (RNSs) can be used to efficiently operate on large numbers, namely, for applications such as cryptography [1] and signal processing [2], [3]. The most computationally demanding operation on RNS is the reverse conversion, but this burden can be overcome by carefully choosing the moduli sets [4]. The well-known three-moduli set $\{2^n-1, 2^n, 2^n+1\}$ has interesting properties but a modest dynamic range ($3n$ bits). Extensions to this moduli set have been proposed for achieving $4n$-bit dynamic ranges, such as the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{n+1}\pm 1\}$ [5], and a generalization of the four-moduli set $\{2^n-1, 2^n+3, 2^n-3, 2^n+1\}$ [6]. Recently, the moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ has been proposed to enlarge the dynamic range to $5n$ bits [7] and the moduli sets $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$ [7] and $\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$ [8] to $6n$ bits. Although none of the last three 4-moduli sets is perfectly balanced, the moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ is the only one with all the moduli in the forms 2^k and $2^k\pm 1$ and with only one modulus in the form of 2^k+1. This is a significant advantage since operations modulo 2^k-1 are simpler to implement than operations modulo 2^k+1 [9]–[11]. Reverse converters, based on the Chinese remainder theorem (CRT), namely, the commonly called new CRTs [12], were proposed for this type of moduli sets [7]. In this brief, we improve the delay and flexibility of these converters and hence deliver more efficiency.

We propose the new moduli set $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$ to enlarge the dynamic range provided by the original moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ to $6n$ bits. This new moduli set has the advantages that all the moduli have the forms 2^k and $2^k\pm 1$, and the exponent of the only modulus with the form 2^k+1 is the smallest in the set (n). Furthermore, a method is proposed for unifying the design of the converters for the moduli sets $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ and $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$.

The proposed method for designing reverse converters applies the mixed-radix (MR) conversion (MRC) successively to moduli subsets with two elements, by adopting the idea of "grouped moduli" presented in [12] and [13]. This method and the designed converters rely on the properties of the power-of-two related moduli that can be efficiently exploited when the moduli are organized in proper small sets of only two elements. With the MRC, the MR digits are computed according to the following equation, where $x_i = |X|_{m_i}$ represents the residue for the ith modulus (m_i) of the target moduli set [14]:

$$a_1 = x_1; a_2 = \left|(x_2 - a_1)|m_1^{-1}|_{m_2}\right|_{m_2}; X = a_1 + m_1 a_2. \quad (1)$$

Experimental evaluation of the proposed converters, and relative assessment of their performance regarding the related state of the art, shows that, we do not only unify the design of the reverse converters for the considered moduli sets but also achieve efficient converters for large dynamic ranges.

The brief is organized as follows. The proposed method and the underlying architecture of the reverse converters are presented in Section II. Section III experimentally evaluates the proposed converters and assesses their efficiency regarding the related state of the art, and finally, Section IV concludes this brief.

II. RNS Reverse Converters

In this section, we propose a method to design reverse converters by decomposing the original moduli set and applying

Manuscript received July 7, 2011; revised October 3, 2011 and December 2, 2011; accepted January 29, 2012. Date of publication March 21, 2012; date of current version April 11, 2012. This work was supported in part by the Fundação para a Ciência e a Tecnologia (Instituto de Engenharia de Sistemas e Computadores—Investigação e Desenvolvimento multiannual funding) through the Programa de Investimentos e Despesas de Desenvolvimento da Administração Central (PIDDAC) Program funds. This paper was recommended by Associate Editor M. M. Mansour.

The authors are with the Department of Electrical and Computer Engineering, Instituto Superior Técnico, Universidade Técnica de Lisboa, 1049-001 Lisboa, Portugal, and also with the Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento em Lisboa, 1000-029 Lisboa, Portugal (e-mail: leonel.sousa@inesc-id.pt; samuel.antao@inesc-id.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2012.2188456

the MRC. For presenting the proposed method and the corresponding reverse converters, the following notations based on [15] are adopted.

1) For an n-bit value γ, bits are referred from the most significant bit (MSB) to the least significant bit (LSB) as $\gamma[n-1],\ldots,\gamma[0]$.
2) $\gamma^l[k]$ refers to an l-bit number such that
$$\gamma^l[k]=\gamma[k+l-1]2^{l-1}+\cdots+\gamma[k+1]2+\gamma[k].$$
3) $\mathcal{O}$ and $\mathcal{Z}$ refer to numbers whose binary representations are all-one and all-zero strings, respectively.
4) The symbol $\bowtie$ operates the concatenation of the binary representations of two numbers.
5) $|\gamma^{-1}|_m$ represents the multiplicative inverse of γ modulo m ($|\gamma\times\gamma^{-1}|_m=1$).

Although $\{2^n+1,2^n-1,2^{2n},2^{2n+1}-1\}$ is a new moduli set, it has been already proved, in the context of different three-moduli sets [7], [16], [17], that all their elements are pairwise coprimes.

A. Proposed Method for Reverse Conversion

Let us represent the target moduli sets by $\{2^n+1,2^n-1,2^{(1+\alpha)n},2^{2n+1}-1\}$, with $\alpha=0,1$ (also consider $\beta=\alpha-1$). The original four-moduli sets are partitioned into two-moduli subsets: $M_1=\{2^n+1,2^n-1\}$ and $M_2=\{2^{(1+\alpha)n},2^{2n+1}-1\}$. Applying the MRC (1) to M_1 ($\{x_1,x_2\}$ denotes the residues for this subset) and to M_2 ($\{x_3,x_4\}$ denotes the residues for this subset)

$$X_{12}=x_1+\omega_{12}\times(2^n+1) \tag{2}$$
$$X_{34}=x_4+\omega_{34}\times(2^{2n+1}-1) \tag{3}$$

with

$$\omega_{12}=\Big|(x_2-x_1)\times\big|(2^n+1)^{-1}\big|_{2^n-1}\Big|_{2^n-1} \tag{4}$$
$$\omega_{34}=\big|(x_3-x_4)\times\big|(2^{2n+1}-1)^{-1}\big|_{2^{(1+\alpha)n}}\big|_{2^{(1+\alpha)n}}. \tag{5}$$

We can compute ω_{12} by using the following value for the multiplicative inverse:

$$\big|(2^n+1)^{-1}\big|_{2^n-1}=2^{n-1} \tag{6}$$

since

$$\big|(2^n+1)\times 2^{n-1}\big|_{2^n-1}=|2\times 2^{n-1}|_{2^n-1}=|2^n|_{2^n-1}=1.$$

Recalling that multiplying modulo 2^n-1 an integer γ, represented with n bits, by 2^s can be accomplished by circularly shifting the n-bit integer by s bits to the left (note that $|2^n|_{2^n-1}=1$) and that $|-\gamma|_{2^n-1}$ corresponds to the one's complement of γ ($\overline{\gamma}$), the expression of ω_{12} in (4) can be simplified using (6)

$$\omega_{12}=\Big|x_2[0]\bowtie x_2^{n-1}[1]+\overline{x_1[0]\oplus x_1[n]}\bowtie\overline{x_1^{n-1}[1]}\Big|_{2^n-1}. \tag{7}$$

Finally, using (7) and (2), we can compute X_{12} as

$$X_{12}=\omega_{12}\bowtie\omega_{12}+x_1. \tag{8}$$

We can compute ω_{34}(5) by using the following value for the multiplicative inverse:

$$\big|(2^{2n+1}-1)^{-1}\big|_{2^{(1+\alpha)n}}=2^{(1+\alpha)n}-1 \tag{9}$$

since

$$\Big|(2^{2n+1}-1)\times(2^{(1+\alpha)n}-1)\Big|_{2^{(1+\alpha)n}}=1.$$

Substituting (9) into (5), we get

$$\omega_{34}=\Big|x_4^{(1+\alpha)n}[0]+\overline{x_3^{(1+\alpha)n}[0]}+1\Big|_{2^{(1+\alpha)n}} \tag{10}$$

and applying (10) and (3), we can compute

$$X_{34}=\omega_{34}^{(1+\alpha)n}[0]\bowtie x_4^{2n+1}[0]+\mathcal{O}^{2n+1}[0]\bowtie\overline{\omega_{34}}+1. \tag{11}$$

To achieve the final binary representation (X), one can consider $\{X_{12},X_{34}\}$ as the representation of X in the moduli set $\{M_{12},M_{34}\}$, composed by the coprime moduli

$$\Big\{(2^n+1)(2^n-1),2^{(1+\alpha)n}(2^{2n+1}-1)\Big\}=\Big\{2^{2n}-1,2^{(\alpha+3)n+1}-2^{(1+\alpha)n}\Big\}.$$

Applying the MRC (1), the binary representation X can be obtained as

$$X=X_{34}+\Omega\times\left(2^{(\alpha+3)n+1}-2^{(1+\alpha)n}\right) \tag{12}$$

with

$$\Omega=\Big||X_{12}-X_{34}|_{2^{2n}-1}\times\Big|(2^{(\alpha+3)n+1}-2^{(1+\alpha)n})^{-1}\Big|_{2^{2n}-1}\Big|_{2^{2n}-1}. \tag{13}$$

We can compute Ω by using the following values for the multiplicative inverses:

$$\big|(2^{3n+1}-2^n)^{-1}\big|_{2^{2n}-1}=2^n \tag{14}$$
$$\big|(2^{4n+1}-2^{2n})^{-1}\big|_{2^{2n}-1}=1 \tag{15}$$

since

$$\big|(2^{3n+1}-2^n)\times 2^n\big|_{2^{2n}-1}=|2-2^{2n}|_{2^{2n}-1}=1$$
$$\big|(2^{4n+1}-2^{2n})\times 1\big|_{2^{2n}-1}=|(2-1)\times 1|_{2^{2n}-1}=1.$$

Given that (11) requires more than $2n$ bits to be computed, let us directly use (8) and (3) to compute $Y=|X_{12}-X_{34}|_{2^{2n}-1}$ as

$$Y=\big|\omega_{12}\bowtie\omega_{12}+x_1-x_4-\omega_{34}\times(2^{2n+1}-1)\big|_{2^{2n}-1}. \tag{16}$$

It is easy to show that $|(2^{2n+1}-1)|_{2^{2n}-1}=1$. For $\alpha=0$

$$\begin{aligned}&|-x_4-\omega_{34}|_{2^{2n}-1}\\&=\Big|\overline{x_4^{2n}[0]}+\underbrace{x_4[2n]\bowtie\cdots\bowtie x_4[2n]}_{2n-1}\\&\qquad\bowtie\mathcal{Z}[0]+\mathcal{O}^n[n]\bowtie\overline{\omega_{34}^n[0]}\Big|_{2^{2n}-1}\\&=\Big|\overline{x_4^{2n}[0]}+\overline{\omega_{34}^n[0]}+\mathcal{O}^{n-1}[n+1]\\&\qquad\bowtie\overline{x_4}[2n]\bowtie\underbrace{x_4[2n]\bowtie\cdots\bowtie x_4[2n]}_{n}\Big|_{2^{2n}-1}\end{aligned} \tag{17}$$

and for $\alpha = 1$

$$|-x_4-\omega_{34}|_{2^{2n}-1} = \left|\overline{x_4^{2n}[0]}+\overline{\omega_{34}^{2n}[0]}+\underbrace{x_4[2n]\bowtie\cdots\bowtie x_4[2n]}_{2n-1}\bowtie\mathcal{Z}[0]\right|_{2^{2n}-1}. \quad (18)$$

Therefore

$$Y = \Big|\omega_{12}\bowtie\omega_{12}+x_1+\overline{x_4^{2n}[0]}+\overline{\omega_{34}^{(1+\alpha)n}[0]} + \underbrace{(\beta\vee x_4[2n])\bowtie\cdots\bowtie(\beta\vee x_4[2n])}_{n-1}\bowtie(\beta\oplus x_4[2n]) \bowtie\underbrace{x_4[2n]\bowtie\cdots\bowtie x_4[2n]}_{n-1}\bowtie(\beta\wedge x_4[2n])\Big|_{2^{2n}-1}. \quad (19)$$

Applying (14) and (15) to (13)

$$\Omega = |Y\times 2^{\beta\times n}|_{2^{2n}-1}. \quad (20)$$

Finally, since $X_{34} < 2^{(\alpha+3)n+1}$ in (11), X can be computed applying (20) to (12)

$$X = \Omega\bowtie X_{34}+\mathcal{O}^{2n}[0]\bowtie\overline{\Omega^{2n}[0]}\bowtie\mathcal{O}^{(1+\alpha)n}[0]+1. \quad (21)$$

In the next section, we present a unified architecture for designing reverse converters for the considered moduli sets. The performance of these architectures is analyzed and compared with the related state of the art.

B. Proposed Architecture

The diagram of blocks in Fig. 1 represents the proposed unified architecture for designing reverse converters for the two considered moduli sets. For the delay and circuit area analysis, only binary and modulus 2^k-1 adders are considered. In this brief, we consider, as in [7], that the delay of a carry–propagate adder (CPA) with end-around carry (EAC) is twice the delay of a regular CPA, while the hardware cost is similar. Considering $\tau_{\rm FA}$ and $\Delta_{\rm FA}$ as the delay and area of a 1-bit full adder (FA), respectively, Table I provides values for the delay and circuit area required by the proposed converters and the ones in the related state of the art [7], [8].

As shown in Fig. 1, the values of ω_{12} are computed according to (7) using a modulo 2^n-1 CPA with EAC, while ω_{34} requires an n-bit or a $2n$-bit binary CPA for computing (10) with $\alpha=0$ or $\alpha=1$, respectively.

For calculating Y, an $(n+1)$-bit binary carry–save adder (CSA), two $2n$-bit CSAs, and a $2n$-bit CPA with EAC are used to compute (19), as shown in Fig. 1. Note that the resulting carry of that $(n+1)$-bit CSA can be accommodated in the last term in (19), rewriting the logical functions accordingly. To compute Ω, a modulo $2^{2n}-1$ multiplication by 2^n is required when $\alpha=0$(20), which is accomplished by rotating to the left the bits of Y by n positions. In parallel, X_{34} is computed with a CPA, as shown in Fig. 1. Although a $((3+\alpha)n+1)$-bit addition seems to be required at a glance in order to compute

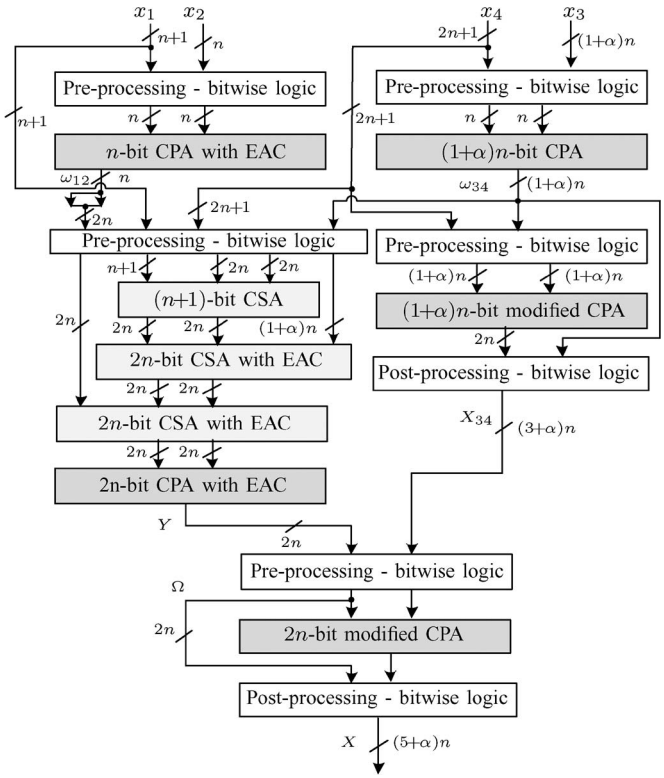

Fig. 1. Diagram of blocks of the reverse converter based on the proposed method; $\alpha=0$ for $m_3=2^n$, and $\alpha=1$ for $m_3=2^{2n}$.

TABLE I
AREA AND DELAY OF THE REVERSE CONVERTERS FOR DIFFERENT MODULI SETS (MODULI 2^n+1 AND 2^n-1 ARE COMMON FOR ALL SETS)

Moduli set	Converter	Delay (τ_{FA})	Area (Δ_{FA})
Converters with a dynamic range of $5n$-bit			
$\{\ldots, 2^n, 2^{2n+1}-1\}$	Proposed	$8n+1$	$13n+2$
	[7]	$12n+5$	$8n+2$
Converters with a dynamic range of $6n$-bit			
$\{\ldots, 2^{2n}, 2^{2n+1}-1\}$	Proposed	$8n+2$	$16n+1$
$\{\ldots, 2^{2n}, 2^{2n}+1\}$	[7]	$8n+3$	$10n+6$
$\{\ldots, 2^{2n+1}-3, 2^{2n}-2\}$	[8]	$14n+7$	$28n+6$

X_{34}, with a deeper analysis of (11), it is possible to conclude that the $2n+1$ MSBs of one of the operands are constant and all equal to "1." Therefore, $(1+\alpha)n$ FAs and $2n+1$ half adders (HAs) are required to compute the $(1+\alpha)n$ LSBs and the $2n+1$ MSBs of X_{34}, respectively. Considering the area of an HA $\Delta_{\rm HA}=(1/2)\Delta_{\rm FA}$, the cost of computing X_{34} is approximately equal to $(2+\alpha)n\times\Delta_{\rm FA}$.

Finally, a large binary adder is required to compute X from X_{34} and Ω according to (21). The $2n$ MSBs and the $(1+\alpha)n$ LSBs of one of the operands in (21) are constant and equal to "1." Therefore, we adopt a carry–select approach, with a total delay of $2n\times\tau_{\rm FA}$, and an area of $3n\Delta_{\rm FA}$. Note that the width of the multiplexer is only $2n$, referring to the constant MSBs, given that the sum with the $(1+\alpha)n$ less significant constant bits is equivalent to moving the input carry $(1+\alpha)n$ positions to the left.

For obtaining the values in Table I, the modulo 2^n-1 adder for computing ω_{12} takes $2n\times\tau_{\rm FA}$, which is, at most, the delay

required to compute ω_{34} in parallel. For computing Y, the delay imposed by the $(n+1)$-bit CSA is hidden in the delay for computing ω_{12}. Moreover, when $\alpha = 0$, the delay of the first $2n$-bit CSA is also hidden given that the computation of ω_{34} is not in the critical path for this case. Thus, this component of the delay that includes two CSAs and a $2n$-bit CPA, both with EAC, is $(4n + 1 + \alpha) \times \tau_{\mathrm{FA}}$. This component of the delay is larger than the delay to compute in parallel X_{34}. At the bottom part of Fig. 1, a CPA is required to compute X, which takes $2n \times \tau_{\mathrm{FA}}$, as stated before.

Regarding the area, the n-bit CPA to compute ω_{12} requires $n\Delta_{\mathrm{FA}}$, and the $(1+\alpha)n$-bit CPA employed for computing ω_{34} requires $(1+\alpha)n \times \Delta_{\mathrm{FA}}$. To compute Y according to (19), one $(n+1)$-bit CSA, two $2n$-bit CSAs, and a $2n$-bit CPA are used. However, as suggested in (19), the $n-1$ MSBs of the last operand are constant when $\alpha = 0$; thus, the area of one of the $2n$-bit CSAs is reduced to $(n+1) \times \Delta_{\mathrm{FA}}$ in this case. Hence, a total of $((6+\alpha)n + 2 - \alpha) \times \Delta_{\mathrm{FA}}$ is required to compute Y. Finally, $(2+\alpha)n \times \Delta_{\mathrm{FA}}$ is required for computing X_{34} and $3n\Delta_{\mathrm{FA}}$ for the final binary adder. The required total area is presented in Table I.

For $\alpha = 0$, we can expect a significant reduction of the delay regarding [7] with the proposed reverse converters at the cost of some extra circuit area. For $\alpha = 1$, the speedup of the proposed converter is incremental regarding [7] but almost two times regarding [8].

III. Experimental Results

In order to experimentally evaluate the practical interest of the proposed converters, not only these converters but also the ones presented in [7] and [8] were implemented. A well-known library of arithmetic units [18], written in synthesizable very high speed integrated circuit hardware description language (HDL) code, was employed to obtain the HDL specification of both the proposed and related art converters.[1] Using the HDL specification of the converters, implementations targeting field-programmable gate array (FPGA) and application-specific integrated circuit (ASIC) were accomplished. A Xilinx Virtex 4 (part xc4vlx200ff1513-11) FPGA is targeted, obtaining the programming bit stream using the Synopsys Synplicity Premier tools (version E-2010.09-SP2) for the synthesis procedure and the Xilinx ISE tools (version 12.4) for placing and routing. For the ASIC technology, the implementation was supported on a TSMC 65-nm general-purpose standard cell library (TCBN65GPLUS, version 200A) tailored for the TSMC 65-nm CMOS logic salicide process (1-poly, 9-metal), using the Cadence RTL Compiler tools (version v09.10-s242_1) for synthesizing the design and the Cadence Encounter and NanoRoute tools (versions v09.12-s159 and v09.12-s013, respectively) for placing and routing. For both FPGA and ASIC technologies, no manual optimizations of any kind were introduced.

Table II presents the experimental results obtained for the delay (τ), the circuit area (Δ), and the metric $\Delta\tau^2$ for the proposed and related art converters. Fig. 2 shows the relative improvements for the same metrics regarding the moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ in [7], which is the one that

[1]We made the HDL specification of the proposed and related art converters publicly available at http://sips.inesc-id.pt/~sfan/prototypes/rnsrevconv/.

TABLE II
Converters' Delay τ [in Nanoseconds], Area Δ [10^3 μm^2 or Slices], and $\Delta\tau^2$ Metric [10^3 μm$^2 \cdot$ ns^2 or 10^3 Slices $\cdot$ ns^2]

Tech.	Dyn. Rng.	Ref.	n=8			n=16			n=32		
			Δ	τ	$\Delta\tau^2$	Δ	τ	$\Delta\tau^2$	Δ	τ	$\Delta\tau^2$
ASIC	$5n$	Prop.	4.2	0.53	1.17	8.7	0.63	3.42	17.1	0.70	8.29
		[7]	3.9	0.60	1.38	7.7	0.71	3.92	15.2	0.82	10.35
	$6n$	Prop.	4.8	0.52	1.28	9.3	0.63	3.68	18.9	0.69	9.10
		[8]	5.6	0.71	2.82	11.7	0.80	7.50	23.6	0.96	21.92
		[7]	3.7	0.23	0.20	7.7	0.71	3.92	14.1	0.34	1.59
FPGA	$5n$	Prop.	261	10.7	30	502	14.1	100	958	16.6	264
		[7]	139	12.7	23	321	16.5	87	854	21.1	380
	$6n$	Prop.	197	13.5	36	789	15.0	177	970	23.0	511
		[8]	622	19.6	238	1177	21.3	536	2144	25.0	1338
		[7]	359	7.2	18	321	16.5	87	1117	10.3	119

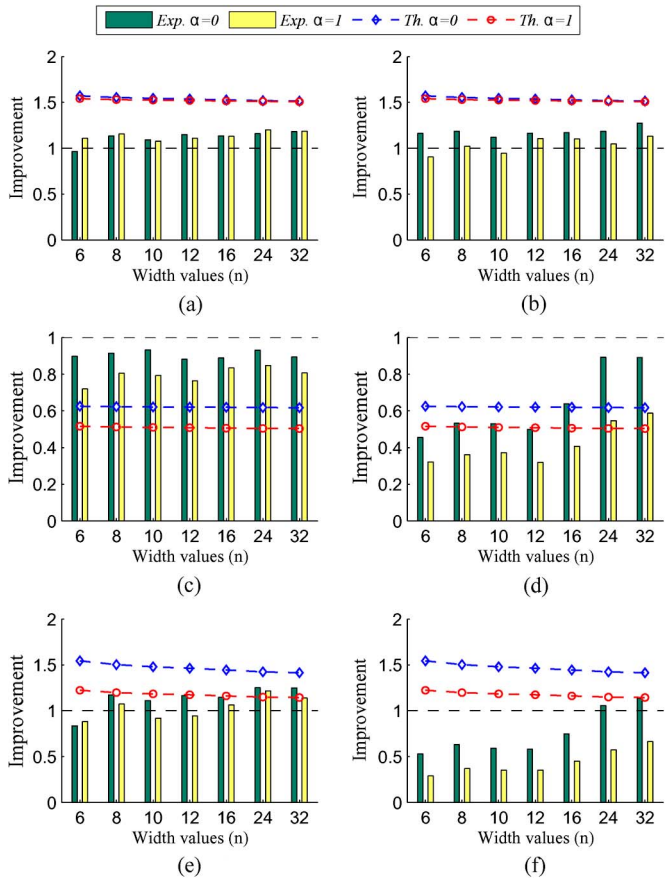

Fig. 2. Assessment (experimental and theoretical) of the proposed converters regarding the related state of the art converter for the moduli set $\{2^n+1, 2^2-1, 2^n, 2^{2n+1}-1\}$ in [7]; values higher than one mean improvement. (a) ASIC delay τ. (b) FPGA delay τ. (c) ASIC area Δ. (d) FPGA area Δ. (e) ASIC $\Delta\tau^2$. (f) FPGA $\Delta\tau^2$.

results in more efficient channel arithmetic [9]. Fig. 2 also shows the predicted results obtained from Table I.

The delay of the ASIC implementation of the proposed converter for the moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ $(\alpha = 0)$ is, on average, reduced by 12% compared to the related state of the art [see Fig. 2(a)]. The penalty in the area forced by the proposed converter is not as pronounced as theoretically expected [see Fig. 2(c)]. Since the proposed converter includes parallel computation flows, the synthesis tool can further optimize the resource usage on the flows that are not in the critical path, which benefits the proposed converter regarding area.

Therefore, the ASIC converter achieves a 25% improvement in the $\Delta\tau^2$ metric, which confirms the proposed converter as a well-balanced delay–area solution for the moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ [see Table II and Fig. 2(e)]. The proposed FPGA implementation delivers slightly higher improvement (18% on average) in conversion delay compared to the related state of the art. Concerning the area, the efficient usage of the FPGA's resources differs from that of the ASIC's in the sense that it is closely related with the mapping of the converters' architecture to the FPGA constructs, namely, the fast carry chains (for example, CPAs); the computation in the related state of the art converter is accomplished with a larger amount of CPAs [$(9n+2)$ FAs are used for the CPA implementation] than in the proposed one (only $7n$ FAs are used for the CPA implementation). Nevertheless, the proposed converter becomes more competitive for large values of n (e.g., $n=24$ and $n=32$).

The results for the moduli set $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}(\alpha=1)$ show that, with the proposed converters, we can extend the dynamic range without imposing extra delay, which confirms the prediction in Table I. The experimental results obtained from the ASIC implementation show that no significant penalty is introduced in the circuit area when the dynamic range is increased from $5n$ to $6n$, improving the $\Delta\tau^2$ metric for some values of n greater than or equal to 8 (up to 21%). The results for the FPGA also show a reduced delay, but regarding the $\Delta\tau^2$ metric, the results are not as competitive as in the ASIC implementation due to the aforementioned circuit area issues. As suggested in Table II, the proposed converter for the moduli set $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$ has worse performance than the one in [7] for the moduli set $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n}+1\}$. This deviation is related to unaccounted effects in the theoretical analysis in Table I, namely, place and routing complexity. However, that particular moduli set requires channel arithmetic units modulo $2^{2n}+1$, which is a clear disadvantage regarding the proposed moduli set. Moreover, when we compare the experimental results for the proposed converter in Table II with the ones for the moduli set $\{2^n+1, 2^n-1, 2^{2n+1}-3, 2^{2n}-2\}$, the proposed converters are significantly better in what concerns both imposed delay and circuit area.

Summarizing, the proposed method unifies the design of converters for the two moduli sets corresponding to different values of α. ASIC implementations in Table II suggest that throughputs ranging from 1435 to 1894 Mconversion/s can be obtained with the proposed converter, for the moduli set with $\alpha=0$ and dynamic ranges $5n=160$ and 40 bits, respectively. For the moduli set with $\alpha=1$, the throughputs even slightly increase to 1441 and 1931 Mconversion/s for the same values of n, and consequently larger dynamic ranges ($6n=192$ and 48 bits). Therefore, the proposed converters have interesting characteristics allowing addressing implementations with $5n$- and $6n$-bit dynamic ranges with approximately the same delay.

IV. Conclusion

In this brief, we have proposed a method, based on the MRC, to unify the design of reverse converters for the four-moduli sets $\{2^n+1, 2^n-1, 2^{(\alpha+1)n}, 2^{2n+1}-1\}$, with $\alpha=0,1$. Experimental results show not only that this method unifies the design of converters for both moduli sets but also that the proposed converters are more efficient than the related state of the art converters for the moduli set $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$. The delay of the proposed converters for $\alpha=0$, implemented on a Virtex 4 FPGA and on a 65-nm CMOS integrated circuit, is improved on average by 18% and 12%, respectively, compared to the related state of the art. Moreover, the figure-of-merit product of the area with the square of the delay is improved by up to 25% when these converters are implemented on ASICs. These results show the significance of the proposed unified method and architectures to design RNS reverse converters for the considered moduli sets, with dynamic ranges of $5n$ and $6n$ bits.

References

[1] S. Antão, J.-C. Bajard, and L. Sousa, "Elliptic curve point multiplication on GPUs," in *Proc. IEEE Int. Conf. ASAP*, Rennes, Apr. 2010, pp. 192–199.

[2] M. A. Soderstrand, W. K. Jenkins, G. A. Jullien, and F. J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. Piscataway, NJ: IEEE Press, 1986.

[3] R. Chaves and L. Sousa, "RDSP: A RISC DSP based on residue number system," in *Proc. Euromicro Symp. DSD*, Belek-Antalya, Turkey, Sep. 2003, pp. 128–135.

[4] P. V. A. Mohan, *Residue Number Systems: Algorithms and Architectures*. Norwell, MA: Kluwer, 2002.

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

[6] L.-S. Didier and P.-Y. Rivaille, "A generalization of a fast RNS conversion for a new 4-modulus base," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 1, pp. 46–50, Jan. 2009.

[7] A. S. 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 Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 823–835, Apr. 2010.

[8] W. Zhang and P. Siy, "An efficient design of residue to binary converter for four moduli set $\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$ based on new CRT II," *Inf. Sci.*, vol. 178, no. 1, pp. 264–279, Jan. 2008.

[9] A. A. Hiasat, "High-speed and reduced-area modular adder structures for RNS," *IEEE Trans. Comput.*, vol. 51, no. 1, pp. 84–89, Jan. 2002.

[10] C. Efstathiou, H. T. Vergos, and D. Nikolos, "Modulo $2^n \pm 1$ adder design using select-prefix blocks," *IEEE Trans. Comput.*, vol. 52, no. 11, pp. 1399–1406, Nov. 2003.

[11] L. Sousa and R. Chaves, "A universal architecture for designing efficient modulo 2^n+1 multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1166–1178, Jun. 2005.

[12] Y. Wang, "Residue-to-binary converters based on new Chinese remainder theorems," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 3, pp. 197–205, Mar. 2000.

[13] A. Skavantzos and T. Stouraitis, "Grouped-moduli residue number systems for fast signal processing," in *Proc. IEEE ISCAS*, Orlando, FL, Jul. 1999, pp. 478–483.

[14] N. B. Chakraborti, J. S. Soundararajan, and A. L. N. Reddy, "An implementation of mixed-radix conversion for residue number applications," *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 762–764, Aug. 1986.

[15] S. S. Erdem and Ç. K. Koç, "A less recursive variant of Karatsuba–Ofman algorithm for multiplying operands of size a power of two," in *Proc. IEEE Symp. ARITH*, Santiago de Compostela, Jun. 2003, pp. 28–35.

[16] R. Chaves and L. Sousa, "$\{2^n+1, 2^{n+k}, 2n-1\}$: A new RNS moduli set extension," in *Proc. Euromicro Symp. DSD*, Rennes, France, Aug. 2004, pp. 210–217.

[17] K. A. Gbolagade, R. Chaves, L. Sousa, and S. D. Cotofana, "Residue-to-binary converters for the moduli set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$," in *Proc. ICAST*, Accra, Ghana, Dec. 2009, pp. 26–33.

[18] R. Zimmermann, "VHDL library of arithmetic units," in *Proc. Int. FDL*, Lausanne, Switzerland, Sep. 1998, pp. 267–272.