Parallel and On-Line Addition in Negative Base and Some Complex Number Systems

Christiane Frougny

Université Paris 8 and
Laboratoire Informatique Théorique et Programmation,
Institut Blaise Pascal,
4 place Jussieu, 75252 Paris Cedex 05, France.
Email: Christiane.Frougny@litp.ibp.fr

Workshop 10: Computer Arithmetic

Abstract. We give parallel and on-line algorithms for addition in number systems where the base is a negative integer, or some complex number of the form $i\sqrt{b}$ or $-1+i$.

1 Introduction

Addition of two numbers in the classical b-ary number system, where b is an integer ≥ 2, has a linear time complexity. In order to save time, several solutions have been proposed. A celebrated one is the Avizienis signed-digit representation [Av], which consists in changing the digit set. Instead of taking digits from the canonical set $\{0, \cdots, b-1\}$, they are taken from a balanced set of the form $\{\bar{a}, \cdots, a\}$, where $\bar{a}$ denotes the digit $-a$, a being an integer such that $a+1 \leq b \leq 2a-1$ (b must be ≥ 3). Such a number system is *redundant*, that is to say, a number may have several representations. This property allows to perform addition in constant time in parallel, because there is no carry propagation. A similar algorithm for base 2 has been devised by Chow and Robertson [CR] using digit set $\{\bar{1}, 0, 1\}$. Here addition is realized in parallel with a window of size 2. These signed-digit representations are really implemented; for instance in the Pentium processor, numbers are represented in base 4 with digit set $\{\bar{2}, \cdots, 2\}$ to perform fast division.

On-line arithmetic consists in performing arithmetic operations in Most Significant Digit First (MSDF) mode [Er], digit serially after a certain delay of latency. This mode of doing allows pipelining 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. To be able to perform on-line addition, it is necessary to use a redundant number system ([Maz]). From a parallel algorithm for addition, it is straighforward to derive an on-line algorithm for the same operation.

In this paper we use a model of computability which is the one of finite state automata. A function is computable by a finite state automaton if it needs only a finite auxiliary storage memory, independently of the size of the data. In that setting, one knows that addition of two integers in the classical b-ary system is computable by a finite state automaton, but that multiplication is not ([Ei]). Actually, the natural f.s.a. one designs to perform addition is a sequential one, processing numbers in the Least Significant Digit First (LSDF) mode. On-line f.s.a. have been introduced by Muller [M]. They are sequential f.s.a. processing data in MSDF mode. The Avizienis and the Chow-Robertson algorithms for parallel addition lead to the construction of on-line f.s.a. for addition ([M], [FS]).

We study here number systems where the base is a negative integer, or a quadratic complex number of the form $\beta = i\sqrt{b}$ or $\beta = -1 + i$. With a negative base, any real number can be expressed without a sign. Representing complex numbers with a complex base implies that those numbers can be manipulated without separating the real and the imaginary part. The bases we are considering are strongly related to negative bases: $(i\sqrt{b})^2 = -b$ and $(-1+i)^4 = -4$. We consider balanced signed-digit sets such that the number systems we obtain are minimally redundant (with respect to the cardinality of the digit set). We give parallel algorithms for addition in these systems, from which we derive the design of on-line finite state automata. We also give some constructions for digit set and/or base conversion.

Complete proofs will appear in a forthcoming paper.[1]

2 Preliminaries

Number representations

Let β be a real or complex number such that $|\beta| > 1$, and let A be a finite set of real or complex digits. A *β-representation* of x with digits in A is a finite or a right infinite sequence $(x_k)_{k \le k_0}$ with $x_k \in A$ such that $x = \sum_{k=k_0}^{-\infty} x_k \beta^k$. It is denoted by

$$(x_{k_0} \cdots x_0 \cdot x_{-1} x_{-2} \cdots)_\beta$$

Automata

Let A be a finite set. The set of finite sequences of elements of A, called *words*, is denoted by A^*. The empty word is ε. Let A and B be two finite sets. An *on-line finite state automaton* $\mathcal{A} = (Q, A \times (B \cup \varepsilon), F, i_0, \omega)$ with delay δ is a directed labelled graph with a finite number of vertices called *states* : Q is the finite set of states, i_0 is the initial state, F is the set of edges, labelled by elements of $A \times (B \cup \varepsilon)$, and $\omega : Q \longrightarrow B^*$ is the *terminal function*. The following properties must be satisfied. First, the automaton is *sequential*, that is to say, if $p \xrightarrow{a/b} q$ and $p \xrightarrow{a/b'} q'$ are two edges, then necessarily $q = q'$ and $b = b'$. Secondly, every path of length δ starting in i_0 is of the form

$$i_0 \xrightarrow{a_1/\varepsilon} i_1 \xrightarrow{a_2/\varepsilon} \cdots \xrightarrow{a_\delta/\varepsilon} i_\delta,$$

[1] We were completing the writing of this paper when we learnt that parallel algorithms for addition in base -2, $i\sqrt{2}$, $2i$ and $-1+i$ have been independently given by Nielsen and Muller in [NM].

the only edge arriving in a state $i_0, \ldots, i_{\delta-1}$ is as above, and all the other edges of $\mathcal{A}$ are labelled by elements of $A \times B$. This means that the automaton starts reading words of length $\leq \delta$ outputting nothing, and after that delay, outputs serially one digit for each input digit.

A function $f : A^* \longrightarrow B^*$ is *computable by an on-line finite state automaton* if there exists such an automaton $\mathcal{A}$ such that the graph of f is equal to the set $\{(x,y) \in A^* \times B^* \mid y = y'y'', (x,y')$ is the label of a path in $\mathcal{A}$ starting in i_0, ending in state q, and $\omega(q) = y''\}$.

The same definition works for functions of infinite words, considering infinite paths in $\mathcal{A}$, but there is no terminal function ω in that case.

An on-line function treats numbers in most significant digit first (MSDF) mode, an on-line finite state automaton processes words sequentially from left to right. We need also the reverse definition: a least significant digit first (LSDF) sequential finite state automaton is the same thing than an on-line finite state automaton, but words are processed from right to left, and the terminal function ω comes as a prefix of the output.

Addition

To perform addition in a given number system with base β and digit set A, the procedure will always be the same: take two numbers $x = x_{n-1} \cdots x_0$ and $y = y_{n-1} \cdots y_0$ such that $x = \sum_{k=0}^{n-1} x_k \beta^k$, $y = \sum_{k=0}^{n-1} y_k \beta^k$, with x_k and y_k in A. If the expansions are infinite, the process is identical. In parallel, compute $z_k = x_k + y_k$. Then z_k is an element of $B = \{c+d \mid c, d \in A\}$, and $x+y = \sum_{k=0}^{n-1} z_k \beta^k$. Addition consists in transforming the representation $z_{n-1} \cdots z_0$ of $x+y$ on B into an equivalent one $s_{n-1+l} \cdots s_0$, with $s_k \in A$, such that $x + y = \sum_{k=0}^{n-1+l} s_k \beta^k$. This is a special case of what is called *normalization*, which consists in the conversion of any representation of a number in base β on a given digit set into an equivalent one on a "canonical" digit set ([F1]).

3 Negative base number systems

Let $\beta = -b$, where b is an integer ≥ 2. It is well known (see [K1, K2], [Ma]) that any real number can be uniquely represented in base $-b$ with digits from the canonical digit set $A = \{0, \cdots, b-1\}$.

Proposition 1. *Addition in base $\beta = -b$, $b \geq 2$, with digits in $A = \{0, \cdots, b-1\}$, is computable by a LSDF sequential finite state automaton (with delay 0).*

Proof. As explained above, we have to convert representations over $B = \{0, \cdots, 2b-2\}$ into equivalent representations over A. Number representations are processed in LSDF mode, that is to say from right to left. The set of states of the automaton is $Q = \{0, 1, \bar{1}\}$. The name of a state indicates what is kept in memory. The initial state is 0. Let $z \in B = \{0, \cdots, 2b-2\}$. Edges are defined by: If $0 \leq z \leq b-1$, let $0 \xrightarrow{z/z} 0$. If $b \leq z \leq 2b-2$, let $0 \xrightarrow{z/z-b} \bar{1}$. If $z = 0$, let $\bar{1} \xrightarrow{0/b-1} 1$. If $1 \leq z \leq b$, let $\bar{1} \xrightarrow{z/z-1} 0$. If $b+1 \leq z \leq 2b-2$, let $\bar{1} \xrightarrow{z/z-b-1} \bar{1}$. If

$0 \leq z \leq b-2$, let $1 \xrightarrow{z/z+1} 0$. If $b-1 \leq z \leq 2b-2$, let $1 \xrightarrow{z/z-b+1} \bar{1}$. The terminal function ω is defined by $\omega(0) = \varepsilon$, $\omega(1) = 1$ and $\omega(\bar{1}) = 1(b-1)$. ∎

Remark. Addition in base $-b$ with digits in A is not computable by an on-line finite state automaton: let us take $b = 2$ and $A = \{0, 1\}$. Then $(01)^n 02$ has to be transformed into $1(10)^{n+1}$, but $(01)^{n+1}$ is transformed into itself, so we have to know the least significant digit to be able to output the most significant digit of the result.

We introduce an other set of digits in order to obtain a redundant number system, analoguous to the Avizienis signed-digit representation [Av]. Let a such that $a+1 \leq b \leq 2a$ and let $D = \{\bar{a}, \cdots, a\}$. Then every real number has a representation in base $-b$ with digits in D. The system is redundant. From now on, we consider redundant digit sets which are the smallest ones for which addition can be realized purely in parallel (Proposition 2) or in parallel with a window of length 2 (Proposition 3).

Proposition 2. *Let $\beta = -b$, where b is an integer ≥ 3, and let $D = \{\bar{a}, \cdots, a\}$ where $a = \lfloor b/2 \rfloor + 1$. Then addition base $-b$ can be realized purely in parallel in constant time. Moreover, addition is computable by an on-line finite state automaton with delay 1 and $2a-1$ states.*

Proof. 1) Let $x = x_{n-1} \cdots x_0$ and $y = y_{n-1} \cdots y_0$ be two numbers written in base β, with x_k and y_k in D for $0 \leq k \leq n-1$, and let $z_k = x_k + y_k$, $z_k \in E = \{\overline{(2a)}, \cdots, (2a)\}$.
If $a \leq z_k \leq 2a$, let $c_{k+1} = \bar{1}$ and $r_k = z_k - b$. If $-2a \leq z_k \leq -a$, let $c_{k+1} = 1$ and $r_k = b + z_k$. If $|z_k| \leq a-1$, let $c_{k+1} = 0$ and $r_k = z_k$.

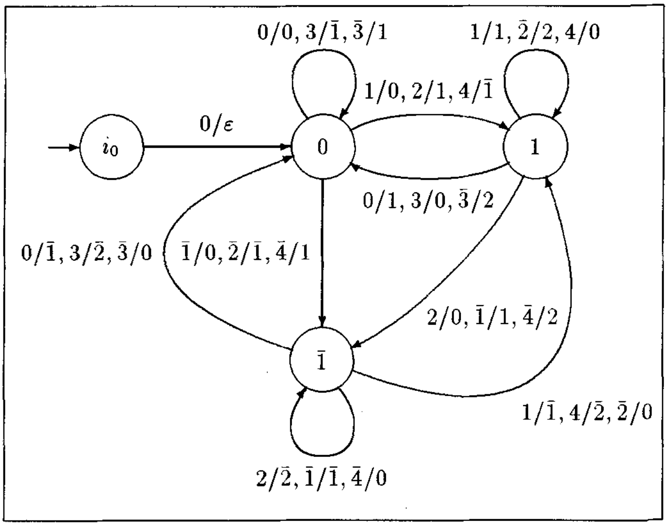
Then in any case, $z_k = \beta c_{k+1} + r_k$. Let us put $s_k = r_k + c_k$ for $0 \leq k \leq n-1$ and $s_n = c_n$. Thus $x + y = \sum_{k=0}^{n} s_k \beta^k$. Since $|c_k| \leq 1$ and $|r_k| \leq b - a \leq a - 1$, we get $s_k \in D$ for $0 \leq k \leq n$.

2) From that algorithm we define an on-line finite state automaton with delay 1 realizing addition. To avoid overflow, we assume that words begin with a 0. Let $z = z_k \in E$ and let $\tau(z) = (c, r) = (c_{k+1}, r_k)$ as determined by the above algorithm. The set of states of the automaton is $Q = \{i_0\} \cup \{-a+1, \cdots, a-1\}$. The initial state is i_0. Edges are defined this way: $i_0 \xrightarrow{0/\varepsilon} 0$, and, for any $q \in Q \setminus \{i_0\}$ and for any $z \in E$, we have an edge $q \xrightarrow{z/c+q} r$, with $(c, r) = \tau(z)$. Since $|c| \leq 1$ and $|q| \leq a-1$, $c + q \in D$ and $r \in Q$. The terminal function associated to any state $q \neq i_0$ is $\omega(q) = q$. ∎

Example

Let $\beta = -3$ and let $D = \{\bar{2}, \cdots, 2\}$. Below is the on-line finite automaton with delay 1 realizing addition in this system.[2] Take $x = 020\bar{2}$ and $y = 02\bar{1}\bar{2}$. Then $x + y = 04\bar{1}\bar{4}$. We have in the automaton $i_0 \xrightarrow{0/\varepsilon} 0 \xrightarrow{4/\bar{1}} 1 \xrightarrow{\bar{1}/1} \bar{1} \xrightarrow{\bar{4}/0} \bar{1}$ and $\omega(\bar{1}) = \bar{1}$, thus $x + y = \bar{1}10\bar{1}$.

[2] I thank Paul Gastin for his set of macros Autograph.



In the case that $\beta = -2$, the previous algorithm does not apply. We give a special algorithm for that case, which can be extended to any even b together with a minimally redundant digit set, and which is analoguous to the Chow and Robertson algorithm for base 2 [CR].

Proposition 3. *Let $\beta = -b$, where $b = 2a$, a being an integer ≥ 1, and let $D = \{\bar{a}, \cdots, a\}$. Then addition base $-b$ can be realized in parallel in constant time with a window of length 2. Addition is computable by an on-line finite state automaton with delay 2, and $4a^2 + 2a + 3$ states.*

Proof. For $0 \leq k \leq n-1$, let $z_k = x_k + y_k \in E = \{\bar{b}, \cdots, b\}$ as above. If $a+1 \leq z_k \leq b$, let $c_{k+1} = \bar{1}$ and $r_k = z_k - b$. If $-b \leq z_k \leq -a-1$, let $c_{k+1} = 1$ and $r_k = b + z_k$. If $z_k = a$, and if $z_{k-1} < 0$ then let $c_{k+1} = \bar{1}$ and $r_k = \bar{a}$ else let $c_{k+1} = 0$ and $r_k = a$. If $z_k = -a$, and if $z_{k-1} > 0$ then let $c_{k+1} = 1$ and $r_k = a$ else let $c_{k+1} = 0$ and $r_k = -a$. If $|z_k| \leq a-1$, let $c_{k+1} = 0$ and $r_k = z_k$.
Then we have $z_k = \beta c_{k+1} + r_k$. Let $s_k = r_k + c_k$ for $0 \leq k \leq n-1$ and $s_n = c_n$. Clearly $x + y = \sum_{k=0}^{n} s_k \beta^k$. We have to show that $s_k \in D$. When $a+1 \leq |z_k| \leq b$, whatever the value of z_{k-1} is, we get $|r_k| \leq a-1$ and $|c_k| \leq 1$, thus $|s_k| \leq a$. If $z_k = a$, and if $z_{k-1} < 0$ then $r_k = \bar{a}$ and $c_k = 0$ or 1, thus $s_k = -a$ or $-a+1$ and thus belongs to D. If $z_k = a$ and $z_{k-1} \geq 0$, then $r_k = a$ and $c_k = -1$ or 0, and so $s_k = a-1$ or a. The case $z_k = -a$ is symmetric. If $|z_k| \leq a-1$, $r_k = z_k$ and $|c_k| \leq 1$, thus $s_k \in D$.

The construction of the on-line automaton will be given in the full paper. ■

In [Ko] it is shown that digit set conversion can be very useful in Computer Arithmetic. Here we have the following.

Corollary 4. *The digit set conversion in base $-b$ between numbers written with digits in the canonical digit set A into their representation with digits in D is computable in parallel in constant time, and computable by an on-line finite* *automaton. The inverse conversion, from D to A, cannot be computed on-line, but in LSDF mode only (with a sequential finite state automaton).*

In the same spirit, in [Al], it is shown that conversion between numbers written in base b, b integer ≥ 2, with digit set $A = \{0, \cdots, b-1\}$ into their representation in base $-b$ with the same digit set is computable by a LSDF sequential finite state automaton. We show the following.

Proposition 5. *Let b be an integer ≥ 2. The conversion from base b and digit set $A = \{0, \cdots, b-1\}$ into base $-b$ and digit set $D = \{\bar{a}, \cdots, a\}$, whith $a+1 \leq b \leq 2a$, is computable by a LSDF sequential finite state automaton. The inverse conversion is also computable by a LSDF sequential finite state automaton.*

4 Complex number systems

Base $\beta = i\sqrt{b}$

Let $\beta = i\sqrt{b}$, where b is an integer ≥ 2. Any complex number is uniquely representable in base β with digits in the canonical digit set $A = \{0, \cdots, b-1\}$ (see [K1], [KS], [Gi]). Note that when b is a perfect square, $b = c^2$, c integer ≥ 2, cases $\beta = ci$ as well as $\beta = -ci$ are included in the general case. Most studied cases are $\beta = 2i$ and $A = \{0, \cdots, 3\}$, strongly related to base -4, and $\beta = i\sqrt{2}$ and $A = \{0, 1\}$ ([K1, K2]).

Let j be an integer ≥ 0, possibly infinite, and let $k \geq 0$. Since $\beta^2 = -b$, we have

$$(a_{2k} \cdots a_0 \cdot a_{-1} \cdots a_{-2j})_\beta =$$

$$(a_{2k}a_{2k-2} \cdots a_0 \cdot a_{-2} \cdots a_{-2j})_{-b} + i\sqrt{b}(a_{2k-1}a_{2k-3} \cdots a_1 \cdot a_{-1} \cdots a_{-2j+1})_{-b}.$$

Thus, if $z = x + iy \in \mathbf{C}$, x and y in $\mathbf{R}$, the β-representation of z can be obtained by interleaving the $-b$-representation of x and the $-b$-representation of $y/\sqrt{b}$.

Proposition 6. *Addition in base $\beta = i\sqrt{b}$, $b \geq 2$, with digits in $A = \{0, \cdots, b-1\}$ is computable by a LSDF sequential finite state automaton. It cannot be computed on-line.*

Proposition 7. *Let $\beta = i\sqrt{b}$, where b is an integer ≥ 2, let $a = \lfloor b/2 \rfloor + 1$ and let $D = \{\bar{a}, \cdots, a\}$. Then addition in base $\beta = i\sqrt{b}$ can be realized purely in parallel in constant time. Addition is computable by an on-line finite state automaton with delay 2 and $(2a-1)^2 + 2$ states.*

Proof. We define $z_k \in E = \{\overline{(2a)}, \cdots, (2a)\}$ as above. If $a \leq z_k \leq 2a$, let $c_{k+2} = \bar{1}$ and $r_k = z_k - b$. If $-2a \leq z_k \leq -a$, let $c_{k+2} = 1$ and $r_k = b + z_k$. If $|z_k| \leq a-1$, let $c_{k+2} = 0$ and $r_k = z_k$. In any case, $|c_k| \leq 1$ and $|r_k| \leq b-a \leq a-1$.

Thus, $z_k = \beta^2 c_{k+2} + r_k$. Let us put $s_k = r_k + c_k$ for $0 \leq k \leq n-1$, $s_n = c_n$, and $s_{n+1} = c_{n+1}$. We have $x + y = \sum_{i=0}^{n+1} s_k \beta^k$ with $|s_k| \leq a$.

The construction of the on-line automaton will be given in the full paper. ∎

We now consider the minimally redundant digit set associated to $\beta = i\sqrt{b}$ when b is even.

Proposition 8. *Let $\beta = i\sqrt{b}$, where $b \geq 2$ is even, let $a = b/2$ and let $D = \{\bar{a}, \cdots, a\}$. Then addition base $\beta = i\sqrt{b}$ can be realized in parallel in constant time with a window of length 3. Addition is computable by an on-line finite state automaton with delay 3 and about b^4 states.*

Proof. Define $z_k \in E = \{-b, \cdots, b\}$ as above. If $a + 1 \leq z_k \leq 2a$, let $c_{k+2} = \bar{1}$ and $r_k = z_k - b$,. If $-2a \leq z_k \leq -a - 1$, let $c_{k+2} = 1$ and $r_k = b + z_k$. If $z_k = a$ and if $z_{k-2} < 0$, let $c_{k+2} = \bar{1}$ and $r_k = \bar{a}$ else let $c_{k+2} = 0$ and $r_k = a$. If $z_k = \bar{a}$ and if $z_{k-2} > 0$, let $c_{k+2} = 1$ and $r_k = a$ else let $c_{k+2} = 0$ and $r_k = \bar{a}$. If $|z_k| \leq a - 1$, let $c_{k+2} = 0$ and $r_k = z_k$. In any case, $|c_k| \leq 1$ and $|r_k| \leq a$. We have $z_k = \beta^2 c_{k+2} + r_k$. Let $s_k = r_k + c_k$ for $0 \leq k \leq n - 1$, $s_n = c_n$, and $s_{n+1} = c_{n+1}$. Then $x + y = \sum_{i=0}^{n+1} s_k \beta^k$.

If $a+1 \leq |z_k| \leq 2a$, then $|r_k| \leq a-1$. Since $|c_k| \leq 1$, $|s_k| \leq a$. If $|z_k| \leq a-1$, we get $|s_k| \leq a$. Let $z_k = a$ and $z_{k-2} < 0$. Thus $r_k = -a$ and $c_k = 1$ or 0, so $s_k \in D$. If $z_k = a$ and $z_{k-2} \geq 0$, then $r_k = a$, $c_k = \bar{1}$ or 0, thus $s_k \in D$. The case $z_k = \bar{a}$ is similar. ∎

Corollary 9. *The digit set conversion in base $\beta = i\sqrt{b}$ between numbers written with digits in A into their representation with digits in D is computable in parallel in constant time, and computable by an on-line finite automaton. The inverse conversion, from D to A cannot be computed on-line, but in LSDF mode only (with a sequential finite automaton).*

Base $\beta = -b + i$

We now consider base $\beta = -b + i$, where b is an integer ≥ 1. It is known ([KS], [Pe]) that any complex number is uniquely representable in base $\beta = -b+i$ with $A = \{0, \cdots, b^2\}$ for canonical digit set. The case $\beta = -b - i$ is the same. Every following property is satisfied by both bases. Recall that addition in base $\beta = -b + i$, $b \geq 1$, with digits in $A = \{0, \cdots, b^2\}$ is computable by a LSDF sequential finite state automaton [Sa].

Remark that $(-1+i)^4 = -4$, but that for any $b \geq 2$, there is no integer $k \neq 0$ such that $(-b + i)^k$ is an integer. Thus, from now on, we treat only of the case $\beta = -1 + i$ and $A = \{0, 1\}$. The minimally redundant digit set for base $-1 + i$ is $\{\bar{1}, 0, 1\}$, but it is an open problem to give a practical algorithm performing addition in parallel for this digit set (see [NM]). Since $(-1 + i)^2 = -2i$, it is natural to take $D = \{\bar{2}, \cdots, 2\}$ as a digit set leading to a redundant number system.

Proposition 10. *Addition in base $\beta = -1 + i$ and digit set $D = \{\bar{2}, \cdots, 2\}$ can be realized in parallel in constant time with a window of length 5. Addition is computable by an on-line finite state automaton with delay 5, and about 5^6 states.*

Proof. Let $z_k \in E = \{\bar{4}, \cdots, 4\}$, $0 \leq k \leq n - 1$. If $z_k = 4$, let $c_{k+4} = \bar{1}$, $r_k = 0$. If $z_k = \bar{4}$, let $c_{k+4} = 1$, $r_k = 0$. If $z_k = 3$, let $c_{k+4} = \bar{1}$, $r_k = \bar{1}$. If $z_k = \bar{3}$, let $c_{k+4} = 1$, $r_k = 1$. If $z_k = 2$ and if $z_{k-4} < 0$, let $c_{k+4} = \bar{1}$ and $r_k = \bar{2}$ else $c_{k+4} = 0$ and $r_k = 2$. If $z_k = \bar{2}$ and if $z_{k-4} > 0$, let $c_{k+4} = 1$ and $r_k = 2$ else $c_{k+4} = 0$ and $r_k = \bar{2}$. If $|z_k| \leq 1$, let $c_{k+4} = 0$ and $r_k = z_k$. We have $z_k = \beta^4 c_{k+4} + r_k$. Let $s_k = c_k + r_k$.

If $3 \leq |z_k| \leq 4$, or $|z_k| \leq 1$, then in any case $|r_k| \leq 1$ and $|c_k| \leq 1$. thus $|s_k| \leq 2$. If $z_k = 2$ and $z_{k-4} < 0$, we have $r_k = \bar{2}$ and $c_k = 1$ or 0, thus $|s_k| \leq 1$. If $z_k = 2$ and $z_{k-4} \geq 0$, then $r_k = 2$ and $c_k = \bar{1}$ or 0, thus $|s_k| \leq 2$. The case $z_k = \bar{2}$ is symmetric. ∎

Remark. Conversion in base $-1+i$ between digit set $A = \{0,1\}$ and $D = \{\bar{2}, \cdots, 2\}$ is not on-line computable, but is computable by a LSDF sequential finite state automaton.

References

[Al] J.-P. Allouche, E. Cateland, W.J. Gilbert, H.-O. Peitgen, J.O. Shallit, G. Skordev, Automatic Maps in Exotic Numeration Systems. Preprint (1995).

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

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

[Ei] S. Eilenberg, *Automata, Languages and Machines*, vol. A, Academic Press, 1974.

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

[F1] Ch. Frougny, Representation of numbers and finite automata. *Math. Systems Theory* **25** (1992), 37–60.

[FS] Ch. Frougny et J. Sakarovitch, Synchronisation déterministe des automates à délai borné. *Theoretical Computer Science*, to appear.

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

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

[K1] D.E. Knuth, An Imaginary Number System. *C.A.C.M.* **3** (1960), 245–247.

[K2] D.E. Knuth, *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*, 2nd ed., Addison-Wesley, 1988.

[Ko] P. Kornerup, Digit-Set Conversions: Generalizations and Applications. *I.E.E.E. Trans. on Computers* **43–5** (1994), 622–629.

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

[Maz] Ch. Mazenc, Systèmes de représentation des nombres et arithmétiques sur machines parallèles, Ph.D. Dissertation, Ecole Normale Supérieure de Lyon, France, 1993.

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

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

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

[Sa] T. Safer, La normalisation en base complexe. Preprint (1995).