On Infinitely Precise Rounding for Division, Square Root, Reciprocal and Square Root Reciprocal

Cristina Iordache and David W. Matula
Dept. of Computer Science and Engineering
Southern Methodist University
Dallas, Texas 75275
Email: cristina, matula@seas.smu.edu

Abstract

Quotients, reciprocals, square roots and square root reciprocals all have the property that infinitely precise p-bit rounded results for p-bit input operands can be obtained from approximate results of bounded accuracy. We investigate lower bounds on the number of bits of an approximation accurate to a unit in the last place sufficient to guarantee that correct round and sticky bits can be determined. Known lower bounds for quotients and square roots are given and/or sharpened, and a new lower bound for root reciprocals is proved. Specifically for reciprocals, quotients and square roots, tight bounds of order $2p + O(1)$ are presented. For infinitely precise rounding of the root reciprocal a lower bound can be found at $3p + O(1)$, but exhaustive testing for small sizes of the operand suggests that in practice $(2+\epsilon)p$ for small ϵ is usually sufficient. Algorithms can be designed for obtaining the round and sticky bits based on the bit pattern of an approximation computed to the required accuracy. We show that some improvement of the known lower bound for reciprocals and division is achievable at the cost of somewhat more complex hardware for rounding. Tests for the exactness of the quotient and square root are also provided.

1. Introduction and Summary

Finite precision computation restricts the domain of values (operands and results of operations) to a discrete range. The possible values can be described in terms of the *base* (β) used, the maximum length of the *significand* and the range of the *exponent* as an *fp-factorization* (see [4]):

$$v = (-1)^s \beta^e f$$

where $s \in \{0, 1\}$ is the *sign bit*, e is the exponent (an integer usually in a finite range) and f is the *significand factor*, given as a base β polynomial $f = \sum_{i=0}^{p-1} d_i[\beta]^i$, with $d_0 \neq 0$ for normalized significands. For IEEE 754 standard floating point $\beta = 2$ [1].

Function evaluation in finite precision requires rounding to a value in the finite precision range. Let $R = (-1)^s 2^e$ $1.b_1 b_2 \ldots b_{p-1} b_p b_{p+1} \ldots$ be the infinitely precise value of the function evaluated. The bits b_i of the infinitely precise binary representation of R are unique and referred to as *correct* bits of R. When R falls between two points of the discrete range, one of them is selected based on the rounding mode and the values of the round bit $r = b_p$ and the sticky bit $s = b_{p+1}$ OR b_{p+2} OR $\ldots$.

Infinitely precise rounding is most difficult when the result is at or near exact p-bit points or midpoints of the p-bit binary discrete range, since very small errors can move the approximation into another interval and change the rounding bits. Especially difficult are the cases when the infinitely precise result is exact at p or $p + 1$ bits, since any error will change the rounding bits. The accuracy required for infinitely precise rounding depends on the distances between infinitely precise results of the function and $p + 1$ bit binary values. The existence of provably longest runs of 0's and 1's after the round bit provide a lower bound on these distances for the difficult cases (exact points and midpoints of the finite range), and thus an upper bound on the required minimum accuracy. This upper bound can be relaxed a little if a different mechanism is used to handle these difficult cases, as will be shown, for example, in the quotient rounding section. The main theoretical results of this paper are summarized in Table 1.

When the accuracy used is not sufficient to guarantee infinitely precise rounding, random or directed testing has been investigated to detect incorrectly rounded cases [2], [3].

Although infinitely precise rounding is possible when the error direction is not known, provided that the accuracy is sufficient, we assume in most of the the following that the

Function	round bit=0		round bit=1		Min. abs. error on the fraction required	At midpoint or exact possible
	Longest run of 0s	Longest run of 1s	Longest run of 0s	Longest run of 1s		
Quotient	$= p-2$	$= p-1$	$= p-1$	$= p-2$	$3 \cdot 2^{-2p}$	N/Y
Reciprocal	$\geq p-3$	$\leq p-1$	$= p-1$	$\leq p-2$	$3 \cdot 2^{-2p}$	N/N
Root Reciprocal	$\leq 2p-1$	$\leq 2p-1$	$\leq 2p-1$	$\leq 2p-1$	2^{-3p+1}	N/N
Square Root	$\leq p-3$	$= p+1$	$\leq p-1$	$= p-1$	$1.25 \cdot 2^{-2p}$	N/Y

Table 1. Structure of non-exact infinitely precise results (exactness considered for nontrivial arguments) and accuracy required for infinitely precise rounding

error is negative. This is a reasonable assumption, since widely used iterative algorithms such as the convergent algorithm and Newton-Raphson guarantee the error direction. Directed error also simplifies our analysis and to a certain extent the hardware implementation.

To save space, some proofs have been omitted here and can be found in [8].

2. Infinitely Precise Rounding of the Quotient

It is known [2] that infinitely precise rounding of the quotient requires accuracy of more than twice the size p of the operands, or equivalently absolute error on the fraction below 2^{-2p}, when the error is directed and the round and sticky bits are directly retrieved from the bits of the approximation. We will show in this section how quotient subestimates can be infinitely precisely rounded when the absolute error does not exceed 3×2^{-2p}.

Without loss of generality, we will assume that D (the $p-$bit divisor) and Q (the quotient) are in the fundamental binade range $[1, 2)$, which means the $p-$bit dividend N is in the range $[D, 2D) \subset [1, 4)$. A practical implementation would divide the significands of N and D (both in the binade $[1, 2)$) and obtain a quotient in the two binade range $(\frac{1}{2}, 2)$ that can be later normalized. Both representations are equivalent and cover all possible values of N and D by simple manipulation of the exponent values.

Throughout the following analysis, we will denote by Q_{p-1} and Q_p the infinitely precise rational quotient N/D in binary representation truncated to p bits $(1.q_1 q_2 \ldots q_{p-1})$ and $p+1$ bits, respectively.

2.1. Properties of the Infinitely Precise Quotient

In summary, the first four observations show that if $Q = N/D$ is an infinitely precise quotient of two p-bit values, then:

- After a round bit $q_p = 0$, at most $p-2$ 0's can follow (unless Q is an exact quotient)
- A run of at most $p-1$ 0's can follow a round bit of 1, also meaning that the sticky bit must be 1

Observation 1 *Q can have a run of $p-1$ 0's after the round bit and a sticky bit of 1. The sticky bit of Q cannot be determined correctly from less than p leading bits of the infinite tail used to define the sticky bit. (i.e. bits $q_{p+1} \ldots q_{2p}$).*

Proof: (will show an example)
Let $d \in [1, \sqrt{5}-1)$ be a value representable by a $\lfloor \frac{p}{2} \rfloor$-bit significand .
Let $D = d + 2^{-p+1}$ and
$Q = 1 + \frac{d}{2} - 2^{-p} + 2^{-(2p-1)} \cdot \frac{1}{D} =$
$= 1.q_1 q_2 \ldots q_{p-1} 1 \underbrace{00\ldots0}_{q_{p+1} \ldots q_{2p-1}} 1xx\ldots.$
Then $Q \cdot D = (1 + \frac{d}{2}) \cdot d + 2^{-p+1}$.
$Q \cdot D \in (\frac{3}{2}, 2)$ is a p-bit value.
□

Observation 2 *A run of p 0's after the round bit cannot occur unless the quotient is exact ($round\_bit = sticky\_bit = 0$). No more than p bits ($q_{p+1} \ldots q_{2p}$) need to be considered for correctly determining the sticky bit of infinite precision quotient Q.*

Proof: Let $Q = Q_p + Q_r$, $Q_r \in [0, 2^{-2p})$ (i.e. all bits $q_{p+1} = q_{p+2} = \ldots = q_{2p} = 0$).
$D \in [1, 2)$, so $Q_r \cdot D < 2^{-(2p-1)}$
$D = 1.d_1 d_2 \ldots d_{p-1} \implies$
$\implies Q_p \cdot D = b_{-1} b_0 . b_1 b_2 \ldots b_{2p-1}$.
Since $Q \cdot D$ is a p-bit value (i.e. $b_m = 0, \forall m > p-1$), Q_r must be 0. This means that the sticky bit (0 in this case) can be correctly determined from bits $q_{p+1} \ldots q_{2p}$ of infinite precision quotient Q. □

Observation 3 *If the round bit $q_p = 1$, then the sticky bit of the infinitely precise quotient Q is also 1. (Q is never at a midpoint).*

Observation 4 *If the round bit $q_p = 0$, then the longest runlength of 0's after the round bit cannot exceed $p-2$ bits, unless Q is an exact quotient.*

In other words, if the round bit $q_p = 0$ and bits $q_{p+1} = q_{p+2} = \ldots = q_{2p-1} = 0$, then the sticky bit of Q is 0.

Proof: $Q = Q_p + Q_r, \qquad Q_r \in [0, 2^{-(2p-1)})$

$$Q \cdot D = \underbrace{Q_{p-1} \cdot D}_{b_{-1}b_0.b_1b_2\ldots b_{2(p-1)}} + \underbrace{Q_r \cdot D}_{<2^{-2(p-1)}}$$

Q cannot be a valid quotient unless $Q_r = 0$ (because $Q \cdot D$ must be a p-bit value). Thus the sticky bit of Q is 0. □

Next, it can be shown that the longest runlength of 1's following a round bit of 0 is $p-1$. This makes it possible to determine the correct rounding bits when the approximation lies on the wrong side of the midpoint and the absolute error does not exceed $2^{-(2p-1)}$.

Observation 5 $(i)\ Q_1 = Q_{p-1} + 2^{-p} - 2^{-(2p-1)} + Q_{r1} = 1.q_1 \ldots q_{p-1}0\ \underbrace{11\ldots1}_{p+1\ldots2p-1}\ 0\,xxx\ldots\ (Q_{r1} < 2^{-2p})$

is a valid quotient pattern (i.e. a run of $p-1$ 1's after a round bit of 0 can occur).

$(ii)\ 1.q_1q_2\ldots q_{p-1}0\underbrace{11\ldots1}_{p+1\ldots2p}xxxx\ldots$ *is not a valid quotient value. (i.e. p 1's after a round bit of 0 cannot occur).*

Also, a runlength of 1's after a round bit of 1 does not exceed $p-2$ bits:

Observation 6 $1.q_1q_2\ldots q_{p-1}1\ \underbrace{11\ldots1}_{p+1\ldots2p-1}\ x\,xxx\ldots$ *is not a valid quotient value.*

2.2. Recovering from Negative Error

We first show that if the absolute error does not exceed 2^{-2p}, the round bit r and sticky bit s can be obtained from bits $p, p+1, \ldots, 2p$ of the subestimate Q' as follows:

$r = q'_p$, $s = q'_p$ OR q'_{p+1} OR ... OR q'_{2p}.

Observation 7 *When the round and sticky bits of the infinitely precise quotient Q are $q_p = 0$, $s = 1$, they can be correctly recovered as the round and sticky bits of the subestimate Q' if and only if the absolute error is $\alpha < 2^{-2p}(1+\tau)$, where*

$\tau \in (\frac{2^{-p+1}}{1-2^{-p}}, \frac{2^{-\lfloor\frac{p-2}{2}\rfloor}}{1-2^{-\lfloor\frac{p}{2}\rfloor}})$.

Since the longest runlength of 0's after a round bit of 1 does not exceed $p-1$, a negative error of absolute value at most 2^{-2p} will not change the round bit (and also note that the combination $r = 1$, $s = 0$ is not possible for an infinitely precise quotient). From this simple observation and Observation 7 proved above, it is easy to see that **when the absolute error does not exceed 2^{-2p}, the round and sticky bits can be computed as**

$r = q'_p$, $s = q'_p$ OR q'_{p+1} OR ... OR q'_{2p}.

(the round bit is the round bit of the subestimate Q', and the sticky bit is obtained as the OR sum of the round bit and the next p bits following it).

If we allow the absolute error to be as large as $2 \cdot 2^{-2p}$, the correct rounding bits can still be easily determined from the first $2p$ bits of the approximation for all cases except exact result, since not all bit combinations are possible for the p-bit tail $q_{p+1}q_{p+2}\ldots q_{2p}$ of a valid infinitely precise quotient of two p-bit values. The exact result case becomes difficult and requires a special detection test because a run of $p-2$ 0's after the round bit followed by a 1 in position $2p-1$ and then again a 0 is a possible bit pattern for an infinitely precise quotient and when the error is in the range $[-2^{-2p+1}, -2^{-2p})$, bits $p, p+1, \ldots, 2p$ of the estimate become all 0.

Let us solve the very easy cases (result not very close to a midpoint or exact point) with the following observation that can be easily proved:

Observation 8 *If $Q' = Q - \alpha = 1.q'_1\ldots q'_{p-1}q'_p\ldots q'_{2p}\ldots$, $\alpha \in [0, 2^{-(2p-1)}]$ is an estimate to quotient Q and the integer value $S = q'_{p+1}q'_{n+3}\ldots q'_{2p}$ is in the range $[1, 2^p-3]$, or $S = 2^p - 2$ and $q'_p = 0$, then the sticky bit of Q is $s = 1$ and the round bit is q'_p, the round bit of Q'.*

Now, the remaining cases can be solved as follows:

- if the subestimate has $q'_p = 1$ and $q'_{p+1} = q'_{p+2} = \ldots = q'_{2p} = 0$ then according to Observation 3 the true round and sticky bits are $r = s = 1$. 9
- if $q'_p = 0$ and $q'_{p+1} = q'_{p+2} = \ldots = q'_{2p} = 1$ then $r = s = 1$ (since a run of p 1's after the round bit cannot occur, see Observation 5).
- if $q'_p = 0$, $q'_{p+1} = q'_{p+2} = \ldots = q'_{2p-1} = 1$ and $q'_{2p} = 0$ then $r = 0$, $s = 1$ (see Observation 8).
- if $q'_p = 1$ and $q'_{p+1} = q'_{p+2} = \ldots = q'_{2p-1} = 1$ then $r = s = 0$ and a unit in the last place must be added to the significand ($r = s = 0$ is the only possibility when the absolute error is limited to 2^{-2p+1}, since the next possible bit pattern is a run of 0's followed by a 1 in position $2p-1$).

The only case that cannot be solved with this method is $q'_p = q'_{p+1} = q'_{p+2} = \ldots = q'_{2p} = 0$.

2.3. Exact Quotient Detection

As shown in the previous subsection, infinitely precise quotient rounding is possible with an estimate accurate to less than $2p$ bits if a test for detecting the exact quotient case is available.

Let us denote by $lnzb(X)$ the position of the last nonzero bit in the significand of X, and by exp_X the exponent in the floating point of X.

So, for $X = 2^{exp_X} 1.x_1 x_2 \dots x_{p-1}$,

$$lnzb(X) = \begin{cases} p-1 & \text{if} \quad x_{p-1} = 1 \\ k\ (\geq 0) & \text{if} \quad x_k = 1, x_{k+1} = \dots = x_{p-1} = 0 \end{cases}$$

Based on the following observation, an exact quotient detection test can be developed:

Observation 9 *Q is an exact quotient of $N, D \in [1,2)$ (i.e. $r = s = 0$, $Q = Q_{p-1} = N/D$) if and only if* $lnzb(Q_{p-1}) + lnzb(D) \leq p - 1 + exp_Q$

For a given $D \in [1,2)$, the distance between two consecutive infinitely precise quotients $Q_1 = \frac{M_1}{D}$, $Q_2 = \frac{M_2}{D}$ is above 2^{-p}:

$|Q_2 - Q_1| = \frac{|M_2 - M_1|}{D} = \frac{2^{-p+1}}{D} \in (2^{-p}, 2^{-p+1}]$.

Thus a p-bit exact quotient Q can always be detected if the absolute error does not exceed 2^{-p}, by rounding up the subestimate $Q' = 2^{exp_Q} 1.q'_1 q'_2 \dots q'_{2p}$ and applying the exact quotient criterion given in Observation 9. However, the maximum absolute error is still limited by the need to determine the correct rounding bits when Q is not a p-bit value.

2.4. Design of a Rounding Algorithm for Absolute Error at most $3 \cdot 2^{-2p}$

Observation 8, rephrased for $\alpha \leq 3 \cdot 2^{-2p}$, still holds for $S = q'_{p+1} q'_{p+2} \dots q'_{2p} \in [1, 2^p - 4]$. Special attention must be given to Q'
$= 1.q'_1 \dots q'_{p-1} q'_p 11 \dots 1 q'_{2p-1} q'_{2p} xxx \dots$, where q'_{2p-1} and q'_{2p} are not both 0. Here are the possible cases:

- $q'_p = 1$, $q'_{2p-1} q'_{2p} = 11$ or 10: Q' is not a valid quotient value, thus set $r = 0$ and s according to the exact quotient test
- $q'_p = 1$, $q'_{2p-1} q'_{2p} = 01$: In this case $rs \neq 01$ (for $\alpha \leq 2^{-2(p-1)}$, see Observation 4). $rs = 00$ or 11 according to the exact quotient test
- $q'_p = 0$, $q'_{2p-1} q'_{2p} = 11$: Q' is not a valid quotient value; the only possibility is $rs = 11$ ($\forall \alpha \leq 2^{-p}$)
- $q'_p = 0$, $q'_{2p-1} q'_{2p} = 01$: $rs = 01$, $\forall \alpha \leq 2^{-2p+1} + 2^{-2p}$

The remaining case, $q'_p = 0$, $q'_{2p-1} q'_{2p} = 10$, needs one more criterion to be resolved for $\alpha > 2^{-2p+1}$. $Q_1 = 1.q_1 \dots q_{p-1} 0 \underbrace{11\dots1}_{p+1\dots2p-1} 0xxx\dots$ can be distinguished from
$Q_2 = 1.q_1 \dots q_{p-1} 1 \underbrace{00\dots0}_{p+1\dots2(p-1)} 10\, xxx \dots$
for $\alpha \leq 2^{-2p+1} + 2^{-2p}$, but not from
$1.q_1 \dots q_{p-1} 1 \underbrace{00\dots0}_{p+1\dots2p-1} 1\, xxx \dots$ (unless the following criterion is used).

Observation 10 *Let $D = 1.d_1 d_2 \dots d_{p-1}$.*
a). If $Q = N/D = 1.q_1 \dots q_{p-1} 0 \underbrace{11\dots1}_{p+1\dots2p-1} 0\, xxx \dots$ is an infinitely precise quotient of two p-bit values, then $q_{p-1} = d_{p-2}$.
b).If $Q = N/D = 1.q_1 \dots q_{p-1} 1 \underbrace{00\dots0}_{p+1\dots2p-1} 1\, xxx \dots$ is an infinitely precise quotient of two p-bit values, then $q_{p-1} \neq d_{p-2}$.

Based on all of the above, an algorithm can be designed for computing quotient Q correctly rounded to p bits, given the first $2p$ significant bits of its subestimate
$Q' = 1.q'_1 q'_2 \dots q'_{p-1} q'_p q'_{p+1} \dots q'_{2p} \dots$, provided that the absolute error is within $2^{-2p+1} + 2^{-2p}$.

3. Determining Round to Nearest Reciprocals

This section focuses on determining the accuracy required for obtaining correctly rounded to nearest reciprocals by simply rounding to nearest an estimate ρ, i.e. $RN(1/y) = RN(\rho)$. We will show that the accuracy requirements (also checked exhaustively for some operand sizes in [7]) are the same as for division, since the worst case (a run of $p-1$ 0's after the round bit) occurs for reciprocals as well.

Let $y = 1.y_1 y_2 \dots y_{p-1}$ be the divisor and (ρ, ϵ) the reciprocal, residue pair such that

$\rho y = 1 + \epsilon$, $|\epsilon| < \frac{1}{2}$.

For $\rho = 0.b_1 b_2 \dots b_q$, $q \geq p+1$, we have
$\epsilon = i \cdot 2^{-(p-1+q)}$ with i an integer. In other words ϵ is an exact low order part of ρy. Also note that

$\epsilon = \frac{\rho - 1/y}{1/y}$ is the relative error for ρ a reciprocal approximation and

$\frac{\epsilon}{y} = \rho - \frac{1}{y}$ is the absolute error (generally a rational number).

These results establish that the rational valued multiplicative inverse $1/1.y_1 y_2 \dots y_{p-1}$ cannot be arbitrarily close to a $p+1$ bit binary value (rounding midpoint) $0.1 b_2 b_3 \dots b_p 1$.

Suppose now that $\rho = 0.b_1 b_2 \dots b_q$ ($q \geq p+1$) is accurate to a unit in the last place (ulp). Our goal is to determine a lower bound N such that if $q \geq N$, and ρ is ulp accurate, then $RN(\frac{1}{y})$ can be determined from ρ. For round-to-nearest, midpoints of the floating point intervals are the

difficult regions and thus a lower bound on the accuracy needed is obtained from the minimum distances between an infinitely precise reciprocal and a midpoint, i.e. the longest runs of 0's after a round bit of 1 and the longest runs of 1's after a round bit of 0.

We already know that an upper bound on the longest run of 0's after a round bit of 1 is $p-1$ (see Observation 2 referring to infinitely precise quotients). This upper bound is still tight for reciprocals:

Observation 11 *A run of $p-1$ 0's after a round bit of 1 can occur for an infinitely precise reciprocal.*

Proof: For $y = 2 - 2^{-p+1}$, we have
$\frac{1}{y} = \frac{1}{2}(1+2^{-p}+2^{-2p}+\ldots) = 0.1\underbrace{00\ldots0}_{2\ldots p}1\underbrace{00\ldots0}_{p+2\ldots 2p}100\ldots$.

After the round bit (position p+1), a run of 0's extends in this case through position $2p$, for a total of $p-1$ positions. □

If $\rho = 0.b_1b_2\ldots b_q$ is affected by error above -2^{-2p-1}, the worst case shown above can be correctly rounded to nearest as $RN(\rho)$ provided that $q \geq 2p+1$. Since the runlengths of 1's after a round bit of 0 are also upper bounded by $p-1$ (see Observation 5 for quotients), an absolute error $\frac{|\epsilon|}{y} < 2^{-2p-1}$ determines the lower bound of $N = 2p+1$ bits needed for the approximation ρ.

Observation 12 *If $\rho = 0.b_1b_2\ldots b_q$ ($q \geq 2p+1$) is an approximation of $\frac{1}{y}$ ($1 \leq y < 2$) and the relative error satisfies $|\epsilon| < 2^{-2p-1}$, then $RN(\frac{1}{y}) = RN(\rho)$.*

Also note that for inputs y very close to 1, or very close to 2, there are simple formulae for RN(1/y) that avoid computing so many accurate digits for ρ.

Observation 13 *Let $y = 1 + i \cdot 2^{-p+1}$. Then for $i < 2^{\frac{p-3}{2}}$, $RN(1/y) = 1 - i \cdot 2^{-p+1}$.*

Proof: Let $\rho = 1 - i \cdot 2^{-p+1}$, then $\epsilon = \rho y - 1 = -i^2 2^{-2p+2}$.

$|\epsilon| < 2^{p-3-2p+2} = 2^{-p-1}$ and the absolute error $\frac{|\epsilon|}{y} < 2^{-p-1}$, so $RN(1/y) = \rho = 1 - i \cdot 2^{-p+1}$. □

Observation 13 implies that if the initial $\lceil \frac{p+1}{2} \rceil$ fraction bits of y are zero (i.e. $y_1y_2\ldots y_{(p/2)+1} = 0$), then the 2's complement of y is the round to nearest reciprocal. This special case rule includes the value $y = 1$ which generally is handled as an exceptional case anyway.

Observation 14 *Let $y = 2 - i \cdot 2^{-p+1}$. Then for $i < 2^{\frac{p-1}{2}}$, $RN(1/y) = \frac{1}{2}(1 + i \cdot 2^{-p})$.*

Proof: Let $\rho = \frac{1}{2}(1 + i \cdot 2^{-p})$, then $\epsilon = \rho y - 1 = -i^2 2^{-2p}$.

$|\epsilon| < 2^{p-1-2p} = 2^{-p-1}$ and the absolute error $\frac{|\epsilon|}{y} < 2^{-p-1}$, so $RN(1/y) = \rho = \frac{1}{2}(1 + i \cdot 2^{-p})$. □

4. Infinitely Precise Rounding: Square Root Reciprocal

Our discussion will be limited without loss of generality to arguments in the two binade range $[1, 4)$, since the exponent can be easily computed. Also, since Newton Raphson or convergent iterations are commonly used to refine a root reciprocal approximation and they both subestimate the result, we shall only consider the problem of determining the correct rounding bits when the error is guaranteed to be negative. However, our analysis of the structure of an infinitely precise root reciprocal can be applied to develop rounding algorithms for other cases as well.

Some notations that will be used in the following are:

$Q = \frac{2}{\sqrt{A}} = Q_p + Q_r = 1.q_1\ldots q_{p-1}q_p + Q_r \in (1, 2)$ ($A \in (1, 4)$) is the scaled, infinitely precise result (we will estimate $\frac{2}{\sqrt{A}}$ so that the result falls in the range $(1, 2)$); q_p is the round bit.

$\tilde{Q} = \tilde{Q}_p + \tilde{Q}_r$ ($\tilde{Q} \leq Q$) is an estimation of $\frac{2}{\sqrt{A}}$.

4.1. Bit Structure of an Infinitely Precise Result

Observation 15 *For any $A \in (1, 4)$, the sticky bit of $\frac{1}{\sqrt{A}}$ is 1.*

Proof: Assume that for some $A \in (1, 4)$, the sticky bit of $\frac{1}{\sqrt{A}}$ is 0, i.e. $Q_r = 0$, $2\frac{1}{\sqrt{A}} = Q_p$.

Then $(A \cdot 2^{p-1}) \cdot (Q_p \cdot 2^p)^2 = 2^{3p+1}$.

$A \cdot 2^{p-1}$ and $Q_p \cdot 2^p$ are both integers, thus they must be both powers of 2. $A \cdot 2^{p-1} \in (2^{p-1}, 2^{p+1})$, $Q_p \cdot 2^p \in [2^p, 2^{p+1})$; the only powers of 2 in these ranges are $A = 2^p$, $Q_{p+1} = 2^p$, but they do not satisfy the equation above: $2^p \cdot (2^p)^2 = 2^{3p} \neq 2^{3p+1}$; the assumption made was false. □

Thus $\frac{1}{\sqrt{A}}$ can have a sticky bit of 0 only for $A = 1$.

The minimum accuracy needed for an estimation so it can be correctly rounded will be determined by the longest runs of 0's or 1's that can be encountered after the round bit in an exact result. The following upper bound can be easily proven:

Observation 16 *The longest runlengths of 0's or 1's after the round bit cannot exceed $2p+1$.*

Proof:

$Q = Q_p + Q_r$; long runlengths are determined by a small, positive (0's) or negative (1's) Q_r.

$A \cdot Q^2 = 4 \Longrightarrow A \cdot Q_p^2 + (2Q_p + Q_r)A \cdot Q_r = 4$.

Let $f(A) = (2Q_p + Q_r) \cdot A = (\frac{4}{\sqrt{A}} - Q_r)A = 4\sqrt{A} - A \cdot Q_r$.

$f(A)$ can be shown to be monotonically increasing:

$4\sqrt{A+d} > 4\sqrt{A}(1+\frac{d}{2A}-\frac{d^2}{8A^2})$;

$d = \begin{cases} 2^{-p+1}, & A \in (1,2) \\ 2^{-p+2}, & A \in [2,4) \end{cases}$

$\Longrightarrow 4\sqrt{A+d} - 4\sqrt{A} > A \cdot 2^{-p}$.

Thus $f(A) \le f(4-2^{-p+2}) = 8\sqrt{1-2^{-p}} - A \cdot Q_r < 8 - 2^{-p+2} - A \cdot Q_r < 8$ (even for Q_r negative, $A \cdot Q_r > -4 \cdot 2^{-p} = -2^{-p+2}$).

$8 \cdot |Q_r| > f(A) \cdot |Q_r| = 4 - A \cdot Q_p^2 > 0$.

$A = 1.a_1 \ldots a_{p-1}$, $Q_p = 1.q_1 \ldots q_{p-1}q_p$ and thus $4 - A \cdot Q_p^2$ must have its last nonzero bit no later than position $p-1+2p = 3p-1$:

$8 \cdot |Q_r| > 2^{-3p+1} \Longrightarrow |Q_r| > 2^{-3p-2}$, i.e. the longest run of 0's or 1's, beginning in position $p+1$, will stop before position $3p+2$. □

In practice, the runs of 0's and 1's are much shorter, but unlike quotients of finite precision values, root reciprocals are irrational numbers and do not have a periodic structure that allows the exact determination of the longest runlength, as a general formula. As can be seen from the following table, the number of bits that must be checked for infinitely precise rounding is a variable step function of p; the longest runlengths are nonmonotonic and highly dependent on p. The accuracy required does not exceed $2p+3$ bits for the values of p listed. This is not surprising, since probabilistic arguments show [5] that about twice the size of the operands is the expected value for the accuracy needed. Observation 16 provides a theoretical bound of $3p+1$ accurate bits needed to guarantee infinitely precise rounding, and trying to lower it is not an easy problem, even for a fixed p. However, we will show that it can still be lowered a little by applying a few theorems from number theory.

Observation 17 *The diophantine equations* $N \cdot Q^2 = 2^{3p+K} \pm 1$ *(*$p > 0$ *integer,* $K \in \{0,1\}$*) have no solutions for* N, Q *integers in the ranges* $2^{p-2} < N < 2^p$, $2^p < Q < 2^{p+1}$.

Based on the above, the following can be proven:

Observation 18 *The runlengths of 0's or 1's after the round bit of* $\frac{1}{\sqrt{A}}$ *(*$A \in (1,4)$ *a p-bit value) do not exceed* $2p-1$.

4.2. Recovery of the Correct Round Bit from an Approximation

The sticky bit of $\frac{1}{\sqrt{A}}$ is known to be 1 for $A \in (1,4)$. An approximation $\tilde{Q} = \frac{2}{\sqrt{A}} + \varepsilon$ of $\frac{2}{\sqrt{A}} = Q_p + Q_r$ has the wrong round bit if $|\varepsilon| \ge |Q_r|$ and they have opposite signs.

Thus a straightforward method to ensure infinitely precise rounding is to compute the function (root reciprocal in this case) to an extra number of bits equal to the maximum runlength $+1$. (i.e. $\varepsilon \le 2^{-3p} < Q_r$ in this case).

A little less accuracy is required if the error is directed (its sign is known) and some checking of the bit values after the round bit position is performed. The absolute error should not exceed the minimum distance between two infinitely precise function values with different round bits.

In our case, $\tilde{Q}$ is a subestimate ($\varepsilon \le 0$). The round bit can be ambiguous for $\tilde{Q} = 1.q_1q_2 \ldots q_{p-1}q_p\,11\ldots1xx\ldots$, which could be an approximation for $Q = 1.q_1q_2 \ldots q_{p-1}(q_p+1)\,00\ldots0xx\ldots$.

If the maximum runlengths of 0's and 1's in infinitely precise function values are known (l_0 and l_1), a lower bound for the distance between two values of different round bits is $2^{-p}(2^{-(l_0+1)} + 2^{-(l_1+1)})$, i.e. the distance between $1.q_1q_2 \ldots q_{p-1}q_p\,\underbrace{00\ldots0}_{l_0}100\ldots = Q_p + 2^{-p-l_0-1}$ and $1.q_1q_2 \ldots q_{p-1}(q_p-1)\,\underbrace{11\ldots1}_{l_0}011\ldots < Q_p - 2^{-p-l_1-1}$.

For the root reciprocal, a theoretical bound would thus be $2 \cdot 2^{-3p} = 2^{-(3p-1)}$. See [8] for a simple algorithm based on this bound.

5. Infinitely Precise Rounding for the Square Root

Tight theoretical bounds on the minimum accuracy needed for a correctly rounded square root (also implicitly used in [6]) are not difficult to establish. The following section derives upper bounds on the longest runs of 0's and 1's following the round bit of an infinitely precise square root, then shows examples where the runlengths of 1's reach the upper bounds derived.

Well known iterative refinement methods guarantee the error direction for the square root as well. Newton Raphson overestimates; a subestimate can be obtained by using the convergent method.

The following notations will be used:

$A = a_{-1}a_0.a_1a_2 \ldots a_{p-1}$ a p-bit value in the range $[1,4)$;

$exp(A)$ = the exponent value of A: 0 for $A \in [1,2)$, 1 for $A \in [2,4)$;

$\sqrt{A} = Q_p + Q_r$, $Q_p = 1.q_1 \ldots q_p \in [1,2)$; $|Q_r| < 2^{-p}$;

$\tilde{Q} = \tilde{Q}_p + \tilde{Q}_r$ denotes an approximation of $\sqrt{A}$.

Also, $lnzb(X)$ will denote the position of the last nonzero bit in the significand of the floating point value X, as defined in subsection 2.3.

5.1. Bit Structure of the Infinitely Precise Square Root

Unlike the square root reciprocal function, the square root can yield exact results for $A \ne 1$: e.g. $\sqrt{1.5625} = \sqrt{(1.1001)_2} = 1.25 = (1.01)_2$. For an absolute error below the minimum distance between function values, an exact result (i.e. $Q_r = 0$) can be detected.

P	round bit=0		round bit=1		Precision required (bits)
	Runlength of 0s	Runlength of 1s	Runlength of 0s	Runlength of 1s	
10	10	7	9	8	18.678
11	13	8	10	13	24.000
12	12	10	11	12	24.000
13	11	12	12	11	25.000
14	16	14	13	11	27.415
15	15	16	16	13	31.000
16	15	15	15	15	31.000
17	15	15	19	14	32.913
18	19	16	17	17	35.678
19	18	17	23	16	36.978
20	22	20	20	16	40.000
21	21	20	20	31	43.002
22	23	22	21	30	45.989
23	25	22	22	29	48.913
24	24	28	24	28	48.913
25	23	25	27	27	50.678

Table 2. Bit structure of root reciprocals

$\sqrt{A+2^{-p+1}} - \sqrt{A} = \frac{2^{-p+1}}{\sqrt{A+2^{-p+1}}+\sqrt{A}} > 2^{-p-1}$, much larger than the minimum accuracy required, as will be seen.

Observation 19 *If $\sqrt{A} = Q_p$, then the following relation between the positions of the last nonzero bits of A and Q_p holds:*

$lnzb(A) - exp(A) = 2 \cdot lnzb(Q_p)$.

Proof: The proof follows readily from $Q_p^2 = A$:

$Q_p = 1.q_1 q_2 \ldots q_{lnzb(Q_p)}$, then the last nonzero bit in the fixed point representation of A occurs at position $2 \cdot lnzb(Q_p)$, i.e. $lnzb(A) - exp(A) = 2 \cdot lnzb(Q_p)$. □

Corollary 20 *The combination $round\_bit = 1, sticky\_bit = 0$ is not possible for an infinitely precise square root.*

Proof: If $round\_bit = 1$, then $lnzb(Q_p) = p$ ($round\_bit = q_p$).

Then $lnzb(A) - exp(A) \leq p - 1 < 2 \cdot lnzb(Q_p) = 2p \implies sticky\_bit \neq 0$. □

The minimum accuracy needed is, of course, determined by the longest runlengths of 0's and 1's, which determine the minimum distance between two possible function values.

Observation 21 *Upper bounds on the runs of 0's or 1's following the round bit are given by:*

- *$p+1$ 0's after a round bit of 1*
- *$p+1$ 1's after a round bit of 0*
- *$p-1$ 0's after a round bit of 0*
- *$p-1$ 1's after a round bit of 1*

Proof:

$A = Q_p^2 + (2Q_p + Q_r)Q_r = Q_p^2 + (2\sqrt{A} - Q_r)Q_r$.

The last nonzero bit of A occurs no later than position $p-1$, and the last nonzero bit of Q_p^2 no later than position $2p$. Thus $(2\sqrt{A} - Q_r)|Q_r| \geq 2^{-2p}$.

Also, $(2\sqrt{A} - Q_r) < 4$, thus $|Q_r| > 2^{-2p-2}$; the longest run ends before position $2p+2$.

When the round bit is 0, we have $|Q_r| > 2^{-2(p-1)-2} = 2^{-2p}$. □

The upper bounds given above are tight for runs of 1's:

Observation 22 *The longest runlength of 1's after a round bit of 0 is $p+1$ bits. The longest runlength of 1's after a round bit of 1 is $p-1$ bits.*

Proof: For $A = 1 + 2^{-p+1}$, we have:

$A = (1+2^{-p})^2 - 2^{-2p}$; thus $(2\sqrt{1+2^{-p+1}} + |Q_r|)|Q_r| = 2^{-2p} \implies$

$|Q_r| < 2^{-2p-1}$, i.e. a run of $p+1$ 1's after a round bit of 0 exists; from Observation 21 this is the longest runlength possible.

For $A = 1 + 2^{-p+2}$:

$A = (1 + 2^{-p+1})^2 - 2^{-2p+2}$; thus $(2\sqrt{1+2^{-p+2}} + |Q_r|)|Q_r| = 2^{-2p+2} \implies$

$|Q_r| < 2^{-2p+1}$, i.e. a run of $p-1$ 1's after a round bit of 1 exists.

□

It can also be shown that the runlengths of 0's are at least 2 bits shorter, i.e. $Q_r > 0$ implies $Q_r > 2^{-2p}$ for a round bit of 1 and $Q_r > 2^{-2p+2}$ for a round bit of 0.

5.2. Infinitely Precise Rounding from an Approximation

As seen in the last subsection, the minimum distance between two possible function values of the form

$1.q_1q_2\ldots q_{p-1}0\,11\ldots1xx\ldots$ and

$1.q_1q_2\ldots q_{p-1}1\,00\ldots0xx\ldots$

is above $2^{-2p-2}+2^{-2p}$.

Between $1.q_1q_2\ldots q_{p-1}1\,11\ldots1xx\ldots$ and

$1.q_1q_2\ldots q_{p-1}0\,00\ldots0xx\ldots$ (possibly an exact result), the distance is at least 2^{-2p}; between a possibly exact result and the next possible bit pattern the distance is at least 2^{-2p+2}.

When the absolute error does not exceed 2^{-2p}, exact result detection is possible without employing the test suggested in Observation 19, which may be expensive to implement in hardware. An algorithm can be found in [8].

5.3. Exact Square Root Tests

The following observations show that when the error direction is known and its magnitude does not exceed the minimum distance between the square roots of consecutive inputs, an exact result can be easily detected by testing the second half of the estimated significand plus two guard bits. If the error magnitude is always below half the minimum distance between the square roots of consecutive inputs, exact result detection is possible regardless of error direction by using three guard bits.

Observation 23 *Let $Q=\sqrt{A}$ be the infinitely precise square root of a p-bit value in the two-binade range $[1,4)$.*

Then $q_{\lfloor\frac{p+1-exp(A)}{2}\rfloor}=\ldots=q_{p-1}=q_p=q_{p+1}=0$ iff $\sqrt{A}$ is exact (i.e. $Q_{p-1}=\sqrt{A}$).

Our exact result test will use bits $\lfloor\frac{p+1-exp(A)}{2}\rfloor,\ldots,p-1,p,p+1$ of the square root approximation. Note that such a test will not work if the absolute error is allowed to reach 2^{-p-1} (which is the lower bound on the distance between square roots of consecutive p-bit inputs):

$\sqrt{A+2^{-(p-1)}}-\sqrt{A}=\frac{2^{-(p-1)}}{\sqrt{A+2^{-(p-1)}}+\sqrt{A}}\in(2^{-p-1},2^{-p})$.

The next two observations show that if the absolute error is below 2^{-p-1}, exact result detection is possible.

Observation 24 *Let $\tilde{Q}$ be an overestimate of $\sqrt{A}$ with absolute error $E\in[0,2^{-p-1})$. Then $\sqrt{A}$ is exact iff bits $\lfloor\frac{p+1-exp(A)}{2}\rfloor,\ldots p-1,p,p+1$ of $\tilde{Q}$ are all zero.*

The analogue of Observation 24 for negative error is:

Observation 25 *Let $\tilde{Q}$ be a subestimate of $\sqrt{A}$ with absolute error $E\in(-2^{-p-1},0)$. Then $\sqrt{A}$ is exact iff bits $\lfloor\frac{p+1-exp(A)}{2}\rfloor,\ldots p-1,p,p+1$ of $\tilde{Q}$ are all 1.*

Observations 24 and 25 show that if the absolute error is below 2^{-p-1}, $\sqrt{A}$ is exact if and only if bits $\lfloor\frac{p+1-exp(A)}{2}\rfloor$, $\ldots,p-1,p,p+1$ of the estimate are all 0 (for positive error) or are all 1 (for negative error). If about one more bit of accuracy is available, exact result detection is possible regardless of the error sign:

Observation 26 *Let $\tilde{Q}$ be an estimate of $\sqrt{A}$, where the error magnitude is $|E|<2^{-p-2}$. Then $\sqrt{A}$ is exact iff $q_{\lfloor\frac{p+1-exp(A)}{2}\rfloor}=\ldots=q_{p-1}=q_p=q_{p+1}=q_{p+2}$.*

References

[1] *IEEE Standard 754 for Binary Floating Point Arithmetic*, ANSI/IEEE Standard No. 754, American National Standards Institute, Washington DC, 1988.

[2] W. Kahan, *Checking Whether Floating Point Division is Correctly Rounded*, monograph, 1987.

[3] W. Kahan, *A Test for Correctly Rounded SQRT*, lecture note, Univ. of California at Berkeley, 1996 (see http://http.cs.berkeley.edu/ wkahan/).

[4] P. Kornerup, D.W. Matula, *Finite Precision Number Systems and Arithmetic*, manuscript

[5] V. Lefevre, J. Muller, A. Tisserand, *Towards Correctly Rounded Transcendentals*, Proc. 13th IEEE Symp. Comput. Arithmetic, 1997, pp. 132-137.

[6] P.W. Markstein, *Computation of Elementary Functions on the IBM RISC System/6000 Processor*, IBM J. Res. Develop., vol. 34, no.1, January 1990, pp. 111-119.

[7] M. Schulte and E.E. Swartzlander, *Exact Rounding of Certain Elementary Functions*, 11th Symp. on Computer Arithmetic, pp. 138-145, 1993.

[8] C.S. Iordache and D.W. Matula, *Infinitely Precise Rounding for Division, Square Root, Reciprocal and Root Reciprocal*, technical report 99-CSE-1, Southern Methodist University, 1999.