Analysis of the Lookup Table Size for Square-Rooting

Behrooz Parhami

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA
E-mail: parhami@ece.ucsb.edu

Abstract

Convergence methods are widely used for division, reciprocation, and square-rooting. With such methods, it is common to use an initial table lookup step for obtaining an approximate result that leads to faster convergence. In the case of division and reciprocation, the required table size has been extensively analyzed and closed form formulas are available for the table length and width, given a desired maximum error. In this paper, we offer similar analyses for square-rooting, deriving necessary and sufficient conditions on the length and width of a lookup table that yields a result with a maximum error of r^{-h}, where r is the radix, or that provides the first h digits of the square root correctly.

Keywords: Computer arithmetic, convergence algorithm, error analysis, starting approximation, table lookup.

1. Introduction

Like division, the most common implementations of square-rooting are based on digit-recurrence or convergence schemes [Parh00]. In digit-recurrence schemes, digits of the square root are determined one at a time, beginning with the most significant one. The latency is proportional to the number k of digits in the square root, with the constant of proportionality ranging from a few gate delays, for schemes based on redundant digit sets and carry-save addition, to slightly more than the delay of a k-bit adder, when a full addition/subtraction is required in each cycle.

Convergence schemes, on the other hand, often require fewer (say, O(log k)) cycles, but each cycle typically involves multiplications and/or divisions. To make such schemes competitive for common word widths, a table lookup step is used to produce an initial approximation to the root, thus reducing the latency by ensuring faster convergence.

The required accuracy or lookup table size for the starting approximation in convergence division and reciprocation has been extensively analyzed in the literature, to the extent that they are even treated in textbooks [Wase82], [Parh87], [Wong92], [DasS94], [Ito97], [Parh00]. The relationship between the accuracy of the starting approximation and the number of iterations in convergence square-rooting has likewise been studied [Schw96], [Ito96], [Ito97].

However, accuracy, though related to the required table size, does not directly lead to the determination of the length and width of the smallest possible lookup table. The reason is that there are two error sources in the initial approximation: (1) Incomplete knowledge of the radicand due to inspecting only part of it, and (2) Reading out a value that is not a full-width word. These two elements must be carefully chosen to minimize the table size (Fig. 1).

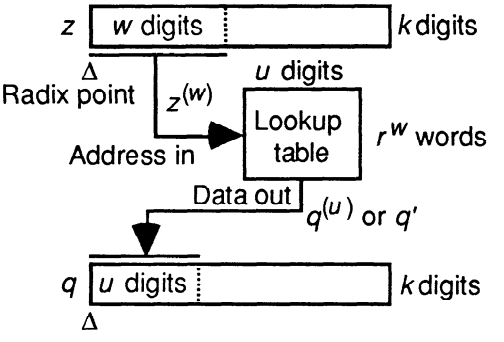

Fig. 1. Approximating $\sqrt{z}$ via table lookup.

Table lookup can also be used at the start of digit-recurrence algorithms for square-rooting. In an early paper on the topic, Nienhaus [Nien89] studied the table size in order to obtain a few (3 or 4) bits of the square root at the start of a digit-recurrence scheme. He did not go beyond 4 bits because he felt that the hardware complexity would become excessive. At about the same time, a small PLA was being used for the determination of the first few bits in radix-4 square-rooting (see the references in [Erce90]).

We use methods akin to those used for our analysis of the lookup table size in convergence division [Parh87] to derive

results for convergence square-rooting. The contributions of this paper consist of a pair of theorems for square-rooting that mirror those in [Parh87]. Besides, an additional theorem is proven that is useful when the approximate square root must be guaranteed to have h correct digits (as opposed to being within r^{-h} of the correct value, from either side).

The main notation used is defined below and in Fig. 1:

q $= (.q_{-1}q_{-2} \ldots q_{-k})_r$; square root, $q \in [1/r, 1)$
r Number representation radix; $r \geq 2$
$x^{(v)}$ Fraction x, truncated to v digits; $x^{(v)}r^v$ is an integer
z $= (.z_{-1}z_{-2} \ldots z_{-k})_r$; unsigned radicand, $z \in [1/r^2, 1)$

Other notation will be defined when needed.

2. Looking up the First h Digits of q

We first focus on the problem of determining the first h digits of $q = \sqrt{z}$ via table lookup. The resulting digits (i.e., q_{-1} through q_{-h}) provide an approximation to q that is guaranteed to be in the interval $(q - r^{-h}, q]$. Note, however, that an approximation guaranteed to be within the one-sided interval $(q - r^{-h}, q]$ does not imply agreement in h digits and is, therefore, a laxer condition than the latter. Still laxer is the requirement that the approximation have an error of less than r^{-h}, as this would allow the approximate value to lie in $(q - r^{-h}, q + r^{-h})$. This last type of approximation will be discussed in Section 3.

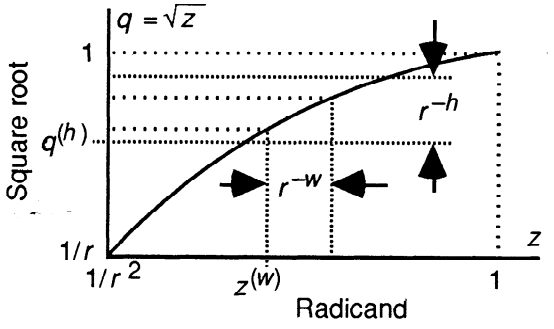

Fig. 2. Requirements for approximation to $\sqrt{z}$ matching the exact value in h digits.

The lookup table supplying the first h digits of q (or $q^{(h)}$) must, of course, be h digits wide. Theorem 1 provides a lower bound of r^{2h} on the length of the lookup table and shows that this bound can be matched, thus solving the problem completely. Note that the need for using $2h$ digits of the radicand to determine h digits of the square root is implicit in the workings of the pencil-and-paper square root extraction algorithm ($2k$-digit radicand, k-digit square root). Because we have assumed fractional operands with specific ranges, we prove this result for completeness.

Theorem 1: To derive the first h radix-r digits of $q = \sqrt{z}$ exactly, it is necessary and sufficient to inspect $2h$ radix-r digits of the radicand z.

Proof: Suppose $2h$ digits of z are inspected to deduce $q^{(h)}$. From the conditions

$$1/r \leq q^{(h)} \leq q = \sqrt{z} < q^{(h)} + r^{-h} \leq 1$$

and the requirement that $q^{(h)}$ not change for the entire interval $[z^{(2h)}, z^{(2h)} + r^{-2h})$ of z values that share the initial $2h$ digits (Fig. 2, with $w = 2h$), we get:

$$(q^{(h)})^2 \leq z^{(2h)} \leq z < z^{(2h)} + r^{-2h} \leq (q^{(h)} + r^{-h})^2$$

These inequalities yield the following bounds for $q^{(h)}r^h$:

$$r^h \sqrt{z^{(2h)} + r^{-2h}} - 1 \leq q^{(h)}r^h \leq r^h \sqrt{z^{(2h)}}$$

Note that the width of the permissible interval for $q^{(h)}r^h$, defined by the inequalities above, is strictly less than 1, so the existence of an integer $q^{(h)}r^h$ in the interval is not automatically guaranteed. A necessary and sufficient condition for the existence of an integer $q^{(h)}r^h$ between the bounds given above is:

$$r^h \sqrt{z^{(2h)} + r^{-2h}} - 1 \leq \lfloor r^h \sqrt{z^{(2h)}} \rfloor$$

Let $z^{(2h)} = (q^{(h)})^2 + \delta$, where δ satisfies $0 \leq \delta \leq 2q^{(h)}r^{-h}$. The upper bound provided for the residual δ is justified by noting that δ is a $2h$-digit number and that increasing its value beyond $2q^{(h)}r^{-h}$, even by only r^{-2h}, would make $z^{(2h)} \geq (q^{(h)} + r^{-h})^2$, thus contradicting the premise that $q^{(h)}$ matches the first h digits of q. The right-hand side of our preceding necessary and sufficient condition is $q^{(h)}r^h$, thereby turning the condition into

$$\sqrt{z^{(2h)} + r^{-2h}} \leq q^{(h)} + r^{-h}$$

which is always satisfied (proof via squaring). To show that $2h$ digits are necessary, consider a radicand z such that:

$$z^{(2h-2)} = (q^{(h-1)})^2 + \delta = (q^{(h-1)})^2 + 2q^{(h-1)}r^{-h}$$

As an example, for $h = 3$ and $r = 10$, we might choose $z^{(4)} = .2034$, leading to $q^{(2)} = .45$ and $\delta = .0009$. If the next two digits of z are 00 (i.e., $z_{-2h+1} = z_{-2h} = 0$), we have:

$$z^{(2h)} = (q^{(h)})^2 + \gamma = z^{(2h-2)} = (q^{(h-1)})^2 + 2q^{(h-1)}r^{-h}$$

Hence, $q^{(h)} = q^{(h-1)}$, as expected. On the other hand, if the next two digits of z are 01, the identities

$z^{(2h)} = z^{(2h-2)} + r^{-2h} = (q^{(h-1)} + r^{-h})^2 = (q^{(h)})^2$

suggest that $q^{(h)} = q^{(h-1)} + r^{-h}$. Since in the two cases above that led to different results for $q^{(h)}$, the respective $z^{(2h)}$ values differed only in z_{-2h}, the insufficiency of inspecting $2h - 1$ digits of z is proven. For our numerical example, if .2034 is extended to .203400, its 3-digit square root is .450, while .203401 leads to .451. ■

Based on Theorem 1, except in the uninteresting case of $r = 2$ and $h = 1$, determining the first h digits of q always requires a lookup table of size r^{2h} words, with the word width being $h - 1$ bits for radix 2 (q_{-1} is always 1 in this case) and h digits otherwise.

3. Approximation in $(q - r^{-h}, q + r^{-h})$

In convergence methods, we really do not need the first h digits of q but rather are interested in an approximate initial value q' that is close enough to q; say, it is in the interval $(q - r^{-h}, q + r^{-h})$. We say that such a value q' offers h digits of convergence. Of course, to be this accurate, q' must have h or more digits. Note that $h - 1$ radix-r digits might have sufficed for an approximation that is based on inspecting all the digits of z (i.e., for $w = k$), since in this case the only source of error is rounding of the exact value of $\sqrt{z}$ to $h - 1$ digits. But the latter is not a useful approximation, given that the table size to look up the exact (rounded) value of $\sqrt{z}$ would be only a factor of $k/(h - 1)$ larger.

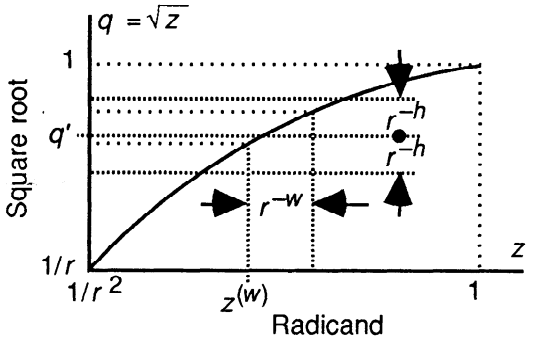

Fig. 3. Requirements for approximation to $\sqrt{z}$ accurate to within r^{-h}.

The requirements for our desired approximation are given in Theorem 2. What distinguishes our result from previous analyses is that we find a lower bound and show that it can be matched, whereas analyses based on worst-case error bounds do not guarantee optimality.

Theorem 2: To find an approximation q' to $q = \sqrt{z}$ that offers h digits of convergence in radix r, meaning that the approximate value lies within the interval $(q - r^{-h}, q + r^{-h})$, it is sufficient to inspect h digits of z. Furthermore, the resulting r^h-entry table of h-digit words is optimal for $r \geq 3$ (for $r = 2$, see Theorem 3).

Proof: Let the table entry q' for $z^{(w)}$ be the h-digit rounded version of the midpoint m defined as:

$$m = (\sqrt{z^{(w)} + r^{-w}} - \sqrt{z^{(w)}})/2$$

Since the rounding error for m is upper bounded by $r^{-h}/2$, it suffices to choose w such that m is less than $r^{-h}/2$ away from the extremes $\sqrt{z^{(w)}}$ and $\sqrt{z^{(w)} + r^{-w}}$. Achieving this goal requires that

$$\sqrt{z^{(w)} + r^{-w}} - \sqrt{z^{(w)}} < r^{-h}$$

which can be written as:

$$\sqrt{z^{(w)}}[(1 + r^{-w}/\sqrt{z^{(w)}})^{1/2} - 1] < r^{-h}$$

In view of the following inequality (easily proven by squaring both sides)

$$(1 + r^{-w}/\sqrt{z^{(w)}})^{1/2} < 1 + r^{-w}/(2\sqrt{z^{(w)}})$$

it is sufficient to guarantee $r^{-w}/2 \leq r^{-h}$. This last inequality is clearly satisfied for $w = h$.

We have already discussed why the table width cannot be reduced below h digits (see the first paragraph in Section 3). We now show that the table length cannot be reduced below r^h words for $r \geq 3$, when digits of z must be dealt with in their entirety; that is, q' cannot be properly chosen based on $h - 1$ digits of z, no matter how wide the table (i.e., even when the rounding error is zero). For this, it is sufficient to show that for some value of $z^{(h-1)}$, we have:

$$\sqrt{z^{(h-1)} + r^{-(h-1)}} - \sqrt{z^{(h-1)}} \geq 2r^{-h}$$

Choosing $z^{(h-1)} = 1/r^2$, turns the preceding inequality into

$$r/4 \geq r^{-1} + r^{-h}$$

which always holds for $r \geq 3$. This concludes the proof that an r^h-entry table of h-digit words is optimal for $r \geq 3$. ■

Theorem 3 shows that in most cases, the table size of r^h words by h digits can be reduced by a factor of 2, including in the important special case of $r = 2$ for which the table size becomes $2^{h-1}(h - 1)$ bits, given the aforementioned factor-of-2 reduction in length combined with the fact that the leading bit $q_{-1} = 1$ need not be stored.

Theorem 3: To find an approximation q' to $q = \sqrt{z}$ that offers h digits of convergence in radix r, meaning that the approximate value lies within the interval $(q - r^{-h}, q + r^{-h})$, it is necessary and sufficient to inspect $h\lceil \log_2 r \rceil - 1$ bits of the binary encoding of z, where each radix-r digit of z is separately encoded as a $\lceil \log_2 r \rceil$-bit binary number.

Proof: We proceed as in the proof of Theorem 2, except that we assume the inspection of $w - 1$ whole digits and the most-significant $\lceil \log_2 r \rceil - 1$ bits of the encoding of the wth digit in z. Everything remains the same, except that the uncertainty in the value of z becomes about $2r^{-w}$ (one r^{-w} contributed by all the uninspected digits beyond the wth and another by the ignored least significant bit of z_{-w}). Our sufficient condition then becomes

$$\sqrt{z' + 2r^{-w}} - \sqrt{z'} < r^{-h}$$

or, equivalently,

$$\sqrt{z'}\,[(1 + 2r^{-w}/\sqrt{z'})^{1/2} - 1] < r^{-h}$$

where z' is the value of z based on the first $h\lceil \log_2 r \rceil - 1$ bits in its binary encoding (note that here we cannot use the notation $z^{(w)}$, given that the first w digits of z are not inspected in full).

Proceeding in the same way as in the proof of Theorem 2, we get the sufficient condition $r^{-w} \le r^{-h}$, which is clearly satisfied for $w = h$.

The proof that the table size cannot be further reduced is similar to that in Theorem 2. Ignoring one more bit of the encoding of z in the table lookup increases the uncertainty in the value of z to about $4r^{-w}$. Thus, the optimality of the preceding result on lookup table size would follow from proving that the following inequality holds for some z':

$$\sqrt{z' + 4r^{-h}} - \sqrt{z'} \ge 2r^{-h}$$

Choosing $z' = 1/r^2$ turns the preceding inequality into:

$$1 \ge r^{-1} + r^{-h}$$

This last inequality cannot hold for $r \ge 3$ or for $r = 2$ if we exclude the uninteresting case of $h = 1$. ■

Based on Theorem 3, in the common case when radix-r digits are encoded in binary, the table size can be reduced by a factor of 2 relative to what Theorem 2 suggests. Interestingly, the fact that for $r = 2$, the table length can be reduced to 2^{h-1} (as observed in the proof of Theorem 2) also follows as a corollary to Theorem 3.

4. Conclusion

Using interval analysis, we have derived the minimal lookup table size for obtaining an initial approximation to the square root function that provides the first h digits of the root (Theorem 1) or that offers h digits of convergence (Theorems 2 and 3). Contents of the required tables are also explicitly given in the respective proofs.

An interesting question that might be considered for further research is whether a maximum error of ε that is not a negative integral power of r could lead to a smaller table. This might be useful, for example, in iterative schemes for word widths that are not powers of 2.

References

[DasS94] DasSarma, D. and D.W. Matula, "Measuring the Accuracy of ROM Reciprocal Tables," *IEEE Trans. Computers*, Vol. 43, No. 8, pp. 932-940, August 1994.

[Erce90] Ercegovac, M.D. and T. Lang, "Radix-4 Square Root Without Initial PLA," *IEEE Trans. Computers*, Vol. 39, No. 8, pp. 1016-1024, August 1990.

[Ito96] Ito, M., N. Takagi, and S. Yajima, "Square Rooting by Iterative Multiply-Additions," *Information Processing Letters*, Vol. 60, No. 5, pp. 267-269, December 9, 1996.

[Ito97] Ito, M., N. Takagi, and S. Yajima, "Efficient Initial Approximation for Multiplicative Division and Square Root by a Multiplication with Operand Modification," *IEEE Trans. Computers*, Vol. 46, No. 4, pp. 495-498, April 1997.

[Nien89] Nienhaus, H.A., "A Fast Square Rooter Combining Algorithmic and Lookup Table Techniques," *Proc. IEEE Southeastcon*, 1989, pp. 1103-1105.

[Parh00] Parhami, B., *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, New York, 2000.

[Parh87] Parhami, B., "On the Complexity of Table Look-Up for Iterative Division," *IEEE Trans. Computers*, Vol. 36, No. 10, pp. 1233-1236, October 1987.

[Schw96] Schwarz, E.M. and M.J. Flynn, "Hardware Starting Approximation Method and Its Application to the Square Root Operation," *IEEE Trans. Computers*, Vol. 45, No. 12, pp. 1356-1369, December 1996.

[Wong92] Wong, D. and M. Flynn, "Fast Division Using Accurate Quotient Approximations to Reduce the Number of Iterations," *IEEE Trans. Computers*, pp. 981-995, 1992.

[Wase82] Waser, S. and M.J. Flynn, *Introduction to Arithmetic for Digital Systems Designers*, Holt, Rinehart, & Winston, 1982.