A Novel Floating-Point Online Division Algorithm

Haixiang Lin and Henk J. Sips

Delft University of Technology
Delft, The Netherlands

abstract

This paper describes a new online division (reciprocal) algorithm for (maximally) redundant floating-point numbers of arbitrary radix. The algorithm works for normalized, quasi-normalized, and pseudo-normailized numbers and can therefore be applied in chained online compuatations. The online delay of the proposed algorithm is the smallest reported so far. The algorithm consista of two steps: the first m digits of the result are generated by a simple table lookup method; the remaining n-m digits are generated by using an adapted Newton-Raphson iteration method. In the second step, the online digits are created by using a fast and simple selection mechanism.

1. Introduction

In online computations, the input operands, as well as the results, flow through arithmetic units in a digit by digit manner, starting with the most significant digit. An online algorithm is said to have an online delay of δ, if for the generation of the j-th digit of the result, $(j+\delta)$ digits of the corresponding operands are required [TRIV77].

During the last decade, online processing has gained much attention [ERCE84]. In the conventional digit-serial arithmetic, in general, all digits must be known in advance before a result (or a part of a result) can be generated. In conventional digit-parallel arithmetic units, high speed multi-operand processing requires full precision bandwidth (as many bits as the word-length) between the arithmetic units in a parallel or/and pipelined computational structure. This is not desirable or feasible as the complexity of the computation increases. In contrast to the conventional digit-parallel approach, online arithmetic reduces the interconnection complexity between the processing units to a minimum of one digit per operand while still providing a good speedup ratio by overlapping or pipelining successive computations. Digit-serial conventional arithmetic also reduces the communication requirements; but it is very slow: the next operation cannot begin until the current operation has been completed.

Fig.1 illustrates the evaluation of an expression. The time diagrams of the conventional method of processing and the online principle are shown in the figure.

This paper deals with the problem of online division. A systematic approach for online reciprocal approximation is proposed. In literature ([TRIV77], [IRWI78], [OWEN81], [OWEN80]), a number of online algorithms dealing with the complete division (Q/P) operation are described. This does not always need to be advantageous than first calculating the reciprocal followed by a multiplication. Those online division algorithms have online delays of $\delta=3$ to 5. It has been shown that $\delta=3$ for radix-8 [OWEN80] and $\delta=4$ for radix-4 normalized numbers [OWEN81]). The online delays of other radices by using this algorithm cannot easily be determined, since no general formula for the online delay has been given as function of the radix. The online division algorithm considered in [TRIV77] has an online delay of $\delta=4$. This is a relative large online delay, since for all other 4 basic functions (addition, subtraction, multiplication, square rooting), algorithms have been found with online delay of 1 [ERCE84]. The proposed online reciprocal algorithm has a smaller online delay. For radix-r redundant numbers with $r\geq 4$, the online delays is 1 and 2 to 3 for normalized and quasi-normalized numbers, respectively. For example, to evaluate the expression $y=(a+b)*(c*d)/\sqrt{(e-f)}$ by using the scheme in Fig. 1, the total online delay can be reduced by 2 to 3 compared to using a complete division unit.

Definition: A non-zero redundant floating-point number P with n digits of mantissa, defined as

$$P=\sum_{j=1}^{n} p_j r^{-j}$$

and represented by a maximally redundant digit set, is said to be

(1) normalized if $r^{-1}\leq|P|<1$;
(2) quasi-normalized if $r^{-2}\leq|P|<1$;
(3) pseudo-normalized if $r^{-q}\leq|P|<1$ with $3\leq q\leq n$.

◊

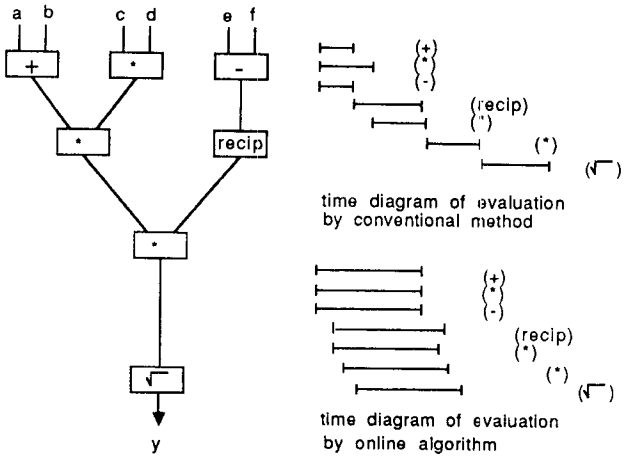

Fig. 1 Computing $y=(a+b)*(c*d)/\sqrt{(e-f)}$. Assume that the online delaysof the different operations are the following: $\delta(+)=1$, $\delta(-)=1$, $\delta(*)=1$, $\delta(\sqrt{})=1$, and $\delta(recip)=1$ to 2.

As is shown in [WATA81], floating-point online arithmetic algorithms yield quasi-normalized results for $r>2$. However, for radix-2 numbers, quasi-normalized results also can be obtained by testing the two first consecutive result digits. When they have opposite signs the second digit is converted and first one is made zero; then the converted digit is compared with the next digit, etc. This process goes on until the two digits to be tested have the same sign or the second digit is zero. In that case the result is quasi-normalized. Of course, the exponent has to be properly adjusted. So, all online processing arithmetic units must be able to handle quasi-normalized numbers in order to chain with other online processing units. The earlier known online division algorithms consider normalized redundant numbers only. In this paper, online division with respect to both normalized and quasi-normalized numbers is considered. The derived results are also valid for pseudo-normalized numbers, however, in practice this leads to realization difficulties, especially in the table look-up procedure. The proposed algorithm is valid for all radices, including radix-2 redundant numbers.

In the following, we will first consider the online digit generation using a RECIP table in Section II. Next the online reciprocal approximation algorithm is given in Section III. In Section IV, the implementational aspects of the proposed online reciprocal unit are considered.

2. Online table lookup method for reciprocal approximation

In this section, an online table lookup method is described. The proposed method produces online reciprocal approximations with a RECIP table.

Definition: Let P, with $r^{-q}\le|P|<1$, be the n-digit mantissa of a redundant floating-point number. The RECIP function f_R is defined as

$$f_R(P(i)) = X(i) \quad where \quad P(i) = \sum_{j=1}^{i+k} p_j r^{-j}$$

$P(i)$ is the i-th approximation to P with k being a positive constant, and $X(i)$ equals to the first i digits of the sum $1/P(i)+sign(P).(r/2).r^{-(i-q)}$, i.e., $X(i)$ is the value of $1/P(i)$ symmetrically rounded to the i-th digit. ◊

The most significant digit x_1 of $X(i)$ has a weight of r^q. $sign(P)$ is the sign function with $sign(P)=1$ if $P>0$, $sign(P)=-1$, otherwise. The first digit, p_1 of P, determines the sign of $P(i)$ for all i. $X(i)$ is represented in the sign-magnitude form, all digits x_k of $X(i)$ have the same sign as P, i.e., $sign(x_k)=sign(P)$ for all k with $1\le k\le i$.

Assume that the RECIP function f_R is implemented by a table lookup unit (RECIP table). It is known that $X(i)$ is one digit more precise than $X(i-1)$. Suppose $X(i-1)$ has been produced in the previous cycle. The next output result will be one digit more accurate if the difference $X(i)-X(i-1)$ is being used to correct $X(i-1)$. This reciprocal approximation can be made online, if for all i $(1\le i\le n)$ the difference $X(i)-X(i-1)$ can be represented by one single digit with a weight of $r^{-(i-q-1)}$. In that case, at the first step $X(1)=x_1$ is produced, then the following digits are equal to $X(2)-X(1)$, $X(3)-X(2)$, respectively.

In the following, we will show that if the parameter k is chosen properly, then the difference $X(i)-X(i-1)$ can be represented by one single digit with a weight of $r^{-(i-q-1)}$. The online delay for this online table lookup method for reciprocal approximation is k.

Lemma 1: Let $X(i-1)=f_R(P(i-1))$ and $X(i)=f_R(P(i))$, then

$$|X(i)-X(i-1)| < r^{-(i-q-2)}\left(\frac{1}{2r}+\frac{1}{2}+r^{-(k-q+2)}+r^{-(k-q+1)}\right) \qquad (1)$$

if $r^{-q}\le|P|<1$

Proof: Given $r^{-q}\le|P|<1$, the approximation to the reciprocal of P can be represented as a redundant number X, consisting of n digits, with x_i, $1\le i\le n$, having a weight of $r^{-(i-q-1)}$. Since $P(i)=p_1p_2\ldots p_{i+k}=P+\varepsilon$ with $|\varepsilon|<r^{-(i+k)}$, it holds

$$\left|\frac{1}{P}-\frac{1}{P(i)}\right| = \left|\frac{1}{P}-\frac{1}{P+\varepsilon}\right| = \left|\frac{\varepsilon}{P(P+\varepsilon)}\right| < \frac{r^{-(i+k)}}{r^{-q}.r^{-q}} = r^{-(i+k-2q)}$$

The error caused by applying symmetrical rounding to $1/P(i)$ at the i-th digit is equal to or less than $(r/2).r^{-(i-q)}$, i.e., $|X(i)-1/P(i)|\le(r/2).r^{-(i-q)}$. Therefore,

$$\left|X(i)-\frac{1}{P}\right| \le \left|X(i)-\frac{1}{P(i)}\right| + \left|\frac{1}{P(i)}-\frac{1}{P}\right| < \frac{r}{2}.r^{-(i-q)}+r^{-(i+k-2q)} \qquad (2)$$

Furthermore,

$$|X(i)-X(i-1)| \le |X(i)-\frac{1}{P}| + |X(i-1)-\frac{1}{P}| < r^{-(i-q-2)}\left(\frac{1}{2r}+\frac{1}{2}+r^{-(k-q+2)}+r^{-(k-q+1)}\right)$$

So, lemma 1 has been proved. ◊

Corollary: Let $X(i-1)=f_R(P(i-1))$ and $X(i)=f_R(P(i))$, then it holds $|X(i)-X(i-1)|<r^{-(i-q-2)}$, if

(1) $k=3$ and $r=2$, when P is normalized;
(2) $k=4$ and $r=2$, when P is quasi-normalized;
(3) $k=2$ and $r=3$, when P is normalized;
(4) $k=3$ and $r=3$, when P is quasi-normalized.
(5) $k=1$ and $r\ge4$, when P is normalized;
(6) $k=2$ and $r\ge4$, when P is quasi-normalized.

Proof: The corollary can be easily verified using Lemma 1. For example, consider P is a normalized radix-2 redundant number and $k=3$. Substitution of $r=2$, $k=3$ and $q=1$ into Eq.(2), leads to $|X(i)-X(i-1)|\le(15/16).r^{-(i-q-2)}$. The other cases can be proved in a similar way.

The following lemma shows that for all i $(1\le i\le n)$, the difference $X(i)-X(i-1)$ can be represented by one single digit with a weight of $r^{-(i-q-1)}$, when the value of k is chosen as that in the corollary of lemma 1.

Lemma 2: Let $X(i-1)=f_R(P(i-1))$ and $X(i)=f_R(P(i))$. If $|X(i)-X(i-1)|<r^{-(i-q-2)}$, then $(X(i)-X(i-1))$ equals to either $x_i.r^{-(i-q-1)}$ or $[-sign(P).r+x_i].r^{-(i-q-1)}$ and $x_i\ne0$ in the latter case.

Proof: Given $|X(i)-X(i-1)| < r^{-(i-q-2)}$, it follows that

$$-x_i.r^{-(i-q-1)} - r^{-(i-q-2)} < X(i)-X(i-1) - x_i.r^{-(i-q-1)} < r^{-(i-q-2)} - x_i.r^{-(i-q-1)}$$

Since $sign(x_i)=sign(P)$ (a property of the function f_R) and $|x_i|<(r-1)$, the above formula can be transformed to

$$-2r.r^{-(i-q-2)} + r^{-(i-q-1)} < X(i)-X(i-1) - x_i.r^{-(i-q-1)} < r^{-(i-q-2)} \quad \text{if } P>0;$$

$$-r^{-(i-q-2)} < X(i)-X(i-1) - x_i.r^{-(i-q-1)} < 2.r^{-(i-q-2)} - r^{-(i-q-1)} \quad \text{if } P<0. \qquad (3)$$

Obviously, $X(i)-X(i-1) - x_i.r^{-(i-q-1)}$ must be an integral multiple of $r^{-(i-q-2)}$, thus Eq.(3) is equivalent to

$$X(i) - X(i-1) - x_i.r^{-(i-q-1)} = \begin{cases} r^{-(i-q-1)} \\ 0 \end{cases} \quad \text{if } P>0$$

$$X(i) - X(i-1) - x_i.r^{-(i-q-1)} = \begin{cases} r^{-(i-q-1)} \\ 0 \end{cases} \quad \text{if } P<0 \qquad (4)$$

Thus, from Eq.(4) it follows:

$$X(i)-X(i-1) = \begin{cases} x_i.r^{-(i-q-1)} \\ [-sign(P).r+x_i].r^{-(i-q-1)} \end{cases} \qquad (5)$$

Since $sign(x_i)=sign(P)$ and $x_i \neq 0$, it holds $|-sign(P).r+x_i| \leq (r-1)$. Thus, lemma 2 shows that $X(i)-X(i-1)$ can be represented by one single digit $\dot{x}_i$ with a weight of $r^{-(i-q-1)}$, provided that the value k is chosen properly (corollary of lemma 1). This means that the reciprocal can be approximated by using $\dot{x}_i=[X(i)-X(i-1)].r^{-(i-q-1)}$ as its i-th digit. Thus, online reciprocal approximation can be generated by using a table lookup unit.

In the following, the problem of how to determine the value of the i-th digit $\dot{x}_i$ of an online reciprocal approximation is considered.

Lemma 3: Let $\dot{x}_i=X(1)$, and $\dot{x}_i=[X(i)-X(i-1)].r^{-(i-q-1)}$ for $i \geq 2$ be the i-th digit of the online reciprocal approximation of P, then it holds

(1) $$\left| \dot{x}_1\dot{x}_2...\dot{x}_n - \frac{1}{P} \right| \leq \frac{r}{2}.r^{-(n-q)} + r^{-(n+k-2q)}$$

(2) $$\dot{x}_i = \begin{cases} x_i & \text{if } x_{i-1}=x'_{i-1} \\ \overline{x}'_i & \text{if } x_{i-1} \neq x'_{i-1} \end{cases} \quad \text{if } i \geq 2$$

where x_i and x_{i-1} are the i-th and $(i-1)$-th digit of $X(i)=f_R(P(i))$ respectively, and x'_{i-1} is the $(i-1)$-th digit of $X(i-1)=f_R(P(i-1))$. $\overline{x}'_i$ is the radix-r complement of x_i; $\overline{x}_i=-sign(x_i).r+x_i$.

Proof:

(1) From the definition of $\dot{x}_i$, it holds

$$\dot{x}_1\dot{x}_2...\dot{x}_n = X(1)+[X(2)-X(1)]+...+[X(n)-X(n-1)]=X(n)$$

Therefore, from Eq.(2) it follows that statement (1) is true.

(2) Lemma 2 says that $X(i)-X(i-1)$ is equal to either $x_i.r^{-(i-q-1)}$ or $[-sign(P).r+x_i].r^{-(i-q-1)}$. Obviously, $X(i)-X(i-1)=x_i.r^{-(i-q-1)}$ means that $X(i)-X(i-1)-x_i.r^{-(i-q-1)}=0$. Since the signs of the corresponding digits of $X(i)$ and $X(i-1)$ are identical, it holds that $x_j=x_j'$ for all j $(1 \leq j \leq i-1)$, where x_j denotes the j-th digit of $X(i)$ and x_j' denotes the j-th digit of $X(i-1)$.
Now, consider the case $X(i)-X(i-1)=[-sign(P).r+x_i].r^{-(i-q-1)}$. This means that $X(i)-X(i-1)-x_i.r^{-(i-q-1)}=-sign(P).r^{-(i-q-2)}$ (see Eq.(5). It can be proved that this always implies $x_{i-1} \neq x_{i-1}'$. Assume that $x_{i-1}=x_{i-1}'$, then it follows that $X(i)-X(i-1)-x_i.r^{-(i-q-1)}$ is an integral multiple of $r^{-(i-q+3)}$. Since $-sign(P).r^{-(i-q-2)}$ cannot be an integral multiple of $r^{-(i-q+3)}$, the assumption leads to a contradiction with the given condition, meaning that $x_{i-1} \neq x_{i-1}'$ (In fact, there are only two possible digits patterns which satisfies $X(i)-X(i-1)-x_i.r^{-(i-q-1)}=-sign(P).r^{-(i-q-2)}$: $x_{i-1}=x_{i-1}'-1$ or $x_1...x_j0(r-1)..(r-1)$ and $x_1'..x_j'10..0$ with $x_k=x_k'$ for all k $(1 \leq k \leq j)$). So, we conclude that the two cases of the value $X(i)-X(i-1)$ can be distinguished by observing the digits x_{i-1} and x_{i-1}', i.e.,

$$X(i) - X(i-1) = \begin{cases} x_i.r^{-(i-q-1)} & \text{if } x_{i-1}=x'_{i-1} \\ [-sign(P).r+x_i].r^{-(i-q-1)} & \text{if } x_{i-1} \neq x'_{i-1} \end{cases}$$

By definition, it holds $\dot{x}_i=[X(i)-X(i-1)].r^{-(i-q-1)}$ and $\overline{x}_i=-sign(x_i).r+x_i$. Therefore, the i-th digit $\dot{x}_i$ of an online reciprocal approximation equals to

$$\dot{x}_i = \begin{cases} x_i & \text{if } x_{i-1}=x'_{i-1} \\ \overline{x}'_i & \text{if } x_{i-1} \neq x'_{i-1} \end{cases} \qquad i \geq 2$$

◊

Lemma 3 shows that the implementation of the online table lookup method is simple. To generate an online reciprocal approximation of m digits with an error less than $(r/2).r^{-(m-q)}$, it requires a RECIP table having $(m+k)$ digits, $a_1a_2...a_{m+k}$, as its input address. At the i-th iteration, $a_1a_2...a_{i+k} = p_1p_2...p_{i+k}$ and $a_j=0$ for all j $(i+1 \leq j \leq m+k)$ is used as the input address. The RECIP table then gives the value of two digits x_{i-1} and x_i. The i-th online digit $\dot{x}_i$ is determined by these two digits and the digit x_{i-1}' (kept in a register) of the previous iteration.

The size of the RECIP table can be reduced if the redundant number $P(i)$ is being transformed into a non-redundant sign-magnitude number representation $P^*(i)$, and using $P^*(i)$ as the input address. The sign of $P^*(i)$ is determined by the sign of the first digit of P. When the radix is small, the reduction of the size of the RECIP table can be significant. For example, for $r=2$, the number of input bit can be reduced by half, since a radix-2 redundant digit requires 2 bits.

The transformation can be done by using a radix-r digit count-down counter. The sign of P $(sign(P) \equiv sign(P^*(i)))$ is stored separate from the magnitude of $P^*(i)$. The magnitude of $P^*(i)$ is stored in the radix-r counter. Denote the most significant digit (MSD) as the first digit of the counter, the next to the MSD as the second, etc. To start the online reciprocal computation, the counter is cleared. The counter is implemented as a radix-r counter which can perform count-down/load operations at different digit positions. At the j-th time step, if $sign(p_j)=sign(P)$ then p_j is counted down from the j-th digit of the counter, otherwise, p_j is loaded into the j-th digit in the counter.

The realization of an online table lookup reciprocal unit is shown in Fig.2.

Another method is on-the-fly conversion, proposed in [ERCE85]. Here, no borrow propagation is necessary.

3. Online division with an adapted Newton-Raphson method

In the previous section, an online table lookup method for reciprocal approximation has been proposed. It has been shown that using a so called RECIP table, a low online delay (the lowest online delay for division known) is obtained. However, the size of the RECIP table grows exponentially with the word-length, e.g., to obtain an accuracy of 2^{-24}, a memory of size $16M.\lceil 24/r \rceil$ bit is required to store the RECIP table. This is impractical. So, we will introduce an adapted Newton-Raphson method in addition to the method of Section II. In the initialization phase, the online table lookup method is applied to generate the first m digits; then the adapted Newton-Raphson method is applied to produce the remaining m-n digits (n=number of digits of the mantissa of a floating point number). Like the online table lookup method, the adapted Newton-Raphson method generates one digit of the online result each iteration.

The iterative Newton-Raphson equation for the reciprocal approximation can be written as $X(i+1)=X(i)*[2-X(i)*P]$, where $X(i)$ is the i-th approximation to the reciprocal $(1/P)$. The Newton-Raphson method converges quadratically. Many online algorithms are based on the Continued Sums/Products principle in a converted form ([DELU70],[KUCK78],[OWEN79]), the Continued Sums/Products method is a linear method. The algorithm to be considered uses the quadratic iterative Newton-Raphson method. Although online results converges linearly (one digit more accurate each iteration), but it will be shown that using a converted faster (quadratic) method reduces the online delay and simplifies the digit selection unit.

The following notations are used: $\Delta x_i=x_i.r^{-(i-q-1)}$ and $\Delta p_i=p_{i+\delta}.r^{-(i+\delta)}$. The value of δ must satisfy the convergence and online requirements of both the online table lookup method and adapted Newton-Raphson method, so $\delta \geq k$ (k is given in the corollary of lemma 1).

1. *Initialization:*

$X(0)=0$;
for $i=1$ to m do
  generates the i-th digit Δx_i with the
  online table lookup method,
  $X(i)=X(i-1)+\Delta x_i$;
  $R(i)=[X(i-1)+\Delta x_i]*\{1-[X(i-1)+\Delta x_i]*[P(i-1)+\Delta p_i]\}$;
od;

2. *Adapted Newton-Raphson iterations:*

select a new online digit: $\Delta x_{m+1}=SEL(R(m))$;
for $i=(m+1)$ to $(n-m)$ do
  $X(i)=X(i-1)+\Delta x_i$;
  $R(i)=[X(i-1)+\Delta x_i]*\{1-[X(i-1)+\Delta x_i]*[P(i-1)+\Delta p_i]\}$;
  select a new online digit: $\Delta x_{i+1}=SEL(R(i))$;
od;

where $SEL(R(i))$ is the selection function; $SEL(R(i))$ equals to the result of $R'(i)+(r/2).r^{-(i-q+1)}$ rounded to the digit with a weight of $r^{-(i-q)}$, with $R'(i)$ represents the L most significant digits which are fully propagated from the $(i+1)$-th to the $(i+L)$-th digit of $R(i)$, i.e., the value $R(i)$ is first truncated to L digits to avoid full carry propagation, then it is symmetrically rounded to the $(i+1)$-th digit (the digit with a weight of $r^{-(i-q)}$). As it will be proved later on, if the parameters m, t and L are chosen properly, $|SEL(R(i))|$ is less than $(r-1/2).r^{-(i-q)}$, so Δx_{i+1} can be represented by one single digit with a weight of $r^{-(i-q)}$.

In the following, the convergence of the adapted Newton-Raphson method and the online property of the method will be proved.

Let $X(i)$ be an approximation to $1/P$ in the i-th iteration, then the Newton-Raphson iteration equation for reciprocals can be written as $X(i)=X(i-1)*(2-X(i-1)*P)$. However, in online computation, not all digits of the input operand P are known during the iterations. Let $P(i)=p_1p_2..p_{i+t}$ be the i-th approximation to P, then the adapted Newton-Raphson iteration equation is $X(i+1)=X(i)*(2-X(i)*P(i))$. Substitution of $X(i)=X(i-1)+\Delta x_i$ leads to $X(i+1)=X(i)+R(i)$. Thus, if we can prove the convergence of the adapted Newton-Raphson method, the used selection function $SEL(R(i))$ will guarantee that $X(i+1)$ is one digit more accurate than $X(i)$. Thereafter, we will prove that $\Delta x_{i+1}=SEL(R(i))$ can be represented by one single digit with a weight of $r^{-(i-q)}$ for all i $(m \leq i \leq n-m)$, i.e., the proposed iteration method is online.

Suppose $X(i)=X(i-1)+\Delta x_i=1/P+c_1.r^{-(i-q)}$ and $P(i)=P(i-1)+\Delta p_i=P+c_2.r^{-(i+t)}$, where $c_1.r^{-(i-q)}$ and $c_2.r^{-(i+t)}$ are the errors of the i-th approximation to $1/P$ and P respectively; t is the online delay of the adapted Newton-Raphson method. Substitution of $X(i)$ and $P(i)$ into the iteration equation, it follows that

$$\begin{aligned} X(i+1) &= X(i).[2-X(i).P(i)] \\ &= \{(1/P)+c_1.r^{-(i-q)}\}.\{2-\{(1/P)+c_1.r^{-(i-q)}\}.\{P+c_2.r^{-(i+t)}\}\} \\ &= (1/P)-\Delta.r^{-(i-q)} \end{aligned} \qquad (6)$$

$$\text{where } \Delta=\frac{c_2}{p^2}.r^{-(t+q)}+\frac{2c_1c_2}{P}.r^{-(i+t)}+Pc_1^2.r^{-(i-q)}+c_1^2c_2.r^{-(2i+t-q)}$$

Since it also holds $X(i+1)=X(i)+R(i)$, we have

$$X(i)+R(i) = (1/P)-\Delta.r^{-(i-q)}$$

Let $\Delta x_{i+1}=SEL(R(i))$, then

$$X(i)+\Delta x_{i+1}+(R(i)-\Delta x_{i+1}) = (1/P)-\Delta.r^{-(i-q)}$$

such that

$$|X(i)+\Delta x_{i+1}-(1/P)| \leq |\Delta|.r^{-(i-q)} + |R(i)-\Delta x_{i+1}| \qquad (7)$$

From the definition of the selection function SEL, it follows that $|R(i)-\Delta x_{i+1}| \leq (r/2).r^{-(i-q+1)}+r^{-(i-q+L-1)}$, where $(r/2).r^{-(i-q+1)}$ is the rounding error (if truncation instead of rounding is applied, a larger error will result which is equal to $r.r^{-(i-q+1)}$), and $r^{-(i-q+L-1)}$ is the truncation error as a result of taking only the digits with a weight larger than $r^{-(i-q+L-1)}$ into consideration by the selection at the i-th iteration. The value of Δ decreases as t and i increases, therefore, the $(i+1)$-th approximation $X(i+1)=X(i)+\Delta x_{i+1}$ will have an error less than $r^{-(i-q)}$, if the values of the parameter m $(i \geq m+1)$, t and L are large enough. Thus, the convergence of the method of selecting Δx_{i+1} by means of the selection function $SEL(R(i))$ has been proved.

In the following, the question of if the approximation generated by the adapted Newton-Raphson method can be made online is considered.

The output $X(i+1)$ is online if Δx_{i+1} can be represented by one single digit with a weight of $r^{-(i-q)}$. We require that $|R(i)| < r^{-(i-q)}-(r/2).r^{-(i-q+1)}=(r-1/2).r^{-(i-q)}$, then it holds that $\Delta x_{i+1}<r.r^{-(i-q)}$, this means that Δx_{i+1} can be represented by one single digit with a weight of $r^{-(i-q)}$. Another consequence of $|R(i)| < (r-1/2).r^{-(i-q)}$ is that only L digits of $R(i)$ have to be taken into consideration by the selection function SEL, since the digits of $R(i)$ from the 1-st through the i-th digit are equal to zero.

Let $X(i)=1/P+c_1.r^{-(i-q)}$, and $X(i+1)=1/P-\Delta.r^{-(i-q)}$ (see Eq.(6)), then $R(i)=X(i)[1-X(i)P(i)]=X(i+1)-X(i)= -\Delta.r^{-(i-q)} -c_1.r^{-(i-q)}$.

Therefore,

$$|R(i)| \le r^{-(i-q)}[|\Delta| + |c_1|] \qquad (8)$$

So, it can be concluded that the output $X(i+1)$ is online if $|\Delta| + |c_1| < r-0.5$.

From Eq.(7) it follows $|c_1| \le |\Delta| + 1/2 + r^{-(L-1)}$. Since $|c_2| < 1$ and $r^{-q} \le |P| < 1$, from Eq.(6) it follows that

$$|\Delta| \le r^{-(t-q)} + 2c_1.r^{-(i+t-q)} + (c_1)^2.r^{-(i-q)} + (c_1)^2.r^{-(2i+t-q)}$$

Furthermore, it holds $i \ge (m+1)$, such that

$$|\Delta| \le r^{-(t-q)} + 2c_1.r^{-(m+t-q+1)} + (c_1)^2.r^{-(m+1-q)} + (c_1)^2.r^{-(2m+t-q+2)} \qquad (9)$$

The aim is to obtain a minimal online delay t. This is an optimization problem: minimize t under the constraint $|\Delta| + |c_1| < r-0.5$ with $|c_1| \le |\Delta| + 1/2 + r^{-(L-1)}$ and Eq.(9).

The above optimization problem is not easy to solve analytically. However, t can be evaluated in a simpler way. Substitution of $|c_1| \le |\Delta| + 1/2 + r^{-(L-1)}$ into $|\Delta| + |c_1| < r-0.5$, it follows that $2.|\Delta| + 1/2 + r^{-(L-1)} < r-0.5$ is a sufficient condition to ensure the output $X(i+1)$ being online. Furthermore, it holds $|c_1| < r-0.5$, substitution of $c_1 = r-0.5$ into the equation of Δ, the expression $2.|\Delta| + 1/2 + r^{-(L-1)} < r-0.5$ becomes a function of parameters t, m and L.

When the values of the parameters t, m and L are chosen properly, the online condition can be satisfied. Table 1 gives several of such values of these parameters as a function of the radix r. It can be easily verified that the value of t in Table 1 is the minimal possible value which still satisfies $|\Delta| + |c_1| < r-0.5$, the values of m and L are lower bounds corresponding to the value of t. In practice, a table lookup initialization system will be restricted to $q<3$, otherwise too large a tables will be necessary.

The proposed online division algorithm uses the online table lookup method to generate the first m digits, and uses the adapted Newton-Raphson method to generate the remaining $(n-m)$ digits. For normalized numbers, the result produced by our algorithm is between 1 and r^1, so if the "decimal-point" (i.e., before the digit with a weight of r^{-1}) is set before the first digit $\dot{x}_1$ of the result, the result is quasi-normalized (of course, the exponent must be increased by 1). For quasi-normalized numbers, $1 < (1/P) \le r^2$. Assuming the "decimal-point" is set before $\dot{x}_1$ (increasing the exponent of the result by 3), then the range of the result becomes between r^{-3} and r^{-1}. With the following quasi-normalization procedure, the online result can be transformed to quasi-normalized numbers.

	t	m	L
r=2	q+2	4+q	3
	q+2	3+q	4
r=3	q+1	2+q	2
r≥4	q	2+q	2

Table I. Several values of the parameters t, m, and L , for which the online condition is satisfied.

Quasi-normalization: Consider the first two digits $\dot{x}_1$ and $\dot{x}_2$. If $\dot{x}_1 = 0$, then right shift the "decimal-point" one position (i.e., decreasing the exponent by 1); if $\dot{x}_1 = 1$, $\dot{x}_2 \ne 0$ and $sign(\dot{x}_1) \ne sign(\dot{x}_2)$, then set $\dot{x}_1 = 0$, complement digit $\dot{x}_2$ and right shift the "decimal-point" one position. In all other cases, the result is already quasi-normalized. The quasi-normalization introduces an extra online delay of 1 digit.

Therefore, the proposed online algorithm gives quasi-normalized floating-point numbers as results. The online delays of the algorithm are $\delta = MAX\{k,t\}$ and $\delta = MAX\{k,t\}$ to $MAX\{k,t\}+1$ for normalized and quasi-normalized inputs, respectively. The values of t given in Table 1 are equal to the corresponding values of k given by the corollary of lemma 1. This means that t can be taken equal to k resulting in a perfect match between the two phases of the algorithm. Thus, for radix-2 numbers the online delays are $\delta = 3$ *and* $\delta = 4$ to 5 for normalized and quasi-normalized inputs respectively. The online delay for radix-r redundant numbers with $r \ge 4$ are $\delta = 1$ and $\delta = 2$ to 3 for normalized and quasi-normalized floating-point inputs, respectively. Among the known online division approaches, the online delay of our algorithm is minimal (the multiplication after the reciprocal evaluation has an online delay of 1 ([TRIV77])). For example, the online delay of the algorithm described in [OWEN81] is $\delta = 4$ and $\delta = 3$ with normalized radix-4 and radix-8 redundant numbers respectively as inputs, and $\delta = 4$ for the online division algorithm described in [TRIV78] for normalized higher radix redundant numbers. Online division algorithms dealing with quasi-normalized numbers have not been reported so far. With the online table lookup principle, the theoretical minimal online delay can be obtained.

4. Implementation of an online reciprocal unit

In the previous sections, an online reciprocal algorithm has been proposed. The algorithm consists of two phases: 1. initialization,by using the online table lookup method to generate the first m digits; 2. adapted Newton-Raphson iteration, by using the adapted Newton-Raphson method to produce the remaining $(m+1)$-th through the n-th digits of the reciprocal.

As has been shown in Section II, the implementation of the online table lookup unit is very simple, it consists of mainly a ROM to store the RECIP table. In this section, the implementation of the adapted Newton-Raphson method is considered.

The formula for the residue $R(i)$ can be written as a recursive equation:

$$R(i) = R(i-1) - 2.X(i-1).P(i-1).\Delta x_i + \Delta x_i.X(i-1).X(i-1).\Delta p_i - 2.X(i-1).\Delta x_i.\Delta p_i - P(i-1).\Delta x_i.\Delta x_i - \Delta x_i.\Delta x_i.\Delta p_i \qquad (10)$$

Since $|R(i)| < (r-1/2).r^{-(i-q)}$, $R(i)$ can be scaled up by a factor r during each iteration, resulting that the most significant digit of $R(i)$ remains in the same position. This simplifies the selection unit. Substitution of $R_s(i) = R(i).r^{-(i-q-1)}$, $\Delta x_i = x_i.r^{-(i-q-1)}$ and $\Delta p_i = p_{i+\delta}.r^{-(i+\delta)}$ into Eq.(10), results in

$$R_s(i) = r.R_s(i-1) - 2.X(i-1).P(i-1).x_i + x_i.X(i-1).X(i-1).p_{i+\delta}.r^{-(\delta+q+1)} - 2.X(i-1).x_i.p_{i+\delta}.r^{-(i+\delta)} - P(i-1).(x_i)^2.r^{-(i-q-1)} - (x_i)^2.p_{i+\delta}.r^{-(2i+\delta-q-1)} \qquad (11)$$

At first sight, Eq.(11) seems quite complicated. However, it can be partitioned into three independent sub-operations which can be implemented in a carry save adder way (i.e., only neighboring carry-propagation during addition and multiplication). The three sub-operations are: 1. the multiplication of $X(i)*P(i)$; 2. the multiplication of $X(i)*X(i)$; 3. the calculation of $R_s(i)$. The block diagrams of these three components are shown in Fig. 3.

If the terms $2X(i-1).x_i.p_{i+\delta}$ and $P(i-1).(x_i)^2.r^{-(i-q-1)}$ are computed with a parallel shift unit, a complex circuitry will result in addition to a large delay in the circuit. To avoid this, the terms $X(i-1)$ and $P(i-1)$ are shifted digit-serially, i.e., each iteration $X(i-1)$ and $P(i-1)$ are shifted right one digit position (equivalent to the multiplication with r^{-1}), such that $X(i-1).r^{-(i+\delta)}$ and $P(i-1).r^{-(i-q-1)}$ can be provided to the digit-multipliers in Fig. 3 (the constant factors $r^{-\delta}$ and $r^{(q+1)}$ can implemented by a fixed offset of wires).

A number of online division algorithms described in the literature demand a complicated digit selection mechanism, but the selection unit for the adapted Newton-Raphson online reciprocal algorithm is simple and fast. As is depicted in Fig. 4, the most significant L digits of $R(i)$ have to be fed to a full propagation adder *(FPA)*, then only the most significant two digits are being used to determine the online digit Δx_{i+1}. For r≥3, it holds that $L=2$, so, full propagation of these 2 digits needs not actually be implemented, since the digit selection logic can be realized with a few gates.

The calculation of $R_s(i)$ requires the longest time among the three components, so the maximal delay per iteration is equal to the time of calculating $R_s(i)$. The delay per iteration is equal to the sum of the delays of *2* digit-multipliers, *2* serial adders and *1* selection operation. Since multiplications and additions are done in a carry save way, the circuit delay is small and high computation speed can be obtained.

5. Conclusions

An algorithm for online division by means of reciprocal evaluation has been described. The algorithm works correctly for maximally redundant floating-point numbers with arbitrary radix. In contrast to other known algorithms, normalized, quasi-normalized, and pseudo-normalized floating point numbers can be handled. For chained online computations, both normalized and quasi-normalized floating-point numbers must be handled; the proposed algorithm meets this criterion. The online delay of the algorithm is the smallest among the known online division algorithms. With the online table lookup method, the theoretical minimum of online delay can be obtained. Using the online table lookup method for the initialization, the adapted Newton-Raphson method perfectly matches this online delay. It has been shown that the digit selection mechanism is simple and fast, and the proposed algorithm can be implemented with fast logic without the need of full carry propagation.

6. References

[TRIV77] K.D. Trivedi and M.D. Ercegovac, "On-line algorithms for division and multiplication", *IEEE Trans. on Computers*, Vol. C-27, no. 7, July 1977.

[ERCE84] M.D. Ercegovac, "An online arithmetic: an overview", SPIE Vol. 495, *Real Time Signal Processing VII*, 1984.

[IRWI78] M.J. Irwin, "A pipelined processing unit for on-line division", *Proceedings 5-th Symposium on Computer Arithmetic*, 1978.

[OWEN81] R.M. Owens, "Compound algorithms for digit online arithmetic", *Proceedings 5-th Symposium on Computer Arithmetic*, 1981.

[OWEN80] R.M. Owens, "Digit online algorithms for pipelined architectures", *Ph.D. Thesis*, Dept. of Computer Science, The Pennsylvania State University, 1980.

[WATA81] O. Watanuki and M.D. Ercegovac, "Floating-point online arithmetic algorithms", *Proceedings 5-th Symposium on Computer Arithmetic*, 1981.

[WATA83] O. Watanuki and M.D. Ercegovac, "Error analysis of certain floating-point on-line algorithms", *IEEE Transactions on Computers*, Vol. C-32, no. 4, April 1983.

[TRIV78] K.D. Trivedi and J.G. Rusnak, "High radix on-line division", *Proceedings 4-th Symposium on Computer Arithmetic*, 1978.

[KUCK78] D.J. Kuck, *The structure of computers and computations*, Vol. I, Wiley and Sons, Inc, 1978.

[OWEN79] R.M. Owens and M.J. Irwin, "On-line algorithms for the design of pipelined architectures", *Proceedings of the Sixth Annual Symposium of Computer Architecture*, Philadelphia, PA, April 1979.

[DELU70] B.G. Delugish, "A class of algorithms for automatic evaluation of certain elementary functions in a binary computer", *Ph.D. Thesis*, Dept. of Computer Science, University of Illenois, 1970.

[ERCE85] M.D. Ercegovac and T. Lang, "On-the-fly conversion of redundant into conventional representation, *UCLA Computer Science Department Technical Report*, August 1985

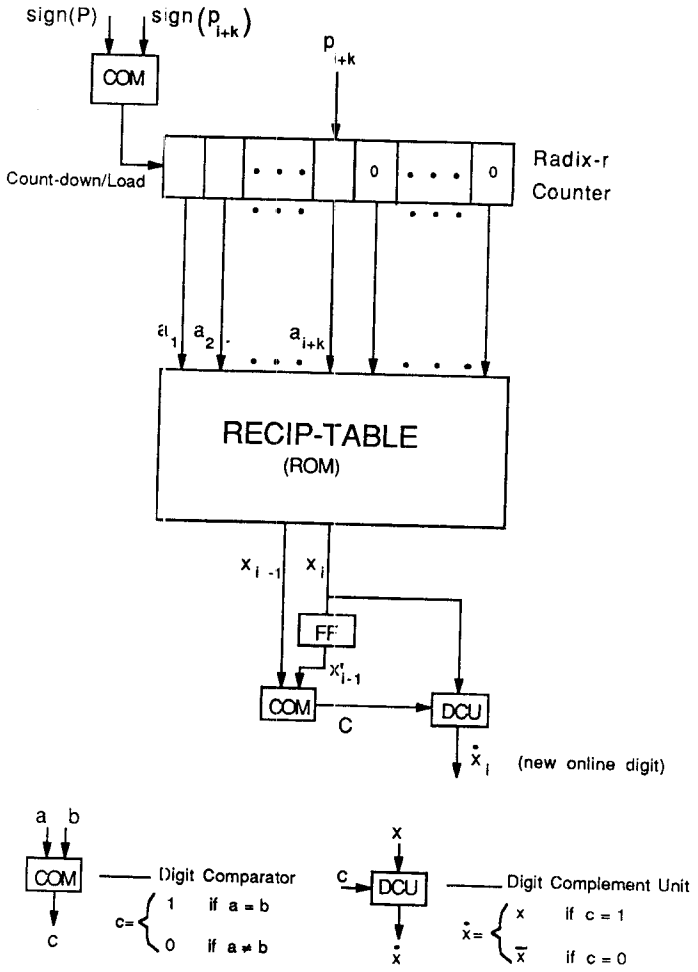

Fig.2 Block diagram of an online table lookup reciprocal unit.

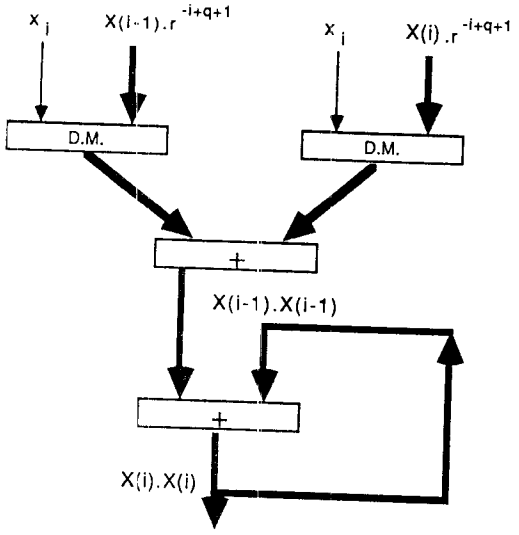

Fig.3a. The multiplier for X(i)X(i). DM is a digit multiplier which operates in carry save mode (without carry propagation).

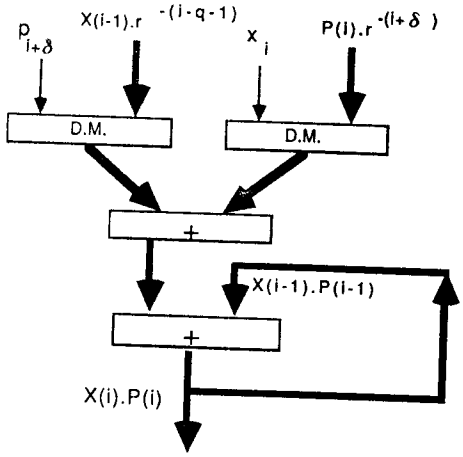

Fig.3b. The multiplier for X(i).P(i).

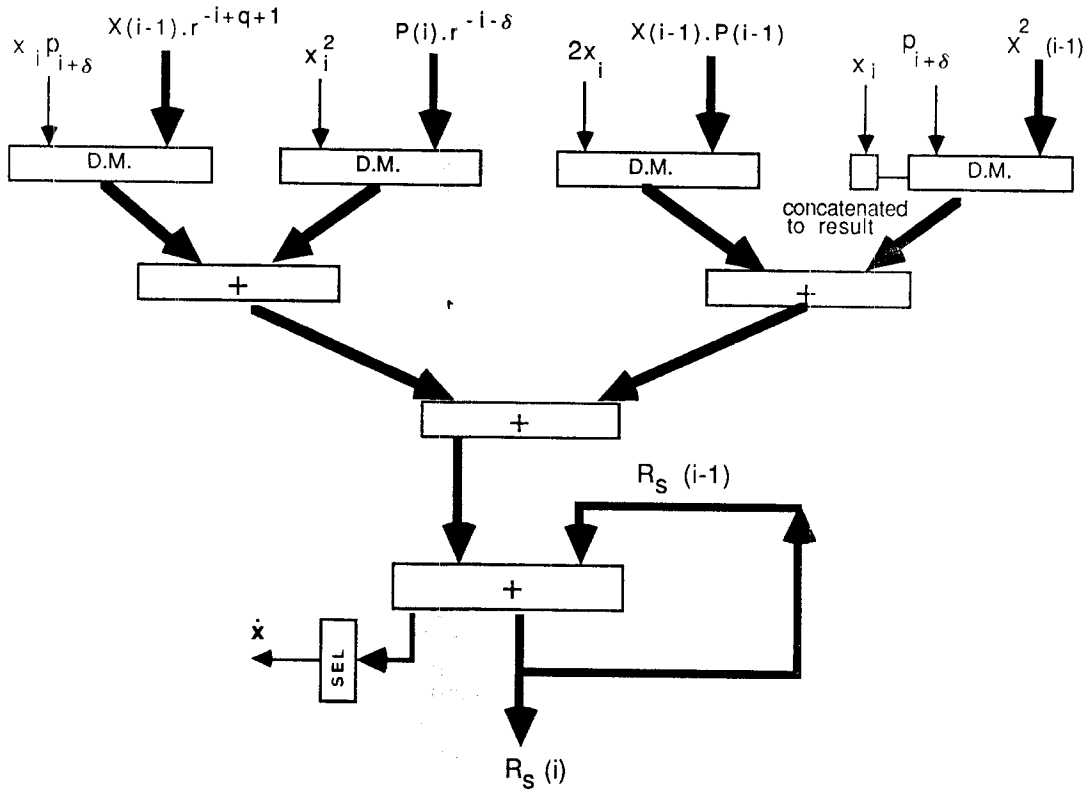

Fig.3c. Caculating the residue $R_s(i)$.

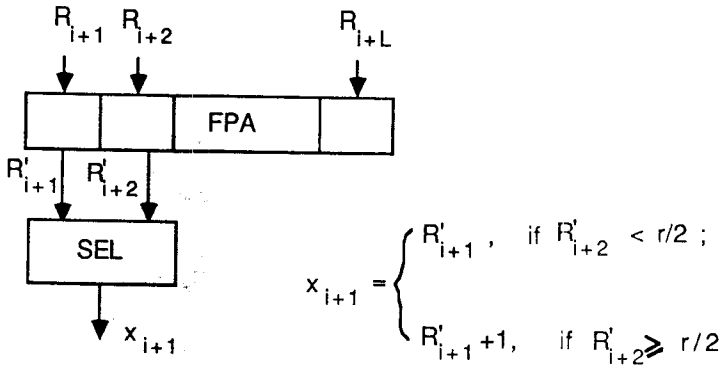

Fig.4 A sketch of the selection unit for the adapted Newton-Raphson method.