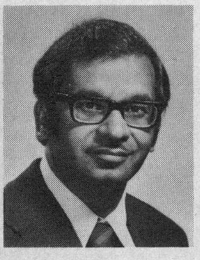
## Acknowledgment

The authors wish to thank M. S. Axelrod and G. A. Maley for their encouragement and discussions during the course of this work.

## References

[1] I. Flores, *The Logic of Computer Arithmetic.* Englewood Cliffs, N. J.: Prentice-Hall, 1963.
[2] O. L. MacSorley, "High-speed arithmetic in binary computers," *Proc. IRE,* vol. 49, pp. 67–91, Jan. 1961.
[3] G. A. Maley and E. J. Skiko, *Modern Digital Computers.* Englewood Cliffs, N. J.: Prentice-Hall, 1964.
[4] A. Svoboda, "Adder with distributed control," *IEEE Trans. Comput.,* vol. C-19, pp. 749–751, Aug. 1970.



**Shanker Singh** (S'67–M'68) was born in Agra, India, on May 5, 1936. He received the B.Sc. and M.Sc. degrees in physics from Agra University, Agra, India, in 1954 and 1957, respectively, the M.Sc. Tech. degree in electronics from the University of Rajasthan, Rajasthan, India, and the Ph.D. degree in electrical engineering from the University of Ottawa, Ottawa, Ont., Canada, in 1959 and 1968, respectively.

From 1960 to 1963 he was an Assistant Professor in the Department of Electrical Engineering, Birla Institute of Science and Technology, Pilani, India. In 1963 he joined the Department of Electrical Engineering, University of Ottawa, to do graduate work. In 1964 he was appointed Lecturer in the same department. In November 1968 he joined the Logic Technology group in the Component Division, IBM, East Fishkill, N. Y. In January 1970 he was promoted to the position of Staff Engineer. Presently he is with the Large Computer Systems group in the newly formed System Product Division of IBM, Poughkeepsie, N. Y. His current areas of interest are switching circuits theory, coding theory, and computer arithmetic. He is now engaged in the design of large computer processors. He is also involved in studies of various problems associated with large-scale integration such as testing, application, and computer hardware organization.

❖

**Ronald Waxman** (S'55–M'56–SM'69) was born in Newark, N. J., on November 28, 1933. He received the B.S.E.E. degree from Newark College of Engineering, Newark, N. J., in 1955 and the M.E.E. degree from Syracuse University, Syracuse, N. Y., in 1963.

From 1955 to the present he has been employed by IBM. Between 1955 and 1956 he worked principally in the area of circuit development. From 1956 to 1958, while on leave of absence from IBM, he served in the U. S. Air Force in an engineering capacity. In 1958 he returned to IBM where he continued his work in circuit development and also participated in the logic design of terminals for an airline reservation system. During the academic year 1962–1963 he attended Syracuse University full time under the IBM Resident Study Program. Since 1963 he has been involved in logic partitioning studies, system application of large-scale integration, and system architecture. He is currently working in the area of computer system development.

Mr. Waxman is a member of the Scientific Research Society of America, Phi Eta Sigma, and Eta Kappa Nu.

# Arithmetic Algorithms in a Negative Base

P. V. SANKAR, S. CHAKRABARTI, AND E. V. KRISHNAMURTHY

***Abstract*—Algorithms are described for the basic arithmetic operations and square rooting in a negative base. A new operation called polarization that reverses the sign of a number facilitates subtraction, using addition. Some special features of the negative-base arithmetic are also mentioned.**

***Index Terms*—Algorithms, basic arithmetic operations, negative base, polarization, square rooting.**

## I. Introduction

THE use of a negative base for representation of numbers has been suggested as early as 1957 by Wadel. Since then, a few papers have appeared on this subject [1]–[9]. This paper is concerned with the development of algorithms for the basic arithmetic operations in a negative base.

Let $a$ be a number in base $-\beta$ (where $\beta$ is a positive integer) given by

$$a = \sum_{i=0}^{m} a_i(-\beta)^i \tag{1}$$

where $0 \leq a_i \leq (\beta-1)$ with $a_m \neq 0$ (normalized form), unless otherwise specified; thus $a$ is negative or positive accordingly, as $m$ is odd or even, respectively.

Manuscript received July 24, 1971; revised July 19, 1972.
P. V. Sankar is with the Molecular Biophysics Unit, Indian Institute of Science, Bangalore-12, India.
S. Chakrabarti and E. V. Krishnamurthy are with the Department of Applied Mathematics, Indian Institute of Science, Bangalore-12, India.

## II. Basic Operations

### A. Addition

Addition in the negative-base number system is similar to that in the positive base but for the fact that we obtain a "twin carry" (a pair of carry digits), while adding two single digits. This is illustrated by the following example in base $-10$:

$$9+1 = 190$$
$$190 = 1(-10)^2+9(-10)+0 = 10_{(10)}$$

where a twin carry (1, 9) arises. In fact, it will be shown later that three different types of twin carry can arise when we add the operand digits and a previous carry. The digits in a twin carry belong to two successive digit positions, as illustrated in the example.

Although one could avoid using twin carry by determining the three possible carries 0, $+1$, or $-1$, we preferred using this since $-1$ has a valid representation in base $-\beta$ as a pair of digits 1, $(\beta-1)$; this avoids defining subtraction at this stage.

In general, the algorithm for the addition of any two numbers $a$ and $b$ (positive or negative) given by

$$a = \sum_{i=0}^{m} a_i(-\beta)^i \tag{2}$$

and

$$b = \sum_{i=0}^{n} b_i(-\beta)^i \tag{3}$$

proceeds finding the sum digit $s_i$ and the twin carry digits $c_{i+1}$, $d_{i+2}$ from a knowledge of $a_i$, $b_i$, $c_i$, and $d_{i+1}$ for each $i=0, 1, 2, \cdots, \max(m, n)+2$ as specified by the logic in Table I (the suffix here indicates the digit position to which $s$, $c$, or $d$ belong). The algorithm terminates when $i=\max(m, n)+2$ giving

$$a+b = \sum_{i=0}^{\max(m,n)+2} s_i(-\beta)^i. \tag{4}$$

*Example 1:* $-\beta=-10$, $a=1614097$ (positive), $b=416034$ (negative), and $s=11911$ (positive).

| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | $\overleftarrow{i}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | $d_i$ |
| | 0 | 9 | 0 | 9 | 1 | 9 | 9 | 0 | $c_i$ |
| | | | 4 | 1 | 6 | 0 | 3 | 4 | $b_i$ |
| | | 1 | 6 | 1 | 4 | 0 | 9 | 7 | $a_i$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 9 | 1 | 1 | $s_i$ |

### *B. Polarization Algorithm and Subtraction*

Since $a-b=a+(-b)$, to subtract $b$ from $a$, it is enough to replace $b$ by the representation of $-b$ in base $-\beta$ and then add this number. The operation of transforming $+b$ to $-b$ or vice versa in $-\beta$ representation is defined as polarization; this term is appropriate since the sign of the number only is reversed.

The polarization has no analog in positive-base arithmetic, as it is required only for subtraction. Thus it has a different function from the complement code used for representing the negative numbers in the positive base.

TABLE I
Logic for Addition

| Cases | $c_i$ | $d_{i+1}$ | $a_i+b_i+c_i$ | $s_i$ | $c_{i+1}$ | $d_{i+2}$ |
|---|---|---|---|---|---|---|
| 1 | $(\beta-1)$ | 1 | $\geq 2\beta$[a] | $(a_i+b_i+c_i) \bmod \beta$ | $(\beta-1)$ | 1 |
| | | | $\geq\beta$ | | 0 | 0 |
| | | | $<\beta$ | | 1 | 0 |
| 2 | 0 or 1 | 0 | $\geq\beta$ | $(a_i+b_i+c_i) \bmod \beta$ | $(\beta-1)$ | 1 |
| | | | $<\beta$ | | 0 | 0 |

[a] This case $(a_i+b_i+c_i)\geq 2\beta$ is not possible for base $-2$.

Let $a$ be the given number and $\bar{a}$ be the polarized form. Consider

$$a = \sum_{i=0}^{m} a_i(-\beta)^i. \tag{5}$$

We can equivalently write (5) as

$$a = -\left[(-\beta)^{m+1}+\sum_{i=1}^{m}(-\beta)^i(\beta+1-a_i)+(\beta-a_0)\right] \tag{6}$$

$$= -[\bar{a}] \tag{7}$$

where

$$\bar{a} = \sum_{i=0}^{m+1} \bar{a}_i(-\beta)^i. \tag{8}$$

The rules for polarization can now be derived from (6)–(8).

*Rule 1:* If $a_i \neq 0$, 1 for all $i$ then $\bar{a}_0=(\beta-a_0)$; $\bar{a}_i=(\beta+1-a_i)$ for $1\leq i\leq m$, and $\bar{a}_{m+1}=1$.

*Rule 2:* If $a_i=0$ for $0\leq i\leq j$ then $\bar{a}_i=0$ for $0\leq i\leq j$; $\bar{a}_{j+1}=(\beta-a_{j+1})$; $\bar{a}_i=(\beta+1-a_i)$ for $j+2\leq i\leq m$; and $\bar{a}_{m+1}=1$.

*Rule 3:* If some intermediate $a_i=0$ or 1 for $0<i\leq m$ then $\bar{a}_i=1$ or 0, respectively, and a twin carry $(-\beta)^{i+2}+(\beta-1)(-\beta)^{i+1}$ is generated; this is to be added to the remaining digits of $a$ before polarizing the $(i+1)$th digit.

This addition of the twin carry along with the polarization can be realized by adding a suitable binary variable $\delta_i$ as specified by the following algorithm.

*Algorithm:* At the start $(i=0)$, set $\delta_i=0$ and for each $i$ $(i=0, 1, 2, \cdots, m)$ follow the rules specified by the decision tree (Fig. 1). At the terminal step, set $\bar{a}_{m+1}=\delta_{m+1}$ and stop.

*Example 2:* $-\beta=-10$, $a=8019$, and $\bar{a}=12001$.

| 4 | 3 | 2 | 1 | 0 | $\overleftarrow{i}$ |
|---|---|---|---|---|---|
| | 0 | 0 | 1 | 0 | $\delta_i$ |
| | 8 | 0 | 1 | 9 | $a_i$ |
| 1 | 2 | 0 | 0 | 1 | $\bar{a}_i$ |

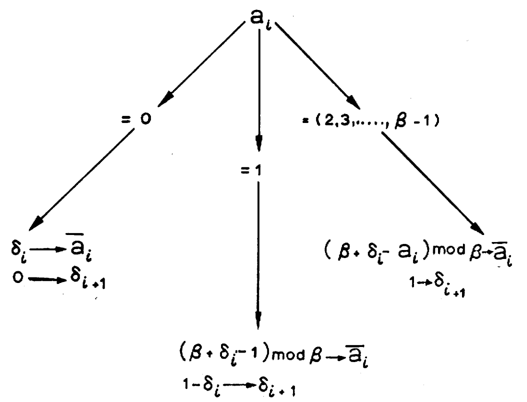



Fig. 1.

*Example 3:* $-\beta = -2$, $a = 110011001011$, and $\bar{a} = 0010001011001$.

| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | $\overleftarrow{i}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | $\delta_i$ |
| | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | $a_i$ |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | $\bar{a}_i$ |

*Remarks:* 1) From the polarization algorithm, we see that either $a(\neq 0)$ or its polarized form $\bar{a}$ (or both) will have 1 as the most significant digit. 2) The polarization either increases or decreases the length of a number by one digit; in the former case we say a polarization overflow has occurred. 3) An alternate way of interpreting subtraction is

$$\begin{aligned} a - b &= a + (-\beta)b + (\beta - 1)b \\ &= a + L(b) + (\beta - 1)b \end{aligned} \tag{9}$$

where $L(b)$ = left shift of $b$ by one digit. From (9) we see that addition is a fundamental operation in the negative base.

In the negative binary, (9) takes the form

$$a - b = a + b + L(b). \tag{10}$$

### C. Multiplication

Let $a$ and $b$ be, respectively, the multiplicand and multiplier (positive or negative) defined by

$$a = \sum_{j=0}^{m} a_j(-\beta)^j$$

$$b = \sum_{i=0}^{n} b_i(-\beta)^i. \tag{11}$$

The multiplication algorithm consists in forming the product of two single digits $P_i{}^j = a_j b_i$ for each $i$ and for all the values of $j$, and then forming the partial products $P_i$ and the final product $P$ (to be defined below) by shifts and additions, as specified by the following recursions.

For each $i$, form

$$P_i{}^j = a_j b_i, \qquad j = 0, 1, 2, \cdots, m \tag{12a}$$

then obtain

$$P_i = \sum_{j=0}^{m} P_i{}^j(-\beta)^j \tag{12b}$$

and

$$P = \sum_{i=0}^{n} P_i(-\beta)^i. \tag{12c}$$

As the sign of a number is implicit in the negative-base representation, the multiplication of two operands of different signs directly results in a product with the correct sign. Since

$$a_{\max} = ((-\beta)^{m+2} - 1)/(\beta + 1) \tag{13a}$$

and

$$b_{\max} = ((-\beta)^{n+2} - 1)/(\beta + 1) \tag{13b}$$

it can easily be proved that $P$ can have at most $(m+n+3)$ digits.

*Example 4:* $-\beta = -10$, $a = 5378$, $b = 37$, and $P = 1911686$.

| 6 | 5 | 4 | 3 | 2 | 1 | 0 | $\leftarrow i, j$ |
|---|---|---|---|---|---|---|---|
| | | | 5 | 3 | 7 | 8 | $a_j$ |
| | | | | | 3 | 7 | $b_i$ |
| | | | | 1 | 5 | 6 | $P_0{}^0$ |
| | | | 1 | 6 | 9 | 0 | $P_0{}^1(-10)$ |
| | | 1 | 8 | 1 | 0 | 0 | $P_0{}^2(-10)^2$ |
| | 1 | 7 | 5 | 0 | 0 | 0 | $P_0{}^3(-10)^3$ |
| | 1 | 7 | 4 | 7 | 4 | 6 | $P_0$ |
| | | | | 1 | 8 | 4 | $P_1{}^0$ |
| | | | 1 | 8 | 1 | 0 | $P_1{}^1(-10)$ |
| | | 0 | 0 | 9 | 0 | 0 | $P_1{}^2(-10)^2$ |
| | 1 | 9 | 5 | 0 | 0 | 0 | $P_1{}^3(-10)^3$ |
| 1 | 9 | 5 | 8 | 9 | 4 | 0 | $P_1(-10)$ |
| 1 | 9 | 1 | 1 | 6 | 8 | 6 | $P$ |

### D. Division

Let us represent the normalized $(n+1)$ digit dividend by $A$, the $(d+1)$ digit divisor by $B$, and the $(m+1)$ digit quotient as $Q$ in base $-\beta$, in floating-point-integral mantissa form. Thus

$$A = (-\beta)^{e_a} \cdot a = (-\beta)^{e_a} \cdot \sum_{j=0}^{n} a_j(-\beta)^j \tag{14a}$$

$$B = (-\beta)^{e_b} \cdot b = (-\beta)^{e_b} \cdot \sum_{j=0}^{d} b_j(-\beta)^j \tag{14b}$$

$$Q = (-\beta)^{e_q} \cdot q = (-\beta)^{e_q} \cdot \sum_{j=0}^{m} q_j(-\beta)^j \tag{14c}$$

where $e_a$, $e_b$, and $e_q$ are the exponents and $a$, $b$, and $q$ are the integral mantissa, respectively.

The nonrestoring division algorithm used in the positive base can be extended to the negative base. This algorithm consists of a sequence of subtractions (polarized addition) and shifts. The steps of the algorithm follow.

*Step 1:* Prefix two leading zeros to the dividend $a$ to prevent initial overflow of the quotient (see Remark 1 following Step 4); this dividend is denoted as the initial partial remainder $R_n$; the number of digits in $R_n$ is taken as $(n+1)$. Prefix one more leading zero to $R_n$ to match the extra digit of the polarized divisor $\bar{b}$ (in the case of polarization overflow).

*Step 2:* Form

$$R_{n-1} = R_n - b(-\beta)^{n-d} = R_n + \bar{b}(-\beta)^{n-d} \tag{15}$$

which will result in

$$q_n = 1$$

and

$$\operatorname{sgn} R_{n-1} \neq \operatorname{sgn} R_n.$$

From now on obtain $q_j$ for each $j = (n-1), (n-2), \cdots, (n-m)$ using the recursion

$$R_{j-1} = R_j + q_j\bar{b}(-\beta)^{j-d} \tag{16}$$

where $q_j$ indicates the number of times $\bar{b}(-\beta)^{j-d}$ is added to $R_j$ until

$$\operatorname{sgn} R_{j-1} \neq \operatorname{sgn} R_j. \tag{17}$$

(In Remark 2 we will explain that this condition (17) can lead to a quotient digit $q_j = +\beta$.)

*Step 3:* Obtain the actual quotient by converting $q_j = +\beta$, if any. This is done as follows.

a) If $q_{j+p} \neq \beta$ is followed by $p$(odd) quotient digits $q_{j+p-1} = q_{j+p-2} = \cdots = q_j = +\beta$, then due to borrow propagation, the actual quotient digits $q_j{}^*$ are given by

$$q_{j+i}{}^* = (\beta - 1), \qquad \text{for } i = 1, 3, \cdots, (p-2)$$
$$q_{j+i}{}^* = 0, \qquad \text{for } i = 0, 2, \cdots, (p-1)$$

and

$$q_{j+p}{}^* = (q_{j+p} - 1).$$

b) If $q_{j+p} \neq \beta$ is followed by $p$(even) quotient digits $q_{j+p-1} = q_{j+p-2} = \cdots = q_j + \beta$, then due to borrow propagation the actual quotient digits $q_j{}^*$ are given by

$$q_{j+i}{}^* = (\beta - 1), \qquad \text{for } i = 1, 3, \cdots, (p-1)$$
$$q_{j+i}{}^* = 0, \qquad \text{for } i = 0, 2, \cdots, (p-2)$$

and

$$q_{j+p}{}^* = q_{j+p}.$$

*Step 4:* If we stop after obtaining $q_{n-m}$

$$e_q = e_a - e_b + n - m - d.$$

*Remark 1—Initial Overflow:* If we assume that the most significant digit of the divisor $b$ is aligned with the most significant digit of the dividend $a$ for division, then since

$$a_{\max} \mathrel{\bar{\div}} (-\beta)^{n+2}/(\beta+1) \tag{18}$$

and

$$b_{\min} \mathrel{\bar{\div}} (-\beta)^{n}/(\beta+1) \tag{19}$$

$$q_{\max} \mathrel{\bar{\div}} (a/b)_{\max} = (-\beta)^2 \tag{20}$$

(where $\bar{\div}$ denotes approximately equal to).

Hence to prevent quotient overflow we prefix two zeros to $a$ (Step 1) and choose $q_n = 1$. If there is no overflow then $q_{n-1} = \beta$ thereby resulting in $q_n = q_{n-1} = 0$ (see Example 5).

*Remark 2—Possibility of $q_j = +\beta$:* Since

$$(R_j)_{\max} = b(-\beta)^{j-d+1} - 1$$

if we choose $q_j = (\beta - 1)$ then

$$R_{j-1} = R_j - (\beta - 1)b(-\beta)^{j-d} \geq 0$$

that will not satisfy the condition (17) unless $q_j = +\beta$.

A similar result can be proved for $R_j$ negative. Also, since the nonrestoring division does not permit $q_j = 0$, the only manner by which this can occur is as an ordered pair $(1, \beta)$.

*Example 5:* $-\beta = -10$, $a = 000136$, $e_a = 0$, $b = 16$, $e_b = 0$, $n+1 = 5$, $d+1 = 2$, $m+1 = 4$, $q = 1(190)21 = 21$, and $e_q = 0$.

| $\overrightarrow{j}$ | | 4 | 3 | 2 | 1 | 0 | $q_j \downarrow$ |
|---|---|---|---|---|---|---|---|
| $R_4$ | 0 | 0 | 0 | 1 | 3 | 6 | |
| $\bar{b}\cdot(-10)^3$ | 0 | 0 | 4 | 0 | 0 | 0 | |
| $R_3$ | 0 | 0 | 4 | 1 | 3 | 6 | 1 |
| $\bar{b}\cdot(-10)^2$ | 0 | 0 | 0 | 4 | 0 | 0 | |
| | | | 4 | 5 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 4 | 9 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 3 | 3 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 3 | 7 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 2 | 1 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 2 | 5 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 2 | 9 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 1 | 3 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| | | | 1 | 7 | 3 | 6 | |
| | | | 0 | 4 | 0 | 0 | |
| $R_2$ | | | 0 | 1 | 3 | 6 | 190 |
| $\bar{b}\cdot(-10)$ | | | | 0 | 4 | 0 | |
| | | | | 1 | 7 | 6 | |
| | | | | | 4 | 0 | |
| $R_1$ | | | | 0 | 1 | 6 | 2 |
| $\bar{b}$ | | | | | 0 | 4 | |
| | | | | 0 | 0 | 0 | 1 |

*E. Square Rooting*

This algorithm, as in the positive base [10]–[13], resembles division, except that the divisor at each step changes according to certain predetermined rules. The most significant digit of the square root is determined by finding out the largest square of a natural number $\leq(\beta-1)$, that is either equal to or less than the first digit (or the first three digits, as the case may be) of the given number. It may be remarked that the pairing of digits from the least significant end is not needed here, as it is in the positive base, since positive numbers in a negative base can have only an odd number of digits that when paired will always leave a single digit at the most significant end. This algorithm proceeds by choosing the divisor from the set of certain prescribed odd numbers in ascending order and subtracting (polarized adding) until the partial remainder changes its sign (or changes the parity of the number of digits); the quotient is given by the number of subtractions in each step.

The principle of this algorithm is as follows. Let $a$ be a number in $-\beta$ representation given by

$$a = \sum_{i=0}^{m} a_i(-\beta)^i. \tag{21}$$

Then

$$\begin{aligned} a^2 = {} & a_m{}^2(-\beta)^{2m} + a_{m-1}{}^2(-\beta)^{2(m-1)} + \cdots + a_0{}^2 \\ & + 2a_m a_{m-1}(-\beta)^{2m-1} + \cdots + 2a_m a_1(-\beta)^{m+1} \\ & + 2a_m a_0(-\beta)^m + \cdots + 2a_1 a_0(-\beta). \end{aligned} \tag{22}$$

Equation (22) can be written as

$$\begin{aligned} a^2 = {} & a_m{}^2(-\beta)^{2m} + [2a_m(-\beta)^m + a_{m-1}(-\beta)^{m-1}]a_{m-1}(-\beta)^{m-1} \\ & + [2(a_m(-\beta)^m + a_{m-1}(-\beta)^{m-1}) + a_{m-2}(-\beta)^{m-2}] \\ & \cdot a_{m-2}(-\beta)^{m-2} + \cdots + [2(a_m(-\beta)^m + a_{m-1}(-\beta)^{m-1} \\ & + \cdots + a_1(-\beta)) + a_0]\, a_0. \end{aligned} \tag{23}$$

If we set

$$a^2 = b = \sum_{i=0}^{2m} b_i(-\beta)^i \tag{24}$$

then we can use (24) to write the recursions for the square-rooting algorithm.

*Steps of the Algorithm:* Start with the given $(2m+1)$ digit number $b$ as the dividend ($m$th partial remainder $R_i{}^0$, $i=m$). To obtain the most significant digit $a_m$ of the square root, subtract (or polarize and add) from the $2m$th digit position of $b$, odd numbers of the form $(2j-1)$ for $j=1, 2, \cdots, (\beta-1)$ until the partial remainder $R_i{}^j(i=m)$ changes sign. ($R_i{}^j$ refers to the partial remainder after each subtraction $j$ corresponding to the given digit position $2i$.) In such a case, the next partial remainder $R_{i-1}{}^0$ equals $R_i{}^j$ and the quotient digit $a_i$ equals $j(i=m)$. For obtaining the successive digits, the following recursions are used for $i=(m-1)$, $(m-2)$, $\cdots, 1, 0$.

$$\begin{aligned} R_i{}^j &= R_i{}^{j-1} - D_i{}^j(-\beta)^{2i} \\ &= R_i{}^{j-1} + \overline{D}_i{}^j(-\beta)^{2i} \end{aligned} \tag{25}$$

until

$$\operatorname{sgn} R_i{}^j \neq \operatorname{sgn} R_i{}^{j-1}$$

where

$$D_i{}^j = 2C_{i+1}(-\beta) + 2j - 1 \tag{26}$$

and

$$C_i = C_{i+1}(-\beta) + a_i \tag{27}$$

and

$$C_{m+1} = 0. \tag{28}$$

Then

$$a_i = j.$$

This algorithm terminates after finding $a_0$. (The algorithm could be continued by adding extra zeros and an appropriate exponent.)

*Example 6:* $-\beta=-10$, $b=14641$, and $\sqrt{b}=a=121$.

| 5 | 4 | 3 | 2 | 1 | $\overleftarrow{i}$ 0 | Quantity | Quotient |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 4 | 6 | 4 | 1 | $b=R_2{}^0$ | |
| 1 | 9 | 0 | 0 | 0 | 0 | $\bar{1}\cdot(-10)^4$ | |
| 0 | 0 | 4 | 6 | 4 | 1 | $R_2{}^1=R_1{}^0$ | $a_2=1$ |
| | 1 | 9 | 9 | 0 | 0 | $\overline{(2a_2(-10)+1)}(-10)^2$ | |
| | 0 | 2 | 5 | 4 | 1 | $R_1{}^1$ | |
| | 1 | 9 | 7 | 0 | 0 | $\overline{(2a_2(-10)+3)}(-10)^2$ | |
| | 0 | 0 | 2 | 4 | 1 | $R_1{}^2=R_0{}^0$ | $a_1=2$ |
| | | 1 | 9 | 7 | 9 | $\overline{2a_2(-10)^2+2a_1(-10)+1}$ | |
| | | 0 | 0 | 0 | 0 | $R_0{}^1$ | $a_0=1$ |

*Remarks:* 1) If $b$ is negative, this algorithm diverges. 2) It is possible to generate either the positive or the negative square root by shifting the initial position of the divisor (to the left or right) through two digits.


## Acknowledgment

The authors wish to thank the Editor and the referees for their valuable comments in revising this paper. Also, two of the authors (P. V. Sankar and S. Chakrabarti) express their gratitude to the Indian Institute of Science, Bangalore, for providing research facilities.


## References


[1] L. B. Wadel, "Negative base number systems," *IRE Trans. Electron. Comput.* (Corresp.), vol. EC-6, p. 123, June 1957.

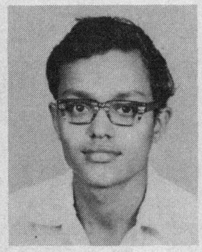
[2] ——, "Conversion from conventional to negative-base number representation," *IRE Trans. Electron. Comput.* (Corresp.), vol. EC-10, p. 779, Dec. 1961.

[3] D. L. Dietmeyer, "Conversion from positive to negative and imaginary radix," *IRE Trans. Electron. Comput.* (Corresp.), vol. EC-12, pp. 20–22, Feb. 1963.

[4] S. Zohar, "Negative radix conversion," *IEEE Trans. Comput.*, vol. C-19, pp. 222–226, Mar. 1970.

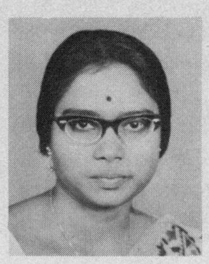
[5] E. V. Krishnamurthy, "Complementary two-way algorithms for negative radix conversions," *IEEE Trans. Comput.*, vol. C-20, pp. 543–550, May 1971.

[6] L. B. Wadel, "Comment on 'negative-radix conversion'," *IEEE Trans. Comput.* (Corresp)., vol. C-20, p. 587, May 1971.
[7] Z. Pawlak, "Another comment on 'negative-radix conversions'," *IEEE Trans. Comput.*, vol. C-20, p. 587, May 1971.
[8] S. Zohar, "Author's reply," *IEEE Trans. Comput.*, vol. C-20, p. 587, May 1971.
[9] M. P. De Regt, "Negative radix arithmetic," *Comput. Design*, vol. 6, pp. 52–63, May 1967.
[10] R. K. Richards, *Arithmetic Operations in Digital Computers*. Princeton, N. J.: Van Nostrand, 1956.
[11] W. S. Humphrey, Jr., *Switching Circuits in Computer Applications*. New York: McGraw-Hill, 1959, pp. 8–9.
[12] D. Cowgill, "Logic equations for a built in square-root method," *IEEE Trans. Electron. Comput.* (Corresp.), vol. 13, pp. 156–157, Apr. 1964.
[13] E. H. Lenaerts, "Automatic square-rooting," *Electron. Eng.*, vol. 27, pp. 287–289, July 1955.

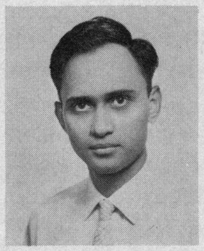
**P. V. Sankar** was born in Madras, India, on June 20, 1948. He received the M.Sc. degree in physics from the University of Madras, Madras, India, in 1969.

He is currently a Research Fellow in the Department of Molecular Biophysics of the Indian Institute of Science, Bangalore, India. His interests lie in the areas of information and computer science and algorithmic graph theory.



**S. Chakrabarti** was born in Calcutta, India, on January 10, 1944. She received the M.Sc. degree in mathematics from the University of Calcutta, Calcutta, India, in 1965.

She is currently a Research Fellow in the Department of Applied Mathematics, Indian Institute of Science, Bangalore, India. Her interests lie in the areas of logical design and numerical techniques.



**E. V. Krishnamurthy** was born in Madras, India, on June 18, 1934. He received the M.Sc. and Ph.D. degrees in physics from the University of Madras, Madras, India, in 1955 and 1959, respectively.

He has been a member of the faculties of the University of Madras, the Indian Statistical Institute, Calcutta, the University of Chicago, Chicago, Ill., the Weizmann Institute of Science, Rehovot, Israel, and the Technion—Israel Institute of Technology, Haifa. He is currently a Professor in the Departments of Applied Mathematics and Molecular Biophysics and the School of Automation in the Indian Institute of Science, Bangalore, India. He has published numerous technical papers in computer and information sciences, reliability theory, theoretical biology, instrumentation, and recreational mathematics.

Dr. Krishnamurthy was elected a Fellow of the Indian Academy of Sciences in 1972.

# Deterministic Division Algorithm in a Negative Base

P. V. SANKAR, S. CHAKRABARTI, AND E. V. KRISHNAMURTHY

*Abstract*—**Described here is a deterministic division algorithm in a negative-base number system; here, the divisor is mapped into a suitable range by premultiplication, so that the choice of the quotient digit is deterministic.**

*Index Terms*—**Deterministic division, negative base, range transformation.**

## I. Introduction

RECENTLY, deterministic division algorithms [1], [2] have been described for conventional and signed-digit number systems; these algorithms transform the divisor to a suitable range by premultiplication, so that the choice of the quotient digit is deterministic, without any need for a trial and error process. It is possible to develop a similar algorithm for division in a negative-base number system [3]. Let us denote the $(n+1)$ digit dividend as $A$, the $(d+1)$ digit divisor as $B$, and the $(m+1)$ digit quotient as $Q$ in floating-point form (integral mantissa) in base $-\beta$. Thus

$$A = (-\beta)^{e_a} \cdot a = (-\beta)^{e_a} \sum_{j=0}^{n} a_j(-\beta)^j \tag{1a}$$

$$B = (-\beta)^{e_b} \cdot b = (-\beta)^{e_b} \sum_{j=0}^{d} b_j(-\beta)^j \tag{1b}$$

$$Q = (-\beta)^{e_q} \cdot q = (-\beta)^{e_q} \sum_{j=0}^{m} q_j(-\beta)^j \tag{1c}$$

where $e_a$, $e_b$, and $e_q$ are the exponents and $a$, $b$, and $q$ are the mantissas, respectively.

## II. Notation and Definitions

The same notations and definitions as in [3] (for negative base) are used. However, for the sake of convenience, we define the following.

Manuscript received July 24, 1971; revised July 19, 1972.
P. V. Sankar is with the Molecular Biophysics Unit, Indian Institute of Science, Bangalore-12, India.
S. Chakrabarti and E. V. Krishnamurthy are with the Department of Applied Mathematics, Indian Institute of Science, Bangalore-12, India.