Vol. 4.Special Issue ICCSII ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

A Detailed Study on the Moduli Number Effect on RNS Timing Performance

[1]Dina Younes, [2] Pavel Steffan

[1]Ph.D. Student, Brno University of Technology, Department of Microelectronics, Brno, Czech Republic

[2]Assoc. Prof., Brno University of Technology, Department of Microelectronics, Brno, Czech Republic

E-mail: [1]xyoune00@stud.feec.vutbr.cz, [2]steffan@feec.vutbr.cz

ABSTRACT

This paper studies the effect of the moduli number within a moduli set on the overall speed of the residue number system (RNS). Choosing a proper moduli set greatly affects the performance of the whole system. The widely known issue is that as the number of moduli increases the speed of the residue arithmetic units (RAUs) increases, whereas the residue-to-binary converters (RCs) become slower and more complex. Thus, we carried out a detailed study on different moduli sets with different moduli number and different dynamic ranges (DRs) and compared timing performance of systems based on them in order to determine the moduli number effect on the overall RNS timing performance and find out the most efficient set for each DR.

Keywords: *Residue number system, moduli set, dynamic range, reverse converters, residue arithmetic units*

1. INTRODUCTION

The carry-free, parallelism, high-speed and security features of the residue number system (RNS) have made it very attractive to be used in digital signal applications (DSPs). The RNS divides the required computations into a number of parallel faster ones according to the number of moduli. These channels are totally independent and perform high-speed computations on smaller residues [1], [2]. This system has been intensively used in applications where addition, subtraction and multiplication are dominant, such as, digital filters, digital communications, discrete Fourier transform (DFT), image processing, and video coding [1], [2], [3]. However, the RNS could not be widely implemented in general-purpose processors, since operations as division, sign detection, magnitude comparison and overflow detection are problematic, and can negatively influence the overall performance of the design. Many solutions for these problematic operations were suggested. Most of them mainly depend on the residue-to-binary converter (reverse converter (RC)). This component represents the main over-head in the whole RNS. On the other hand, choosing a proper moduli set is another essential issue for building an efficient RNS with a sufficient dynamic range (DR). The most famous moduli set is $\{2^n - 1, 2^n, 2^n + 1\}$ [4]. This set has been known as a means of simplifying the calculations necessary to implement the RC. However, this set has modulo $(2^n + 1)$ channel that represents the bottleneck of the system. Its arithmetic circuits suffer from the longest delay among all three channels. In general, arithmetic circuits modulo $(2^k - 1)$ are more efficient than those modulo $(2^k + 1)$, therefore, it is better to reduce the number of moduli of the form $(2^k + 1)$ [3]. Thus, in order to simplify the complexity caused by modulo $(2^n + 1)$ in the set $\{2^n - 1, 2^n, 2^n + 1\}$ [4], new moduli sets $\{2^{n-1} - 1, 2^n - 1, 2^n\}$ [5] and $\{2^n - 1, 2^n, 2^{n+1} - 1\}$ [6], that substitute this modulo with another of the form $(2^k - 1)$, were suggested. These three sets have a 3*n*-bit DR, which is sufficient for applications that require medium DRs (less than 22-bits). However, many DSP applications require larger DRs, therefore, new moduli sets $\{2^n - 1, 2^n, 2^{2n+1} - 1\}$ [7] and $\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$ [8] that provide 4*n*-bit DR and $\{2^n, 2^{2n} - 1, 2^{2n} + 1\}$ [9] that provides 5*n*-bit DR, were suggested.

Although the DR is larger, the delay of the residue arithmetic units (RAUs) based on these sets has considerably increased, due to utilizing moduli with greater magnitudes. In order to eliminate this drawback and maintain the large DR, sets of four and five moduli have been suggested, such as $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ [10-I], $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ [10-II], $\{2^{n/2} - 1, 2^{n/2} + 1, 2^n + 1, 2^{2n+1} - 1\}$ [20],$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ [11], $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ [12-I], $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$ [13], $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$ [14], $\{2^n - 1, 2^{2n}, 2^n + 1, 2^{2n} + 1\}$ [12-II], $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n+1} + 1\}$ [15], $\{2^n, 2^{n/2} - 1, 2^{n/2} + 1, 2^n + 1, 2^{2n-1} - 1\}$ [16], $\{2^n + 1, 2^n - 1, 2^{2n}, 2^{2n+1} - 1\}$ [21] and $\{2^{2n+1}, 2^{2n} + 1, 2^n + 1, 2^n - 1\}$ [22]. Each of these sets has its own advantages and disadvantages. Some of them offer higher DRs than others, while others have more parallelism. Some can result in more efficient RCs, while others in more efficient RAUs. However, the main concentration of publications that introduce these sets is on the RCs. Moreover, each of them compares its proposed set with other two or three. Thus, we have observed a lack to a detailed comparative study that fairly compares these sets, in terms of the DR, number of moduli, time and hardware requirements for implementing RCs and RAUs based on them. Consequently, we have decided to do this research, deeply study each of these sets, evaluate and compare as fair as possible their RCs and RAUs. The studied moduli sets are classified based on the DR they provide (3*n*, 4*n*, 5*n* and 6*n*).

The effect of moduli number on the overall system's timing performance is a well-known fact. More moduli lead to more parallelism, which theoretically means faster computations within the independent RAUs. However, this advantage transforms into a disadvantage regarding designing the RCs. Thus, the second part of our research was dedicated for studying the relation between the moduli number and the overall system timing performance. According to the RAUs and RCs, time and hardware requirements are computed in Sections 4, 5, 6 and 7, we estimated the most efficient and inefficient moduli sets for three DRs (medium, large and very large). For the sake of a fair comparison, we have adopted the

unit gate model [17]. Furthermore, we have used same basic blocks among all designs (e.g. adders and multipliers).

The rest of this paper is organized as follows; a brief overview of RNS is presented in Section 2. The basic blocks that were used during our study and comparisons, in addition to their time and hardware requirements, are illustrated in Section 3. Sections 4, 5, 6 and 7 present the time and hardware requirements for implementing the modular adders, modular multipliers and RCs based on each of the studied moduli sets for DR = 3*n*, 4*n*, 5*n* and 6*n*, respectively. Section 8 demonstrates a comparison between all studied sets for three precise DRs, 12 bits (medium DR), 24 bits (large DR) and 60 bits (very large DR). This section compares the timing performance of all components in systems based on the moduli sets under study. Finally, the conclusions and future work are drawn in Section 9.

2. RNS OVERVIEW

The RNS is defined by a set of positive pairwise relatively prime numbers $\{m_1, m_2,\ldots, m_n\}$ called moduli. The dynamic range (DR) is defined as $M = m_1 \times m_2 \times \ldots \times m_n$. In this system, any integer X in the range $[0, M-1]$ can be uniquely represented by an ordered set of residues $(x_1, x_2,\ldots, x_n)$. Each residue x_i is represented by,

$$x_i = X \bmod m_i = \langle X \rangle_{m_i} \quad ; \quad 0 \le x_i < m_i \tag{1}$$

In this system, arithmetic operations (addition, subtraction and multiplication) are performed totally in parallel on those totally independent residues.

$$(x_1, x_2,\ldots, x_n) \circ (y_1, y_2,\ldots, y_n) = \left(\langle x_1 \circ y_1 \rangle_{m_1}, \langle x_2 \circ y_2 \rangle_{m_2}, \ldots, \langle x_n \circ y_n \rangle_{m_n}\right) \quad ; \quad \circ \equiv (+,-,\times) \tag{2}$$

A residue number $(x_1, x_2,\ldots, x_n)$ can be converted back into its weighted equivalent, by using one of the residue-to-binary conversion algorithms (reverse conversion algorithms), such as, the Chinese Reminder Theorem (CRT), the Mixed-Radix Conversion (MRC), the new CRT-I, the new CRT-II, etc. [1], [2].

The CRT can be implemented in parallel. However, it needs a large modular adder, which can be very difficult for hardware implementation. According to the CRT, a weighted number X can be calculated from its residues $(x_1, x_2,\ldots, x_n)$ by the following equation,

$$X = \left\langle \sum_{i=1}^{n} \left| x_i N_i \right|_{m_i} M_i \right\rangle_M \quad \text{Where,} \quad M_i = \frac{M}{m_i} \quad , \quad N_i = \left\langle M_i^{-1} \right\rangle_{m_i} \quad , \quad i = 1,2,\ldots,n \tag{3}$$

$\left\langle M_i^{-1} \right\rangle_{m_i}$ denotes the multiplicative inverse of M_i modulo m_i.

The MRC does not need any modular adder, but it is a sequential algorithm, which makes it not suitable for systems with more than four moduli within the set. By using the MRC, a residue number $(x_1, x_2,\ldots, x_n)$ can be converted back into its weighted equivalent X by,

$$X = v_n \prod_{i=1}^{n-1} m_i + \cdots + v_3 m_2 m_1 + v_2 m_1 + v_1$$
$$\text{Where,} \quad v_1 = x_1, \; v_2 = \left\langle (x_2 - v_1) \left\langle m_1^{-1} \right\rangle_{m_2} \right\rangle_{m_2}, \tag{4}$$
$$v_n = \left\langle \left(\left((x_n - v_1) \left\langle m_1^{-1} \right\rangle_{m_n} - v_2 \right) \times \left\langle m_2^{-1} \right\rangle_{m_n} - \cdots - v_{n-1} \right) \left\langle m_{n-1}^{-1} \right\rangle_{m_n} \right\rangle_{m_n}$$

The new CRT-I is a modification of the original CRT, where the size of the final modular adder is reduced. By using this algorithm, a residue number $(x_1, x_2,\ldots, x_n)$ can be converted back into its weighted equivalent X by,

$$X = x_1 + m_1 \times \left\langle \begin{matrix} k_1(x_2 - x_1) + k_2 m_2 (x_3 - x_2) + \cdots + \\ + k_{n-1} m_2 m_3 \cdots m_{n-1} (x_n - x_{n-1}) \end{matrix} \right\rangle_{m_2 m_3 \cdots m_n}$$
$$\text{Where,} \quad \langle k_1 \times m_1 \rangle_{m_2 m_3 \cdots m_n} = 1 \tag{5}$$
$$\langle k_2 \times m_1 \times m_2 \rangle_{m_3 \cdots m_n} = 1$$
$$\langle k_{n-1} \times m_1 \times m_2 \times \cdots \times m_{n-1} \rangle_{m_n} = 1$$

As can be noticed in equation 5, the final modular adder is reduced by one modulo. This can bring a great benefit when the first modulo is of the 2^k form, and the multiplication of the rest moduli is of the $(2^k - 1)$ form.

The new CRT-II even further reduces the size of the final modular adder. A residue number $(x_1, x_2,\ldots, x_n)$ can be converted back into its weighted equivalent X by the new CRT-II by,

$$X = Z + m_1 m_2 \langle k_1 (Y - Z) \rangle_{m_3 m_4}$$
$$Z = x_1 + m_1 \langle k_2 (x_2 - x_1) \rangle_{m_2} \tag{6}$$
$$Y = x_3 + m_3 \langle k_3 (x_4 - x_3) \rangle_{m_4}$$
$$\text{Where,} \quad \langle k_1 m_1 m_2 \rangle_{m_3 m_4} = 1, \; \langle k_2 m_1 \rangle_{m_2} = 1, \; \langle k_3 m_3 \rangle_{m_4} = 1$$

3. CONSIDERATIONS AND BASIS OF THE COMPARISONS

The comparison between moduli sets has been done based on time and hardware requirements for implementing the RCs and RAUs. For the sake of a fair comparison, the unit gate model has been used [17].

Before beginning, the considerations that have been taken into account must be listed. In order to make the comparison process universal, cost effective and high-speed designs were not considered. That is the reason why some time and area values are different from those in the original publications. The time and area requirements (referred to as 'T' and 'A') for each of the following components are as follows, gates and multiplexers: the time and area requirements for an inverter (NOT gate) were ignored. Each 2-input monotonic gate (AND, OR, NAND, NOR): $T_{AND} = 1$, $A_{AND} = 1$. Each 2-input XOR, XNOR gate: $T_{XOR} = 2$, $A_{XOR} = 2$. A 2:1 multiplexer: $T_{MUX} = 2$, $A_{MUX} = 3$.

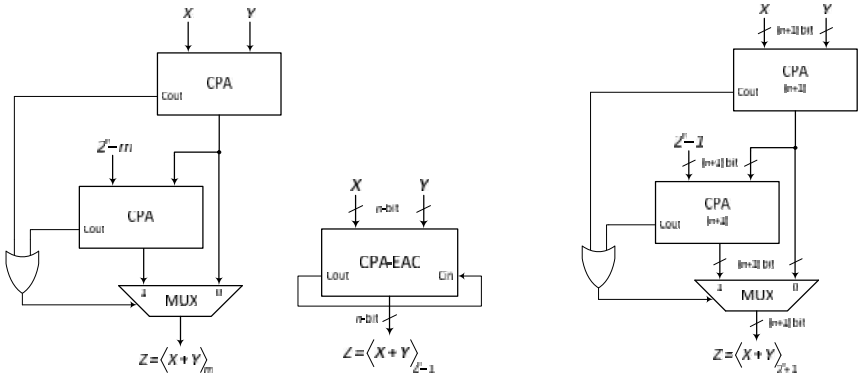

Fig. 1: Structures of the utilized modular adders
(a) General modular adder. (b) Modulo ($2^n - 1$) adder. (c) Modulo ($2^n + 1$) adder.

Binary adders: A half adder (HA): $T_{HA} = 2$, $A_{HA} = 3$. A full adder (FA): $T_{FA} = 4$, $A_{FA} = 7$. Any binary adder was considered as a carry propagate adder for *n*-bit (CPA). A CPA with end-around carry (CPA-EAC) has a delay twice that of a regular CPA, and the same area. The structures of the used modular adders are shown in Fig. 1. The structure of the utilized general modular adder consists of two (*n*-bit) CPAs, an OR gate and a 2:1 multiplexer of *n*-bit [3]. The ($2^n - 1$) modular adder (1st complement adder) is considered as a CPA-EAC [2]. The used ($2^n + 1$) modular adder is the one based on the series method reported in [18], which consists of two binary adders of (*n*+1)-bit, an OR gate and a 2:1 multiplexer of (*n*+1) bit. The structure of this adder is very similar to that of the general modular adder. However, we have used it, due to its wide utilization and simplicity.

Binary multipliers: The used binary multiplier for *n*-bit is an array multiplier [2]. The structures of the utilized modular multipliers are shown in Fig. 2. The utilized general modular multiplier is the partitioned-operand modulo-*m* multiplier, which is constructed based on the product-partitioning method presented in [2]. The utilized modular multipliers are based on those reported in [19]. Therefore, modulo ($2^n - 1$) multiplier consists of a binary multiplier of *n*-bit and modulo ($2^n - 1$) adder. The utilized modulo ($2^n + 1$) multiplier consists of a binary multiplier of (*n*+1) bit and modulo ($2^n + 1$) subtractor.

The delay and complexity of circular shifting or bits rearrangement was ignored, since it is just a rearrangement of wires, which does not virtually present any additional delay or hardware cost. The last issue to be observed is that the computed delay in the following tables represents the delay of the critical channel (modulo), whereas the hardware requirements (complexity) are those for all channels together. The values of the least delays and hardware requirements are bold and underlined in order to highlight them.

4. MODULI SETS WITH 3N DYNAMIC RANGE

The first category of the studied moduli sets is the one that provides a DR of 3*n*-bits. The three sets that have been investigated are, $\{2^n - 1, 2^n, 2^n + 1\}$ [4], $\{2^{n-1} - 1, 2^n - 1, 2^n\}$ [5] and $\{2^n - 1, 2^n, 2^{n+1} - 1\}$ [6]. They provide the following DRs, (3*n*), (3*n* – 1), and (3*n* + 1), respectively. However, we have considered that these sets belong to the 3*n*-DR category. The most famous and widely used moduli set is the one introduced in [4]. Nevertheless, as aforementioned before, modulo ($2^n + 1$) represents a bottleneck to the whole system. In order to get rid of the complexities resulted from using this modulo, the authors of [5] and [6] substituted it with another modulo of ($2^k - 1$) form.

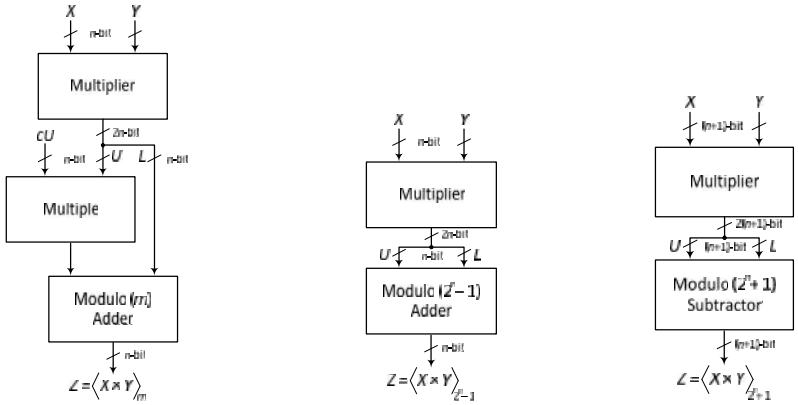

Fig. 2: Structures of the utilized modular multipliers
(a) General modular multiplier. (b) Modulo ($2^n - 1$) multiplier. (c) Modulo ($2^n + 1$) multiplier. Where, *U* and *L* are the upper and lower *n* bits of the product ($X \times Y$), respectively, (a) $c = \langle 2^n \rangle_m$ and $L < m$. (c) modulo ($2^n + 1$) subtracts ($L - U$).

Table 1: Comparison between reverse converters, modular adders and modular multipliers for systems based on sets that provide DR = 3*n*

Moduli Set	DR	*n* odd/ even	mod #	RC		Critical Channel	Modular Adders		Modular Multipliers	
				Delay	*Complexity*		*Delay*	*Complexity*	*Delay*	*Complexity*
$\{2^n-1, 2^n, 2^n+1\}$ [4]	$3n$	any	3	$16n+8$	**$31n+13$**	(2^n+1)	$8n+11$	$38n+18$	$16n+12$	$24n^2+7n+15$
$\{2^{n-1}-1, 2^n-1, 2^n\}$ [5]	$3n-1$	any	3	$24n-2$	$54n-45$	(2^n-1)	**$8n$**	**$21n-7$**	**$16n-7$**	**$24n^2-35n+12$**
$\{2^n-1, 2^n, 2^{n+1}-1\}$ [6]	$3n+1$	any	3	**$8n+30$**	$110n+159$	$(2^{n+1}-1)$	$8n+8$	$21n+7$	$16n+9$	$24n^2-27n+4$

Table 1 details each of these sets, its DR, possible *n* values that can be used in this set, the number of its moduli, the critical channel that presents the longest delay, time and hardware requirements for implementing RCs, modular adders and modular multipliers.

a. Reverse Converters

Both RCs based on the sets [4] and [5] are based on the CRT. Whereas in [6], three architectures of the RC were proposed, they depend on MRC, CRT and CRT with ROM. During our research and comparisons, we have chosen the second architecture that is based on the CRT. The computed delay in [5] is different from the reported one, due to the reason that it was calculated based on the aforementioned considerations, (e.g. the carry look-ahead adder (CLA) was replaced by a CPA).

According to our study, the fastest converter was [6] (for $n \geq 3$), whereas [4] was the one with the least cost (for $n \geq 3$). However, considering the balanced strategy, the RC based on [4] is the most efficient one.

b. Residue Arithmetic Units

The RAU includes modular adders, subtractors and multipliers. Since a subtractor can be constructed by an adder and a few inverters, we have decided to compare just adders and multipliers. The critical channels of the studied sets are (2^n+1), (2^n-1) and $(2^{n+1}-1)$ in [4], [5] and [6], respectively.

As shown in table 1, the superiority of the adders and multipliers based on the set in [5] is clear. According to this table, the critical channel among all sets is (2^n+1); both arithmetic components based on the set [4] have the longest delay. However, this channel makes the RC simpler. As mentioned above, the RC based on the set [4] is the most efficient one.

5. MODULI SETS WITH 4N DYNAMIC RANGE

In this section, we have investigated five sets; the three-moduli sets $\{2^n-1, 2^n, 2^{2n+1}-1\}$ [7] and $\{2^n-1, 2^n+1, 2^{2n}+1\}$ [8], and the four-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$, $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ [10] and $\{2^{n/2}-1, 2^{n/2}+1, 2^n+1, 2^{2n+1}-1\}$ [20]. For simplicity, we have denoted $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ as [10-I] and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ as [10-II]. [10-I] and [20] can only be used with even values of *n*, whereas, [10-II] can only be used with odd ones. This presents a kind of limitation for using such sets. All these sets have $(4n+1)$ DRs, except [8] that has $4n$ DR. Table 2 details each of these sets, its DR, possible *n* values that can be used in this set, the number of its moduli, the critical channel that presents the longest delay, time and hardware requirements for implementing RCs, modular adders and modular multipliers.

a. Reverse Converters

The RC based on the set [7] depends on the MRC. Whereas, the one based on the set [8] depends on the new CRT-I. The RCs based on the sets [10-I] and [10-II] have combinatorial architectures. Each of them is partitioned into two new parts. The first part is converted by using the RC based on the set $\{2^n-1, 2^n, 2^n+1\}$. Then the final binary equivalent is calculated from the result of the first part and the fourth residue by using the MRC. The four-moduli set [20] depends on the new CRT-I. As shown in table 2, the RC based on the three-moduli set [8] has the least delay and complexity. The reason of this superiority is obvious, the final modular adder is of (2^k-1) form. The second best set regarding the RC is [20]. According to [20], the moduli within this set have simple multiplicative inverses. This considerably reduces the complexity of the RC, which is obvious in table 2. On the other hand, the delay and complexity of the RCs, based on both four-moduli sets [10-I] and [10-II], are the greatest.

Table 2: Comparison between reverse converters, modular adders and modular multipliers for systems based on sets that provide DR = $4n$

Moduli Set	DR	*n* odd/ even	mod #	RC		Critical Channel	Modular Adders		Modular Multipliers	
				Delay	*Complexity*		*Delay*	*Complexity*	*Delay*	*Complexity*
$\{2^n-1, 2^n, 2^{2n+1}-1\}$ [7]	$4n+1$	any	3	$40n+20$	$69n+20$	$(2^{2n+1}-1)$	$16n+8$	**$38n+25$**	$32n+9$	$48n^2+9n+4$
$\{2^n-1, 2^n+1, 2^{2n}+1\}$ [8]	$4n$	any	3	**$32n+8$**	**$62n+8$**	$(2^{2n}+1)$	$16n+11$	$58n+36$	$32n+12$	$48n^2+62n+15$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ [10-I]	$4n+1$	even	4	$46n+28$	$15n+5+7/2(n^2-3n-4)$	(2^n+1)	**$8n+11$**	$45n+25$	**$16n+12$**	**$32n^2+19n+19$**
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ [10-II]	$4n+1$	odd	4	$48n+62$	$7n^2+102n+108$	$(2^{n+1}+1)$	$8n+19$	$55n+53$	$16n+28$	$32n^2+45n+60$
$\{2^{n/2}-1, 2^{n/2}+1, 2^n+1, 2^{2n+1}-1\}$ [20]	$4n+1$	even	4	$32n+37$	$68n+37$	$(2^{2n+1}-1)$	$16n+8$	$43n+43$	$32n+9$	$44n^2+59n+34$

Table 3: Comparison between reverse converters, modular adders and modular multipliers for systems based on sets that provide DR = 5*n*

Moduli Set	DR	*n* odd/ even	mod #	RC		Critical Channel	Modular Adders		Modular Multipliers	
				Delay	*Complexity*		*Delay*	*Complexity*	*Delay*	*Complexity*
$\{2^n, 2^{2n}-1, 2^{2n}+1\}$ [9]	5*n*	even	3	**$32n+4$**	**$44n+8$**	$(2^{2n}+1)$	$16n+11$	$55n+18$	$32n+12$	$72n^2+25n+15$
$\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ [11]	5*n*	any	4	$32n+12$	$95n+39$	$(2^{2n}+1)$	$16n+11$	$72n+36$	$32n+12$	$56n^2+51n+30$
$\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ [12-I]	5*n*+1	any	4	$48n+20$	$74n+14$	$(2^{2n+1}-1)$	$16n+8$	$52n+25$	$32n+9$	$56n^2+31n+19$
$\{2^n-1, 2^n, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1\}$ [13]	5*n*	odd	5	$32n+20$	$184n-9$	$(2^n+2^{(n+1)/2}+1)$	**$8n+11$**	$72n+37$	$24n+13$	$56n^2+29n+28$
$\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}+1\}$ [15]	5*n*	even	5	$72n+4l+28$	$7\times(5n^2+43n+m)/6+112n-7$	$(2^{n+1}+1)$	$8n+19$	$62n+36$	**$16n+28$**	**$40n^2+25n+72$**
$\{2^n, 2^{n/2}-1, 2^{n/2}+1, 2^n+1, 2^{2n-1}-1\}$ [16]	5*n*–1	even	5	$52n+4$	$97n+11$	$(2^{2n-1}-1)$	$16n-8$	**$50n+29$**	$32n-23$	$52n^2-20n+42$

b. Residue Arithmetic Units

Since [10-I] has the critical channel with the least magnitude, its RAUs are the fastest. In addition, the multiplier based on this set has the least area requirements. However, the least cost adder is the one based on the three-moduli set [7]. On the other hand, using four moduli in [20] did not help in speeding up the RAUs, since the magnitude of the critical modulo is greater than those of other sets. According to our results, using two small moduli of the form $(2^{n/2} \pm 1)$ has not benefit the system, since the magnitude of the critical channel remains big.

6. MODULI SETS WITH 5N DYNAMIC RANGE

In this section, we have investigated six sets; the three-moduli set $\{2^n, 2^{2n}-1, 2^{2n}+1\}$ [9], the four-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ [11] and $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ [12-I], and the five moduli sets $\{2^n-1, 2^n, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1\}$ [13] , $\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}+1\}$ [15] and $\{2^n, 2^{n/2}-1, 2^{n/2}+1, 2^n+1, 2^{2n-1}-1\}$ [16]. The sets [9], [15] and [16] can only be used with even values of *n*, whereas, [13] can only be used with odd ones. Each of the sets [9], [11], [13] and [15] provides 5*n* DR, whereas [12-I] and [16] provide (5*n* + 1) and (5*n* – 1) DRs, respectively.

Table 3 details each of these sets, its DR, possible *n* values that can be used in this set, the number of its moduli, the critical channel that presents the longest delay, time and hardware requirements for implementing RCs, modular adders and modular multipliers.

a. Reverse Converters

The RC based on the set [9] depends on the CRT. This RC has a very simple and efficient structure consisting of a (4*n*-bit) CSA-EAC and modulo $(2^{4n}-1)$ adder. The set [11] depends on the new CRT-I. In case of combining its first and third moduli, we get the set [9]. However, the RC based on [11] has greater delay and complexity than that of [9]. The RC based on the set [12-I] depends on the new CRT-II. The memory-less RC based on the set [13] depends on the CRT. The set [15] is partitioned into two new parts. The first part is converted by using the RC based on $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$. Then, the final binary equivalent is calculated from the result of the first part and the fifth residue (modulo $(2^{n+1}+1)$) by the MRC.

The delay and complexity of this RC do not only depend on the value of *n*, but also on two more variables *l* and *m*; where, *l* denotes the number of the levels of a CSA tree with $((n/2)+1)$ inputs, and $m = n-4$, $9n-12$ or $5n-8$ for $n = 6k-2$, $6k$ or $6k+2$, respectively [15]. The set [16] is also partitioned into two new parts, $\{2^n(2^{n/2}-1)(2^{n/2}+1)(2^n+1), 2^{2n-1}-1\}$. The first part is converted by using the new CRT-I. Then the result of the first part is combined with the fifth residue by using the MRC. According to table 3, the most efficient RC, in terms of delay and complexity, is the one based on the three-moduli set [9]. Contrary, the most inefficient RC is the one based on the five-moduli set [15].

b. Residue Arithmetic Units

As shown in table 3, the adder with the least delay is the one based on [13]. At first glance, modulo $(2^{n+1}+1)$ adder should have less delay than that of $(2^n+2^{(n+1)/2}+1)$, but it does not for the following reasons, the structure of $(2^{n+1}+1)$ adder depends on a (2^k+1) adder, its length is $(n+2)$ bits. Whereas the structure of $(2^n+2^{(n+1)/2}+1)$ adder depends on a general modular one, its length is $(n+1)$ bits. Since the utilized modulo (2^k+1) adder has a similar structure as that of the general modular one, as mentioned in section 3, thus, the delay of $(2^n+2^{(n+1)/2}+1)$ adder is less than that of $(2^{n+1}+1)$. In case of using another structure of (2^k+1) adder during our study, the delay of $(2^n+2^{(n+1)/2}+1)$ may be longer than that of $(2^{n+1}+1)$, however, the area consumption will be different too. The cost of the adder based on [13] is the greatest one. The least cost adder is the one based on the five-moduli set [16]. Concerning the multipliers, the one based on the five-moduli set [15] was superior over the other ones. In contrast to the general modular adders, general modular multipliers have considerably longer delay than special modular ones.

7. MODULI SETS WITH 6N DYNAMIC RANGE

In this section, we have investigated four sets, $\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$ [14], $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n}+1\}$ [12-II], $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$ [21] and $\{2^{2n+1}, 2^{2n}+1, 2^n+1, 2^n-1\}$ [22] that consist of four moduli.

Table 4: Comparison between reverse converters, modular adders and modular multipliers for systems based on sets that provide DR = 6*n*

Moduli Set	DR	*n* odd/ even	mod #	RC		Critical Channel	Modular Adders		Modular Multipliers	
				Delay	*Complexity*		*Delay*	*Complexity*	*Delay*	*Complexity*
$\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$ [14]	6*n*+1	any	4	56*n* + 39	188*n* + 57	$(2^{2n+1}-3)$	16*n* + 11	92*n* + 37	48*n* + 13	$144n^2 + 62n + 28$
$\{2^n-1, 2^{2n}, 2^n+1, 2^{2n}+1\}$ [12-II]	6*n*	any	4	**32*n* + 12**	**88*n* + 24**	$(2^{2n}+1)$	16*n* + 11	72*n* + 19	**28*n* + 12**	$80n^2 + 40n + 30$
$\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$ [21]	6*n*+1	any	4	40*n*	98*n* + 7	$(2^{2n+1}-1)$	**16*n* + 8**	**52*n* + 25**	32*n* + 9	**$80n^2 + 20n + 19$**
$\{2^{2n+1}, 2^{2n}+1, 2^n+1, 2^n-1\}$ [22]	6*n*+1	any	4	**32*n* + 12**	88n + 28	$(2^{2n}+1)$	16*n* + 11	72*n* + 34	**28*n* + 12**	$80n^2 + 72n + 27$

During our research we did not come across a moduli set of more than four moduli that provides a DR of 6*n*-bits. All these sets provide (6*n* + 1) DR, except [12-II] that provides 6*n* DR. Table 4 details each of these sets, its DR, possible *n* values that can be used in this set, the number of its moduli, the critical channel that presents the longest delay, time and hardware requirements for implementing RCs, modular adders and modular multipliers.

a. Reverse Converters

The RC based on the set [14] depends on the new CRT-II. According to [12], the authors of [14] used moduli $(2^{2n} - 2, 2^{2n+1} - 3)$ in order to have simple multiplicative inverses and get rid of the ROM tables and multiplications required by the new CRT-II. However, the use of these two moduli results in increasing the delay and complexity of the RC. The RC based on the set [12-II] depends on the new CRT-II. According to the authors, this set is referred to as a conversion friendly set, which is evident in table 4. In [21], the four-moduli set was partitioned into two new moduli subsets, and the MRC was used for the conversion process. The RC in [22] is based on the new CRT-I. As shown in table 4, the fastest RCs are the ones based on [12-II] and [22]. Moreover, the complexity of their RCs is very similar too. On the other hand, the RC with the longest delay and largest complexity is the one based on [14].

b. Residue Arithmetic Units

The critical modulo in the set [14] has almost the greatest magnitude among all other sets and its arithmetic circuits are general modular ones, therefore, their delay and cost are the greatest. According to table 4, the fastest and the least complex modular adder is [21]. The fastest multiplier is one based on [12-II] and [22]. The least complex multiplier is the one based on [21]. The inefficiency of using general moduli as in [14] is clear. Its RC and RAUs have the longest delays and the highest complexities.

8. COMPARING ALL SETS TOGETHER

In our previous paper [23], we have presented a detailed comparison between different moduli sets based on (delay × complexity) ratio for each component according to three precise DRs (12 bit, 24 bit and 60 bit). In this paper, we compared different moduli sets based on the delay of each component. Thus, we can study how the moduli number affects timing performance of the overall system. Moreover, in our previous paper [23], the little differences in the DRs were ignored, e.g. 3*n*, 3*n* – 1 and 3*n* + 1 were considered as one category (3*n*). For example, for DR = 12 bit, the accurate DRs of [4], [5] and [6] for *n* = 4 are 12, 11 and 13 bits, respectively. This presents some inconsistency during the comparison. Therefore, in this paper, for enhancing and comparing in a more fairly way we have taken into account these little differences and computed the approximated delays for each set. However, this did not change the results neither affect the estimated most efficient and inefficient sets for each DR. The main concentration during the comparisons was timing performance.

The delays of the RCs, modular adders and modular multipliers for each of the three precise DRs, (12-bit, 24-bit and 60-bit) are shown in Fig. 3, 4 and 5, respectively.

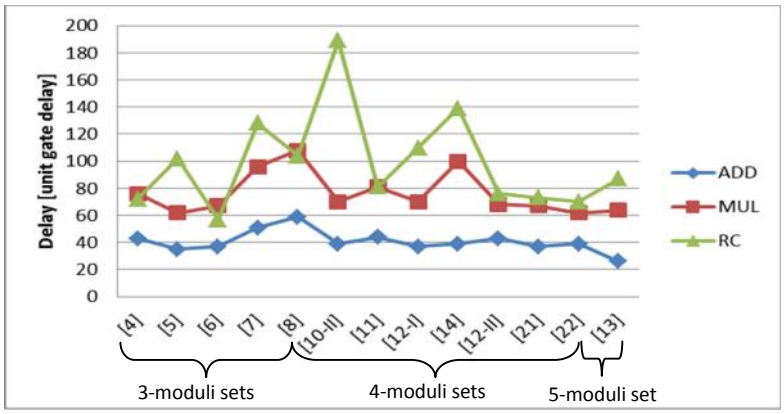

Fig. 3: The delays of each of the basic components based on the moduli sets for DR = 12 bit (medium DR)

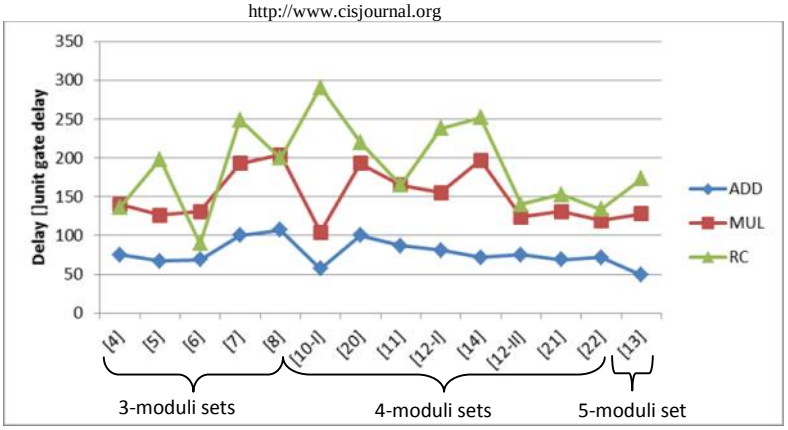

Fig. 4: The delays of each of the basic components based on the moduli sets for DR = 24 bit (large DR)

In each set, *n* has been chosen in order to provide the required DR. For example, for DR = 12-bit, *n* was (4, 3, 3, 2) for sets with DR (3*n*, 4*n*, 5*n*, 6*n*). However, in some cases there was some inconsistency (e.g. for sets with DR = 5*n*, *n* = 3 provides a DR greater than the required 12-bit). Nevertheless, as aforementioned before, we have dealt with this issue and estimated the approximate delay for the required DR.

Not all sets are illustrated in the graphs, due to the reason that some of these sets can only be used with even values of *n* ([9], [10-I], [20], [15] and [16]) or odd values of *n* ([10-II] and [13]), which does not fit the chosen value of *n* in order to acquire the required DR.

Figure 3 shows the delays of the RCs, modular adders and modular multipliers based on each of the sets [4], [5], [6], [7], [8], [10-II], [11], [12-I], [14], [12-II], [21], [22] and [13], for DR = 12-bit (medium DR). In order to acquire this DR, *n* was chosen (4, 3, 3, 2) for sets with DR (3*n*, 4*n*, 5*n*, 6*n*), respectively. The inconsistencies were dealt with as mentioned above. According to Fig. 3, the adder with least delay was the one based on the five-moduli set [13]. However, the unexpected thing was that the second fastest adder is the one based on the three-moduli set [5]. The fastest multipliers were the ones based on the three-moduli set [5] and four-moduli set [22]. Whereas, the slowest ones were based on the three-moduli set [8] and the four-moduli set [14]. Concerning the RCs, the fastest one is [6] and the slowest one is based on the four-moduli set [10-II]. Considering the delay of the three components, it is obvious that the three-moduli set [6] and the four-moduli set [22] are the most efficient, since the components based on these sets have relatively small delays. Thus, for DR = 12 bit, we see that the number of moduli does not affect the overall speed of the system.

Figure 4 shows the delays of the RCs, modular adders and multipliers based on each of the sets [4], [5], [6], [7], [8], [10-I], [20], [11], [12-I], [14], [12-II], [21], [22] and [13], for DR = 24-bit (large DR). In order to acquire this DR, *n* was chosen (8, 6, 5, 4) for sets with DR (3*n*, 4*n*, 5*n*, 6*n*), respectively. The inconsistencies were dealt with as mentioned above. Figure 4 shows that the delay trends of the basic components are similar to those in DR = 12 bit. The fastest adder is the one based on the five-moduli set [13]. The four-moduli set [10-I] has the fastest multiplier and one of the best adders. However, its RC is the worst. The fastest RC is the one based on the three-moduli set [6]. Considering the delay of the three components, we can say that the most efficient moduli sets for this DR = 24-bit are the three-moduli sets [4] and [6], and the four-moduli sets [12-II] and [22], since the components based on these sets have relatively small delays.

Figure 5 shows the delays of the RCs, modular adders and multipliers based on each of the sets [4], [5], [6], [7], [8], [10-II], [9], [11], [12-I], [14], [12-II], [21], [22], [15] and [16], for DR = 60-bit (very large DR). In order to acquire this DR, *n* was chosen (20, 15, 12, 10) for sets with DR (3*n*, 4*n*, 5*n*, 6*n*), respectively. The inconsistencies were dealt with as mentioned above. Figure 5 shows that the fastest adder and multiplier are the ones based on the five-moduli set [15]. However, their RC is the slowest one. The fastest RC is the one based on the three-moduli set [6]. This set also has reasonably fast adder and multiplier.

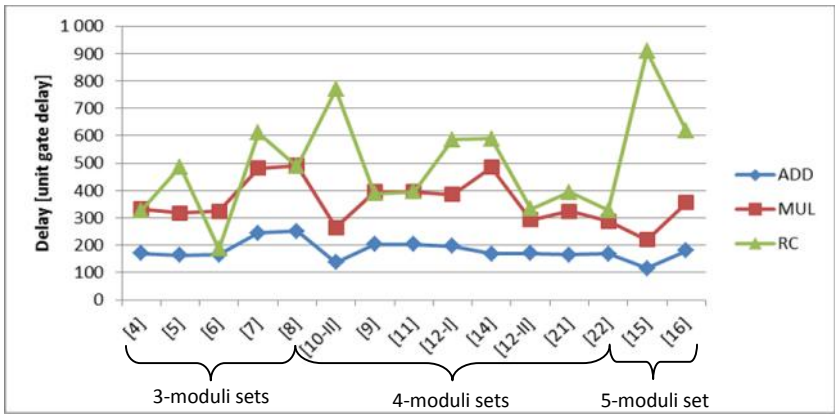

Fig. 5: The delays of each of the basic components based on the moduli sets for DR = 60 bit (very large DR)

Considering the delay of the three components, we can say that the most efficient moduli sets for this DR = 60-bit are the three-moduli sets [4] and [6], and the four-moduli sets [12-II] and [22], as the components based on these sets have relatively small delays.

Since the most competent sets are not the five-moduli ones for all three DRs, we conclude that the number of moduli does not affect on the overall delay of the system considering all its components. There is no point for choosing a five-moduli set if the overall timing performance will be worse than that based on three or four-moduli sets.

In our previous paper [23], the most efficient sets for medium and large DRs, in terms of (time × complexity) ratio, were the three moduli sets [5] and [6]. In this paper, we found out that the most efficient sets for the same DRs, in terms of delay, are the three moduli set [4], [6] and the four-moduli set [12-II] and [22]. Again, in [23], the most competent set with the relatively best (D × C) ratios for very large DR was the four moduli set [12-II]. In this paper, the most efficient sets for the same DR are the three-moduli sets [4], [6], and the four moduli sets [12-II] and [22].

Another important issue to consider is that five-moduli sets show better timing performance in medium and large DRs than that in very large DR. Although their RAUs were of the best ones for the very large DR, their RCs were the worst.

According to our research, the unexpected issue we have ascertained is that five-moduli sets do not show any superiority over other sets taking into account the three components of RNS (modular adders, modular multipliers and RCs).

9. CONCLUSIONS AND FUTURE WORK

This paper presented a detailed study on the effect of moduli number on the overall speed of the RNS. The study consists of two main parts, the first one compares between different moduli sets that provide same DR ($3n$, $4n$, $5n$ and $6n$). Time and hardware requirements of the RCs and RAUs based on each set are presented and described in a universal way using unit gate model.

The second part is devoted for comparing timing performance of the aforementioned components based on the studied sets, for three specific dynamic ranges; medium (12-bits), large (24-bits) and very large (60-bit). This part of the study was carried out in order to find-out the most efficient set for each dynamic range regardless the number of its moduli, and the dynamic range it was initially suggested for. According to the commonly known issue, which says, as the moduli number increases the speed of the RAUs increases and the RC becomes slower and more complex, we expected to have a kind of linear relationship between the number of moduli and the delay of the aforementioned components. However, the results in our study lead to the fact that the moduli number does not have such an essential role in the overall speed of the RNS, since timing performance of the components based on five-moduli sets were not better than those based on three and four-moduli sets.

Our next step will be dedicated for studying the effect of the moduli number on the hardware complexity of the overall system in RNS.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Industry and Trade of the Czech Republic under the MPO ČR č. FR-TI3/485 project and Prospective applications of new sensor technologies and circuits for processing of sensor signals, No.FEKT-S-11-16 project.

REFERENCES

[1] M. Lu, Arithmetic and Logic in Computer Systems. John Wiley & Sons, Inc., 2004.

[2] A. Omondi and B. Premkumar, Residue Number Systems: Theory and Implementation. Imperial College Press, 2007.

[3] K. Navi, A. S. Molahosseini and M. Esmaeildoust, "How to teach residue number system to computer scientists and engineers," IEEE Trans. on Education, vol. 54, pp. 156 – 163, February 2011.

[4] S. J, Piestrak, "A high-speed realization of a residue to binary number system converter," IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 42, pp. 661 – 663, October 1995.

[5] W. Wang, M. N. S. Swamy, M. O. Ahmad and Y. Wang, "A high-speed residue-to-binary converter for three-moduli (2^k, $2^k - 1$, $2^{k-1} - 1$) RNS and a scheme for its VLSI implementation," IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 47, pp. 1576 – 1581, December 2000.

[6] 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, pp. 775 – 779, September 2007.

[7] A. S. Molahosseini, K. Navi and M. K. Rafsanjani, "A new residue to binary converter based on mixed-radix conversion," 3rd International Conference on ICTTA, pp. 1 – 6, April 2008.

[8] W. Wang, M. N. S. Swamy, M. O. Ahmad and Y. Wang, "A study of the residue-to-binary converters for the three-moduli sets," IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications, vol. 50, pp. 235 – 243, February 2003.

[9] A. Hariri, K. Navi and R. Rastegar, "A new high dynamic range moduli set with efficient reverse converter," Computers & Mathematics with Applications Journal, vol. 55, pp. 660 – 668, February 2008.

[10] P. V. A. Mohan and A. B. 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. on Circuits and Systems-I: Regular Papers, vol. 54, pp. 1245 – 1254, June 2007.

[11] B. Cao, C. H. Chang and T. Srikanthan, "An efficient reverse converter for the 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$," IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications, vol. 50, pp. 1296 – 1303, October 2003.

[12] 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. on Circuits and Systems-I: Regular Papers, vol. 57, pp. 823 – 835, April 2010.

[13] A. A. Hiasat, "VLSI implementation of new arithmetic residue to binary decoders," IEEE Trans. on VLSI Systems, vol. 13, pp. 153 – 158, January 2005.

[14] 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," Information Sciences Journal, vol. 178, pp. 264 – 279, March 2008.

[15] B. Cao, C. H. Chang and T. Srikanthan, "A residue-to-binary converter for a new five-moduli set," IEEE Trans. on Circuits and Systems-I: Regular Papers, vol. 54, pp. 1041 – 1049, May 2007.

[16] A. S. Molahosseini, C. Dadkhah and K. Navi, "A new five-moduli set for efficient hardware implementation of the reverse converter," IEICE Electronics Express, vol. 6, pp. 1006 – 1012, July 2009.

[17] A. S. Molahosseini, K. Navi, O. Hashemipour and A. Jalali, "An efficient architecture for designing reverse converters based on a general three-moduli set," Elsevier Journal of Systems Architecture, vol. 54, pp. 929 – 934, October 2008.

[18] S. Timarchi and K. Navi, "Improved modulo 2^n+1 adder design," World Academy of Science, Engineering and Technology, vol. 39, July 2008.

[19] A. A. Hiasat, "New memoryless, mod $(2^n \pm 1)$ residue multiplier," Electronics Letters, vol. 28, pp. 314 – 315, January 1992.

[20] A. S. Molahosseini, F. Teymouri and K. Navi, "A new four-modulus RNS to binary converter," Proc. of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 4161 – 4164, June 2010.

[21] L. Sousa and S. Antão, "MRC-based RNS reverse converters for the four-moduli sets $\{2^n + 1, 2^n - 1, 2^n, 2^{2n+1} - 1\}$ and $\{2^n + 1, 2^n - 1, 2^{2n}, 2^{2n+1} - 1\}$," IEEE Trans. on Circuits and Systems II: Express Briefs, vol. 59, pp. 244 – 248, April 2012.

[22] R. Aliabadian, A. Aliabadian, A. Bolhasani, S. Z. Hosseini and A. Golsorkhtabar, "A novel high dynamic range 4-module set $\{2^{2n+1}, 2^{2n} + 1, 2^n + 1, 2^n - 1\}$ with efficient reverse converter and review improving modular multiplication's dynamic range with this module set," International Conference on Computer Communication and Informatics (ICCCI), pp. 1 – 6, January 2012.

[23] D. Younes and P. Steffan, "A comparative study on different moduli sets in residue number system," International Conference on Computer Systems and Industrial Informatics (ICCSII), pp. 1 – 6, December 2012.

AUTHOR PROFILES

D. Younes received her B.Sc. and M.Sc. degrees in Computer and Automatic Control Engineering from Tishreen University, Latakia, Syria in 2008. Currently, she is a Ph.D. student at the Faculty of Electrical Engineering and Communication at Brno University of Technology, Brno, Czech Republic. Her main research interests is residue number system based building blocks for applications in digital signal processing. She is a part time instructor in the Department of Microelectronics. She is author and co-author of several research national and international conference and journal papers.

P. Steffan received his B.Sc. and M.Sc. degrees in Microelectronics from Brno University of Technology, Brno, Czech Republic in 2000 and 2003, respectively. He also received his Ph.D. degree in Optimization of microelectronic systems for smart microsystems from Brno University of Technology in 2007.
Since 2011, he is Associate Professor at the Department of Microelectronics, Faculty of Electrical Engineering and Communication, Brno University of Technology. His research interests are technical area measurement, processing and digitization of sensors signals. Currently, he is teaching courses of digital circuits and microprocessor technology. He is author and co-author of over than 100 research national and international conference and journal papers.