Circuits Syst Signal Process (2011) 30:1445–1461
DOI 10.1007/s00034-011-9326-5

# On the Design of Modulo $2^n \pm 1$ Subtractors and Adders/Subtractors

**E. Vassalos · D. Bakalis · H.T. Vergos**

Received: 8 January 2010 / Revised: 19 May 2011 / Published online: 22 June 2011


**Abstract** Novel architectures for designing modulo $2^n + 1$ subtractors and combined adders/subtractors are proposed in this manuscript. Both the normal and the diminished-one representations of the operands are considered. Unit gate estimates and CMOS VLSI implementations reveal that the proposed modulo $2^n + 1$ subtractors for operands in the normal representation are more efficient than those previously proposed. The proposed diminished-one modulo $2^n + 1$ subtractors have a complexity similar to that of the corresponding diminished-one adders. Modulo $2^n - 1$ subtractors and adders/subtractors are also considered for the sake of completeness and a comparison between alternative architectures is provided.

**Keywords** Residue number system · Modulo $2^n \pm 1$ arithmetic circuits · Subtraction · Addition · Normal and diminished-one modulo $2^n + 1$ number representations

## 1 Introduction

The Residue Number System (RNS) is a non-weighted, carry-free number system [2, 25], well-suited to applications in which the operations are limited to addition, subtraction, multiplication and squaring.

E. Vassalos · D. Bakalis (✉)
Electronics Laboratory, Physics Department, University of Patras, Patras, 26500, Greece
e-mail: bakalis@physics.upatras.gr

E. Vassalos
e-mail: vassalos@upatras.gr

H.T. Vergos
Computer Engineering and Informatics Department, University of Patras, Patras, Greece
e-mail: vergos@ceid.upatras.gr

In an RNS, an operand is represented by its residues over a moduli set. Every arithmetic operation is carried out in parallel units each performing a computation on narrow residues instead of the wide operand. In this way, significant speedup over a traditional system that follows the binary representation may be achieved. As a consequence, the RNS has been considered for the design of communication components [17, 22, 24, 26], cryptographic circuits [3], digital signal processors and reconfigurable datapaths [8, 9, 29], DCT processors [13], FIR filters [7, 21, 28] and image processing units [23, 33].

Almost all RNSs use some modulus of the $2^n \pm 1$ form, mainly due to the existence of efficient architectures for the required conversions from/to the RNS/binary system and for performing the arithmetic operations in modulo $2^n \pm 1$ arithmetic.

The complexity of a modulo $2^n + 1$ arithmetic unit is determined by the representation chosen for the input operands. Several representations have therefore been considered: the normal weighted one, the diminished-one [19], the signed-LSB [15] and the SUT [32] representations are such examples. In the following, we consider the first two representations, since they are the most widely adopted. The normal representation uses the operands binary value. Therefore, it has the disadvantage that it requires $(n + 1)$ bits while it uses only the $2^n + 1$ combinations of them. In the diminished-one representation, each operand is represented by a value which is decreased by one compared to the value of its normal representation. As a result, only $n$ bits are used in the computation units, leading to more efficient modulo $2^n + 1$ arithmetic circuits. However, zero operands and results have to be treated separately. Several architectures for modulo $2^n + 1$ arithmetic circuits have been recently presented for both representations [1, 4, 10, 12, 15, 20, 30–32, 35–37, 39].

Operands in modulo $2^n - 1$ arithmetic require $n$ bits in order to be represented, thus the modulo $2^n - 1$ channel can be implemented to be as efficient as the modulo $2^n$ (conventional arithmetic) channel [1, 4–6, 11, 15, 16, 32, 39].

Subtraction is an operation very frequently met in a variety of applications. Examples in digital signal processing (DSP) include linear-phase FIR filters with antisymmetric impulse responses, such as digital Hilbert transformers and differentiators, reduced-complexity complementary and parallel FIR structures, and adaptive DSP algorithms, such as channel equalizers and echo cancellers [27, 38]. However, little work has been presented on the design of modulo $2^n \pm 1$ subtractors [5, 31, 32].

Since the majority of DSP algorithms require a significant number of addition and subtraction operations, two solutions are possible for implementing these algorithms over an RNS. They are, either to include both adder and subtractor circuits, or to include a single unit capable of performing either addition or subtraction depending on a mode signal. The first solution almost doubles the required hardware, whereas the second one does not allow additions and subtractions to be executed in parallel. Application examples include digital signal processors [9, 29] and reconfigurable datapaths [6, 8].

In this paper we deal with the problem of designing efficient modulo $2^n + 1$ subtractors and combined adders/subtractors in a unified way. We consider both the cases of normal and diminished-one operands' representation. We also consider, although straightforward, the design of modulo $2^n - 1$ subtractors and adders/subtractors just for the sake of completeness. Finally, we evaluate and compare the presented archi-

tectures against previous proposals and present area, delay and average power results based on both a unit gate model and CMOS VLSI implementations.

The rest of the paper is organized as follows. In the next section, novel architectures for designing modulo $2^n + 1$ subtractors and adders/subtractors are presented. In Sect. 3, we comment on designing modulo $2^n - 1$ subtractors and combined adders/subtractors. Evaluations and comparisons are given in Sect. 4. Finally, conclusions are drawn in the last section.

## 2 Modulo $2^n + 1$ Subtractors and Adders/Subtractors

In this section we present novel architectures for designing modulo $2^n + 1$ subtractors, as well as modulo $2^n + 1$ adders/subtractors. The first two and the last two subsections consider that the operands follow the normal and the diminished-one representation, respectively.

### 2.1 Modulo $2^n + 1$ Subtractors for the Normal Representation

Let $A = a_n \cdots a_0$ and $B = b_n \cdots b_0$ denote two $(n + 1)$-bit operands that follow the normal representation, with $0 \leq A, B < 2^n + 1$. Let also $|x|_m$ denote the residue of a $k$-bit operand $x$, when divided by $m$. The difference, $D$, of $A$ and $B$ taken modulo $2^n + 1$ can be computed as follows:

$$\begin{aligned} D = |A - B|_{2^n+1} &= \left|A + 2\left(2^n + 1\right) - B\right|_{2^n+1} \\ &= \left|A + \left(2^{n+1} - 1\right) - B + 3\right|_{2^n+1} = \left|A + \bar{B} + 3\right|_{2^n+1} \end{aligned} \tag{1}$$

where $\bar{B}$ denotes the one's complement of $B$. Relation (1) indicates that the modulo $2^n + 1$ difference of $A$ and $B$ is equivalent to the sum of $A$ and $\overline{B}$ taken modulo $2^n + 1$ as long as a correction term equal to 3 is also taken into account.

It has recently been shown [37] that the modulo $2^n + 1$ sum of two $(n + 1)$-bit operands $X$ and $Y$ in the normal representation can be carried out by an inverted end-around carry (IEAC) $n$-bit parallel adder, augmented by an inverted end-around-carry carry save adder (IEAC CSA). Specifically, the $n$ least significant bits of $X$ and $Y$, $X_{n-1:0}$ and $Y_{n-1:0}$, along with an $n$-bit correction term, $C$, that depends on the values of the most significant bits of $X$ and $Y$, $x_n$ and $y_n$, are added by an IEAC CSA which consists of $n$ full adders (FAs) and an inverter. It holds that $C = 1 \cdots 1(\overline{x_n \wedge y_n})(\overline{x_n \oplus y_n})$, where $\wedge$ denotes the logical AND and $\oplus$ denotes the logical exclusive-OR. The two $n$-bit outputs of the carry save adder are then driven to the IEAC adder that produces the $n$ least significant bits of the result. The most significant bit of the result is derived by detecting whether the two inputs of the IEAC adder are complementary or not. According to [37], the area overhead for computing the most significant bit of the result is negligible and the delay overhead is zero.

We can use the normal modulo $2^n + 1$ addition architecture of [37] for the normal modulo $2^n + 1$ subtraction as well, using as inputs the operands $A$ and $\overline{B}$ of (1). Furthermore, the constant value of 3 in (1) can be merged with the correction term $C$. This leads to a new correction term $C'$ that is $(n + 1)$-bit wide and its value is equal to

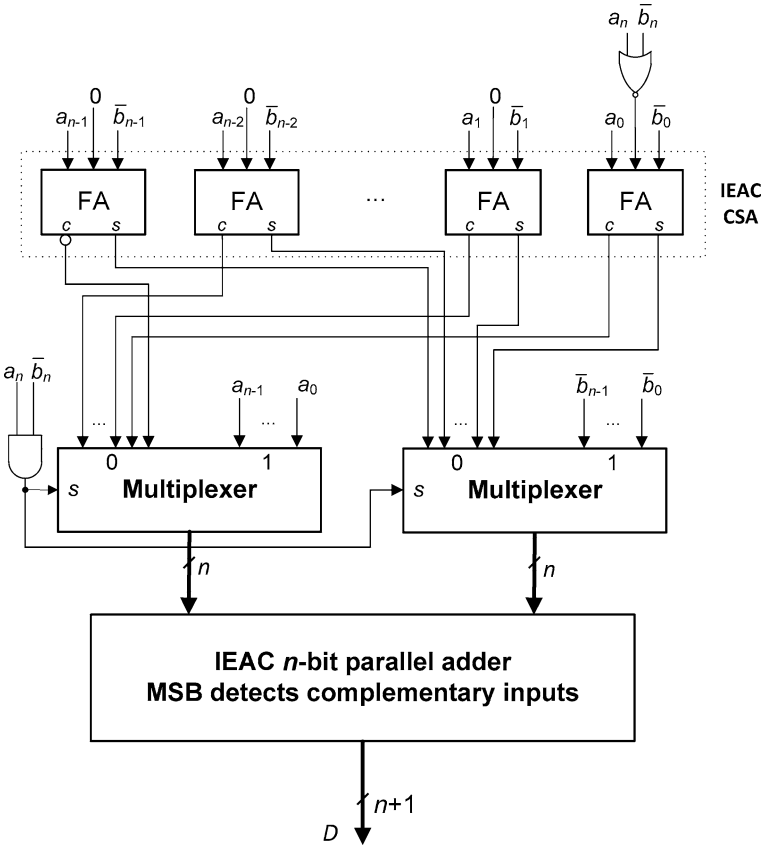


**Fig. 1** Modulo $2^n + 1$ subtractor for operands in the normal representation

$(a_n \wedge \bar{b}_n)0 \ldots 0(\bar{a}_n \wedge b_n)$. When the correction term $C'$ is equal to $2^n = |-1|_{2^n+1}$, the carry save addition is not required [37] and the inputs of the IEAC adder should be driven directly by the $n$ least significant bits of $A$ and $\bar{B}$. This is justified by the fact that an IEAC CSA produces at the output the sum of its input operands increased by one. Hence, an IEAC CSA addition of $A$, $B$, and $-1$ results in $A + B + (-1) + 1 = A + B$ and can be avoided.

According to the above, the architecture presented in Fig. 1 is derived for a modulo $2^n + 1$ subtractor. Two $n$-bit 2-to-1 multiplexers with a common select signal equal to $(a_n \wedge \bar{b}_n)$ are used between the IEAC CSA and the IEAC adder.

Considering that, in modulo $2^n + 1$ arithmetic, $a_n(b_n)$ and $a_i(b_i)$, $0 \leq i < n$, cannot be simultaneously at 1, several simplifications can be applied to the modulo subtractor circuit of Fig. 1:

- The $(n-1)$ leftmost FAs of the IEAC CSA can be simplified to half adders (HAs) since $C'$ consists of $(n-1)$ zeros. Furthermore, the sum outputs $a_i \oplus \overline{b_i}$, $0 < i < n$, of those HAs can be directly driven to the corresponding inputs of the IEAC adder, bypassing the multiplexer that is shown on the right. When $a_n \wedge \bar{b}_n = 0$, then the

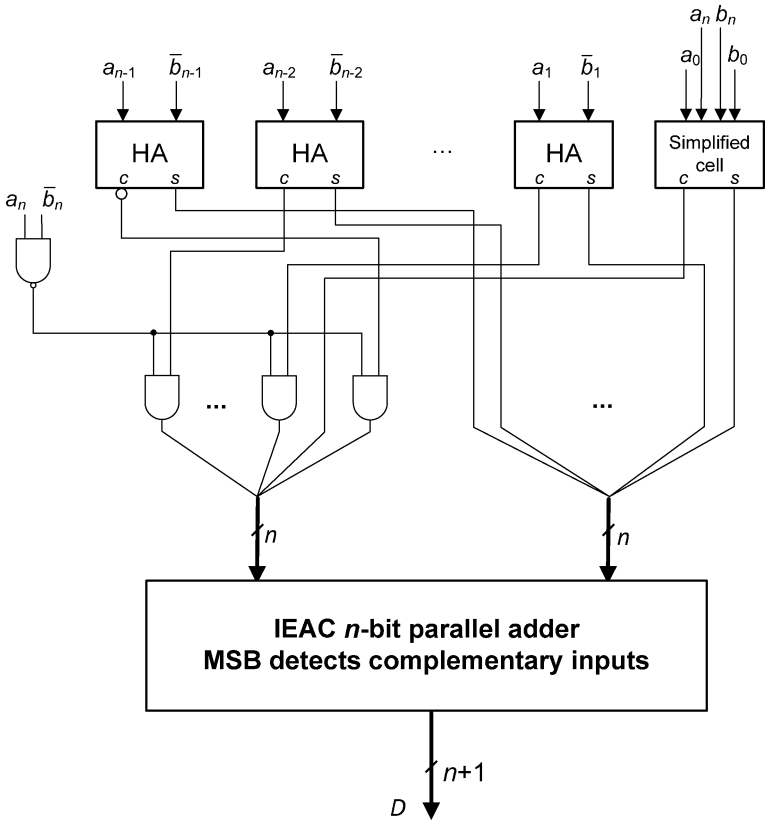


**Fig. 2** Proposed modulo $2^n + 1$ subtractor for operands following the normal representation

required values, that is $a_i \oplus \overline{b_i}$, are obviously driven to the IEAC adder. If, on the other hand, $a_n \wedge \overline{b}_n = 1$, then $a_n = 1$ and $a_i = 0, 0 \leq i < n$, and the required values are also driven to the diminished-one adder since $a_i \oplus \overline{b_i} = 0 \oplus \overline{b_i} = \overline{b_i}$.

- The right input of the left multiplexer of Fig. 1, instead of the $a_{n-1} \cdots a_0$, can be driven by the $0 \cdots 0$ value due to the fact that this input of the multiplexer is selected when $a_n = 1$. Hence, the leftmost multiplexer can be replaced by $n$ 2-input AND logic gates and an inverter.
- The rightmost FA along with the 2-input NOR logic gate and the corresponding 1-bit 2-to-1 multiplexers can also be simplified since $a_n(b_n)$ and $a_0(b_0)$ cannot be simultaneously at 1. The logic equations of the outputs of this simplified part of the circuit are: $s = (\overline{a_0 \oplus (b_0 \vee b_n)}) \vee (a_n \wedge \overline{b_0})$ and $c = (a_0 \wedge \overline{b_0}) \vee (\overline{a_n} \wedge b_n)$, where $\vee$ denotes the logical OR function.

The simplified circuit that is finally derived for the modulo $2^n + 1$ subtractor is given in Fig. 2.

*Example 1* Let us consider, as an example, a value of $n$ equal to 8 and that $A = 85_{10} = 001010101_2$ and $B = 12_{10} = 000001100_2$. Then $D = |A - B|_{257} =$

$|85 - 12|_{257} = 73_{10}$. According to (1) it holds that $D = |A + \overline{B} + 3|_{257} = |85 + 499 + 3|_{257} = |587|_{257} = 73_{10}$. Let $A_{n-1:0} = 01010101_2$, $\bar{B}_{n-1:0} = 11110011_2$ and $C'_{n-1:0} = 00000000_2$ denote the three inputs of the IEAC CSA. The 8-bit sum and carry outputs are then equal to $10100110_2 = 166_{10}$ and $10100011_2 = 163_{10}$, respectively. Since the most significant bit of the $C'$ term is equal to 0, these two values are driven to the IEAC adder. This adder increments the integer sum of its input vectors when the carry output of their integer addition is 0, and leaves it unchanged otherwise [39]. In our example, the IEAC adder produces $01001001_2$ and since its inputs are not complementary, the most significant bit of the result is equal to 0. Hence the output is $D = 001001001_2 = 73_{10}$.

### 2.2 Normal Modulo $2^n + 1$ Addition/Subtraction

The modulo $2^n + 1$ addition of [37] and the modulo $2^n + 1$ subtraction of the previous subsection can be combined into a single circuit that can perform addition or subtraction depending on the value of an input signal $M$. Addition (subtraction) is performed when $M = 0$ ($M = 1$). The combined adder/subtractor is based on an IEAC parallel adder and an IEAC CSA.

In case of addition, the IEAC CSA accepts as inputs the $A_{n-1:0}$, $B_{n-1:0}$ and $C$ vectors. The outputs of the IEAC CSA are driven directly to the IEAC adder. In case of subtraction, the IEAC CSA accepts as inputs the $A_{n-1:0}$ and $\bar{B}_{n-1:0}$ vectors and the $n$ least significant bits of the $C'$ vector. The outputs of the IEAC CSA are driven to two $n$-bit multiplexers with a common select signal equal to the most significant bit of the $C'$ vector ($a_n \wedge \overline{b_n}$) and the outputs of the multiplexers are driven to the IEAC adder. Both cases can be covered by the circuit given in Fig. 3. A series of 2-input XOR logic gates are used to invert the $n$ least significant bits of the $B$ input operand in case of subtraction. Furthermore, the two correction terms $C$ and $C'$are combined into an $(n+1)$-bit correction term $C'' = c''_n \cdots c''_0$, where $c''_n = M \wedge a_n \wedge \overline{b_n}$, $c''_{n-1} = \cdots = c''_2 = \overline{M}$, $c''_1 = \overline{M} \wedge (\overline{a_n \wedge b_n})$ and $c''_0 = \overline{M} \wedge (\overline{a_n \oplus b_n}) \vee M \wedge \overline{a_n} \wedge b_n$. Bit $c''_n$ is used as the select signal of the two multiplexers.

Similar simplifications to those presented in the previous subsection can also be performed on the two multiplexers of Fig. 3. Furthermore, the two rightmost FAs along with their driving logic can also be simplified since $a_n(b_n)$ and $a_i(b_i)$, $0 \leq i \leq 1$, cannot be simultaneously at 1. The above modifications result in a more efficient circuit, in terms of area and delay that is shown in Fig. 4. The simplified logic equations for the outputs of each of the two rightmost cells are:

$$
\begin{aligned}
s_0 &= (a_n \wedge b_n) \vee (a_0 \wedge b_n) \vee (a_0 \wedge b_0) \vee \left(\overline{M} \wedge a_n \wedge b_0\right) \vee \left(M \wedge a_n \wedge \overline{b_0}\right) \\
&\quad \vee \left(\overline{a_n \vee b_n \vee a_0 \vee b_0}\right), \\
c_0 &= \left(M \wedge \overline{a_n} \wedge b_n\right) \vee \left(\bar{b}_n \wedge a_0 \wedge \bar{b}_0\right) \vee \left(\overline{M} \wedge \overline{a_n} \wedge b_0\right), \\
s_1 &= (a_1 \wedge b_1) \vee \left(\overline{a_1} \wedge \overline{a_n} \wedge b_n\right) \vee \left(M \wedge a_n \wedge \overline{b_1}\right) \vee \left(\overline{a_1 \vee b_n \vee b_1}\right), \quad \text{and} \\
c_1 &= \left(a_1 \wedge \overline{b_1}\right) \vee \left(\overline{M} \wedge b_1\right).
\end{aligned}
$$

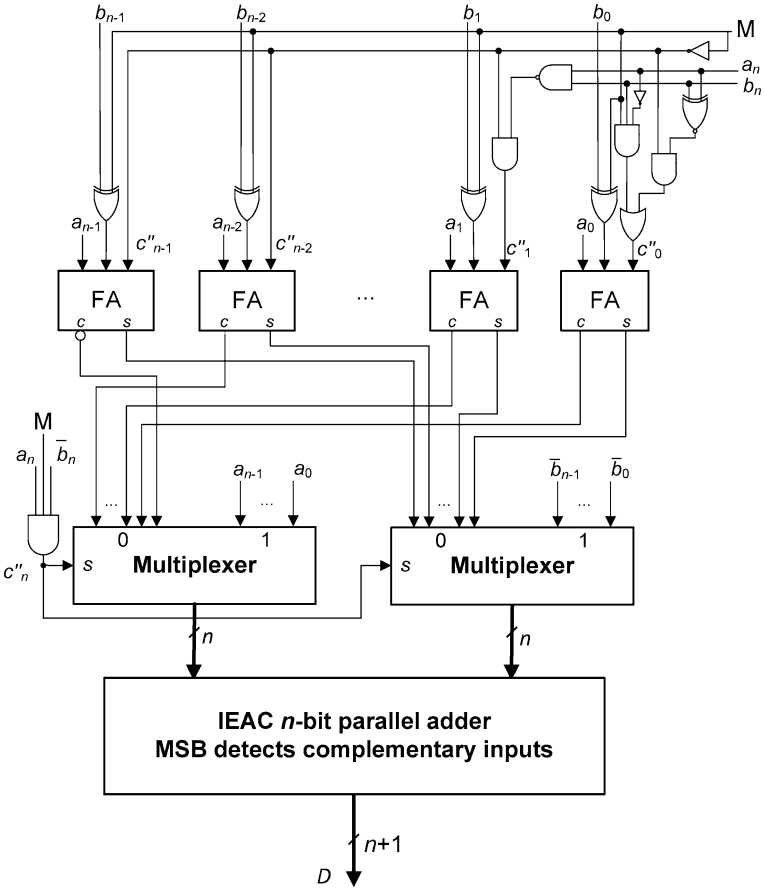


**Fig. 3** Modulo $2^n + 1$ adder/subtractor for operands in the normal representation

## 2.3 Modulo $2^n + 1$ Subtractors for the Diminished-one Representation

Let $A^* = a^*_{n-1} \cdots a^*_0$ and $B^* = b^*_{n-1} \cdots b^*_0$ denote the diminished-one representations of $A$ and $B$, respectively, with $0 < A, B < 2^n + 1$. Both $A^*$ and $B^*$ are $n$ bits wide, while $A^* = A - 1$ and $B^* = B - 1$. A diminished-one subtractor of $A$ and $B$ is a circuit that accepts $A^*$ and $B^*$ and produces the diminished-one representation $D^*$ of the difference $D = |A - B|_{2^n+1}$. It holds that

$$\begin{aligned} D^* &= |A - B - 1|_{2^n+1} = \left|\left(A^* + 1\right) - \left(B^* + 1\right) - 1\right|_{2^n+1} \\ &= \left|A^* + \left(2^n - 1\right) - B^* + 1\right|_{2^n+1} = \left|A^* + \bar{B}^* + 1\right|_{2^n+1} \end{aligned} \tag{2}$$

Ignoring zero operands and results, it is well known [39] that a modulo $2^n + 1$ adder for diminished-one operands is equivalent to an IEAC adder, which, when driven by two $n$-bit operands $X$ and $Y$, computes $|X + Y + 1|_{2^n+1}$. Hence, the modulo $2^n + 1$ subtraction for operands in the diminished-one representation indicated in (2) can be performed by an $n$-bit IEAC adder driven by $A^*$ and $\bar{B}^*$.

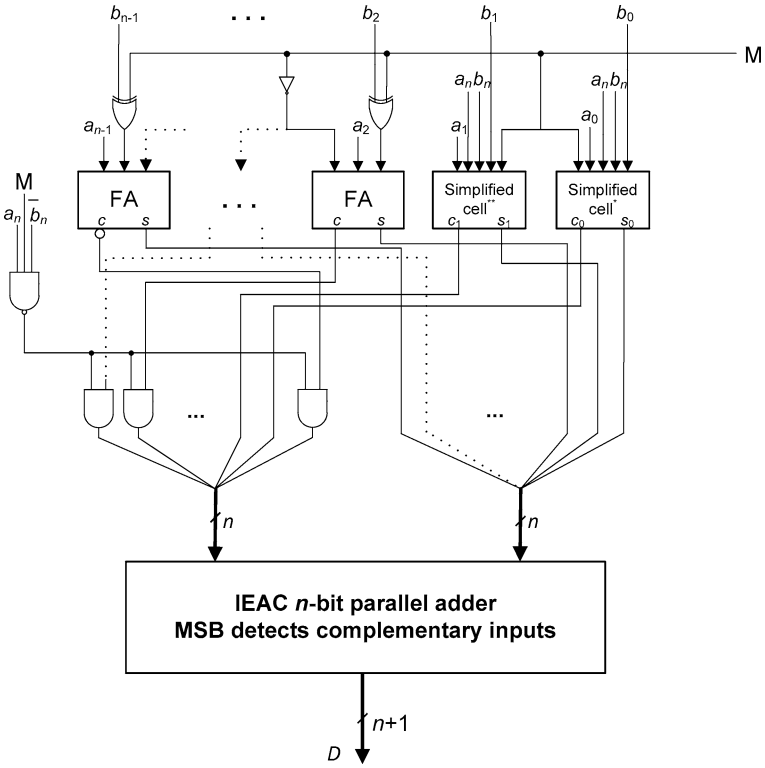


**Fig. 4** Proposed modulo $2^n + 1$ adder/subtractor for operands in the normal representation

*Example 2* Let us consider that $n = 8$, $A = 85_{10}$ and $B = 12_{10}$. Then $A^* = A - 1 = 01010100_2 = 84_{10}$, $B^* = B - 1 = 00001011_2 = 11_{10}$, and $D^* = |A - B - 1|_{257} = 72_{10}$. The values $A^*$ and $\bar{B}^* = 11110100_2$ are driven to an 8-bit IEAC adder whose result is equal to $01001000_2 = 72_{10}$.

In the following, we focus on the cases where $A$ or $B$ or the result are equal to zero. Arithmetic circuits that deal with operands in the diminished-one representation usually utilize an extra bit per input/output operand, along with the $n$ bits of its diminished-one representation for indicating the case of a zero value [10].

Let us denote as $A_z$ and $B_z$ the zero indication bits of $A$ and $B$ respectively, and as $D_z$ the zero indication bit of $D$. The values of $D_z$ and $D^*$ for the four different combinations of $A_z$ and $B_z$ are given in Table 1. The third line of Table 1 is justified as follows: When $A = 0$ and $B \neq 0$, then $D = |A - B|_{2^n+1} = |-B|_{2^n+1} \neq 0$. Hence, $D_z = 0$ and

$$\begin{aligned} D^* &= |-B-1|_{2^n+1} = \left|\left(2^n+1\right) - \left(B^*+1\right) - 1\right|_{2^n+1} \\ &= \left|\left(2^n-1\right) - B^*\right|_{2^n+1} = \left|\bar{B}^*\right|_{2^n+1} = \bar{B}^* \end{aligned} \tag{3}$$

**Table 1** Truth table for diminished-one modulo $2^n + 1$ subtraction

| $A_z$ | $B_z$ | $D$ | $D_z$ | $D^*$ |
|---|---|---|---|---|
| 0 | 0 | $\|A - B\|_{2^n+1}$ | ♦ | $\|A^* - B^* - 1\|_{2^n+1}$ |
| 0 | 1 | $\|A\|_{2^n+1}$ | 0 | $A^*$ |
| 1 | 0 | $\|-B\|_{2^n+1}$ | 0 | $\bar{B}^*$ |
| 1 | 1 | 0 | 1 | 0 |

♦ Depends on the values of $A^*$ and $B^*$

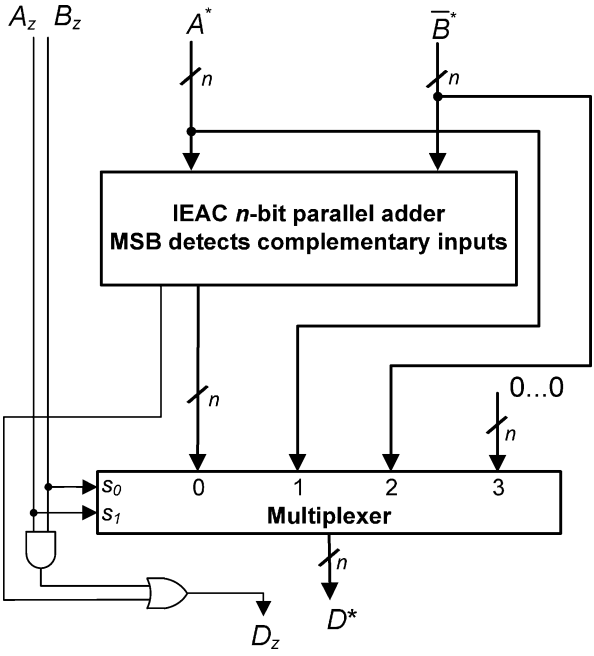


**Fig. 5** A modulo $2^n + 1$ subtractor for operands in the diminished-one representation with zero-handling

Figure 5 presents an architecture for a modulo $2^n + 1$ subtractor for operands that follow the diminished-one representation, which is capable of handling zero operands and results. It is based on an $n$-bit IEAC adder and an $n$-bit 4-to-1 multiplexer. $A_z$ and $B_z$ are used as the multiplexer select signals. The zero indication of the result is equal to 1 when: (a) $A = B = 0$, or (b) $A = B$ and $A, B \neq 0$. The first case can be detected by a 2-input AND gate whereas the second case can be detected by checking whether $A^*$ and $\bar{B}^*$ are bitwise complementary vectors, or equivalently, by utilizing the enhanced IEAC adder used in the previous subsection for the normal operands.

Unfortunately, the 4-to-1 multiplexer resides on the critical path of the circuit and therefore contributes to the delay of the modulo subtraction operation. A similar problem appears in the modulo $2^n + 1$ adder's case as well. To remove this additional delay, [10] presented an adder architecture that embeds the treatment of zero operands within the parallel prefix structure of the IEAC adder and cancels the need for the 4-to-1 multiplexer. Since the proposed modulo $2^n + 1$ subtractor for diminished-one operands is similarly built around an IEAC adder, we can also eliminate the 4-to-1 multiplexer by using as the adder the one presented in [10]. Then the resulting modulo subtractor circuit takes the form of the one in Fig. 6. The adder can use as inputs,

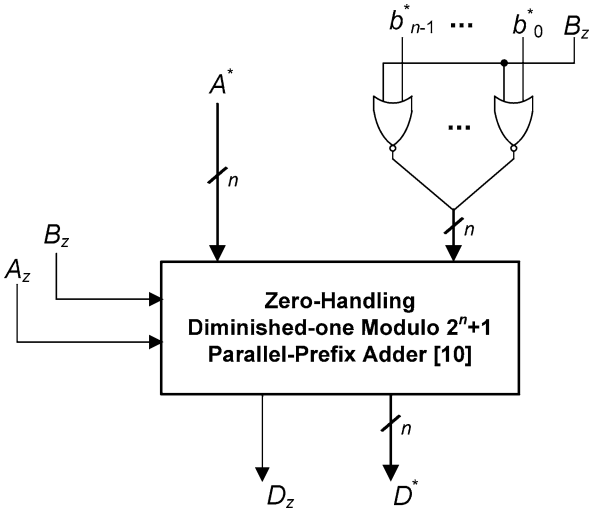


**Fig. 6** Proposed diminished-one modulo $2^n + 1$ subtractor with zero-handling capability

the $A^*$ and $\bar{B}^*$ $n$-bit operands along with the $A_z$ and $B_z$ zero indications. However, when $B_z = 1$ ($B = 0$), the second input of the adder should be driven with the all-zeros value and not with the all-ones value, in order to get the correct result. Hence, $n$ 2-input NOR gates have to be used. The first input of every NOR gate is connected to $B_z$, while the second input is connected to $b_i^*$, $0 \leq i < n$.

### 2.4 Diminished-one Modulo $2^n + 1$ Addition/Subtraction

A combined modulo $2^n + 1$ adder/subtractor for operands in the diminished-one representation can be straightforwardly derived. This uses an extra operation mode input $M$. Addition (subtraction) is selected by setting $M$ to 0 (1). The circuit is presented in Fig. 7. It utilizes the zero-handling diminished-one modulo $2^n + 1$ parallel prefix adder presented in [10]. The first $n$-bit input is connected to the $A^*$ operand whereas the second $n$-bit input is connected to the outputs of an $n$-bit 2-to-1 multiplexer with a select signal equal to $M$. When $M = 0$, $B^*$ is driven to the input of the adder, whereas when $M = 1$, $\overline{b_i \vee B_z}$, $0 \leq i < n$, are driven to the input of the adder as required for the subtraction operation according to the analysis given in Sect. 2.3.

## 3 Modulo $2^n - 1$ Subtractors and Adders/Subtractors

Although straightforward, for the sake of completeness, the design of modulo $2^n - 1$ subtractors and combined adders/subtractors is briefly presented in the following.

Let $A = a_{n-1} \cdots a_0$ and $B = b_{n-1} \cdots b_0$ denote two $n$-bit modulo $2^n - 1$ operands, such that $0 \leq A, B < 2^n - 1$. The difference, $D$, of $A$ and $B$ modulo $2^n - 1$ is equal to

$$D = |A - B|_{2^n-1} = \left|A + \left(2^n - 1\right) - B\right|_{2^n-1} = \left|A + \bar{B}\right|_{2^n-1} \tag{4}$$

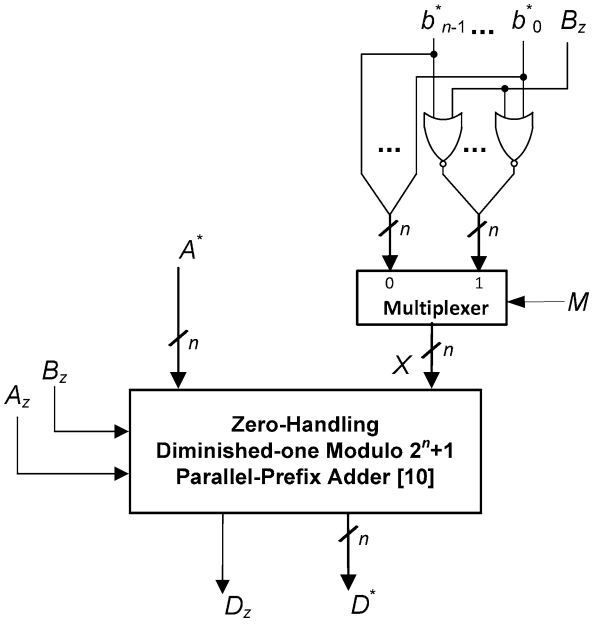


**Fig. 7** Proposed modulo $2^n + 1$ adder/subtractor for operands in the diminished-one representation with zero-handling capability

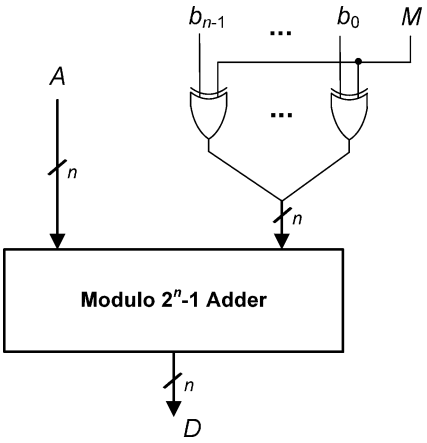


**Fig. 8** Modulo $2^n - 1$ adder/subtractor

where $\overline{B}$ denotes the one's complement of operand $B$. It is obvious that the difference $D$ is actually an addition of $A$ and $\overline{B}$ modulo $2^n - 1$, resulting that way in a straightforward hardware implementation based on a modulo $2^n - 1$ adder.

Furthermore, a combined modulo $2^n - 1$ adder/subtractor can be also easily derived by utilizing $n$ 2-input XOR logic gates in order to invert the bits of operand $B$ in case of subtraction ($M = 1$), as shown in Fig. 8.

**Table 2** Area and delay estimates according to the unit-gate model

| Architecture | Delay | Area |
|---|---|---|
| Subtractors | | |
| Normal modulo $2^n+1$ [31] | $2\log n+11$ | $3n\log(n(n+1))+3\log(n+1)+8n+31$ |
| Normal modulo $2^n+1$ (proposed) | $2\log n+7$ | $(9/2)n\log n+(9/2)n+12$ |
| Diminished-one modulo $2^n+1$ (proposed) | $2\log n+7$ | $3n\log n+4n+6$ |
| Modulo $2^n-1$ [5] | $3\log n+6$ | $3n\log n+5n+7$ |
| Modulo $2^n-1$ (modulo adder-based) | $2\log n+3$ | $3n\log n+4n$ |
| Adders/subtractors | | |
| Normal modulo $2^n+1$ (proposed) | $2\log n+9$ | $(9/2)n\log n+(21/2)n+25$ |
| Diminished-one modulo $2^n+1$ (proposed) | $2\log n+9$ | $3n\log n+7n+6$ |
| Modulo $2^n-1$ (modulo adder-based) | $2\log n+5$ | $3n\log n+6n$ |

## 4 Evaluation and Comparisons

In this section we compare the proposed circuits against previous proposals and we present experimental results.

Table 2 summarizes area and delay requirements, in equivalent gates, of all different architectures under comparison, assuming the unit gate model [34]. This model assumes that each monotonic two-input gate counts as one equivalent gate for both area and delay, while an XOR/XNOR gate counts as two equivalent gates for area and delay. We assume that all binary adders follow the Kogge–Stone [18] architecture while the IEAC adders and the modulo $2^n-1$ adders follow the architectures presented in [36] and [16], respectively. For the diminished-one modulo $2^n+1$ case, the parallel-prefix adder architecture with a carry increment stage, presented in [10], is considered.

Since we are not aware of any other work on combined modulo $2^n\pm1$ adders/subtractors, no comparison with other proposals is possible. However, it is obvious that the combined adders/subtractors have a slight area and delay overhead compared to the corresponding modulo adders or subtractors.

The proposed subtractors for operands in the normal representation consist of: (a) $(n-1)$ half adders for the IEAC CSA and a simplified combinational cell at the least significant bit position, (b) an enhanced IEAC adder for the final addition [37], and (c) $n$ 2-input gates for driving the appropriate inputs to the IEAC adder. We compare our proposal against the recently proposed subtractors for the normal representation [31]. The architecture of [31] was based on the following equation:

$$|A-B|_{2^n+1}=\begin{cases}|A-B|_{2^n+1} & \text{if } A-B\geq 0\\ |A-B+2^n+1|_{2^n+1} & \text{if } A-B<0\end{cases} \tag{5}$$

To implement (5), [31] first converts the input operands $A$ and $B$ from unsigned numbers to the signed ones. It then computes both terms of (5) in parallel, using one 2-input binary adder and one 3-input binary adder. Finally, one $(n+1)$-bit 2-to-1 multiplexer is used to select between the outputs of the two binary adders and derive

**Table 3** Unit-gate area and delay estimates of modulo $2^n + 1$ subtractors for operands in the normal representation

| $n$ | Area | | | Delay | | |
|---|---|---|---|---|---|---|
| | [31] (gate eqs.) | Proposed (gate eqs.) | Savings (%) | [31] (gate eqs.) | Proposed (gate eqs.) | Savings (%) |
| 4 | 122 | 66 | 45.8 | 15 | 11 | 26.7 |
| 8 | 253 | 156 | 38.2 | 17 | 13 | 23.5 |
| 16 | 559 | 372 | 33.5 | 19 | 15 | 21.1 |

the correct result. Some further logic is also required that generates the selection signal of the multiplexer.

Comparing the subtractors of [31] with the proposed ones, we can notice that the 2-input binary adder of [31] and the IEAC adder have similar area and delay requirements. The architecture of [31] further requires a 3-input binary adder and a $(n + 1)$-bit 2-to-1 multiplexer, whereas the proposed architecture, besides the IEAC adder, requires an IEAC CSA which mainly consists of half adders, and $n$ 2-input logic gates. Hence, we expect that the proposed subtractors will result in significantly less area and delay than those of [31].

We present in Table 3 the area and delay estimates, according to the unit-gate model, of the normal modulo $2^n + 1$ subtractors for three different values of $n$, that is, $n = 4, 8$ and 16, assuming the proposed architecture as well as the architecture of [31]. We have also described in HDL the corresponding circuits. After validating the correct operation of the HDL descriptions, each design was synthesized and mapped to a 90-nm power-characterized CMOS standard-cell library, assuming typical process parameters. Finally, the area and delay estimates were derived. To obtain the average dynamic power estimations, we followed a simulation-driven approach. We applied $2^{16}$ random input vectors at a 500-MHz frequency at each design netlist and measured the average power dissipation using a commercial power estimator. The same vectors were applied to the corresponding netlists of the architectures under comparison. The attained area, delay and power estimates for the normal modulo $2^n + 1$ subtraction circuits are presented in Table 4. The derived results indicate that the proposed modulo $2^n + 1$ subtractors offer significant savings in area and average power dissipation compared to the circuits of [31]. Reductions of up to 49 and 50% in the required implementation area and the average power consumed are reported in Table 4, while reductions in delay of up to 19% are also observed.

The proposed modulo $2^n + 1$ subtractors for operands in the diminished-one representation, for a specific value of $n$, consist of: (a) $n$ NOT logic gates for inverting the $B^*$ operand, and (b) an IEAC adder. If zero-handling is also required, then the proposed subtractors consist of: (a) $n$ 2-input NOR gates for controlling the $B^*$ operand, and (b) a diminished-one modulo $2^n + 1$ adder with embedded zero-handling capability.

To the best of our knowledge, there is no other architecture proposed for subtraction of operands that follows the diminished-one representation. It is however obvious that the proposed modulo $2^n + 1$ subtractors offer comparable area, delay

**Table 4** CMOS VLSI area, delay and power estimates of modulo $2^n + 1$ subtractors for operands in the normal representation

| $n$ | Area | | | Delay | | | Average power | | |
|---|---|---|---|---|---|---|---|---|---|
| | [31] ($\mu m^2$) | Proposed ($\mu m^2$) | Savings (%) | [31] (ns) | Proposed (ns) | Savings (%) | [31] (mW) | Proposed (mW) | Savings (%) |
| 4 | 2094 | 1064 | 49.2 | 0.37 | 0.30 | 18.9 | 0.90 | 0.45 | 50.0 |
| 8 | 3926 | 2499 | 36.3 | 0.45 | 0.37 | 17.8 | 1.56 | 0.98 | 37.2 |
| 16 | 8385 | 6101 | 27.2 | 0.52 | 0.44 | 15.4 | 3.24 | 2.39 | 26.2 |

**Table 5** Unit-gate area and delay estimates of modulo $2^n - 1$ subtractors

| $n$ | Area | | Delay | |
|---|---|---|---|---|
| | [5] (gate eqs.) | Modulo adder-based (gate eqs.) | [5] (gate eqs.) | Modulo adder-based (gate eqs.) |
| 4 | 51 | 40 | 12 | 7 |
| 8 | 119 | 104 | 15 | 9 |
| 16 | 279 | 256 | 18 | 11 |

**Table 6** CMOS VLSI area, delay and power estimates of modulo $2^n - 1$ subtractors

| $n$ | Area | | Delay | | Average power | |
|---|---|---|---|---|---|---|
| | [5] ($\mu m^2$) | Modulo adder-based ($\mu m^2$) | [5] (ns) | Modulo adder-based (ns) | [5] (mW) | Modulo adder-based (mW) |
| 4 | 932 | 596 | 0.36 | 0.19 | 0.49 | 0.29 |
| 8 | 1969 | 1584 | 0.39 | 0.24 | 1.10 | 0.77 |
| 16 | 4745 | 4197 | 0.48 | 0.30 | 2.49 | 1.83 |

and power characteristics with those of the corresponding adders, since their only overhead against the latter is few logic gates.

A modulo $2^n - 1$ subtractor can be designed using: (a) $n$ NOT logic gates for inverting the $B$ operand, and (b) a modulo $2^n - 1$ adder, such as the one presented in [16]. Another architecture was proposed in [5] for FPGA implementations. The modulo $2^n - 1$ subtractors of [5] are based on the following equation:

$$|A - B|_{2^n-1} = \begin{cases} K & \text{if } C_{\text{out}} = 1 \\ K - 1 & \text{if } C_{\text{out}} = 0 \end{cases} \tag{6}$$

where $K$ is the $n$-bit result and $C_{\text{out}}$ is the carry-out bit of the binary subtraction $A - B = A + \bar{B} + 1 = 2^n C_{\text{out}} + K$. The corresponding circuits are implemented by using a binary adder and a mux-based decrementer. Since a modulo $2^n - 1$ adder has similar area and delay requirements with those of a binary adder/subtractor in CMOS VLSI implementations, we expect the modulo $2^n - 1$ subtractors that are based on the modulo $2^n - 1$ adders to be more suitable for CMOS VLSI implementations

that those proposed in [5]. We present in Tables 5 and 6 the area, delay and average power dissipation estimates for both architectures, based on the unit-gate model closed forms of Table 2 and synthesized HDL descriptions. The results validate that, for CMOS VLSI implementations, the modulo adder-based subtractors of Sect. 3 are more efficient.

An alternative method for designing modulo $2^n \pm 1$ subtractors would be by utilizing redundant number systems. In [32], modulo $2^n \pm 1$ adders and subtractors are proposed that use the Stored-Unibit Trasfer (SUT) redundant number representation. This redundant number representation results in constant time addition and subtraction since the carry propagation is limited to a single digit position. However, both the inputs and outputs of the circuits of [32] follow the SUT representation. To this end, converters from/to binary/SUT representation have to be used. Unfortunately, these converters are inefficient in both terms of area and delay since they require carry propagation through all the digits [14] and therefore this representation is not commonly adopted.

## 5 Conclusions

Moduli choices of the $2^n \pm 1$ forms have received significant attention in building RNS-based systems. We have presented novel modulo $2^n + 1$ subtractor architectures, for operands that follow either the normal or the diminished-one representation. Experimental results have validated that the proposed modulo $2^n + 1$ subtractors for operands in the normal representation offer significantly less area and consume significantly less power than those previously reported [31], while also being faster. The proposed modulo $2^n + 1$ subtractors for operands in the diminished-one representation are capable of handling zero-operands and stem from adding few logic gates over the corresponding diminished-one modulo $2^n + 1$ adders. As a result, their area, delay and power characteristics are very close to those of the adders and therefore are of a high efficiency. Arguments on the design of modulo $2^n - 1$ subtractors have been also given. The modulo $2^n - 1$ adder-based subtractors were shown to be more efficient in CMOS VLSI implementations than those presented in [5]. Finally, modulo $2^n \pm 1$ combined adder/subtractor circuits have been introduced that are suitable for applications where the hardware overhead of having separate circuits for modulo addition and subtraction is intolerable.

**Acknowledgements** This work was supported by the Caratheodory Programme of the University of Patras (D.178).

## References

1. D. Adamidis, H.T. Vergos, RNS multiplication/sum-of-squares units. IET Comput. Digit. Tech. **1**(1), 38–48 (2007)
2. P.V. Ananda Mohan, *Residue Number Systems: Algorithms and Architectures* (Kluwer, Norwell, 2002)
3. J.C. Bajard, L. Imbert, A full RNS implementation of RSA. IEEE Trans. Comput. **53**(6), 769–774 (2004)

4. D. Bakalis, H.T. Vergos, Shifter circuits for $\{2^n+1, 2^n, 2^n-1\}$ RNS. Electron. Lett. **45**(1), 27–29 (2009)
5. S. Bi, W. Gross, W. Wang, A. Al-Khalili, M.N.S. Swamy, An area-reduced scheme for modulo $2^n-1$ addition/subtraction, in *Proc. International Workshop System-on-Chip for Real-Time Applications* (2005)
6. N. Burgess, The flagged prefix adder and its applications in integer arithmetic. J. VLSI Signal Process. **31**(3), 263–271 (2002)
7. G. Cardarilli, A. Nannarelli, M. Re, Reducing power dissipation in FIR filters using the residue number system, in *Proc. IEEE Midwest Symposium on Circuits and Systems* (2000), pp. 320–323
8. G. Cardarilli, A. Re, A. Nannarelli, M. Re, Residue number system reconfigurable datapath, in *Proc. IEEE International Symposium on Circuits and Systems* (2002), pp. 756–759
9. R. Chaves, L. Sousa, RDSP: a RISC DSP based on residue number system, in *Proc: Euromicro Symposium on Digital System Design* (2003), pp. 128–135
10. C. Efstathiou, H.T. Vergos, D. Nikolos, Handling zero in diminished-one modulo $2^n+1$ adders. Int. J. Electron. **90**(2), 133–144 (2003)
11. C. Efstathiou, H.T. Vergos, D. Nikolos, Modified booth modulo $2^n-1$ multipliers. IEEE Trans. Comput. **53**(3), 370–374 (2004)
12. C. Efstathiou, H.T. Vergos, G. Dimitrakopoulos, D. Nikolos, Efficient diminished-1 modulo $2^n+1$ multipliers. IEEE Trans. Comput. **54**(4), 491–496 (2005)
13. P.G. Fernandez, A. Lloris, RNS-based implementation of $8\times 8$ point 2D-DCT over field-programmable devices. Electron. Lett. **39**(1), 21–23 (2003)
14. G. Jaberipur, B. Parhami, Stored-transfer representations with weighted digit-set encodings for ultrahigh-speed arithmetic. IET Circuits Dev. Syst. **1**(1), 102–110 (2007)
15. G. Jaberipur, B. Parhami, Unified approach to the design of modulo-$(2^n\pm 1)$ adders based on signed-lsb representation of residues, in *Proc. 19th IEEE Symposium on Computer Arithmetic* (2009), pp. 57–64
16. L. Kalampoukas, D. Nikolos, C. Efstathiou, H.T. Vergos, J. Kalamatianos, High-speed parallel prefix modulo $2^n-1$ adders. IEEE Trans. Comput. **49**(7), 673–680 (2000)
17. T. Keller, T.H. Liew, L. Hanzo, Adaptive redundant residue number system coded multicarrier modulation. IEEE J. Sel. Areas Commun. **18**(11), 2292–2301 (2000)
18. P.M. Kogge, H.S. Stone, A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE Trans. Comput. **C-22**(8), 786–793 (1973)
19. L.M. Leibowitz, A simplified binary arithmetic for the Fermat number transform. IEEE Trans. Acoust. Speech Signal Process. **24**(5), 356–359 (1976)
20. S.H. Lin, M.H. Sheu, VLSI design of diminished-one modulo $2^n+1$ adder using circular carry selection. IEEE Trans. Circuits Syst. II. **55**(9), 897–901 (2008)
21. Y. Liu, E. Lai, Moduli set selection and cost estimation for RNS-based FIR filter and filter bank design. Des. Autom. Embed. Syst. **9**(2), 123–139 (2004)
22. A.S. Madhukumar, F. Chin, Enhanced architecture for residue number system-based CDMA for high-rate data transmission. IEEE Trans. Wirel. Commun. **3**(5), 1363–1368 (2004)
23. F. Marino, E. Stella, A. Branca, N. Veneziani, A. Distante, Specialized hardware for real-time navigation. Real-Time Imaging **7**(1), 97–108 (2001)
24. U. Meyer-Baese, A. Garcia, F. Taylor, Implementation of a communications channelizer using FPGAs and RNS arithmetic. J. VLSI Signal Process. **28**(1), 115–128 (2001)
25. A. Omondi, B. Premkumar, *Residue Number Systems: Theory and Implementation* (Imperial College Press, London, 2007)
26. M. Panella, G. Martinelli, An RNS architecture for quasi-chaotic oscillators. J. VLSI Signal Process. **33**(1), 199–220 (2003)
27. K. Parhi, *VLSI Digital Signal Processing Systems* (Wiley, New York, 1999)
28. J. Ramirez, U. Meyer-Baese, High performance, reduced complexity programmable RNS-FPL merged FIR filters. Electron. Lett. **38**(4), 199–200 (2002)
29. J. Ramirez, A. Garcia, S. Lopez-Buedo, A. Lloris, RNS-enabled digital signal processor design. Electron. Lett. **38**(6), 266–268 (2002)
30. L. Sousa, R. Chaves, A universal architecture for designing efficient modulo $2^n+1$ multipliers. IEEE Trans. Circuits Syst. I. **52**(6), 1166–1178 (2005)
31. S. Timarchi, K. Navi, M. Hosseinzade, New design of RNS subtractor for modulo $2^n+1$, in *Proc. International Conference on Information and Communication Technology* (2006), pp. 2803–2808
32. S. Timarchi, K. Navi, Arithmetic circuits of redundant SUT-RNS. IEEE Trans. Instrum. Meas. **58**(9), 2959–2968 (2009)

33. T. Toivonen, J. Heikkila, Video filtering with Fermat number theoretic transforms using residue number system. IEEE Trans. Circuits Syst. Video Technol. **16**(1), 92–101 (2006)
34. A. Tyagi, A reduced-area scheme for carry-select adders. IEEE Trans. Comput. **42**(10), 1163–1170 (1993)
35. H.T. Vergos, C. Efstathiou, Design of efficient modulo $2^n + 1$ multipliers. IET Comput. Digit. Tech. **1**(1), 49–57 (2007)
36. H.T. Vergos, C. Efstathiou, D. Nikolos, Diminished-one modulo $2^n + 1$ adder design. IEEE Trans. Comput. **51**(12), 1389–1399 (2002)
37. H.T. Vergos, D. Bakalis, C. Efstathiou, Fast modulo $2^n + 1$ multi-operand adders and residue generators. Integr. VLSI J. **43**(1), 42–48 (2010)
38. L. Wanhammar, *DSP Integrated Circuits* (Academic Press, Reading, 1999)
39. R. Zimmermann, Efficient VLSI implementation of modulo $2^n \pm 1$ addition and multiplication, in *Proc. Symposium on Computer Arithmetic* (1999), pp. 158–167