Economic Pseudodivision Processes for Obtaining Square Root, Logarithm, and Arctan

B. P. SARKAR AND E. V. KRISHNAMURTHY

Abstract—Modified Meggitt methods (pseudodivision methods) are suggested for evaluating logarithm, arctan, and square root. The modifications described here consist in restricting the magnitude of the pseudopartial remainder such that the pseudoquotient assumes a form close to the minimal representation in the radix of choice. These methods will become useful for large-scale integrated system design.

Index Terms—Arctan, function evaluation logarithm, pseudodivision, square root.

I. INTRODUCTION

Meggitt [1] (see also Ralston and Wilf [2]) has described certain digit-by-digit methods for the evaluation of elementary functions like logarithm, arctan, and square root. These processes resemble repeated subtraction division (hence pseudodivision) and the accuracy of the process depends on the number of significant digits obtained for the pseudoquotient. In this note we shall consider a modification of these methods that makes these operations faster, requiring a smaller number of operations per quotient digit. This modification is based on restricting the magnitude of the pseudopartial remainder in the pseudodivision process which consequently results in the pseudoquotient's assuming a form close to the minimal representation in the given radix of choice. Incidentally, these modifications will be of interest for software or hardware realizations in conventional as well as signed-digit computer systems [3].

Since Meggitt has described the methods in great detail, we shall describe only the salient features of the modified method and indicate the departures from Meggitt's method, where required.

II. EVALUATION OF LOG $(1+y/x)$

The modified method consists of choosing the signed digits q_j such that

$$y + x = x \prod_{j=0} (1 + \alpha\beta^{-j})^{|q_j|} \tag{1}$$

Manuscript received November 24, 1967.

B. P. Sarkar was with the Computer Science Unit, Indian Statistical Institute, Calcutta, India. He is now with the Department of Applied Mathematics and School of Automation, Indian Institute of Science, Bangalore, India.

E. V. Krishnamurthy is with the Department of Applied Mathematics and School of Automation, Indian Institute of Science, Bangalore, India.

where y and x are given n-precision positive integers in radix β and

$$\alpha = \begin{cases} +1 & \text{if } q_j \text{ is positive} \\ -1 & \text{if } q_j \text{ is negative.} \end{cases} \tag{2}$$

Thus,

$$\log(1 + y/x) = \sum_{j=0} |q_j| \log(1 + \alpha\beta^{-j}), \tag{3}$$

all the logarithms being to the base β.

The calculation now consists of two parts; in the first part the digits q_j are obtained, while in the second part the product in (3) is formed using stored values of $\log(1+\alpha\beta^{-j})$.

Calculation of q_j: We suppose that the quantity

$$y - x\left\{\prod_{k=0}^{j-1} (1 + \alpha\beta^{-k})^{|q_k|} - 1\right\}$$

has been calculated where $q_0, q_1, \cdots, q_{j-1}$ are chosen such that a particular condition, called the magnitude condition, to be described later, is satisfied. For calculating q_j successive calculations of

$$y_{\pm a}^{(j)} = y - x\left\{\left[\prod_{k=0}^{j-1} (1+\alpha\beta^{-k})^{|q_k|}\right](1+\alpha\beta^{-j})^{|a|} - 1\right\} \tag{4}$$

for $a = 0, 1, 2, \cdots, q_j$ are made with sign of

$$a = \text{sign of } y_0^{(j)}. \tag{5}$$

(Note that we use parentheses for j when j is used as an index for the recursion cycle and no parentheses when j is used as an exponent.)

Define now

$$x_{\pm a}^{(j)} = x\left\{\prod_{k=0}^{j-1} (1 + \alpha\beta^{-k})^{|q_k|}\right\} (1 + \alpha\beta^{-j})^{|a|}. \tag{6}$$

The successive y's and x's can be obtained from the recurrence relations shown separately in the following for positive and negative quotient digits.

Recurrence Relations for Positive $y_0^{(j)}$: Start with

$$y_0^{(0)} = y, \qquad x_0^{(0)} = x$$

and define

$$y_{a+1}^{(j)} = y_a^{(j)} - \beta^{-j} x_a^{(j)} \tag{7a}$$

$$x_{a+1}^{(j)} = x_a^{(j)} + \beta^{-j} x_a^{(j)}. \tag{7b}$$

Then iterate until sign $(y_a^{(j)}) \neq$ sign $(y_{a+1}^{(j)})$ whence q_j is defined by the *magnitude condition*

$$q_j = \begin{cases} a + 1 & \text{if } y_a^{(j)} + y_{a+1}^{(j)} > 0 \ (\text{or } y_a^{(j)} > \tfrac{1}{2} x_a^{(j)} \beta^{-j}) \\ a & \text{otherwise.} \end{cases} \tag{7c}$$

In either case, the next cycle $(j+1)$ starts with

$$y_0^{(j+1)} = y_{q_j}^{(j)} \tag{7d}$$

and

$$x_0^{(j+1)} = x_{q_j}^{(j)}. \tag{7e}$$

Defining $Z_{\pm a}^{(j)} = \beta^j y_{\pm a}^{(j)}$, we can write the recurrence relations in (7) as follows. Start with

$$Z_0^{(0)} = y, \qquad x_0^{(0)} = x$$

and define

$$Z_{a+1}^{(j)} = Z_a^{(j)} - x_a^{(j)} \cdots \tag{8a}$$

$$x_{a+1}^{(j)} = x_a^{(j)} + \beta^{-j} x_a^{(j)} \cdots. \tag{8b}$$

Iterations continue until sign $(Z_a^{(j)}) \neq$ sign $(Z_{a+1}^{(j)})$ and the magnitude condition (7c) becomes

$$q_j = \begin{cases} a+1 & \text{if } Z_a^{(j)} + Z_{a+1}^{(j)} > 0 \text{ (or } Z_a^{(j)} > \frac{1}{2}x_a^{(j)}\text{)} \\ a & \text{otherwise.} \end{cases} \tag{8c}$$

The next cycle $(j+1)$ starts with

$$Z_0^{(j+1)} = \beta Z_{q_j}^{(j)} \cdots \tag{8d}$$

and

$$x_0^{(j+1)} = x_{q_j}^{(j)} \cdots. \tag{8e}$$

Recurrence Relations for Negative $y_0^{(j)}$*:* Following the notations used in (8a)–(8e), we can write the recurrences in this case as follows:

$$Z_{-a-1}^{(j)} = Z_{-a}^{(j)} + x_{-a}^{(j)} \cdots \tag{9a}$$

$$x_{-a-1}^{(j)} = x_{-a}^{(j)} - \beta^{-j} x_{-a}^{(j)} \cdots. \tag{9b}$$

Iterations continue until sign $(Z_{-a}^{(j)}) \neq$ sign $(Z_{-a-1}^{(j)})$. q_j is defined at this stage by the magnitude condition

$$-q_j = \begin{cases} -a-1 & \text{if } Z_{-a-1}^{(j)} + Z_{-a}^{(j)} < 0 \text{ (or } Z_{-a}^{(j)} < -\frac{1}{2}x_{-a}^{(j)}\text{)} \\ -a & \text{otherwise.} \end{cases} \tag{9c}$$

The next cycle $(j+1)$ begins with

$$Z_0^{(j+1)} = \beta Z_{-q_j}^{(j)}$$

and

$$x_0^{(j+1)} = x_{-q_j}^{(j)}.$$

Note that since x and y are assumed positive to start with, $Z_0^{(j)}$ can be positive for $j=0$ onwards and can be negative starting with $j=1$.

It may also be noted from the recurrence relations (8) and (9) that the magnitude condition determining the quotient digit q_j is different from that used by Meggitt [1], i.e., $Z_{q_j}^{(j)} \geq 0 > Z_{q_j+1}^{(j)}$. We have to consider the effect of dictating the magnitude of pseudopartial remainder (ppr) on the magnitude of the pseudoquotient digit. It is seen from (8b) and (9b) that the increment on $x_a^{(j)}$ is negligible for large j and also that the magnitude of quotient digit for large j cannot exceed $\beta/2$. For small j, however, we have to investigate in more detail as is done in the following analysis. We shall give a separate analysis for each of the four possible cases that may arise. We shall study the magnitude of q_{j+1}.

Case 1: Suppose the previous ppr $Z_{q_j}^{(j)}$ is positive and satisfies the magnitude condition (8c), i.e.,

$$Z_{q_j}^{(j)} \leq \tfrac{1}{2} x_{q_j}^{(j)}.$$

We have then

$$Z_0^{(j+1)} = \beta Z_{q_j}^{(j)} \leq \frac{\beta}{2} x_{q_j}^{(j)}.$$

Also since

$$x_a^{(j+1)} = x_0^{(j+1)} \{1 + \beta^{-(j+1)}\}^a \tag{10a}$$

we get using (8a), (8b), and (8e)

$$Z_a^{(j+1)} = Z_0^{(j+1)} - x_0^{(j+1)} \beta^{j+1} [\{1 + \beta^{-(j+1)}\}^a - 1]. \tag{10b}$$

Obviously q_{j+1} is maximum when $Z_a^{(j+1)}$ changes sign and equating the right-hand side of (10b) to zero, we get

$$a_{\max} = (q_{j+1})_{\max} \leq \left\lceil \frac{\log_\beta \left(1 + \dfrac{\beta}{2\beta^{(j+1)}}\right)}{\log_\beta (1 + \beta^{-(j+1)})} \right\rceil. \tag{11}$$

Here ⌈ ⌉ denotes the higher integral part of the factor inside.

Case 2: We assume that the previous ppr $Z_{q_j}^{(j)}$ is positive, not satisfying the magnitude condition (8c); and hence we iterated one more step, $Z_{q_j+1}^{(j)}$ thereby turning negative. Then,

$$0 > Z_{q_j+1}^{(j)} > -\tfrac{1}{2} x_{q_j}^{(j)} > -\tfrac{1}{2} x_{q_j+1}^{(j)}.$$

Thus

$$0 > Z_{q_j+1}^{(j)} > -\frac{1}{2} \frac{x_{q_j+1}^{(j)}}{(1 + \beta^{-j})}$$

and using (9d) and (9e),

$$0 > Z_0^{(j+1)} > -\frac{\beta}{2} \frac{x_0^{(j+1)}}{(1 + \beta^{-j})}.$$

We also get using (9)

$$x_{-a}^{(j+1)} = x_0^{(j+1)} \{1 - \beta^{-(j+1)}\}^a \tag{12a}$$

$$Z_{-a}^{(j+1)} = Z_0^{(j+1)} + x_0^{(j+1)} \beta^{(j+1)} [1 - \{1 - \beta^{-(j+1)}\}^a]. \tag{12b}$$

Using the same argument as in Case 1,

$$a_{\max} \leq \left\lceil \frac{\log_\beta \left(1 - \dfrac{1}{2\beta^j (1 + \beta^{-j})}\right)}{\log_\beta (1 - \beta^{-(j+1)})} \right\rceil. \tag{13}$$

Case 3: Let the previous ppr $Z_{-q_j}^{(j)}$ be negative, satisfying the magnitude condition (9c) so that

$$Z_0^{(j+1)} \geq -\frac{\beta}{2} x_0^{(j+1)}.$$

The expressions for $x_{-a}^{(j+1)}$ and $Z_{-a}^{(j+1)}$ are the same as in (12a) and (12b) and we obtain

$$a_{\max} \leq \left\lceil \frac{\log_\beta \left(1 - \dfrac{1}{2\beta^j}\right)}{\log_\beta (1 - \beta^{-(j+1)})} \right\rceil. \quad (14)$$

Case 4: Let the previous ppr $Z_{-q_j}^{(j)}$ be negative not satisfying the magnitude condition (9c) and hence we iterated one more step, $Z_{-(q+1)}^{(j)}$ thereby turning positive. As a consequence we have

$$0 < Z_{-(q_j+1)}^{(j)} < \frac{1}{2}\left(\frac{x_{-(q_j+1)}^{(j)}}{1 - \beta^{-j}}\right).$$

Thus

$$0 < Z_0^{(j+1)} < \frac{\beta}{2}\left(\frac{x_0^{(j+1)}}{1 - \beta^{-j}}\right).$$

Using the expressions (10a) and (10b) for $x_a^{(j+1)}$ and $Z_a^{(j+1)}$ we get

$$a_{\max} \leq \left\lceil \frac{\log_\beta \left(1 + \dfrac{1}{2\beta^j(1 - \beta^{-j})}\right)}{\log_\beta (1 + \beta^{-(j+1)})} \right\rceil. \quad (15)$$

For $j=0$, i.e., at the start, the magnitude of q_0 is determined by the initial restrictions on x and y. For example, if $1/\beta < y/x < \beta$, then $q_0 \leq s$ where $2^{s+1} \geq \beta + 1$. For radix 10, $q_0 \leq 3$ under such a restriction. As regards the magnitude of other digits, (12)–(15) give upper limits. For radix 10, these equations yield a $q_j = 6$; however, functional tabulation of the expression inside ⌈ ⌉ gives a value very close to 5, indicating that $|q_j| \leq 5$. Trial programs run on an IBM 1401 support this view.

Calculation of Logarithm from the q_j: The second part of the calculation consists in finding $\log(1 + y/x)$ using (3). Hence it is required to store the values of $\log(1 + \alpha\beta^{-j})$ for both $\alpha = +1$ and $\alpha = -1$, unlike Meggitt's scheme where we store values only for $\alpha = +1$.

III. Evaluation of $\mathrm{Tan}^{-1}(\frac{y}{x})$

The modified method consists in finding signed integers q_j such that

$$(x + iy) \prod_{j=0} (1 + i\alpha\beta^{-j})^{|q_j|} = R \quad (16)$$

where y and x are given n-precision positive integers in radix β and R is real. Further,

$$\alpha = \begin{cases} +1 & \text{if } q_j < 0 \\ -1 & \text{if } q_j > 0. \end{cases} \quad (17)$$

Then,

$$\log(x + iy) = \log R - \sum_{j=0} |q_j| \log(1 + i\alpha\beta^{-j}). \quad (18)$$

Setting

$$x + iy = re^{i \tan^{-1}(\frac{y}{x})}$$

and equating imaginary parts

$$\tan^{-1}\left(\frac{y}{x}\right) = \sum_{j=0} \alpha |q_j| \tan^{-1} \beta^{-j}. \quad (19)$$

The calculation of $\tan^{-1}(\frac{y}{x})$ then consists of two parts as in the case of logarithm, the first part being the calculation of q_j and the second part the calculation of actual $\tan^{-1}(\frac{y}{x})$ using (4) and the stored values of $\tan^{-1}\beta^{-j}$.

Defining $x_a^{(j)}$ and $y_a^{(j)}$ by the relation

$$x_a^{(j)} + iy_a^{(j)} = (x + iy) \prod_{k=0}^{j-1} (1 + \alpha i\beta^{-k})^{|q_k|}(1 + \alpha i\beta^{-j})^{|a|}$$

and also, as in the case of logarithm $Z_0^{(j+1)} = \beta Z_{q_j}^{(j)}$, $x_0^{(j+1)} = x_{q_j}^{(j)}$, and $Z_a^{(j)} = \beta^j y_a^{(j)}$, the following recurrence relations are obtained for positive and negative ppr.

Recurrence Relations for Positive $Z_0^{(j)}$: Start with

$$Z_0^{(0)} = y \quad \text{and} \quad x_0^{(0)} = x$$

and define

$$Z_{a+1}^{(j)} = Z_a^{(j)} - x_a^{(j)} \quad (20a)$$

$$x_{a+1}^{(j)} = x_a^{(j)} + \beta^{-2j} Z_a^{(j)}. \quad (20b)$$

The magnitude condition determining q_j is the same as in (8c).

Recurrence Relations for Negative $Z_0^{(j)}$: Define

$$Z_{-a-1}^{(j)} = Z_{-a}^{(j)} + x_{-a}^{(j)} \quad (21a)$$

$$x_{-a-1}^{(j)} = x_{-a}^{(j)} - \beta^{-2j} Z_{-a}^{(j)}. \quad (21b)$$

The magnitude condition determining $-q_j$ is the same as in (9c).

To determine the magnitude of the quotient digits, we note that in the positive case both $x_a^{(j)}$ and $Z_a^{(j)}$ are positive, while in the negative case $x_{-a}^{(j)}$ is positive but $Z_{-a}^{(j)}$ is negative. The magnitude condition then ensures that $|q_j| \leq [\frac{\beta}{2}]$ where [ ] denotes the integral part. Further, since

$$0 \leq \tan^{-1}\left(\frac{y}{x}\right) \leq \frac{\pi}{2},$$

it is obvious that $q_0 \leq 2$.

IV. Evaluation of Square Root

To obtain the square root of (y/x), the signed integers q_j' are found so that

$$y = x\left[\sum_{j=0} \alpha |q_j| \beta^{-j}\right]^2 \cdots \quad (22)$$

and thus

$$\sqrt{(y/x)} = \sum_{j=0} \alpha |q_j| \beta^{-j} \cdots. \quad (23)$$

Note that $\alpha=+1$ if $q_j>0$ and $\alpha=-1$, if $q_j<0$.

Calculation of the q_j is again a pseudodivision process. Having obtained $q_0, q_1, \cdots, q_{j-1}$, to obtain q_j successive calculation of

$$y_a^{(j)} = y - x\left[\sum_{k=0}^{j-1} \alpha \mid q_k \mid \beta^{-k} + \mid a \mid \alpha\beta^{-j}\right]^2$$

is done for $a=0, +1, +2, \cdots$, or $a=0, -1, -2, \cdots$, until the magnitude condition described earlier is satisfied. Defining

$$x^{(j)} = 2x\left[\sum_{k=0}^{j-1} \alpha \mid q_k \mid \beta^{-k} + \mid a \mid \alpha\beta^{-j}\right] + x\beta^{-j} \qquad (24a)$$

and

$$Z_a^{(j)} = \beta^j y_a^{(j)} \cdots, \qquad (24b)$$

the following recurrence relations are obtained for positive and negative partial remainders.

Recurrence Relations for Positive $Z_0^{(j)}$:

$$Z_{a+1}^{(j)} = Z_a^{(j)} - x_a^{(j)} \cdots \qquad (25a)$$

$$x_{a+1}^{(j)} = x_a^{(j)} + 2x\beta^{-j} \cdots. \qquad (25b)$$

We also get

$$Z_a^{(j)} = Z_0^{(j)} - ax_0^{(j)} - a(a-1)x\beta^{-j} \qquad (26a)$$

$$x_a^{(j)} = x_0^{(j)} + 2ax\beta^{-j} \cdots \qquad (26b)$$

when the magnitude condition is satisfied for the jth cycle; we set for the $(j+1)$th cycle

$$Z_0^{(j+1)} = \beta Z_q^{(j)} \cdots \qquad (27a)$$

and

$$x_0^{(+1)} = \left[x_q^{(j)} - \frac{\beta-1}{\beta} x\beta^{-j}\right] \cdots. \qquad (27b)$$

Recurrence Relations for Negative $Z_0^{(j)}$: The recursions for negative $Z_0^{(j)}$ take the following form:

$$Z_{-a-1}^{(j)} = Z_{-a}^{(j)} + x_{-a}^{(j)} \cdots \qquad (28a)$$

$$x_{-a-1}^{(j)} = x_{-a}^{(j)} - 2x\beta^{-j} \cdots. \qquad (28b)$$

Also,

$$Z_{-a}^{(j)} = Z_0^{(j)} + ax_0^{(j)} - a(a-1)x\beta^{-j} \qquad (29a)$$

$$x_{-a}^{(j)} = x_0^{(j)} - 2ax\beta^{-j} \cdots \qquad (29b)$$

when the magnitude condition is satisfied for the jth cycle; we set for $(j+1)$ cycle

$$Z_0^{(j+1)} = \beta Z_{-q_j}^{(j)} \cdots \qquad (30a)$$

and

$$x_0^{(j+1)} = x_{-q_j}^{(j)} - \frac{\beta-1}{\beta} x\beta^{-j} \cdots. \qquad (30b)$$

As regards the magnitude of the quotient digits, for large j, $|q_j| \leq \frac{\beta}{2}$. For small j we can carry out an analysis similar to the case of logarithm. In this case, however, the limits obtained in the case of logarithm may be exceeded by a small number. The details can be worked out as in the case of log. Only the broad outline is given below for the four cases as under logarithm.

Case 1: The previous ppr $Z_{q_j}^{(j)}$ is positive and satisfies the magnitude conditon (8c). Then, using the fact

$$Z_0^{(j+1)} = \beta Z_{q_j}^{(j)} \leq \frac{\beta}{2} x_{q_j}^{(j)}$$

together with the following equation obtained from (26a) and (26b).

$$Z_a^{(j+1)} = Z_0^{(j+1)} - ax_0^{(j+1)} - a(a-1)x\beta^{-(j+1)}, \qquad (31)$$

we can determine maximum value of a from the Diophantine system

$$\left(\frac{\beta}{2} - a\right)x_0^{(j+1)} + \left[\frac{\beta-1}{\beta} - \frac{a^2}{\beta} + \frac{a}{\beta}\right]\frac{x\beta^{-j}}{x_0^{(j+1)}} = 0.$$

Case 2: The previous ppr $Z_{q_j}^{(j)}$ is positive, not satisfying the magnitude condition (8c) and hence we iterated one more step, $Z_{q_j+1}^{(j)}$ thereby turning negative. Then for determining maximum a we use the inequality

$$0 > Z_0^{(j+1)} > -\frac{\beta}{2}\left[x_{q_j+1}^{(j)} - 2x\beta^{-j}\right]$$

together with the following equation obtained from (28a) and (28b):

$$Z_{-a}^{(j+1)} = Z_0^{(j+1)} + ax_0^{(j+1)} - a(a-1)x\beta^{-(j+1)}, \qquad (32)$$

The Diophantine system to determine maximum a is

$$\left(a - \frac{\beta}{2}\right)x_0^{(j+1)} - \left[\frac{\beta-1}{2} + \frac{a^2}{\beta} - \frac{a}{\beta} - \beta\right]x\beta^{-j} = 0.$$

Case 3: The previous ppr $Z_{-q_j}^{(j)}$ is negative, satisfying the magnitude condition (9c). Then the condition

$$Z_0^{(j+1)} \geq -\frac{\beta}{2}\left[x_0^{(j+1)} + \frac{\beta-1}{\beta} x\beta^{-j}\right]$$

together with (32) give the Diophantine system

$$\left(a - \frac{\beta}{2}\right)x_0^{(j+1)} - \left[\frac{\beta-1}{2} + \frac{a^2}{\beta} - \frac{a}{\beta}\right]x\beta^{-j} = 0.$$

from which we can determine maximum value of a.

Case 4: The previous ppr $Z_{-q_j}^{(j)}$ is negative, not satisfying (9c) and hence we iterated one more step, $Z_{-q_j-1}^{(j)}$ thereby turning positive. Then from the condition

$$Z_0^{(j+1)} < \frac{\beta}{2}\left[x_0^{(j+1)} - \frac{\beta-1}{\beta} x\beta^{-j}\right]$$

and (31), one can determine the maximum value of a by solving the Diophantine system

$$\left(\frac{\beta}{2} - a\right) x_0^{(j+1)} + \left[\frac{\beta - 1}{2} - \frac{a^2}{\beta} + \frac{a}{\beta} + \beta\right] x\beta^{-j} = 0.$$

V. Implementing the Check

An easy way of implementing the check as to whether the magnitude condition is satisfied at any particular stage is discussed below.

Case 1—Positive $Z_0^{(j)}$: Compute Z_{a+1} from Z_a and store Z_a. If sign of Z_{a+1}=sign of Z_a, compute x_{a+1} and replace Z_a by Z_{a+1}. Otherwise, check whether $Z_{a+1}+Z_a>0$. If so, compute x_{a+1}, and $q_j=a+1$. Otherwise, set $q_j=a$ and erase Z_{a+1}.

Case 2—Negative $Z_0^{(j)}$: Compute Z_{-a-1} from Z_{-a} and store Z_{-a}. If sign of Z_{-a-1}=sign of Z_{-a}, compute x_{-a-1} and store Z_{-a-1} in place of Z_{-a} and continue. Otherwise, check whether $Z_{-a-1}+Z_{-a}<0$. If so, compute x_{-a-1} and $q_j=-a-1$. Otherwise, set $q_j=-a$ and erase Z_{-a-1}.

Concluding Remarks

It can be shown by a simple calculation that the above algorithms require on the average 3.5 operations per quotient digit, while Meggitt's scheme requires 5.5 operations per quotient digit. It is to be noted that in the present algorithm the quotient digits are made to assume a form close to the minimal representation in the radix of choice. This, in particular, is very much like the class of division methods suggested by Robertson [4]. For an alternative minimal square rooting algorithm, suitable for binary radix, in which the number of nonzero digits is minimal, reference is made to Metze [5].

Note Added in Proof: Since the communication of this paper in 1967, the authors have come across an excellent article by Linhardt and Miller [6] that describes how digit-by-digit computation can speed up function computation by more than three times. Other interesting papers in this area are by Levy *et al.* [7] and Beelitz *et al.* [8], which are concerned with the system utilization of large-scale integration.

References

[1] J. E. Meggitt, "Pseudodivision and multiplication processes," *IBM J. Res. Dev.*, vol. 6, pp. 210–226, Apr. 1962.

[2] A. Ralston and H. S. Wilf, *Mathematical Methods for Digital Computers*. New York: Wiley, 1960.

[3] A. Avižienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 389–400, Sept. 1961.

[4] J. E. Robertson, "A new class of digital division methods," *IRE Trans. Electron. Comput.*, vol. EC-7, pp. 218–222, Sept. 1958.

[5] G. Metze, "Minimal square rooting," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 181–185, Apr. 1965.

[6] R. J. Linhardt and H. S. Miller, "Digit-by-digit transcendental-function computation," *RCA Rev.*, vol. 30, pp. 209–247, June 1969.

[7] S. Y. Levy *et al.*, "System utilization of large-scale integration," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 562–566, Oct. 1967.

[8] H. R. Beelitz *et al.*, "System architecture for large-scale integration," in *1967 Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 30. Washington, D. C.: Thompson, 1967, pp. 185–200.