On-line multiplication in real and complex base

Christiane Frougny
LIAFA, CNRS UMR 7089
2 place Jussieu, 75251 Paris Cedex 05, France
and Université Paris 8
Christiane.Frougny@liafa.jussieu.fr

Athasit Surarerks
Computer Engineering Department
Faculty of Engineering
Chulalongkorn University
Phayathai road, Bangkok 10330, Thailand
athasit@cp.eng.chula.ac.th

Abstract

Multiplication of two numbers represented in base β is shown to be computable by an on-line algorithm when β is a negative integer, a positive non-integer real number, or a complex number of the form $i\sqrt{r}$, where r is a positive integer.

1 Introduction

On-line arithmetic, introduced in [24], is a mode of computation where operands and results flow through arithmetic units in a digit serial manner, starting with the most significant digit. To generate the first digit of the result, the δ first digits of the operands are required. The integer δ is called the *delay* of the algorithm. This technique allows the pipelining of different operations such as addition, multiplication and division. It is also appropriate for the processing of real numbers having infinite expansions : it is well known that when multiplying two real numbers, only the left part of the result is significant. On-line arithmetic is used for special circuits such as in signal processing, and for very long precision arithmetic. One of the interests of on-line computable functions is that they are continuous for the usual topology on the set of infinite sequences on a finite digit set.

To be able to perform on-line computation, it is necessary to use a redundant number system, where a number may have more than one representation (see [24, 5]). An example of such a system is the so-called signed-digit number system. It is composed of an integer base $\beta \geq 2$ and a signed-digit set of the form $\{-a, \dots, a\}$, with $\beta/2 \leq a \leq \beta - 1$. In this sytem addition can be performed in constant time in parallel [1, 4]. On-line multiplication is also feasible [24, 5]. Parallel addition is used internally in the multiplication algorithm.

On-line algorithms for addition, subtraction, multiplication, division and square-root in integer base are well studied, see [2, 6, 19].

In this paper we study the multiplication in base β when β is a negative integer, or a non-integer real number, or a complex number of the form $i\sqrt{r}$.

It is known that any real number can be represented in negative integer base without a sign [12, 13, 15], and that, with a signed-digit set, addition is computable in constant time in parallel, and is computable by an on-line finite state automaton [7]. We show that the on-line multiplication used in the signed-digit number system can be applied in negative base. Negative base is related to some complex number systems, see below.

When the base β is a real number > 1, by the greedy algorithm of Rényi [23], one can compute a representation in base β of any real number belonging to the interval $[0, 1]$, called its β-*expansion*, and where the digits are elements of the canonical digit set $A_\beta = \{0, \dots, \lfloor\beta\rfloor\}$ if β is not an integer, or of $A_\beta = \{0, \dots, \beta - 1\}$ if β is an integer. In such a representation, when β is not an integer, not all the patterns of digits are allowed (see [21] for instance). For instance in base $\beta = (1 + \sqrt{5})/2$ the golden ratio, the β-expansion of the number $x = 3 - \sqrt{5}$ is $1001000\ldots$. The pattern 11 is forbidden. Different β-representations of x are $0111000\ldots$, or $100(01)(01)(01)\ldots$ for instance. This system is thus naturally redundant.

Representation of numbers in non-integer base is encountered for instance in coding theory, see [10], and in the modelization of quasicrystals, see [3].

In [8] we have studied the problem of the conversion in a real base β from a digit set $D = \{0, \dots, d\}$ with $d \geq \lfloor\beta\rfloor$ to the canonical digit set A_β, without changing the numerical value. Addition and multiplication by a fixed positive integer are particular cases of digit set conversion. We have proved that the digit set conversion is on-line computable.

Moreover, if the base β is a Pisot number[1] this conversion is realizable by an on-line finite automaton. This means that in the Pisot case, digit set conversion needs only a finite storage memory, independent of the size of the data. However, addition is not computable in constant time in parallel.

In this work, we prove that multiplication of real numbers represented in real base β is on-line computable with a certain delay explicitly computed.

Our algorithm can be applied to the particular case that β is an integer. With digit set $\{0, \dots, \beta\}$, this is the Carry-Save representation. The delay of our on-line multiplication algorithm in the Carry-Save representation is greater than the delay of on-line multiplication in the signed-digit representation, but the Carry-Save representation takes less memory.

We then consider the Knuth complex number system, which is composed of the base $\beta = i\sqrt{r}$, with $r \geq 2$ an integer, and canonical digit set $\{0, \dots, r-1\}$. Every complex number has a representation in this system [12]. This allows a unified treatment of the real and imaginary parts of a complex number. We show that in the Knuth number system with a signed-digit set $\{-a, \dots, a\}$, with $r/2 \leq a \leq r-1$, multiplication is on-line computable. It is known that addition is computable in constant time in parallel, and is computable by an on-line finite state automaton [7]. The case $\beta = 2i$ together with the signed-digit set $\{-2, \dots, 2\}$, has been considered in [17, 16] with practical applications.

Other complex numeration systems have been considered. For instance the Penney complex number system consists of the complex base $\beta = -1 + i$ and the canonical digit set $\{0, 1\}$. Every complex number is representable in this system, and the representation is unique and finite for the Gaussian integers [22]. It is shown in [25] that in base $-1+i$ with signed-digit set $\{-1, 0, 1\}$ multiplication is on-line computable. The techniques are different from the ones used in the Knuth number system.

2 Preliminaries

2.1 On-line computability

Let A and B be two finite digit sets, and denote by $A^{\mathbb{N}}$ the set of infinite sequences of elements of A. Let

$$\begin{aligned} \varphi : A^{\mathbb{N}} &\rightarrow B^{\mathbb{N}} \\ (a_j)_{j\geq 1} &\mapsto (b_j)_{j\geq 1} \end{aligned}$$

The function φ is said to be *on-line computable with delay* δ if there exists a natural number δ such that, for each $j \geq 1$ there exists a function $\Phi_j : A^{j+\delta} \rightarrow B$ such that $b_j = \Phi_j(a_1 \cdots a_{j+\delta})$, where $A^{j+\delta}$ denotes the set of sequences of length $j+\delta$ of elements of A. This definition extends readily to functions of several variables.

[1]A Pisot number is an algebraic integer such that its algebraic conjugates are strictly less than 1 in modulus. The golden ratio and the natural integers are Pisot numbers.

It is well known that some functions are not on-line computable, like addition in the binary system with canonical digit set $\{0, 1\}$. Addition is considered as a conversion χ from $\{0, 1, 2\}$ to $\{0, 1\}$. Denote by v^ω the infinite concatenation $vvv\ldots$, and by v^n the word v concatenated n times. Since $\chi(01^n20^\omega) = 10^\omega$ and $\chi(01^n0^\omega) = 01^n0^\omega$ for any $n \geq 1$, one sees that the most significant digit of the result depends on the least significant digits of the input.

Recall that a distance ρ can be defined on $A^{\mathbb{N}}$ as follows: let $v = (v_j)_{j\geq 1}$ and $w = (w_j)_{j\geq 1}$ be in $A^{\mathbb{N}}$, $\rho(v, w) = 2^{-r}$ where $r = \min\{j \mid v_j \neq w_j\}$ if $v \neq w$, $\rho(v, w) = 0$ otherwise. The set $A^{\mathbb{N}}$ is then a compact metric space. This topology is equivalent to the product topology. Then any function from $A^{\mathbb{N}}$ to $B^{\mathbb{N}}$ which is on-line computable with delay δ is 2^δ-Lipschitz, and is thus uniformly continuous [8].

Let D be a digit set. We say that *multiplication is on-line computable with delay* δ *in base* β *on the digit set* D if there exists a function

$$\begin{aligned} \mu : D^{\mathbb{N}} \times D^{\mathbb{N}} &\rightarrow D^{\mathbb{N}} \\ ((x_j)_{j\geq 1}, (y_j)_{j\geq 1}) &\mapsto (p_j)_{j\geq 1} \end{aligned}$$

such that

$$\sum_{j\geq 1} p_j \beta^{-j} = \sum_{j\geq 1} x_j \beta^{-j} \times \sum_{j\geq 1} y_j \beta^{-j}$$

which is on-line computable with delay δ. Note that *apriori*, because of redundancy, the result of such a process is not unique, but the algorithms we shall consider later on are deterministic, and thus compute a function.

In the following, we will make the assumption that the operands begin with a run of δ zeroes. This allows to ignore the delay inside the computation.

2.2 Number representation

A survey on numeration systems can be found in [14, Chapter 7]. Let D be a finite digit set of real or complex digits and let β be a real or complex number such that $|\beta| > 1$. A β*-representation* of a real or complex number x with digits in D is a finite or a right infinite sequence $(x_j)_{j\leq n}$ with $x_j \in D$ such that $x = \sum_{j=n}^{-\infty} x_j \beta^j$. It is denoted by

$$(x_n \cdots x_0 \cdot x_{-1} x_{-2} \cdots)_\beta.$$

In this paper we consider only representations of the form $(x_j)_{j\geq 1} \in D^{\mathbb{N}}$ representing a number x equal to $\sum_{j\geq 1} x_j \beta^{-j}$. When a representation ends with infinitely many zeroes, it is said to be *finite*, and the zeroes are usually omitted.

2.3 Signed-digit number system

The base is a positive integer $\beta > 1$. With a signed-digit set of the form $S = \{-a, \dots, a\}$, $\beta/2 \le a \le \beta - 1$ the representation is redundant and addition can be performed in constant time in parallel, and is computable by an on-line finite automaton, [1, 4, 19].

2.4 Negative base numeration systems

Let the base be a negative integer $\beta < -1$. It is well known (see [12, 13, 15]) that any real number can be represented without a sign in base β with digits from the canonical digit set $A = \{0, \dots, |\beta| - 1\}$. With a signed-digit set of the form $T = \{-a, \dots, a\}$, with $|\beta|/2 \le a \le |\beta| - 1$ the representation is redundant and addition can be performed in constant time in parallel, and is computable by an on-line finite automaton [7].

2.5 Representations in real base

Let β be a real number > 1, generally not an integer. Any real number $x \in [0, 1]$ can be represented in base β by the following greedy algorithm, [23] : denote by $\lfloor . \rfloor$ and by $\{.\}$ the integral part and the fractional part of a number. Let $r_0 = x$ and for $j \ge 1$ let $x_j = \lfloor \beta r_{j-1} \rfloor$ and $r_j = \{\beta r_{j-1}\}$. Thus $x = \sum_{j \ge 1} x_j \beta^{-j}$, where the digits x_j are elements of the *canonical* digit set $A_\beta = \{0, \dots, \lfloor \beta \rfloor\}$ if $\beta \notin \mathbb{N}$, $A_\beta = \{0, \dots, \beta - 1\}$ otherwise. The sequence $(x_j)_{j \ge 1}$ of $A_\beta^{\mathbb{N}}$ is called the β-*expansion* of x. When β is not an integer, a number x may have several different β-representations on A_β: this system is naturally redundant. The β-expansion obtained by the greedy algorithm is the greatest one in the lexicographic order.

It is shown in [8] that addition in real base is on-line computable. When the base is a Pisot number, addition is computable by an on-line finite automaton.

2.6 Knuth number system

Here the base is a complex number of the form $\beta = i\sqrt{r}$, where r is an integer ≥ 2. Any complex number is representable in base β with digits in the canonical digit set $A = \{0, \dots, r - 1\}$ (see [12, 11, 9]). If r is a square then every Gaussian integer has a unique finite representation of the form $a_k \cdots a_0 \cdot a_{-1}$, $a_i \in A$.

Since $\beta^2 = -r$, we have

$$z = \sum_{j \ge 1} a_j \beta^{-j} = \sum_{k \ge 1} a_{2k} (-r)^{-k} + i\sqrt{r} \sum_{k \ge 0} a_{2k+1} (-r)^{-k-1}.$$

Thus,

$$\Re(z) = x = \sum_{k \ge 1} a_{2k} (-r)^{-k}$$

and

$$\Im(z) = y = \sqrt{r} \sum_{k \ge 0} a_{2k+1} (-r)^{-k-1}.$$

So the β-representation of z can be obtained by intertwinning the $(-r)$-representation of x and the $(-r)$-representation of $y/\sqrt{r}$.

Most studied cases are $\beta = 2i$ and $A = \{0, \dots, 3\}$, strongly related to base -4, and $\beta = i\sqrt{2}$ and $A = \{0, 1\}$ ([12, 13, 20, 17, 16]).

With a signed-digit set of the form $R = \{-a, \dots, a\}$, $r/2 \le a \le r - 1$, addition is computable in constant time in parallel, and is computable by an on-line finite state automaton [7].

3 On-line multiplication algorithm in integer base

3.1 Classical on-line multiplication algorithm

First we recall the classical algorithm for on-line multiplication in the signed-digit number system, see [24, 5]. We give our own presentation.

THEOREM **1** *Multiplication of two numbers represented in integer base $\beta > 1$ with digits in $S = \{-a, \dots, a\}$, $\beta/2 \le a \le \beta - 1$, is computable by an on-line algorithm with delay δ, where δ is the smallest positive integer such that*

$$\frac{\beta}{2} + \frac{2a^2}{\beta^\delta(\beta - 1)} \le a + \frac{1}{2}. \tag{1}$$

Proof. Denote by X_j the partial sum $\sum_{1 \le i \le j} x_i \beta^{-i}$ (and respectively Y_j and P_j), and denote by round(z) the closest integer to z

Classical on-line multiplication algorithm M_{SD}**.**
Input: two sequences $x = (x_j)_{j \ge 1}$ *and* $y = (y_j)_{j \ge 1}$ *of* $S^{\mathbb{N}}$ *such that* $x_1 = \dots = x_\delta = 0$ *and* $y_1 = \dots = y_\delta = 0$.
Output: a sequence $p = (p_j)_{j \ge 1}$ *of* $S^{\mathbb{N}}$ *such that* $\sum_{j \ge 1} p_j \beta^{-j} = \sum_{j \ge 1} x_j \beta^{-j} \times \sum_{j \ge 1} y_j \beta^{-j}$.

```
begin
    p_1 ← 0, ..., p_δ ← 0
    W_δ ← 0
    j ← δ + 1
    while j ≥ δ + 1 do
        W_j ← β(W_{j-1} − p_{j-1}) + y_j X_j + x_j Y_{j-1}
        p_j ← round(W_j)
        j ← j + 1
end
```

First let us prove by induction that for any $j \ge \delta$

$$W_j \beta^{-j} = X_j Y_j - P_{j-1}.$$

By Line 5 of the algorithm,

$$W_j\beta^{-j} = \beta^{-j+1}(W_{j-1} - p_{j-1}) + \beta^{-j}(y_jX_j + x_jY_{j-1}).$$

By induction hypothesis, $W_j\beta^{-j} = X_{j-1}Y_{j-1} - P_{j-2} - \beta^{-j+1}p_{j-1} + \beta^{-j}(y_jX_j + x_jY_{j-1})$, and the result follows from the fact that $X_j = X_{j-1} + x_j\beta^{-j}$, and the similar relations for Y_j and P_j.

Thus at step $n \geq \delta$, $X_nY_n = \beta^{-n}W_n + P_{n-1} = \beta^{-n}(W_n - p_n) + P_n$. Since $|W_n - p_n| \leq \frac{1}{2}$,

$$|X_nY_n - P_n| \leq \frac{\beta^{-n}}{2}$$

and the algorithm is convergent. The sequence $p_1 \cdots p_n$ is a β-representation of the most significant half of the product X_nY_n.

Now it remains to prove that the digits p_j's computed in Line 6 of the algorithm are in the digit set S. It is enough to show that $|W_j| \leq a + \frac{1}{2}$.
From Line 5 and the fact that $|X_j|$ and $|Y_{j-1}|$ are less than $\frac{a}{\beta^\delta(\beta-1)}$ follows that

$$|W_j| < \frac{\beta}{2} + \frac{2a^2}{\beta^\delta(\beta-1)} \leq a + \frac{1}{2}$$

by (1). ■

Note that additions and multiplications by a digit in Line 5 are performed in constant time in parallel.

COROLLARY **1** *The delay δ for Algorithm M_{SD} takes the following values. If $\beta = 2$ and $a = 1$, $\delta = 2$. If $\beta = 3$ and $a = 2$, $\delta = 2$. If $\beta = 2a \geq 4$ then $\delta = 2$. If $\beta \geq 4$ and if $a \geq \lfloor \beta/2 \rfloor + 1$, $\delta = 1$.*

3.2 On-line multiplication in negative base

Now we consider the case where the base is a negative integer $\beta < -1$ and the digit set is $T = \{-a, \ldots, a\}$, $|\beta|/2 \leq a \leq |\beta| - 1$.

PROPOSITION **1** *Multiplication of two numbers represented in negative base $\beta < -1$ and digit set $T = \{-a, \ldots, a\}$, $|\beta|/2 \leq a \leq |\beta| - 1$, is computable by the classical on-line algorithm M_{SD} with delay δ, where δ is the smallest positive integer such that*

$$\frac{|\beta|}{2} + \frac{2a^2}{|\beta|^\delta(|\beta|-1)} \leq a + \frac{1}{2}. \qquad (2)$$

Proof. At step $n \geq \delta$ of the algorithm we get that $X_nY_n - P_n = \beta^{-n}(W_n - p_n)$. Since $|W_n - p_n| \leq \frac{1}{2}$,

$$|X_nY_n - P_n| \leq \frac{|\beta|^{-n}}{2}.$$

To show that p_j is in T, we have to show that $|W_j| \leq a + \frac{1}{2}$. From Line 5 and the fact that $|X_j|$ and $|Y_{j-1}|$ are less than $\frac{a}{|\beta|^\delta(|\beta|-1)}$ follows that

$$|W_j| < \frac{|\beta|}{2} + \frac{2a^2}{|\beta|^\delta(|\beta|-1)} \leq a + \frac{1}{2}$$

by (2). ■

4 On-line multiplication in real base

Let $D = \{0, \ldots, d\}$ be a digit set containing A_β, that is, $d \geq \lfloor \beta \rfloor$.

THEOREM **2** *Multiplication of two numbers represented in base β with digits in D is computable by an on-line algorithm with delay δ, where δ is the smallest positive integer such that*

$$\beta + \frac{2d^2}{\beta^\delta(\beta-1)} \leq d + 1. \qquad (3)$$

Proof. Clearly a number δ satisfying (3) exists, because $d \geq \lfloor \beta \rfloor$.

Real base on-line multiplication algorithm $M_{\mathbb{R}}$.
Input: two sequences $x = (x_j)_{j\geq1}$ and $y = (y_j)_{j\geq1}$ of $D^{\mathbb{N}}$ such that $x_1 = \cdots = x_\delta = 0$ and $y_1 = \cdots = y_\delta = 0$ [2].
Output: a sequence $p = (p_j)_{j\geq1}$ of $D^{\mathbb{N}}$ such that $\sum_{j\geq1} p_j\beta^{-j} = \sum_{j\geq1} x_j\beta^{-j} \times \sum_{j\geq1} y_j\beta^{-j}$.

begin
1. $p_1 \leftarrow 0, \ldots, p_\delta \leftarrow 0$
2. $W_\delta \leftarrow 0$
3. $j \leftarrow \delta + 1$
4. **while** $j \geq \delta + 1$ **do**
5. $\quad W_j \leftarrow \beta(W_{j-1} - p_{j-1}) + y_jX_j + x_jY_{j-1}$
6. $\quad p_j \leftarrow \lfloor W_j \rfloor$
7. $\quad j \leftarrow j + 1$

end

As above, at step $n \geq \delta$, $X_nY_n - P_n = \beta^{-n}(W_n - p_n)$. Since $0 \leq W_n - p_n < 1$ we get

$$0 \leq X_nY_n - P_n < \beta^{-n}$$

and the algorithm is convergent.

It remains to prove that the p_j's are in D, *i.e.* $0 \leq p_j \leq d$. It is enough to show that $W_j < d + 1$. From Line 5 and the fact that X_j and Y_{j-1} are less than $\frac{d}{\beta^\delta(\beta-1)}$ follows that

$$W_j < \beta + \frac{2d^2}{\beta^\delta(\beta-1)} \leq d + 1$$

by (3). ■

[2] This implies that $\sum_{j\geq1} x_j\beta^{-j}$ and $\sum_{j\geq1} y_j\beta^{-j}$ are in $[0, 1]$.

EXAMPLE **1** *Let $\beta = \varphi = (1+\sqrt{5})/2$ be the golden ratio. Then the canonical digit set is $A_\varphi = \{0, 1\}$. Multiplication on A_φ is on-line computable with delay $\delta = 5$ by (3). This delay does not seem to be optimal, we conjecture that 4 is optimal. We give in Table 1 below the detail of a computation with $x = y = \mathbf{.}0^5 10101$. The numerical value of x is equal to $\varphi^{-5}(\varphi^{-1} + \varphi^{-3} + \varphi^{-5})$. The result is $p = \mathbf{.}0^{10} 101000100001$. Computations are represented in base φ, in a symbolic way.*

j	$(W_j)_\varphi$	p_j
6	.000001	0
7	.00001	0
8	.0010001001	0
9	.010001001	0
10	.101000100001	0
11	1.01000100001	1
12	.1000100001	0
13	1.000100001	1
14	.00100001	0
15	.0100001	0
16	.100001	0
17	1.00001	1
18	.0001	0
19	.001	0
20	.01	0
21	.1	0
22	1.0	1
23	.0	0

Table 1. On-line multiplication in base φ with delay 5

Note that the result is not the greedy β-expansion in general. For instance with $x = \mathbf{.}0^5 10100101010101$ and $y = \mathbf{.}0^5 010100101010101$, Algorithm $M_\mathbb{R}$ gives the result $p = \mathbf{.}0^{11} 10100001100001010001001$, which contains the forbidden pattern 11.

EXAMPLE **2** *Let $\beta = (3+\sqrt{5})/2$ be the square of the golden ratio. Then the canonical digit set is $A_\beta = \{0, 1, 2\}$. Multiplication on A_β is on-line computable with delay $\delta = 3$. This delay is optimal for our algorithm : suppose that the delay 2 is achievable, and take $x = y = \mathbf{.}002222$. The result given by Algorithm $M_\mathbb{R}$ would then be $p = \mathbf{.}00010301011011$ which is not on the alphabet A_β.*

5 Carry-Save *versus* Signed-Digit

In this section β is an integer > 1. Take $D = \{0, \ldots, \beta\}$, then the representation on D is redundant. We call it the Carry-Save representation, because it is used in computer arithmetic under that name in the case that $\beta = 2$ for internal additions in multipliers, see [18].

By the real base algorithm $M_\mathbb{R}$, multiplication in base 2 on $\{0, 1, 2\}$ is on-line computable with delay $\delta = 3$. This delay is optimal, as shown by the following example. Suppose that the delay is 2, and take $x = .00222$ and $y = .00212$. The result computed by the algorithm would be equal to $p = .0001301$.

If $\beta \geq 3$, multiplication on $D = \{0, \ldots, \beta\}$ is on-line computable with the optimal delay $\delta = 2$.

Relation (3) is never satisfied for β integer and $d = \beta - 1$, which is not surprising because it is known that multiplication in integer base on the canonical digit set is not on-line computable.

Internal additions and multiplications by a digit in Algorithm $M_\mathbb{R}$ can be performed in parallel when β is an integer. This is well known when $\beta = 2$. We give below the algorithm for addition in the general case.

PROPOSITION **2** *Addition in the Carry-Save representation can be performed in constant time in parallel.*

Proof. *Input: $x_{n-1} \cdots x_0$ and $y_{n-1} \cdots y_0$ with x_i and y_i in $D = \{0, \ldots, \beta\}$ for $0 \leq i \leq n-1$.*
Output: $s_n \cdots s_0$ with s_i in D such that

$$\sum_{0 \leq i \leq n} s_i \beta^i = \sum_{0 \leq i \leq n-1} x_i \beta^i + \sum_{0 \leq i \leq n-1} y_i \beta^i.$$

```
begin
1.      In parallel for 0 ≤ i ≤ n − 1 do
2.        z_i ← x_i + y_i
3.        if z_i = 2β then { c_{i+1} ← 2; r_i ← 0}
4.        if z_i = 2β − 1 then
              if z_{i−1} ≥ β then
                  { c_{i+1} ← 2; r_i ← −1}
              else { c_{i+1} ← 1; r_i ← β − 1}
5.        if z_i = 2β − k (2 ≤ k ≤ β − 1) then
                 { c_{i+1} ← 1; r_i ← β − k}
6.        if z_i = β then { c_{i+1} ← 1; r_i ← 0}
7.        if z_i = β − 1 then
              if z_{i−1} ≥ β then
                 { c_{i+1} ← 1; r_i ← −1}
              else { c_{i+1} ← 0; r_i ← β − 1}
8.        if 0 ≤ z_i ≤ β − 2 then
                 { c_{i+1} ← 0; r_i ← z_i}
9.        s_i ← c_i + r_i
10.     s_n ← c_n
end
```

Clearly,

$$\sum_{0 \leq i \leq n} s_i \beta^i = \sum_{0 \leq i \leq n-1} x_i \beta^i + \sum_{0 \leq i \leq n-1} y_i \beta^i.$$

One has to prove that the digits s_i's are elements of D.
Since $-1 \le r_i \le \beta - 1$ and $0 \le c_i \le 2$ then $-1 \le s_i \le \beta + 1$. The case $s_i = -1$ can happen only if $r_i = -1$ and $c_i = 0$, which is impossible.
The case $s_i = \beta + 1$ can happen only if $r_i = \beta - 1$ and $c_i = 2$, which is impossible as well. ■

The delay for multiplication in the signed-digit representation is better than in the Carry-Save representation, but note that the digit-set in the signed-digit representation has cardinality $2a + 1$, to be compared to $\mathrm{card}\,(D) = \beta + 1$, and the digits in D being nonnegative take less memory to be stored.

6 On-line multiplication in the Knuth complex number system

THEOREM **3** *Multiplication of two complex numbers represented in base $\beta = i\sqrt{r}$, with r an integer ≥ 2, and digit set $R = \{-a, \ldots, a\}$, $r/2 \le a \le r - 1$, is computable by an on-line algorithm with delay δ, where δ is the smallest odd integer such that*

$$\frac{r}{2} + \frac{4a^2}{r^{\frac{\delta-1}{2}}(r-1)} \le a + \frac{1}{2}. \tag{4}$$

Proof. On-line multiplication algorithm $M_{\mathbb{C}}$.
Input: two sequences $x = (x_j)_{j\ge1}$ and $y = (y_j)_{j\ge1}$ of $R^{\mathbb{N}}$ such that $x_1 = \cdots = x_\delta = 0$ and $y_1 = \cdots = y_\delta = 0$.
Output: a sequence $p = (p_j)_{j\ge1}$ of $R^{\mathbb{N}}$ such that $\sum_{j\ge1} p_j\beta^{-j} = \sum_{j\ge1} x_j\beta^{-j} \times \sum_{j\ge1} y_j\beta^{-j}$.

begin
1. $p_1 \leftarrow 0, \ldots, p_\delta \leftarrow 0$
2. $W_\delta \leftarrow 0$
3. $j \leftarrow \delta + 1$
4. **while** $j \ge \delta + 1$ **do**
5. $\quad W_j \leftarrow \beta(W_{j-1} - p_{j-1}) + y_jX_j + x_jY_{j-1}$
6. $\quad p_j \leftarrow \mathrm{sign}\,(\Re(W_j))\lfloor|\Re(W_j)| + \frac{1}{2}\rfloor$
7. $\quad j \leftarrow j + 1$

end

The digit p_j will belong to R if $|\Re(W_j)| < a + \frac{1}{2}$. By Line 6, for all j, $\Re(|W_j - p_j|) \le \frac{1}{2}$ and $\Im(W_j - p_j) = \Im(W_j)$. Thus, by Line 5,

$$|\Re(W_j)| \le \sqrt{r}|\Im(W_{j-1})| + a(|\Re(X_j) + \Re(Y_{j-1})|$$

and

$$|\Im(W_j)| \le \frac{\sqrt{r}}{2} + a(|\Im(X_j) + \Im(Y_{j-1})|.$$

First suppose that δ is odd. Then

$$|\Re(X_j)| < \frac{a}{r^{\frac{\delta-1}{2}}(r-1)}$$

and

$$|\Im(X_j)| < \sqrt{r}\frac{a}{r^{\frac{\delta+1}{2}}(r-1)}$$

and the same holds true for Y_{j-1}. Thus

$$|\Re(W_j)| \le \frac{r}{2} + \frac{4a^2}{r^{\frac{\delta-1}{2}}(r-1)} < a + \frac{1}{2}$$

by (4).

Suppose now that a better even delay δ' could be achieved. Then

$$|\Re(X_j)| < \frac{a}{r^{\frac{\delta'}{2}}(r-1)}$$

and

$$|\Im(X_j)| < \sqrt{r}\frac{a}{r^{\frac{\delta'}{2}}(r-1)}$$

thus

$$|\Re(W_j)| < \frac{r}{2} + \frac{2a^2(r+1)}{r^{\frac{\delta'}{2}}(r-1)}.$$

This delay will work if

$$\frac{r}{2} + \frac{2a^2(r+1)}{r^{\frac{\delta'}{2}}(r-1)} \le a + \frac{1}{2}. \tag{5}$$

Suppose that the delay in (4) is of the form $\delta = 2k + 1$ and the delay in (5) is of the form $\delta' = 2k'$, and set

$$C = \frac{(r-1)(2a+1-r)}{4a^2}.$$

Then k is the smallest positive integer such that

$$k > \frac{\log(2/C)}{\log(r)}$$

and k' is the smallest positive integer such that

$$k' > \frac{\log((r+1)/C)}{\log(r)}$$

and obviously $k < k'$.

Since for $n \ge \delta$

$$X_nY_n - P_n = \beta^{-n}(W_n - p_n),$$

$$|\Re(W_n - p_n)| \le 1/2$$

and

$$|\Im(W_n - p_n)| = |\Im(W_n)| \le \frac{\sqrt{r}}{2} + \sqrt{r}\frac{2a^2}{r^{\frac{\delta+1}{2}}(r-1)}$$

the algorithm is convergent, and $p_1 \cdots p_n$ is a β-representation of the most significant half of X_nY_n. ■

COROLLARY **2** *The delay δ for Algorithm $M_{\mathbb{C}}$ takes the following values. If $r = 2$ and $a = 1$, $\delta = 7$. If $r = 8$ or $r = 9$ and $a = r - 1$ then $\delta = 3$. If $r = 10$ and $a \geq 7$ then $\delta = 3$. In the other cases, for $r \leq 10$ the delay is $\delta = 5$.*

EXAMPLE **3** *Let $\beta = 2i$ and $R = \{-2, -1, 0, 1, 2\}$. By Corollary 2 the delay δ is equal to 5. Let $x = .0^5 1\bar{2}0\bar{1}201$ and $y = .0^5 1\bar{1}00121$. The result is $p = .0^{10} 1111\bar{1}1\bar{1}2\bar{1}\bar{1}\ldots$.*

j	$(W_j)_{2i}$	p_j
6	$.000001$	0
7	$.0001112$	0
8	$.001112$	0
9	$.01112\bar{1}1$	0
10	$.11110000\bar{1}2$	0
11	$1.1110120\bar{2}$	1
12	$1.11\bar{1}1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
13	$1.1\bar{1}1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
14	$1.\bar{1}1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
15	$\bar{1}.1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	$\bar{1}$
16	$1.\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
17	$\bar{1}.2\bar{1}\bar{1}\bar{1}\bar{1}21$	$\bar{1}$
18	$2.\bar{1}\bar{1}\bar{1}\bar{1}21$	2
19	$\bar{1}.\bar{1}\bar{1}\bar{1}21$	$\bar{1}$
20	$\bar{1}.\bar{1}\bar{1}21$	$\bar{1}$

Table 2. On-line multiplication in base $2i$ with delay 5

Acknowledgements

We are pleased to thank the anonymous referee who indicated us references [17] and [16].

References

[1] A. Avizienis, Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on electronic computers* **10** (1961), 389–400.

[2] J.-C. Bajard, J. Duprat, S. Kla and J.-M. Muller, Some operators for on-line radix 2 computations. *J. of Parallel and Distributed Computing* **22** (1994), 336-345.

[3] Č. Burdík, Ch. Frougny, J.-P. Gazeau and R. Krejcar, Beta-integers as natural counting systems for quasicrystals. *J. of Physics A: Math. Gen.* **31** (1998), 6449–6472.

[4] C.Y. Chow and J.E. Robertson, Logical design of a redundant binary adder. *Proc. 4th Symposium on Computer Arithmetic*, I.E.E.E. Computer Society Press (1978), 109–115.

[5] M.D. Ercegovac, On-line arithmetic: An overview. *Real time Signal Processing VII* **SPIE 495** (1984), 86–93.

[6] M.D. Ercegovac, An on-line square-rooting algorithm. *Proc. 4th Symposium on Computer Arithmetic*, I.E.E.E. Computer Society Press (1978).

[7] Ch. Frougny, On-line finite automata for addition in some numeration systems. *Theoretical Informatics and Applications* **33** (1999), 79–101.

[8] Ch. Frougny, On-line digit set conversion in real base. *Theoret. Comp. Sci.* **292** (2003), 221–235.

[9] W. Gilbert, Radix representations of quadratic fields. *J. Math. Anal. Appl.* **83** (1981), 264–274.

[10] W.H. Kautz, Fibonacci codes for synchronization control. *I.E.E.E. Trans. Infor. Th.* **11** (1965), 284–292.

[11] I. Kátai and J. Szabó, Canonical number systems. *Acta Sci. Math.* **37** (1975), 255–280.

[12] D.E. Knuth, An imaginary number system. *C.A.C.M.* **3** (1960), 245–247.

[13] D.E. Knuth, *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*, 2nd ed., Addison-Wesley, 1988.

[14] M. Lothaire, *Algebraic Combinatorics on Words*, Cambridge University Press, 2002.

[15] D.W. Matula, Basic Digit Sets for Radix Representation. *J.A.C.M.* **29** (1982), 1131–1143.

[16] R. McIlhenny, Complex number on-line arithmetic for reconfigurable hardware : algorithms, implementations, and applications. PhD Dissertation, Computer Science Department, UCLA, 2002.

[17] R. McIlhenny and M.D. Ercegovac, On-line algorithms for complex number arithmetic. *Proc. 32nd Asilomar Conference on Signals, Systems, and Computers* (1998).

[18] J.-M. Muller, *Arithmétique des ordinateurs*, Masson, 1989.

[19] J.-M. Muller, Some characterizations of functions computable in on-line arithmetic. *I.E.E.E. Trans. on Computers* **43** (1994), 752–755.

[20] A.M. Nielsen and J.-M. Muller, Borrow-Save Adders for Real and Complex Number Systems. In Proceedings of the Conference *Real Numbers and Computers*, Marseilles, 1996, 121–137.

[21] W. Parry, On the β-expansions of real numbers. *Acta Math. Acad. Sci. Hungar.* **11** (1960), 401–416.

[22] W. Penney, A "binary" system for complex numbers. *J.A.C.M.* **12** (1965), 247–248.

[23] A. Rényi, Representations for real numbers and their ergodic properties. *Acta Math. Acad. Sci. Hungar.* **8** (1957), 477–493.

[24] K.S. Trivedi and M.D. Ercegovac, On-line algorithms for division and multiplication. *I.E.E.E. Trans. on Computers* **C 26** (1977), 681–687.

[25] A. Surarerks, On-line arithmetics in real and complex base. Ph.D. Dissertation, University Paris 6, LIAFA report 2001/06, june 2001.