Arithmetic-Based Binary-to-RNS Converter Modulo $\{2^n \pm k\}$ for jn-Bit Dynamic Range

Pedro Miguens Matutino, Ricardo Chaves, and Leonel Sousa

***Abstract*— In this brief, a read-only-memoryless structure for binary-to-residue number system (RNS) conversion modulo $\{2^n \pm k\}$ is proposed. This structure is based only on adders and constant multipliers. This brief is motivated by the existing $\{2^n \pm k\}$ binary-to-RNS converters, which are particular inefficient for larger values of n. The experimental results obtained for $4n$ and $8n$ bits of dynamic range suggest that the proposed conversion structures are able to significantly improve the forward conversion efficiency, with an AT metric improvement above 100%, regarding the related state of the art. Delay improvements of 2.17 times with only 5% area increase can be achieved if a proper selection of the $\{2^n \pm k\}$ moduli is performed.**

***Index Terms*— Arithmetic, binary-to-RNS, forward conversion, residue number systems.**

I. Introduction

Residue number system (RNS) is a nonweighted numbering system, which uses remainders to represent numbers. Its modular characteristics offer the potential for high-speed and parallel processing based on carry-free arithmetic [1]. The basic arithmetic operations (add, subtract, and multiply) are independently implemented over multiple channels, defined by the moduli set that supports each particular RNS. The RNS moduli set is set up by defining the moduli, where m_i represents positive relatively prime integers. A number X is represented in RNS by its residues $x_i = \langle X \rangle_{m_i}$, where x_i is the remainder of the division of X by m_i. Conversion from weighted number system to RNS (binary to-RNS or forward conversion), and vice versa (RNS-to-binary or reverse conversion), is required in order to implement a complete RNS-based processing system. Subsequently, RNS is usually used on computational intensive applications, such as digital signal processing, filtering, convolution, correlation, fast Fourier transform computation, and cryptography [2]–[4].

Initially, research on RNS was mainly focused on the three-modulus set $\{2^n - 1, 2^n, 2^n + 1\}$ [5]. More recently, different RNS moduli sets have been proposed in order to increase the dynamic range (DR) and/or reduce the width of the RNS channels, such as $\{2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3\}$ in [6] and [7], $\{2^n - 1, 2^{n+\beta}, 2^n + 1\}$ in [8], $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ in [9], and $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ in [10]. More recently, the moduli set with a DR up to $(8n + 1)$ bits has been proposed. This set [11] is composed by the moduli $\{2^{n+\beta}, 2^{n-1}, 2^{n+1}, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1, 2^{n+1} + 1\}$, requiring arithmetic units modulo $\{2^n \pm 1\}$ [12], and generic arithmetic units modulo $\{2^n \pm k\}$ [13] for the modulo $\{2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$ channels.

The use of the moduli $\{2^n \pm k\}$, with unrestricted k values, is rather useful to obtain larger RNS moduli sets [4], resulting in circuits with improved metrics. With larger moduli sets for the same DR, the operands are smaller, and consequently more compact arithmetic units are achieved, with reduced delay and circuit area requirements. However, most of the forward converters presented in the state of the art [14]–[19] are limited in terms of the number of bits per channel, or are unable to scale for larger DR, given their exponential growth with the number of bits per channel, mostly having structures based on lookup tables, such as read only memories (ROMs).

The structures proposed in [15] are based on weighted reduction, considering a serial, serial–parallel, or a fully parallel approaches. An optimized version of these structures, considering the periodicity property, was proposed in [16]. However, the periodicity property can only be used to improve the conversion when shorter period values for modulo $\{2^n \pm k\}$ can be found. More recently, a novel conversion structure has been proposed in [17] that overcomes this dependency using the distributive property instead of periodicity, allowing for unrestricted modulo values. In [18], multimoduli architectures are proposed, using a weight selection algorithm, with binary additions and ROMs. These architectures allow performing the same arithmetic operations for different moduli within the same structure, but suffer from the same lack of scalability for larger DR. Furthermore, since the brief herein presented is focused on simple single-modulo conversion structures, these are not herein considered. In [19], the analysis and implementation of a computer-aided design (CAD) tool is presented, capable of generating a structural description of binary-to-RNS converters, for a DR up to 21 bits. The proposed method actually allows achieving forward converters for any DR, as herein shown.

Herein, a novel ROM-less generic forward conversion structure for a DR of $m = jn$-bit, using $\{2^n \pm k\}$ moduli, is proposed, considering $n \geq 2$. The proposed approach splits the jn input bits into j input sets, and computes the respective residue value using modular additions and constant multiplications. The use of constant multipliers in the proposed scheme does not impose the exponential area increase as the ROM-based topologies [14] proposed in the related state of the art. To evaluate the performance of the proposed structure, the experimental results were obtained. These results suggest that the proposed forward conversion topology allows improvements of 85% in area and 15% in delay when compared with the ROM-based modulo $\{2^n \pm k\}$ binary-to-RNS converters proposed in [14]. Moreover, improvements up to 50% on delay can be achieved when comparing with the ROM-less-based converters proposed in [19], with a minimal increase in circuit area resources, 10% higher on average.

This brief is organized as follows. Section II introduces the formulation adopted to design the modulo $\{2^n \pm k\}$ binary-to-RNS

Manuscript received May 23, 2013; revised September 30, 2013, December 26, 2013, and February 24, 2014; accepted March 25, 2014. Date of publication April 15, 2014; date of current version February 20, 2015. This work was supported in part by the National Funds through the Fundação para a Ciência e a Tecnologia under project PEst-OE/EEI/LA0021/2013, in part by the FARNuSyC - Framework for Automatic RNS-Based Computation Project under Grant EXPL/EEI-ELC/1572/2013, and in part by the PROTEC Program under the Research Grant SFRH/PROTEC/49763/2009.

P. M. Matutino is with the High Institute of Engineering of Lisbon, Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento em Lisboa, Instituto Superior Técnico, Universidade de Lisboa, Lisbon 1649-004, Portugal (e-mail: pmmm@sips.inesc-id.pt).

R. Chaves and L. Sousa are with the Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento em Lisboa, Instituto Superior Técnico, Universidade de Lisboa, Lisbon 1649-004, Portugal (e-mail: ricardo.chaves@inesc-id.pt; las@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/TVLSI.2014.2314174

conversion structure described in Section III. Section IV presents the experimental results, and compares the proposed topology with the related state of the art. The conclusion is presented in Section V.

II. Formulation of RNS Conversion Modulo $\{2^n \pm k\}$

Considering a integer X with m-bit inputs, herein represented as $\{X_{[m-1]}, \ldots, X_{[1]}, X_{[0]}\}$ with $X = \sum_{i=0}^{m-1} 2^i \cdot X_{[i]}$, a forward converter for modulo $\{2^n \pm k\}$ transforms X into a residue value r with w_{mod} bits, $\{r_{[w_{\text{mod}}-1]}, \ldots, r_{[1]}, r_{[0]}\}$, with $w_{\text{mod}} = \lceil \log_2(2^n \pm k) \rceil$; $w_{\text{mod}} = n$ for modulo $\{2^n - k\}$ and $w_{\text{mod}} = n + 1$ for modulo $\{2^n + k\}$, when $0 < k < 2^n$. The residue modulo $\{2^n \pm k\}$ of the input value X can be achieved by computing the integer division of X by $2^n \pm k$, but it is a costly operation to obtain the remainder.

The approach herein considered to derive $\langle X \rangle_{2^n \pm k}$ is based solely on simple modular arithmetic operations. Given the propriety

$$\langle 2^n \rangle_{2^n \pm k} = \langle 2^n \pm k \mp k \rangle_{2^n \pm k} = \langle \mp k \rangle_{2^n \pm k} \tag{1}$$

and with $X_v \equiv X_{[(v+1)\cdot n-1:v\cdot n]}$, and considering X as the binary representation of an integer to be converted, with a DR of jn bits, where $X_{[msb:lsb]}$ represents the msb to lsb bits of integer X. The residue modulo $\{2^n - k\}$ of the value X with jn bits can be computed as

$$\begin{aligned}
\langle X \rangle_{2^n-k} &= \big\langle 2^{(j-1)n} X_{[jn-1:(j-1)n]} + \cdots + 2^{2n} X_{[3n-1:2n]} \\
&\quad + 2^n X_{[2n-1:n]} + X_{[n-1:0]} \big\rangle_{2^n-k} \\
&= \big\langle 2^{(j-1)n} X_{j-1} + \cdots + (2^n - k + k)(2^n - k + k) X_2 \\
&\quad + (2^n - k + k) X_1 + X_0 \big\rangle_{2^n-k} \\
&= \big\langle k^{j-1} X_{j-1} + \cdots + k^2 X_2 + k X_1 + X_0 \big\rangle_{2^n-k} \\
&= \left\langle \sum_{i=0}^{j-1} k^i X_i \right\rangle_{2^n-k}.
\end{aligned} \tag{2}$$

Identically, the residue modulo $\{2^n + k\}$ of X can be computed as

$$\begin{aligned}
\langle X \rangle_{2^n+k} &= \big\langle - k^{j-1} X_{j-1} + \cdots + k^2 X_2 - k X_1 + X_0 \big\rangle_{2^n+k} \\
&= \left\langle \sum_{i=0}^{\lfloor \frac{j-1}{2} \rfloor} k^{2i} X_{2i} - \sum_{i=0}^{\lfloor \frac{j-2}{2} \rfloor} k^{2i+1} X_{2i+1} \right\rangle_{2^n+k}.
\end{aligned} \tag{3}$$

The modular subtractions in (3) can be computed as

$$\begin{aligned}
\langle -X_l \rangle_{2^n+k} &= \big\langle 2^n - 1 - X_l + k + 1 \big\rangle_{2^n+k} \\
&= \big\langle \overline{X}_l + k + 1 \big\rangle_{2^n+k}.
\end{aligned} \tag{4}$$

Given (4), (3) can be rewritten as

$$\begin{aligned}
\langle X \rangle_{2^n+k} &= \left\langle \sum_{i=0}^{\lfloor \frac{j-1}{2} \rfloor} k^{2i} X_{2i} + \sum_{i=0}^{\lfloor \frac{j-2}{2} \rfloor} k^{2i+1} \left(\overline{X}_{2i+1} + k + 1 \right) \right\rangle_{2^n+k} \\
&= \left\langle \sum_{i=0}^{\lfloor \frac{j-1}{2} \rfloor} k^{2i} X_{2i} + \sum_{i=0}^{\lfloor \frac{j-2}{2} \rfloor} k^{2i+1} \overline{X}_{2i+1} \right. \\
&\quad \left. + \sum_{i=0}^{\lfloor \frac{j-2}{2} \rfloor} k^{2i+1} (k+1) \right\rangle_{2^n+k}
\end{aligned} \tag{5}$$

where

$$\left\langle \sum_{i=0}^{\lfloor \frac{j-2}{2} \rfloor} k^{2i+1} (k+1) \right\rangle_{2^n+k}$$

is a constant (cst).

III. Hardware Structures

To the best of the author's knowledge, the existing generic modulo $\{2^n \pm k\}$ binary-to-RNS converters are based on the weight-selection approach [14]–[19].

In the first approach proposed in [14], each n-bit segment of the binary input value is passed through a lookup table, outputting the corresponding residue value. Each table has an n-bit input and an n-bit output. The outputs of these lookup tables are then added by an adder-tree structure and further reduced modulo $\{2^n \pm k\}$ by an additional lookup table and a final $\{2^n \pm k\}$ modulo adder. This conversion structure is herein referred as Piestrak [14]. Given the exponentially area increases of the lookup tables, implemented by ROMs, this structure is only efficient for small values of n. In [15], a conversion structure considering weight selection based only on multiplexers and modular adders is proposed, herein referred to as Premkumar. The same author extend his work in [16] using the periodicity property, which can also be employed in the remaining state of the art. However, this technique can only be used to improve the conversion when shorter period values for modulo $\{2^n \pm k\}$ can be found. More recently, a new converter was proposed in [17], allowing the best conversion implementations for DR up to 64 bits with modulo up to 6 bits. This architecture is based on a depth-bounded carry-save addition, with a carry-free reduction of the sum and carry vectors. The final reduction is accomplished using a modified modulo adder and a lookup table. For larger moduli and DRs, the usage of the lookup table makes it once more not scalable. In [20], another weight-selection-based conversion approach is proposed. However, in this case, the resulting structure is based solely on full adders, allowing it to scale to larger moduli channels and DRs. The authors also made available a CAD tool to facilitate the implementation of the proposed conversion structures [19]. However, the developed tool only considers DR up to 21 bits. In order to obtain conversion structures for the considered DRs, an augmented version of this CAD tool was developed, considering the proposed method [20]. The version implemented by us now allows for the unrestricted DR, and was designed to use the same optimized arithmetic structures used in the herein proposed conversion structures. The conversions structures generated by the new tool are herein referred as Soudris.

Given the inadequacy of most of the related state-of-the-art structures, for larger values of n, two new generic binary-to-RNS conversion structures are herein proposed for modulo $\{2^n \pm k\}$, only requiring arithmetic operations. These conversion units only require constant multiplication and addition units: 1) Proposed I structure targets a more compact structure with a serialized modular reduction approach, which is an extension of the previous work presented in [21], but allowing now to compute X modulo $\{2^n \pm k\}$ with DR of jn bits instead of the previous $4n$ bits; 2) Proposed II structure targets faster conversion considering a parallel approach. In these proposed structures, a restriction to the width of k (w_k) is considered, allowing to simplifying the modular reduction step, namely $w_k \equiv \lceil \log_2 k \rceil \leq n/2$. Even with this restriction of w_k, the k^i constant value in (2) and (3) can be greater than $2^n \pm k$ for $i \geq 2$, since for $k < 2^{n/2}$ results in $k^i < 2^{n \cdot i/2}$. In these cases, the resulting modulo constant $\langle k^i \rangle_{2^n \pm k}$ can be precomputed and reduced modulo $\{2^n \pm k\}$, providing a constant value with w_{k^i} bits, where $w_{k^i} = \lceil \log_2(\langle k^i \rangle_{2^n \pm k}) \rceil$. The maximum width of k values is represented by $w_{k_{\max}} = \max(w_{k^i}),\ \forall\ 0 \leq i \leq j-1$.

Proposed II structure for modulo $\{2^n \pm k\}$ forward conversion is based on a parallel approach. In this approach, each partial operation $(k^i \cdot X_i)$ is first reduced modulo $\{2^n \pm k\}$, and only then added to obtain the final residue value. Thus, computation is performed in two stages: the first stage computes the constant multiplication vectors

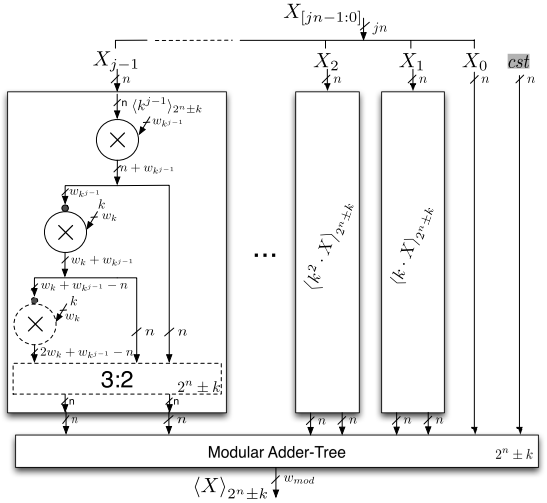

Fig. 1. Proposed II binary-to-RNS converter modulo $\{2^n \pm k\}$.

and the second stage performs the addition of all those vectors. This structure performs the modular reduction of each calculation, using the modulo $\{2^n \pm k\}$ Carry-Save-Adder [22], instead of adding all terms and reducing then iteratively at the end, as in Proposed I.

The considered modular k^i constant multiplication block computes $\langle\langle k^i\rangle_{2^n\pm k} X_i\rangle_{2^n\pm k}$ as

$$\begin{aligned}\left\langle\left\langle k^i\right\rangle_{2^n\pm k}\cdot X_i\right\rangle_{2^n\pm k} &= \left\langle k^i_{[w_{k^i}-1:0]}\cdot X_{[(i+1)\cdot n-1:i\cdot n]}\right\rangle_{2^n\pm k}\\ &= \left\langle P^1_{[n+w_{k^i}-1:0]}\right\rangle_{2^n\pm k}\\ &= \left\langle 2^n\cdot P^1_{[n+w_{k^i}-1:n]}+P^1_{[n-1:0]}\right\rangle_{2^n\pm k}\\ &= \left\langle \mp k\cdot P^1_{[n+w_{k^i}-1:n]}+P^1_{[n-1:0]}\right\rangle_{2^n\pm k}\\ &= \left\langle P^2_{[w_{k^i}+w_k-1:0]}+P^1_{[n-1:0]}\right\rangle_{2^n\pm k}. \end{aligned} \quad (6)$$

However, when $w_{k^i}+w_k > n$, the value P^2 has more than n bits, requiring an additional reduction step, computed as

$$\begin{aligned}&\left\langle\left\langle k^i\right\rangle_{2^n\pm k}\cdot X_i\right\rangle_{2^n\pm k}\\ &\quad= \left\langle \mp k\cdot P^2_{[w_{k^i}+w_k-1:n]}+P^2_{[n-1:0]}+P^1_{[n-1:0]}\right\rangle_{2^n\pm k}\\ &\quad= \left\langle P^3_{[w_{k^i}+2w_k-n-1:0]}+P^2_{[n-1:0]}+P^1_{[n-1:0]}\right\rangle_{2^n\pm k}. \end{aligned} \quad (7)$$

Therefore, the k^i constant modular multiplication can be implemented with two constant multipliers when $(w_{k^i}+w_k)\le n$, or with an additional constant multiplier and a 3:2 modular compressor [22], when $(w_{k^i}+w_k) > n$. The resulting modular k^i constant multiplication block is shown in Fig. 1. The additional resources for the more complex implementation case are represented with dashed lines. After the computation of $\langle\langle k^i\rangle_{2^n\pm k} X_i\rangle_{2^n\pm k}$, the resulting carry-and-save vectors are feed into a modular adder tree to obtain the final result $\langle X\rangle_{2^n\pm k}$. In the particular case of the binary-to-RNS modulo $\{2^n+k\}$, an additional input is required in the modular adder tree to compute the addition of the correction factor cst, from (6).

IV. Experimental Results

In order to fully evaluate the proposed binary-to-RNS converters and the related state of the art, all structures were described in Very High Speed Integrate Circuits Hardware Description Language and mapped to an application-specified integrated circuit technology, in particular for the United Microelectronics Corporation (UMC) 0.13-μm CMOS technology from UMC [23]. The ROM results have been obtained by synthesizing the available ROM sizes in the synchronous via-1 ROM compiler for the UMC 0.13-μm high-speed logic process technology, and the others were estimated based on the real obtained values. Both synthesis and mapping results were performed using Design Vision E-2010.12-SP4 from Synopsys.

In order to take into account the impact of different values for n and k, the presented experimental results were obtained for a variation of $n \in [6, 32]$. For the value k, the best case is obtained for $k = 3$, and for the worst case, the values $k = 2^{n/2}-1$ and $\langle k^i\rangle_{2^n\pm k} = 2^n-1$ are considered. The worst case value for k correspond to the worst case scenario for Proposed I and Proposed II structures implying the most complex and costly multipliers. Since for the Soudris structure the worst case scenario cannot be easily determined, and to simplify the analysis, the same value of k, $k = 2^{n/2}-1$, is used as the worst case value. For the Premkumar structure, the presented values are for $k = 3$, since no significant variations occur for other values of k. In order to properly evaluate the cost of the several conversion structures, RNS with four and eight moduli channels are considered, resulting in a DR of $4n$ and $8n$, respectively. The obtained experimental results for circuit area and delay are shown in Fig. 2(a), (b), (d), and (e).

As expected from the theoretical analysis [22], the ROM-based conversion structure proposed in [14] imposes significantly worst area metrics even for small DR. As the DR increases, and with it the size of the ROM, the area of ROM-based conversion structures significantly increases, due to the exponential increase of the ROM area. The Premkumar [15] structure has worst area metrics, as the Piestrak converter, for small modulo and DR. However, the Premkumar structure has less area than the Piestrak for larger modulo, for modulo channels with more than 10 bits. Nevertheless, the Premkumar structure requires up to 6.85 times more area resources than Proposed II. The Soudris conversion structure always requires more area resource for $k = 3$ when compared with Proposed II. However, when considering the worst case of k, the Soudris converters require less circuit area than the proposed converters. Note that the worst case scenario for the herein proposed conversion structures is not realistic since it considers that all the constant multipliers are in the most complex form. Still, less area demanding conversion units can be obtained with the proposed conversion structures, when compared with the related state of the art, in particular for larger values of n. Given the parallel approach of Proposed II structure, when compared with Proposed I, slightly higher area requirements are observed. From the obtained area results, it can also be concluded that the area increases with the number of moduli channels (j), since more constant multipliers and adders are needed, in particular for the more parallel Proposed II structure. The obtained experimental results suggest that the ROM-based related art (Piestrak) can in fact be faster, but only for values of n up to 18 bits. The modular adder-based conversion structure from Premkumar has worst delay metric than Piestrak for n up to 18 bits, but achieves better delay metrics for n greater than 20 bits. The Soudris conversion structure has the worst delay for the considered structures, nevertheless for n greater than 22 bits better delay metrics than the Piestrak conversion structure can be achieved. Regarding the proposed conversion structures, Proposed II conversion structure is able to achieve lower delay metrics than the related state of the art for n as low as 8 bits. Proposed II structure is always faster for n greater than 18 bits, even for the worst case of k, for acceptable area costs. The delay improvement inversion point depends on the number of modulo channels (j) and the considered k values. For example, the Premkumar converter allows to achieve better delay metric regarding ours in the worst cases. However, requiring on average 5.42 times more area resources. On average, the obtained results, for $k = 3$ and for the worst value of k, suggest that Proposed I structure requires on average 91%–87% less area,

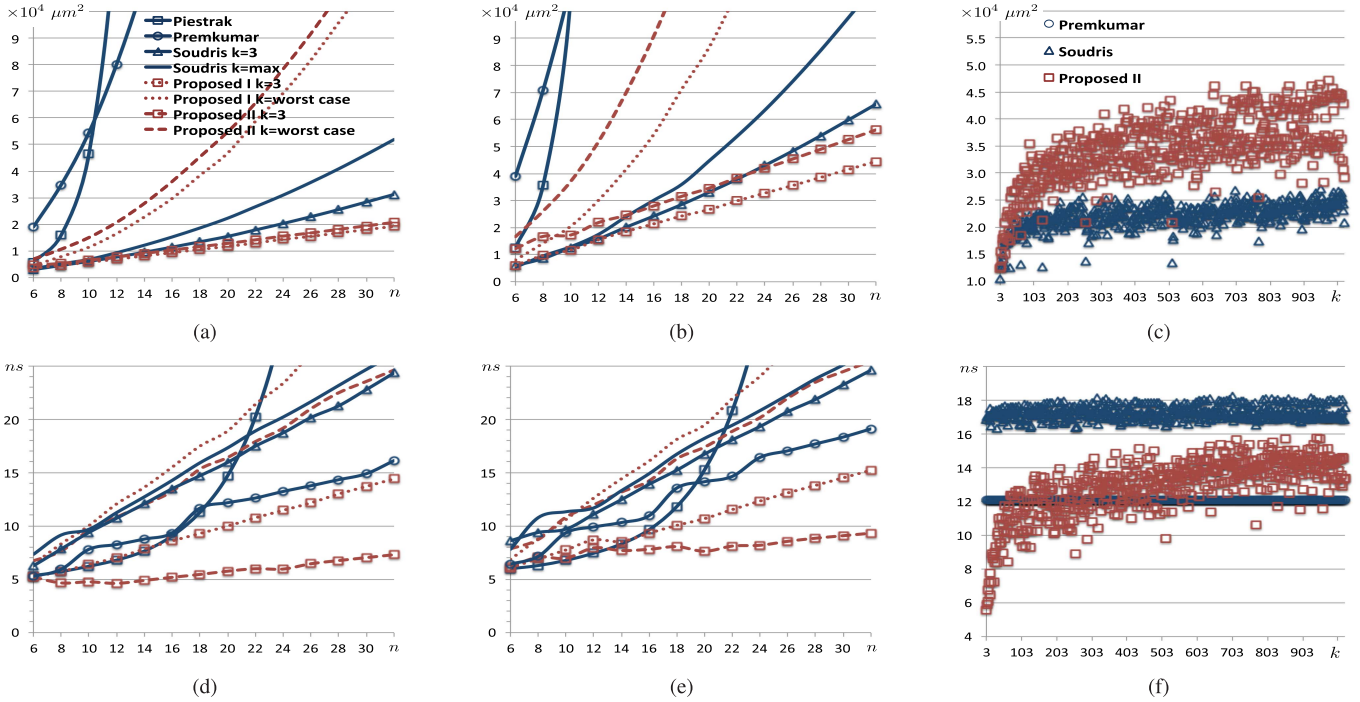

Fig. 2. Experimental results for binary-to-RNS converter with areas. (a) DR $4n$ - modulo $\{2^n \pm k\}$, (b) DR $8n$ - modulo $\{2^n \pm k\}$, and (c) $j = 4$, $n = 20$, varying k and delays, (d) DR $4n$ - modulo $\{2^n \pm k\}$, (e) DR $8n$ - modulo $\{2^n \pm k\}$, and (f) $j = 4$, $n = 20$, varying k.

and is on average 10% faster to 16% slower than the Piestrak and Premkumar conversion structures, respectively. Regarding the Soudris structure, it allows for a 12% faster conversion at a cost of 44% more area resources. The experimental results suggest that Proposed II structure is on average 85%–80% smaller, and 30%–5% faster than the Piestrak and the Premkumar structures, respectively. When compared with the Soudris structure, the experimental results suggest that Proposed II structure is on average 40% faster at a cost of 56% more circuit area. Note that this average is obtained from $k = 3$ and $k =$ worst case, which does not allow to fully interpret the resulting metrics. To better understand the variation of these metrics with the variation of the k value, the Premkumar, the Soudris, and Proposed II conversion structures were synthesized for all possible k values for $n = 20$, considering a DR of $4n$ bits and modulo $\{2^n - k\}$, as shown in Fig. 2(c) and (f). From these results, it can be seen that the Premkumar structure has a stable value for the delay metric, meaning that the obtained results are less dependent of the values of k. However, this invariance is achieved at significantly higher area costs, up to 18 times more area resources than Proposed II, being outside of the considered scale in Fig. 2(c). The obtained results also suggest that Proposed II structure is always faster than Soudris, in particular for smaller values of k. For the first 100 k values, the structure herein proposed achieves an average delay improvement of 48% when compared with the Soudris structure, at a cost of 36% more area resources. If the best 16 k values of each structure are considered, the delay of Proposed II structure is on average 2.17 times faster with just 5% additional area resources, resulting in an AT performance metric improvement of 108%. When considering the energy consumption per conversion, Proposed II structure suggests the largest variation, according to the selected k value, as shown in Fig. 3. Nevertheless, the obtained results suggest that with an adequate selection of the moduli set, (i.e., the k values) significantly less energy is required for the conversion. For example, for a moduli set with four channels, the Soudris and the Premkumar

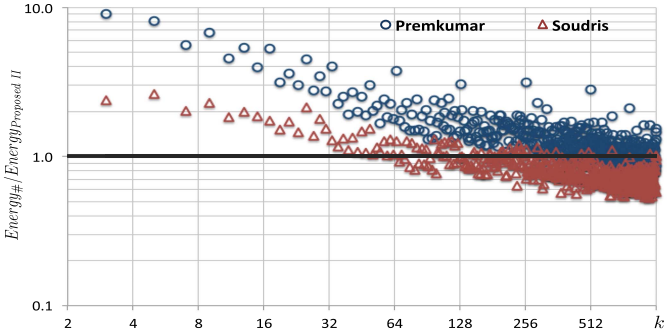

Fig. 3. Ratio of the energy consumption for $j = 4$, $n = 20$, varying k.

structures require 128% and 594% more energy, respectively, than Proposed II structure, with a modulo conversion requiring an average of 110 nJ. The results also suggest that, if the moduli set is extended to 16 channels, the Soudris and the Premkumar structures will require 50% and 264% more energy than Proposed II structure, respectively. From this, it can be concluded that Proposed II conversion structure allows for significantly more efficient conversion metrics, in particular if the k values are adequately selected.

V. Conclusion

In this brief, two generic and scalable modulo $\{2^n \pm k\}$ binary-to-RNS conversion structures are proposed for jn-bit DRs. The proposed approach splits the jn input bits into j input sets, and computes the respective residue value using modular additions and constant multiplications, implementing ROM-less structures. To assess the gains achieved by the proposed structures, the experimental results were obtained for $4n$- and $8n$-bit DRs. The obtained experimental results suggest that the proposed conversion approach allows for average delay improvements between 10% and 40%, with a worst

average area cost between 44% and 56%, regarding the best existing state of the art. Nevertheless, if a proper selection of the used k values is performed, 2.17 times faster conversion operations with only 5% extra area resources can be achieved, with AT performance metric improvements above 100%.

REFERENCES

[1] N. Szabo and R. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*. New York, NY, USA: McGraw-Hill, 1967.

[2] G. Cardarilli, A. Nannarelli, and M. Re, "Residue number system for low-power DSP applications," in *Proc. 41st ACSSC*, 2007, pp. 1412–1416.

[3] J. Bajard and L. Imbert, "A full RNS implementation of RSA," *IEEE Trans. Comput.*, vol. 53, no. 6, pp. 769–774, Jun. 2004.

[4] S. Antão, J.-C. Bajard, and L. Sousa, "RNS based elliptic curve point multiplication for massive parallel architectures," *Comput. J.*, vol. 55, no. 5, pp. 629–647, 2011.

[5] F. E. P. D. Gallaher and P. Srinivasan, "The digit paralell method for fast RNS to weighted number system conversion for specific moduli $\{2^n-1, 2^n, 2^n+1\}$," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 1, pp. 53–57, Jan. 1997.

[6] P. A. Mohan, "Reverse converters for the moduli sets $\{2^{2N}-1,\ 2^N,\ 2^{2N}+1\}$ and $\{2^N-3,\ 2^N+1,\ 2^N-1,\ 2^N+3\}$," in *Proc. SPCOM*, Dec. 2004, pp. 188–192.

[7] M.-H. Sheu, S.-H. Lin, C. Chen, and S.-W. Yang, "An efficient VLSI design for a residue to binary converter for general balance moduli $\{2^n-3,\ 2^n+1,\ 2^n-1,\ 2^n+3\}$," *IEEE Trans. Circuits Syst., Exp. Briefs*, vol. 51, no. 3, pp. 152–155, Mar. 2004.

[8] R. Chaves and L. Sousa, "$\{2^n+1, 2^{n+k}, 2^n-1\}$: A new RNS moduli set extension," in *Proc. EUROMICRO Syst. Digit. Syst. Des.*, Sep. 2004, pp. 210–217.

[9] A. Premkumar and A. P. Vinod, "A memoryless reverse converter for the 4-moduli superset $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$," *J. Circuits, Syst. Comput.*, vol. 10, no. 2, pp. 85–99, 2000.

[10] B. Cao, C.-H. Chang, and T. Srikanthan, "An efficient reverse converter for the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ based on the new Chinese remainder theorem," *IEEE Trans. Circuits Syst., Fundam. Theory Appl.*, vol. 50, no. 10, pp. 1296–1303, Oct. 2003.

[11] H. Pettenghi, R. Chaves, and L. Sousa, "RNS reverse converters for moduli sets with dynamic ranges up to (8n+1)-bit," *IEEE Trans. Circuits Syst., Reg. Papers*, vol. 60, no. 6, pp. 1–14, May 2012.

[12] R. Zimmermann, "Efficient VLSI implementation of modulo $\{2^n \pm 1\}$ addition and multiplication," in *Proc. 14th IEEE Symp. Comput. Arithmetic*, Jun. 1999, pp. 158–167.

[13] P. Matutino, H. Pettenghi, R. Chaves, and L. Sousa, "RNS arithmetic units for modulo $\{2^n \pm k\}$," in *Proc. 15th Euromicro Conf. DSD*, Sep. 2012, pp. 795–802.

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

[15] A. Premkumar, "A formal framework for conversion from binary to residue numbers," *IEEE Trans. Circuits Syst., Analog Digit. Signal Process.*, vol. 49, no. 2, pp. 135–144, Feb. 2002.

[16] A. Premkumar, E. Ang, and E.-K. Lai, "Improved memoryless RNS forward converter based on the periodicity of residues," *IEEE Trans. Circuits Syst., Exp. Briefs*, vol. 53, no. 2, pp. 133–137, Feb. 2006.

[17] J. Low and C.-H. Chang, "A new approach to the design of efficient residue generators for arbitrary moduli," *IEEE Trans. Circuits Syst., Reg. Papers*, vol. 60, no. 9, pp. 2366–2374, Aug. 2013.

[18] H. Pettenghi, L. Sousa, and J. Ambrose, "Efficient implementation of multi-moduli architectures for binary-to-RNS conversion," in *Proc. 17th ASP-DAC Asia South Pacific*, 2012, pp. 819–824.

[19] D. Soudris, M. Dasigenis, S. Vasilopoulou, and A. Thanailakis, "A CAD tool for architecture level exploration and automatic generation of RNS converters," in *Proc. IEEE ISCAS*, vol. 4. May 2001, pp. 730–733.

[20] D. Soudris, M. Dasygenis, and A. Thanailakis, "VLSI methodology for the design of RNS and QRNS full adder based converters," *IEE Proc. Circuits, Devices Syst.*, vol. 149, no. 4, pp. 241–250, Aug. 2002.

[21] P. M. Matutino, H. Pettenghi, R. Chaves, and L. Sousa, "Multiplier-based binary-to-RNS converter modulo $\{2n \pm k\}$," in *Proc. 26th Conf. DCIS*, Nov. 2011, pp. 125–130.

[22] P. M. Matutino, R. Chaves, and L. Sousa, "Theoretical analysis of modulo $\{2^n \pm k\}$ units," INESC-ID, Lisbon, Portugal, Tech. Rep., 2013.

[23] "High performance standard cell library (UMC 0.13 μm)," Virtual Silicon Technology Inc., Palo Alto, CA, USA, Tech. Rep., 2004.