Efficient Initial Approximation for Multiplicative Division and Square Root by a Multiplication with Operand Modification

Masayuki Ito, Naofumi Takagi, *Member, IEEE*, and Shuzo Yajima, *Senior Member, IEEE*

Abstract—An efficient initial approximation method for multiplicative division and square root is proposed. It is a modification of the piecewise linear approximation. The multiplication and the addition required for the linear approximation are replaced by only one multiplication with a slight modification of the operand. The same accuracy is achieved. The modification of the operand requires only a bit-wise inversion and a one-bit shift, and can be implemented by a very simple circuit. One clock cycle may be saved, because the addition is removed. The required table size is also reduced, because only one coefficient instead of two has to be stored.

Index Terms—Computer arithmetic, division, initial approximation, linear approximation, reciprocal, square root.

1 Introduction

With the increasing availability of high-speed multipliers, multiplicative methods have become attractive to the fast division and square root. In general, such methods adopt an initial approximation and improve it by a converging algorithm, e.g., Newton-Raphson method and Goldschmidt's algorithm [1], [2]. Efficient initial approximations reduce the number of iterations of converging algorithms and achieve high-speed division and square root. In this paper, we focus on generating initial approximations for multiplicative division and square root on the mantissa of a floating point number.

For the initial approximation, look-up tables are commonly used. The simplest way is directly reading an initial approximation through table look-up using some most significant bits of an operand as the index [3]. Another efficient method is a piecewise linear approximation, which requires a multiplication and an addition [4]. The two coefficients of the linear function are read out of a look-up table. The approximation is about twice as many bits of accuracy as that achieved by the direct approximation, when the same bits of an operand are used as the index. Therefore, the multiplication and the addition, as well as the increase of the table size, for the approximation are worth the first iteration of a quadratic converging algorithm, such as Newton-Raphson's, which requires two multiplications for division or three multiplications for square root.

In this paper, we propose a new initial approximation method, which is a modification of the piecewise linear approximation. We replace the multiplication and the addition required for the linear approximation by one multiplication with a slight modification of the operand. The same accuracy is achieved. For the modification of the operand, we need only a bit-wise inversion and a one-bit shift of a part of it. These operations are also required in the converging algorithms and can be performed by a very simple circuit.

• M. Ito is with Hitachi Ltd., Kokubunji 185, Japan.
• N. Takagi is with the Department of Information Engineering, Nagoya University, Chikusa-ku, Nagoya 464-01, Japan. E-mail: ntakagi@nuie.nagoya-u.ac.jp.
• S. Yajima is with the Department of Information Science, Kyoto University, Kyoto 606-01, Japan.

Manuscript received June 16, 1995; revised July 24, 1996.
For information on obtaining reprints of this article, please send e-mail to: transcom@computer.org, and reference IEEECS Log Number C96306.

One clock cycle may be saved, because the addition is removed. Furthermore, the required table size is also reduced, because only one coefficient instead of two has to be stored.

In the following, we will propose a new initial approximation method for division and for square root, in Sections 2 and 3, respectively. In Section 4, we will compare it with conventional methods.

2 Initial Approximation for Division

Newton-Raphson method and Goldschmidt's algorithm are widely used for multiplicative division [1], [2]. They need an initial approximation to the reciprocal of a given divisor $Y = [1.y_1y_2 \cdots y_n]$, where $1 \le Y < 2$.

The piecewise linear approximation for reciprocal adopts a linear function $-A_1 \times Y + A_0$ [4]. The two coefficients A_1 and A_0 are read through table look-up addressed by the m most significant bits of the divisor Y, i.e., $[y_1y_2 \cdots y_m]$. The look-up table keeps the coefficients for 2^m intervals of Y. One multiplication and one addition are required.

Differentiating the error function $\epsilon(Y) = -A_1 \times Y + A_0 - \frac{1}{Y}$ yields $\epsilon'(Y) = \frac{1}{Y^2} - A_1$. Hence, the positive maximum error is $\epsilon(Y)_{max} = E\left(\frac{1}{\sqrt{A_1}}\right) = A_0 - 2\sqrt{A_1}$. To minimize the worst absolute error in each interval $[p, p + 2^{-m})$, where $p = [1.y_1y_2 \cdots y_m]$, the errors at both endpoints, i.e., $\epsilon(p)$ and $\epsilon(p + 2^{-m})$, should have the same value and should be equal to $-\epsilon(Y)_{max}$. Consequently, the coefficients should be

$$\begin{cases} A_1 = \dfrac{1}{p \cdot \left(p + 2^{-m}\right)} \\ A_0 = \dfrac{p + 2^{-m-1} + \sqrt{p \cdot \left(p + 2^{-m}\right)}}{p \cdot \left(p + 2^{-m}\right)}. \end{cases} \tag{1}$$

Then,

$$\epsilon(Y)_{max} = A_0 - 2\sqrt{A_1} = \frac{p + 2^{-m-1} - \sqrt{p \cdot \left(p + 2^{-m}\right)}}{p \cdot \left(p + 2^{-m}\right)} <$$
$$\frac{1}{p^2} \cdot \left(p + 2^{-m-1} - p\left(1 + \frac{2^{-m-1}}{p} - \frac{2^{-2m-3}}{p^2} + \cdots\right)\right) < \frac{1}{p^3} \cdot 2^{-2m-3},$$

when calculations are carried out with infinite precision. A_1 and A_0 satisfy $\frac{1}{4} < A_1 < 1$ and $1 < A_0 < 2$, respectively. When t_1 and t_0 bits are kept for A_1 and A_0, respectively, in the table, A_1 and A_0 have the forms $\left[0.a_1^1a_2^1 \cdots a_{t_1}^1\right]$ and $\left[1.a_1^0a_2^0 \cdots a_{t_0}^0\right]$, respectively. The table is of size $2^m \times (t_1 + t_0)$ bits. The total error ϵ_L, considering the errors due to storing only t_1 and t_0 bits of A_1 and A_0, respectively, satisfies

$$\left|\epsilon_L\right| < \frac{1}{p^3} \cdot 2^{-2m-3} + Y \cdot 2^{-t_1-1} + 2^{-t_0-1}. \tag{2}$$

The last two terms are from the rounding errors.

Since the upper part of Y is always p in the interval $[p, p + 2^{-m})$, $-A_1 \times Y + A_0$ can be rewritten as $-A_1 \times q + A_0'$ where $q = Y - p$ (= $[0.0 \cdots 0y_{m+1}y_{m+2} \cdots y_n]$) and $A_0' = A_0 - A_1 \times p$ [5]. A_0' satisfies $\frac{1}{2} < A_0' < 1$. The total error ϵ_I, considering the errors due to storing only t_1' and t_0' bits of A_1 and A_0', respectively, satisfies

$$\left|\epsilon_I\right| < \frac{1}{p^3} \cdot 2^{-2m-3} + 2^{-m} \cdot 2^{-t_1'-1} + 2^{-t_0'-2}. \tag{3}$$

t_1' may be about the half of t_1. When we prepare a dedicated multiplier for initial approximation, its size may be smaller. We refer to this method as improved linear approximation method.

Now, we propose a new initial approximation method. We can find the following relation between A_1 and A_0 in (1):

$$A_0 \simeq A_1 \times \left(2p + 2^{-m}\right). \tag{4}$$

Therefore, $A_1 \times (2p + 2^{-m} - Y)$ produces almost the same value as $-A_1 \times Y + A_0$. This means that one multiplication $B_1 \times \hat{Y}$ achieves the same accuracy, where $B_1 \simeq A_1$ and $\hat{Y} = 2p + 2^{-m} - Y$.

Since $\hat{Y} = p + 2^{-m} - q = \left[1.y_1 y_2 \cdots y_m \tilde{y}_{m+1} \tilde{y}_{m+2} \cdots \tilde{y}_n\right] + 2^{-n}$, where $\tilde{1} = 0$ and $\tilde{0} = 1$, we can form $\hat{Y}$ by only a bit-wise inversion (complementation) of the lower part of Y. (We ignore the last term 2^{-n}.)

Since the error of $A_1 \times \hat{Y}$ is

$$\begin{aligned} & A_1 \cdot \left(2p + 2^{-m} - Y\right) - \frac{1}{Y} \\ &= \frac{p + 2^{-m} - q}{p \cdot \left(p + 2^{-m}\right)} - \frac{1}{p+q} \\ &= \frac{1}{p} - \frac{q}{p^2} \cdot \left(1 + \frac{2^{-m}}{p}\right)^{-1} - \frac{1}{p}\left(1 + \frac{q}{p}\right)^{-1} \\ &= \frac{2^{-2m-2}}{p^3} - \frac{\left(q - 2^{-m-1}\right)^2}{p^3} + O\left(2^{-3m}\right), \end{aligned} \tag{5}$$

B_1 should be

$$\begin{aligned} B_1 &= A_1 - \frac{2^{-2m-3}}{p^4} \\ & \left(= \frac{1}{p \cdot \left(p + 2^{-m}\right)} - \frac{2^{-2m-3}}{p^4}\right). \end{aligned} \tag{6}$$

We use an m-bits-in t-bits-out table for B_1. The table is of size $2^m \times t$ bits. The total error ϵ_P considering the rounding error of B_1 satisfies

$$\left|\epsilon_P\right| < \frac{1}{p^3} \cdot 2^{-2m-3} + Y \cdot 2^{-t-1}. \tag{7}$$

The proposed method with an adequate t produces an approximation with the same accuracy as the conventional linear approximations, without addition. Furthermore, the required table size is also reduced, because only one coefficient instead of two has to be stored. The modification of the operand is only a bit-wise inversion. Note that a bit-wise inversion is also required in Newton-Raphson's and Goldschmidt's iteration for subtracting an intermediate result from 2. Note also that we do not need whole $\hat{Y}$ but need down to about the tth position of $\hat{Y}$.

Fig. 1 illustrates an implementation of the proposed method. The required hardware is a ROM of size $2^m \times t$ bits and an operand modifier which consists of only $t - m$ inverters. Note that the operand modifier may be a modification of the complementer which is used for the converging algorithms. No other dedicated hardware is required when we use an existing multiplier which is also used for the converging algorithms. When we prepare a dedicated multiplier for initial approximation, its size may be $(t + 1)$-bits by t-bits.

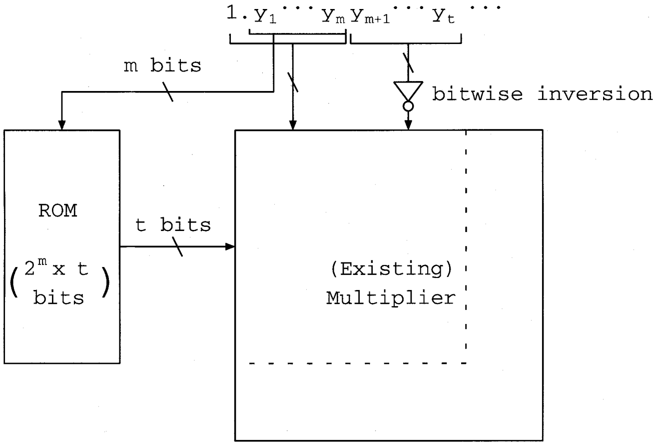

Fig. 1. An implementation of the proposed method for reciprocal.

We can increase the accuracy of the approximation by adding B_0 to $B_1 \times \hat{Y}$, where B_0 is read from another table indexed by both an upper part of p and that of q. Another table look-up and addition are required. We have discussed this method in [6] and [7]. (In [7], A_1 is used instead of B_1.) This method may be efficient in a system which has a multiply-adder.

3 Initial Approximation for Square Root

In square root, $\sqrt{X}$ or $\sqrt{2X}$ is calculated accordingly as the exponent part of the original floating point number is even or odd, where $X = [1.x_1 x_2 \cdots x_n]$ and $1 \le X < 2$.

Newton-Raphson method for calculating the square root reciprocal and Goldschmidt's algorithm are widely used for multiplicative square root [1], [2]. They need an initial approximation to the square root reciprocal of a given operand, i.e., X or $2X$.

The piecewise linear approximation method for square root reciprocal adopts a linear function $-C_1 \times X + C_0$. (When the exponent part is odd, $2X$ is used instead of X.) The two coefficients C_1 and C_0 are read through table look-up addressed by the $(m - 1)$ most significant bits of X, i.e., $[x_1 x_2 \cdots x_{m-1}]$, together with the least significant bit of the exponent part. From similar consideration to the case of reciprocal, C_1 and C_0 for the interval $[u, u + 2^{-m+1})$ should be

$$\begin{cases} C_1 = 2^{m-1}\left(\frac{1}{\sqrt{u}} - \frac{1}{\sqrt{u + 2^{-m+1}}}\right) \\ \quad \left(= \frac{1}{2u\sqrt{u}}\left(1 - \frac{3 \cdot 2^{-m-1}}{u} + \frac{5 \cdot 2^{-2m-1}}{u^2} - \cdots\right)\right) \\ C_0 = \frac{1}{2}\left(C_1 \cdot u + \sqrt{u} + \frac{3}{2}\left(2C_1\right)^{\frac{1}{3}}\right) \\ \quad \left(= \frac{1}{2\sqrt{u}}\left(3 - \frac{3 \cdot 2^{-m-1}}{u} + \frac{7 \cdot 2^{-2m-3}}{u^2} - \cdots\right)\right). \end{cases} \tag{8}$$

C_1 and C_0 for the interval $[2u, 2u + 2^{-m+2})$ should be the values that obtained by substituting $2u$ and $m - 1$ for u and m in (8), respectively. When t_1 and t_0 bits are kept for C_1 and C_0, respectively, in the table, C_1 and C_0 have the forms $\left[0.0c_1^1 c_2^1 \cdots c_{t_1}^1\right]$ and $\left[c_1^0.c_2^0 \cdots c_{t_0}^0\right]$, respectively. (We assume $2C_1$ is stored instead of C_1 for $[2u, 2u + 2^{-m+2})$.) The table is of size $2^m \times (t_1 + t_0)$ bits. The total error δ_L considering the rounding errors of C_1 and C_0 satisfies

$$\left|\delta_L\right| < \frac{3}{u^2\sqrt{u}} \cdot 2^{-2m-4} + X \cdot 2^{-t_1-2} + 2^{-t_0}. \qquad (9)$$

The table size for C_1 may be reduced to about the half, as the case of reciprocal. The approximation function is rewritten as $-C_1 \times v + C_0'$ where $v = X - u$ (= $[0.0 \cdots 0x_m x_{m+1} \cdots x_n]$) and $C_0' = C_0 - C_1 \times u$. The total error δ_I, considering the errors due to storing only t_1' and t_0' bits of C_1 and C_0', respectively, satisfies

$$\left|\delta_I\right| < \frac{3}{u^2\sqrt{u}} \cdot 2^{-2m-4} + 2^{-m+1} \cdot 2^{-t_1'-2} + 2^{-t_0'-2}. \qquad (10)$$

Now, we propose a new initial approximation method. We can find the following relation between the two coefficients C_1 and C_0 in (8).

$$C_0 \simeq C_1 \times \left(3u + 3 \cdot 2^{-m}\right). \qquad (11)$$

Therefore, $2C_1 \times \frac{1}{2}\left(3u + 3 \cdot 2^{-m} - X\right)$ can produce almost the same value as $-C_1 \times X + C_0$. This means that one multiplication $D_1 \times \hat{X}$ achieves the same accuracy, where $D_1 \simeq 2C_1$ and $\hat{X} = \frac{1}{2} \cdot \left(3u + 3 \cdot 2^{-m} - X\right)$.

Since

$$\hat{X} = u + 2^{-m} + 2^{-m-1} - \tfrac{v}{2} = \left[1.x_1 x_2 \cdots x_{m-1}\tilde{x}_m x_m \tilde{x}_{m+1}\tilde{x}_{m+2}\cdots\tilde{x}_n\right] + 2^{-n-1},$$

where $\tilde{1} = 0$ and $\tilde{0} = 1$, we can form $\hat{X}$ only by a bit-wise inversion and a one-bit shift. (We ignore the last term 2^{-n-1}.)

Since the error of $2C_1 \times \hat{X}$ is

$$\begin{aligned} &2C_1 \times \frac{1}{2}\left(3u + 3 \cdot 2^{-m} - X\right) - \frac{1}{\sqrt{X}} \\ &= 2^{m-1}\left(\frac{1}{\sqrt{u}} - \frac{1}{\sqrt{u + 2^{-m+1}}}\right)\left(2u - v + 3 \cdot 2^{-m}\right) - \frac{1}{\sqrt{u+v}} \\ &= \frac{5 \cdot 2^{-2m}}{8u^2\sqrt{u}} - \frac{3\left(v - 2^{-m}\right)^2}{8u^2\sqrt{u}} + O\left(2^{-3m}\right), \end{aligned} \qquad (12)$$

D_1 should be

$$D_1 = 2C_1 - \frac{7 \cdot 2^{-2m-4}}{u^3\sqrt{u}} \quad \left(= \frac{1}{u\sqrt{u}}\left(1 - \frac{3 \cdot 2^{-m-1}}{u} + \frac{33 \cdot 2^{-2m-4}}{u^2} - \cdots\right)\right). \qquad (13)$$

D_1 for the interval $[2u, 2u + 2^{-m+2})$ should be the value that obtained by substituting $2u$ and $m - 1$ for u and m in (13), respectively.

We use an m-bits-in t-bits-out table for D_1. The table is of size $2^m \times t$ bits. The total error δ_P considering the rounding error of D_1 satisfies

$$\left|\delta_P\right| < \frac{3}{u^2\sqrt{u}} \cdot 2^{-2m-4} + X \cdot 2^{-t-1}. \qquad (14)$$

The proposed method with an adequate t produces an approximation with the same accuracy as the conventional linear approximations, without addition. The required table size is reduced. The modification of the operand is only a bit-wise inversion and a one-bit shift which are also required in Newton-Raphson's and Goldschmidt's iteration for subtracting an intermediate result from 3 and dividing it by 2.

The required hardware is a ROM of size $2^m \times t$ bits and an operand modifier which consists of only $(t - m)$ inverters. (One-bit shift may be implemented by wiring.) No other dedicated hardware is required when we use an existing multiplier which is also used for the converging algorithms.

We can increase the accuracy of the approximation by adding D_0 to $D_1 \times \hat{X}$, where D_0 is read from another table indexed by both an upper part of u and that of v [6].

The piecewise linear approximation may directly produce $\sqrt{X}$ or $\sqrt{2X}$ with single-precision (24-bit) accuracy. The piecewise linear approximation for square root adopts a linear function $E_1 \cdot X + E_0$. The two coefficients E_1 and E_0 for the interval $[u, u + 2^{-m+1})$ should be

$$\begin{cases} E_1 = 2^{m-1}\left(\sqrt{u + 2^{-m+1}} - \sqrt{u}\right) \\ \quad \left(= \frac{1}{2\sqrt{u}}\left(1 - \frac{2^{-m-1}}{u} + \frac{2^{-2m-1}}{u^2} - \cdots\right)\right) \\ E_0 = \frac{\sqrt{u}}{2} - E_1 \cdot \frac{u}{2} + \frac{1}{8E_1} \\ \quad \left(= \frac{\sqrt{u}}{2}\left(1 + \frac{2^{-m-1}}{u} - \frac{2^{-2m-1}}{u^2} + \frac{2^{-2m-3}}{u^2} - \cdots\right)\right). \end{cases} \qquad (15)$$

E_1 and E_0 for the interval $[2u, 2u + 2^{-m+2})$ should be the values that obtained by substituting $2u$ and $m - 1$ for u and m in (15), respectively. When t_1 and t_0 bits are kept for E_1 and E_0, respectively, in the table, the table is of size $2^m \times (t_1 + t_0)$ bits. The total error ξ_L considering the rounding errors of E_1 and E_0 satisfies

$$\left|\xi_L\right| < \frac{1}{u\sqrt{u}} \cdot 2^{-2m-4} + X \cdot 2^{-t_1-2} + 2^{-t_0-2}. \qquad (16)$$

The table size for E_1 may be reduced to about the half, by the improved linear approximation method.

We can find the following relation between the two coefficients E_1 and E_0 in (15):

$$E_0 \simeq E_1 \times \left(u + 2^{-m}\right). \qquad (17)$$

One multiplication $F_1 \times \check{X}$ achieves the same accuracy, where $F_1 \simeq 2E_1$ and $\check{X} = \frac{1}{2}\left(u + 2^{-m} + X\right)$. Since $\check{X} = u + 2^{-m-1} + \frac{v}{2} = \left[1.x_1 x_2 \cdots x_{m-1} x_m \tilde{x}_m x_{m+1} \cdots x_n\right]$, $\check{X}$ can be formed only by a bit inversion and a one-bit shift. F_1 should be

$$F_1 = 2E_1 - \frac{3 \cdot 2^{-2m-4}}{u^2\sqrt{u}} \quad \left(= \frac{1}{\sqrt{u}}\left(1 - \frac{2^{-m-1}}{u} + \frac{5 \cdot 2^{-2m-4}}{u^2} - \cdots\right)\right). \qquad (18)$$

F_1 for the interval $[2u, 2u + 2^{-m+2})$ should be the value that obtained by substituting $2u$ and $m - 1$ for u and m in (18), respectively.

We use an m-bits-in t-bits-out table for F_1. The total error ξ_P considering the rounding error of F_1 satisfies

$$\left|\xi_P\right| < \frac{1}{u\sqrt{u}} \cdot 2^{-2m-4} + X \cdot 2^{-t-1}. \qquad (19)$$

The proposed method with an adequate t produces an approximation with the same accuracy as the conventional linear approximations, without addition. The required table size is reduced. The modification of the operand is only a bit inversion and a one-bit shift.

We can increase the accuracy (or reduce m) by adding F_0 to $F_1 \times \check{X}$, where F_0 is read from another table indexed by both an upper part of u and that of v [6].

TABLE 1
COMPARISON OF INITIAL APPROXIMATION METHODS FOR RECIPROCAL

Method	Required ROM size (bits)	Required Operations	Accuracy (bits)
Linear	$2^m \times ((2m + 3) + (2m + 3))$	Mul, Add	$2m + 2$
Improved	$2^m \times ((m + 3) + (2m + 2))$	Mul, Add	$2m + 2$
Proposed	$2^m \times (2m + 3)$	OM, Mul	$2m + 2$
Modified	$2^m \times \left(\frac{5m}{2} + 4\right) + 2^m \times \left(\frac{m}{2} + 1\right)$	OM, Mul, Add	$\frac{5m}{2}$

Mul: Multiplication, Add: Addition, OM: Operand Modification

TABLE 2
COMPARISON OF REQUIRED ROM SIZE FOR DOUBLE-PRECISION DIVISION (BITS)

Method	Number of following Newton-Raphson Iterations		
	3	2	1
Linear	$2^3 \times (8 + 8) = 128$	$2^6 \times (15 + 15) = 1{,}920$	$2^{13} \times (28 + 28) = 448K$
Improved	$2^3 \times (4 + 7) = 88$	$2^6 \times (9 + 14) = 1{,}472$	$2^{13} \times (14 + 27) = 328K$
Proposed	$2^3 \times 8 = 64$	$2^6 \times 14 = 896$	$2^{13} \times 28 = 224K$

4 COMPARISON

Table 1 shows a comparison of initial approximation methods for reciprocal, i.e., the (original) piecewise linear approximation (“Linear”), the improved linear approximation (“Improved”), the proposed method (“Proposed”), and the modified version of the proposed method discussed in [6] (“Modified”), with respect to the ROM size, the required operations besides table look-up, and the obtained accuracy. The upper m bits of the divisor Y is used as the index for table look-up. “Modified” requires two tables indexed by different bits of Y. B_1 is indexed by the same m bits as the other methods and $\frac{5m}{2} + 4$ bits are stored. B_0 is indexed by the upper $\frac{m}{2}$ bits of both p and q and $\frac{m}{2} + 1$ bits are stored [6]. Note that the multiplier for “Improved” may be smaller.

Comparisons for square root reciprocal and for square root are similar to that in Table 1.

The proposed method produces an approximation with the same accuracy as the conventional linear approximations. Since addition is removed, one clock cycle may be saved. Furthermore, the required table size is about the half of that for the original linear approximation, and about the two thirds of that for the improved linear approximation.

Table 2 compares the table size required by “Linear,” “Improved,” and “Proposed” for double-precision (53-bit) division, with respect to the number of following Newton-Raphson iterations. (We assume that the reciprocal is computed with 54-bit accuracy.) Linear approximation methods may directly produce a reciprocal for single-precision (24-bit) division. For this, the proposed method requires a table of size $2^{12} \times 26 = 104K$ bits.

For double-precision square root, the proposed method for square root reciprocal requires a table of size $2^6 \times 14 = 896$ bits when followed by two Newton-Raphson iterations and a table of size $2^{13} \times 28 = 224K$ bits when followed by one iteration. The proposed method for square root directly produces single-precision square root by means of a look-up table of size $2^{10} \times 24 = 24K$ bits.

5 CONCLUDING REMARKS

We have proposed a new initial approximation method for multiplicative division and square root. It requires only one multiplication with a slight modification of the operand and produces an approximation with the same accuracy as the conventional linear approximations which require one multiplication and one addition. One clock cycle may be saved, and the required ROM size is also reduced.

REFERENCES

[1] I. Koren, *Computer Arithmetic Algorithms*. Prentice Hall, 1993.

[2] A.R. Omondi, *Computer Arithmetic Systems: Algorithms, Architecture, and Implementations*. Prentice Hall, 1994.

[3] D. DasSarma and D.W. Matula, “Measuring the Accuracy of ROM Reciprocal Tables,” *IEEE Trans. Computers*, vol. 43, no. 8, pp. 932-940, Aug. 1994.

[4] *Numerical Methods*. G. Dahlquist, A. Bjorck, and N. Anderson, eds., Prentice Hall, 1974.

[5] P.M. Farmwald, “High Bandwidth Evaluation of Elementary Functions,” *Proc. Fifth Symp. Computer Arithmetic*, pp. 139-142, 1981.

[6] M. Ito, N. Takagi, and S. Yajima, “Efficient Initial Approximation Methods for Division and Square Root Using a Multiply-Add Unit,” Technical Report HPC 55-10, Information Processing Soc. of Japan, Mar. 1995.

[7] M. Ito, N. Takagi, and S. Yajima, “Efficient Initial Approximation and Fast Converging Methods for Division and Square Root,” *Proc. 12th Symp. Computer Arithmetic*, pp. 2-9, July 1995.