Signed-Digit Numbe Representations for Fast Parallel Arithmetic*

ALGIRDAS AVIZIENIS†, MEMBER, IRE

Summary—**This paper describes a class of number representations which are called *signed-digit* representations. Signed-digit representations limit carry-propagation to one position to the left during the operations of addition and subtraction in digital computers. Carry-propagation chains are eliminated by the use of redundant representations for the operands. Redundancy in the number representation allows a method of fast addition and subtraction in which each sum (or difference) digit is the function only of the digits in two adjacent digital positions of the operands. The addition time for signed-digit numbers of any length is equal to the addition time for two digits. The paper discusses the properties of signed-digit representations and arithmetic operations with signed-digit numbers: addition, subtraction, multiplication, division and roundoff. A brief discussion of logical design problems for a signed-digit adder concludes the presentation.**

I. Introduction

THIS PAPER describes a class of number representations which are called *signed-digit* representations. Signed-digit representations limit carry propagation to one position to the left during the operations of addition and subtraction in digital computers. Carry-propagation chains are eliminated by the use of redundant representations for the operands. In a *conventional number representation* with an integer radix $r>1$ each digit is allowed to assume exactly r values: $0, 1, \cdots, r-1$. In a *redundant representation* with the same radix r each digit is allowed to assume more than r values.

Previous methods of carry elimination formed redundant representations by the combination of explicitly identified digits, such as stored carries or borrows, with conventional number representations.[1–6] A difficulty in redundant representations of the stored-carry type is the lack of unique representation for the zero algebraic value of a number; the former sign digit (now indicator digit) does not always indicate the true sign of a number and overflow detection becomes more complicated. Furthermore, the number of states used to represent the values of one digital position is doubled by the binary stored-carry digit; this amount of redundancy is excessive.

In the method described here, each digit of a positional constant radix number representation with an integer radix r is allowed to assume q values

$$r+2 \leq q \leq 2r-1,$$

that is, more than the r values allowed in the conventional representation.[7] Both positive and negative digit values are allowed for this purpose. Redundancy in the number representation allows a method of fast addition and subtraction in which each sum (or difference) digit is the function only of the digits in two adjacent digital positions of the operands. These operations are called *totally-parallel* addition and subtraction. The requirement for totally-parallel addition and subtraction determines the minimum redundancy ($r+2$ values) which is necessary in the representation of one digit. The upper limit for the redundancy of digit values results from the requirement for a unique representation of the zero algebraic value of a number. To satisfy this requirement the magnitude of allowed digit values may not exceed $r-1$.

Requirements of totally-parallel addition and subtraction and of a unique representation for the zero value are satisfied by a class of redundant representations for radices $r>2$ which are called *signed-digit* representations. The digits of a signed-digit representation individually assume both positive and negative integer values and contain the sign information for the number; no special sign digit is necessary. This property leads to the name of "signed-digit" representations. The number of digit values in a radix $r>2$ representation ranges from a required minimum of $r+2$ to an allowable maximum of $2r-1$.

The following sections of this paper describe the properties of individual digits and of complete numbers in signed-digit representations. After development of these properties, arithmetic operations for signed-digit representations are discussed. Examples are given of

* Received by the PGEC, March 23, 1961; revised manuscript received, May 22, 1961.

† Jet Propulsion Lab., California Inst. Tech., Pasadena, Calif.

[1] A. W. Burks, *et al.*, "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," Institute for Advanced Study, Princeton, N. J.; pt. 1, vol. I, 2nd ed. revised; September, 1947.

[2] J. E. Robertson, "Preliminary Design of an Arithmetic Unit for Use with a Self-Checking Binary Parallel Digital Computer," Digital Computer Lab., University of Illinois, Urbana, Ill., Internal Rept. No. 19; June, 1950.

[3] Staff of the Digital Computer Lab., "On the Design of a Very High-Speed Computer," University of Illinois, Urbana, Ill., Rept. No. 80, ch. 8; October, 1957.

[4] G. Metze, "A Study of Parallel One's Complement Arithmetic Units with Separate Carry or Borrow Storage," Ph.D. dissertation, University of Illinois, Urbana, Ill., 1958; Digital Computer Lab., University of Illinois, Rept. No. 81; November 11, 1957.

[5] G. Metze and J. E. Robertson, "Elimination of carry propagation in digital computers," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959; pp. 389–396.

[6] J. E. Robertson, "Redundant number systems for digital computer arithmetic," notes for the University of Michigan Engrg. Summer Conf., "Topics in the Design of Digital Computing Machines," Ann Arbor, Mich.; July 6–10, 1959.

[7] A. Avizienis, "A Study of Redundant Number Representations for Parallel Digital Computers," Ph.D. dissertation, University of Illinois, Urbana, Ill., 1960; Digital Computer Laboratory, University of Illinois, Rept. No. 101; May 20, 1960.

signed-digit addition, subtraction, multiplication and division. The paper concludes with some logical design considerations for a signed-digit adder.

II. Properties of Signed-Digit Representations

In this section the class of signed-digit representations is derived according to four requirements which are postulated as necessary for number representations in fast parallel arithmetic.

A. Requirements for Signed-Digit Representations

The purpose of signed-digit representations is to allow addition and subtraction of two numbers in which no serial signal propagation is required along the adder; that is, the time duration of the operation is independent of the length of the operands and is equal to the time required for the addition or subtraction of two digits. Furthermore, the signed-digit representations must have a unique representation of zero algebraic value of a number. The desired principal properties of signed-digit representations are specified by the following description of a signed-digit number:

A *signed-digit number* is represented by $n+m+1$ digits z_i $(i=-n, \cdots, -1, 0, 1, \cdots, m)$ and has the algebraic value

$$Z = \sum_{i=-n}^{m} z_i r^{-i},$$

where the values of r and z_i are such that the following requirements are satisfied:

1) The radix r is a positive integer.
2) The algebraic value $Z=0$ has a unique representation.
3) There exist transformations between the conventional representation and the signed-digit representation for every algebraic value Z within a specified range.
4) Totally-parallel addition and subtraction is possible for all digits in corresponding positions of two representations.

The arithmetic operations of totally-parallel addition and subtraction of two digits z_i and y_i from the corresponding ith positions of the representations of numbers Z and Y are defined as follows:

Definition 1: Addition of digits z_i and y_i is totally-parallel if the following two conditions are satisfied:

1) The *sum digit* s_i (ith digit of the sum $S=Z+Y$) is a function only of the *augend digit* z_i, *addend digit* y_i and the *transfer*[8] *digit* t_i from the $(i+1)$th position on the right: $s_i=f(z_i, y_i, t_i)$.

[8] The term "transfer digit" is used here instead of the commonly used terms "carry" or "borrow" for two reasons:

1) the transfer digit may assume both positive and negative values in either addition or subtraction;
2) unlike the "carry" or "borrow" of conventional addition or subtraction, the transfer digit is never propagated past the first adder position on the left.

2) The transfer digit t_{i-1} to the $(i-1)$th position on the left is a function only of the augend digit z_i and the addend digit y_i: $t_{i-1}=f(z_i, y_i)$.

Definition 2: Totally-parallel subtraction of the *subtrahend digit* y_i from the *minuend digit* z_i is performed as the totally-parallel addition of the additive inverse of y_i, *i.e.*, $z_i-y_i=z_i+(-y_i)$.

Definitions 1 and 2 impose limiting conditions on the values which the digits z_i and y_i may assume. The addition of two digits is performed in two successive steps. First, an outgoing transfer digit t_{i-1} and an *interim sum digit* w_i are formed:

$$z_i + y_i = rt_{i-1} + w_i. \tag{1}$$

Then the sum digit s_i is formed:

$$s_i = w_i + t_i. \tag{2}$$

Definition 1 will be satisfied if the range of values which s_i may assume in (2) does not exceed the allowed range of values for the digits z_i and y_i in (1). The block diagram of a totally-parallel adder for signed-digit representations is shown in Fig. 1.

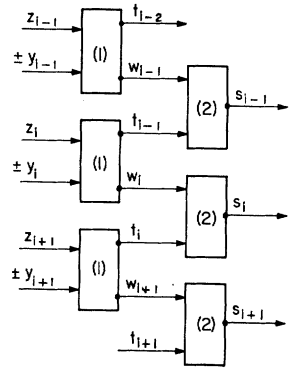

Fig. 1—Section of a totally-parallel adder for signed-digit representation.

Definition 2 will be satisfied if, for every allowed nonzero value of the digit y_i, there exists an additive inverse in the set of all allowed values of y_i;

for every $y_i=a$, there exists $y_i=-a$ such that $a+(-a)=0$. (3)

The requirement for unique representation of the zero value of a number will be satisfied by the condition

$$|z_i| \leq r-1. \tag{4}$$

The requirement for conversion from the conventional sign and magnitude, m-digit representation to signed-digit representation (m digits long), will be satisfied if the procedure of totally-parallel addition, applied to the nonsign digits x_i of the conventional representation,

$$x_i = rt_{i-1} + w_i \tag{5}$$

$$z_i = w_i + t_i, \tag{6}$$

will yield an allowed value of digit z_i for every positive and negative value of the digit x_i ($x_i = 0, 1, \cdots, r-1$) in the conventional radix r representation. Conversion from signed-digit to conventional representation is always possible, since condition (4) allows the interpretation of any signed-digit representation as the sum of two conventional sign and magnitude representations, one consisting of the positive and one of the negative digits of the signed-digit representation.

B. Required and Allowed Values of Digits

The conditions (1)–(6) of the preceding section determine the required and allowed values of one digit z_i for an arbitrary integer radix $r \geq 2$. As the first step, the required and allowed values of t_i and w_i in (1) and (2) must be established. The conditions (1)–(4) establish[9] the set of values

$$t_i = -1, 0, 1 \tag{7}$$

as the least sufficient set of values for the transfer digit t_i, and the condition

$$|w_i| \leq r - 2 \tag{8}$$

as the upper limit for the magnitude of the interim sum w_i if t_i is limited by (7). Values of $|t_i| > 1$ have no practical application for two-digit operations, but they are necessary for simultaneous totally-parallel addition and subtraction of several digits which will be discussed later. An immediate result of (8) is the restriction of allowed values of radix r to the integer values

$$r > 2 \tag{9}$$

because, for $r = 2$, the only allowed value of $w_i = 0$ does not satisfy (5) for the value $x_i = 1$.

Requirements (5), (7) and (8) establish that at least r values of w_i are required if all $2r-1$ positive and negative values of x_i are to be transformed according to (5), and that the values $w_i = -1, 0, 1$ should be included for all $r > 2$. The relationship between the greatest value $w_{\max}$ and smallest value $w_{\min}$ of w_i is:

$$w_{\max} - w_{\min} \geq r - 1, \tag{10}$$

where the equality sign applies if only r values of w_i are chosen. *The required set of values of* w_i therefore must consist of r integers, which include -1, 0, 1 and satisfy (8) and (5) for all $2r-1$ values of x_i. More than one such set may exist, but the choice of a sequence symmetric around zero is preferable in further development. *The set of all allowed values of* w_i is unique and consists of $2r-3$ integers from $-(r-2)$ to $r-2$.

The values of digits w_i and t_i and the totally-parallel subtraction requirement stated in (3) determine which values of z_i are required and how many values of z_i are allowed for a digit z_i of a radix $r \geq 3$ signed-digit representation. The digit z_i assumes the minimum number of values when the required set of w_i is chosen as a sequence of r integers $\{w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}\}$ in which additive inverses exist for all (when r is odd) or all but one (when r is even) nonzero values of w_i. In both cases the required values of the sum digit s_i in (2) consist of a sequence of $r+2$ integers: $\{w_{\min}-1, w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}, w_{\max}+1\}$.

For an odd radix $r_0 \geq 3$ we choose $w_{\max} = -w_{\min} = \frac{1}{2}(r_0 - 1)$; here the additive inverse exists for every value of w_i and s_i, and *the required (minimum) set of values for digits* z_i *or* y_i in (1) consists of the sequence of r_0+2 integers

$$\{-\tfrac{1}{2}(r_0+1), \cdots, -1, 0, 1, \cdots, \tfrac{1}{2}(r_0+1)\}. \tag{11}$$

For an even radix $r_e \geq 4$, either $(w_{\min}-1)$ or $(w_{\max}+1)$ in the set of the required values of s_i has no additive inverse. This additive inverse is required as a subtrahend input to the adder (y_i) in order to satisfy (3). *The required (minimum) set of values for the subtrahend digit* y_i consists of the sequence of r_e+3 integers

$$\{-(\tfrac{1}{2}r_e+1), \cdots, -1, 0, 1, \cdots, \tfrac{1}{2}r_e+1\}. \tag{12}$$

The minimum set of values for the sum digit s_i and the augend (or minuend) digit z_i requires only r_e+2 integers and either $\frac{1}{2}r_e+1$ or $-(\frac{1}{2}r_e+1)$ is omitted from (12) to give this set. Only r_e+2 digit values require storage, but the adder must accept r_e+3 values of the subtrahend digit y_i. For instance, given $r=4$, the values -2, -1, 0, 1, 2, 3 are sufficient to represent any sum or difference, but the adder must also accept the subtrahend digit value $y_i = -3$ as the additive inverse of $y_i = 3$ during a subtraction.

C. Types of Signed-Digit Representations

The sets (11) and (12) are the required (minimum) sets of digit values which satisfy the requirements for signed-digit representations. They are the only allowed sets for radix 3 (values -2, -1, 0, 1, 2) and radix 4 (values -3, -2, -1, 0, 1, 2, 3). For all $r > 4$ there exists more than one set of allowed digit values. Since the maximum allowed absolute value of a digit is $r-1$ according to (4), all sequences of integers,

$$\{-a, -(a-1), \cdots, -1, 0, 1, \cdots, a-1, a\}$$
$$\tfrac{1}{2}(r_0+1) \leq a \leq r_0 - 1 \text{ or } \tfrac{1}{2}r_e + 1 \leq a \leq r_e - 1; \tag{13}$$
$$r_0 \text{ is an odd integer } r_0 \geq 3,$$
$$\text{and } r_e \text{ is an even integer } r_e \geq 4,$$

will satisfy the requirements for signed-digit representation. *The maximum allowed set of values* occurs when $a = r-1$ and consists of $2r-1$ integers. For instance, two sets exist for radix 5, one with 7 values (-3 to 3) and one with 9 values (-4 to 4). Four sets exist for radix 10, from 13 values (-6 to 6) to 19 values (-9 to 9).

All signed-digit representations, *i.e.*, representations which satisfy the requirements 1) to 4) of Section II-A, may now be described in terms of the allowed values of radix r and digits z_i as follows:

[9] Derivations of the results presented here appear in the Appendix

Definition 3: The *class of signed-digit representations* consists of all positional, constant radix representations of algebraic values

$$Z = \sum_{i=-n}^{m} z_i r^{-i}$$

in which the digits z_i assume one of the allowed sets of values given by (13) and the radix is a positive integer $r>2$. The *redundancy* of a signed-digit representation is *minimal* when $a=\frac{1}{2}(r_0+1)$ or $a=\frac{1}{2}r_e+1$, and the redundancy is *maximal* when $a=r_0-1$ or $a=r_e-1$ in (13).

A consequence of restriction (4), $|z_i| \leq r-1$, is an upper limit to the maximum weighted sum of all digits z_i $(i=k+1, \cdots, m)$ to the right of an arbitrary digit z_k in the signed-digit representation of the algebraic value Z, which is

$$\sum_{i=k+1}^{m} |z_i|_{\max} r^{-i} \leq r^{-k}(1 - r^{-m+k}) < r^{-k}. \quad (14)$$

Important consequences of (14) are as follows:

1) The sign of the algebraic value Z is indicated by the sign of the most significant nonzero digit.
2) The algebraic value Z is zero if, and only if, all digits of its signed-digit representation have the value $z_i=0$.
3) Given a signed-digit representation of the algebraic value Z, the signed-digit representation of $-Z$ is formed by changing the sign of the value of every nonzero digit z_i.

By repetitive formation and propagation of transfer digits, a signed-digit representation may be converted to a *canonical form* in which the values of all digits z_i are in the chosen set of the values of w_i; that is, no more transfer digits can be formed. A maximum of m additions of the number to zero may be required to put an m-digit number into the canonical form. If the set of values of w_i is the minimal (required) set, we obtain the *minimal canonical form*, which is nonredundant, since its digits assume only r values.

D. Range of Numbers and Overflow Indication

Generally, the range for algebraic values Z of fixed-point numbers (or mantissas in floating point arithmetic) is chosen as $1>Z>-1$ (or 1 or -1 may be included) in conventional number representations. Two considerations determine the choice of a convenient range for signed-digit representations: 1) the entire conventional range $1\geq Z\geq -1$ should be covered, and 2) a simple, immediate method of overflow indication should exist.

These two requirements will be satisfied if the absolute range limits for the algebraic value

$$Z = \sum_{i=0}^{m} z_i r^{-i}$$

of a signed-digit representation are defined to be

$$1 + |z_i|_{\max} \sum_{i=2}^{m} r^{-i} \geq Z \geq -1 - |z_i|_{\max} \sum_{i=2}^{m} r^{-i}. \quad (15)$$

Overflow is detected from the inspection of the two most significant digits z_0 and z_1 as follows:

positive overflow occurs if $z_0 = 1$ and

$$z_1 \geq 1, \text{ or if } z_0 > 1, \quad (16)$$

negative overflow occurs if $z_0 = -1$ and

$$z_1 \leq -1, \text{ or if } z_0 < -1. \quad (17)$$

In a redundant representation one algebraic value may be represented in more than one way. For this reason overflow will be indicated for some representations of numbers near the limits, but within the allowed maximum range, of (15). For the maximum allowed range of (15), and the overflow detection rules (16) and (17), there exist three ranges of overflow indication:

1) *Certain overflow indication*, irrespective of representation, will occur if Z exceeds the maximum range of (15)

$$|Z| > 1 + \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \quad (18)$$

2) *Potential overflow indication*, dependent on the specific representation of Z, covers the range

$$1 + \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max} \geq |Z| \geq 1 + \frac{1}{r} - \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \quad (19)$$

3) *No overflow indication* for any representation of Z will occur in the range

$$|Z| < 1 + \frac{1}{r} - \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \quad (20)$$

The no-overflow range is minimum when $|z_i|_{\max} = r-1$. In this case, the range of (20) reduces to

$$|Z| < 1 + r^{-m}. \quad (21)$$

The range $1\geq Z\geq -1$ is therefore covered by every signed-digit representation and is considered as the allowed range for the scaling of numbers. It must be noted that an overflow indication may not occur until the algebraic value of a number reaches the certain overflow indication range of (18). Other ranges and more precise overflow detection may be chosen in an analogous manner.

E. Conversion Procedures

The conversion of a conventional (sign and magnitude) representation to a signed-digit representation of the same radix $r>2$ occurs according to (5) and (6).

Given the conventional representation of

$$Z = (1 - 2x_s) \sum_{i=-n}^{m} x_i r^{-i},$$

in which the digits x_i range over r values $0, 1, \cdots, r-1$, and $x_s = 0$ if Z is positive and $x_s = 1$ if Z is negative, the conversion to signed-digit representation is performed as follows:

1) Choose the set of allowed values ($w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}$) of the interim sum w_i.
2) Interpret each digit x_i ($i = -n, \cdots, -1, 0, 1, \cdots, m$) to be negative if the sign digit $x_s = 1$.
3) Consider each digit x_i as the sum of two digits in signed-digit representation and recode according to (5)

$$w_i = x_i - rt_{i-1},$$

where

$$t_{i-1} = 0 \text{ if } w_{\min} \leq x_i \leq w_{\max}$$
$$t_{i-1} = 1 \text{ if } x_i > w_{\max}$$
$$t_{i-1} = -1 \text{ if } x_i < w_{\min}.$$

4) Form the digits z_i of the signed-digit representation: $z_i = w_i + t_i$ for $i = -n, \cdots, -1, 0, 1, \cdots, m$, and $z_{-n-1} = t_{-n-1}$.

The conversion from signed-digit to a conventional representation may be performed by several methods. A direct approach is to consider the signed-digit number as the sum of two numbers in conventional representation of the same length, one of which is positive and the other negative. Negative and positive digits are separated to form these two numbers, which then can be added in an adder for conventional representations to yield the desired conventional representation. Conversion can also be executed in a serial manner, starting with the least significant digit of the signed-digit representation.[7]

F. Simultaneous Addition of Several Digits

Previous discussion of totally-parallel addition and subtraction was limited to simultaneous addition of two digits and therefore to two numbers only. However, rules (1) and (2) may be extended to simultaneous totally-parallel addition of an arbitrary number n of digits if values of the transfer digit $|t_i| > 1$ are allowed and the radix r is sufficiently large.

The conditions (1), (2), (4) and (10) yield the upper and lower limits for the greatest absolute value of t_i as follows:

$$\frac{n-1}{r-1} |z_i|_{\max} \leq |t_i|_{\max} \leq |z_i|_{\max} - |w_i|_{\max}. \quad (22)$$

The values of t_i are chosen as a sequence of integers ranging from $-|t_i|_{\max}$ to $|t_i|_{\max}$, and the values of w_i must satisfy (2) and (4); *i.e.*, $|w_i + t_i| \leq r-1$ for all values of t_i and w_i. Consequently, the number n of digits which may be added simultaneously is a function of the radix r and of the maximum allowed magnitudes of digits z_i and w_i

$$n \leq r - \frac{|w_i|_{\max}}{|z_i|_{\max}} (r-1). \quad (23)$$

If the greatest $|z_i|_{\max}$ and the least $|w_i|_{\max}$ (see Appendix) are used,

$$n \leq \tfrac{1}{2}(r_0 + 1) \quad \text{and} \quad n \leq \tfrac{1}{2} r_e \quad (24)$$

is the upper limit for n. For instance, a maximum of two digits may be added simultaneously for radix 4, three digits for radix 5, and five digits for radix 10.

A total of n numbers in signed-digit representation may be simultaneously added (or subtracted) in the totally-parallel mode and will yield a sum or difference in signed-digit representation if (22) and (23) are satisfied.

G. Modified Signed-Digit Representations

The requirement of totally-parallel addition and subtraction restricts signed-digit representations to radices $r > 2$. Furthermore, at least $r+2$ values are required for the sum digit s_i in (2), and the subtraction requirement increases the required number of subtrahend digit values to $r+3$ for even radices, as shown in (12). The digit addition rules (1) and (2) may be modified to allow the propagation of the transfer digit over two digital positions to the left.[10] If this type of *two-transfer* addition is allowed, the radix $r = 2$ may be used and only $r+1$ values are required for the sum digit. Two-transfer addition is executed in three successive steps (the adder diagram is shown in Fig. 2):

$$1) \quad z_i' + y_i' = rt_{i-1}' + w_i' \quad (25)$$
$$2) \quad w_i' + t_i' = rt_{i-1}'' + w_i'' \quad (26)$$
$$3) \quad s_i' = w_i'' + t_i''. \quad (27)$$

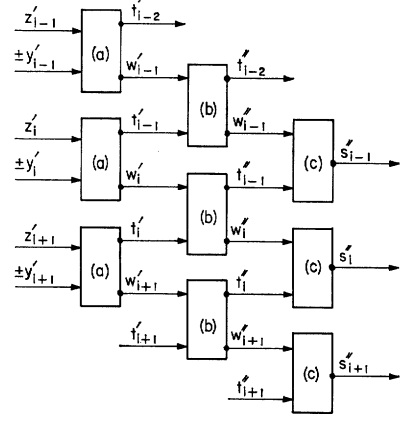

Fig. 2—Section of a two-transfer adder for modified signed-digit representation.

[10] J. E. Robertson, private communication.

The digits z_i', y_i' and s_i' are digits of a *modified* signed-digit representation.

For any radix $r \geq 2$, w_i' assumes $r+1$ values. For any *even* radix r_e, the values of w_i' range from $\frac{1}{2}r_e$ to $-\frac{1}{2}r_e$. In step 1) a transfer digit is generated whenever possible, *i.e.*, whenever $|z_i'+y_i'| \geq \frac{1}{2}r_e$, $|t_{i-1}'| = 1$; in step 2) a transfer digit is generated only if it is mandatory, *i.e.*, $|t_{i-1}''| = 1$ only if $|w_i'+t_i'| = \frac{1}{2}r_e+1$. Under these conditions t_i' and t_i'' cannot be both $+1$ or both -1, and s_i' assumes only r_e+1 values which are the same as the values of w_i'

$$\left\{-\tfrac{1}{2}r_e, \cdots, -1, 0, 1, \cdots, \tfrac{1}{2}r_e\right\}. \tag{28}$$

These r_e+1 values are the required values of one digit z_i' or y_i' for any even radix $r_e \geq 2$ if addition is performed according to (25)–(27). For instance, for the radix $r=2$ the required values are -1, 0, 1; for $r=4$ the required values range from -2 to 2; and for $r=10$ from -5 to 5.

For any *odd* radix $r_0 \geq 3$ the same procedure of addition applies if r_0+1 values of w_i' are chosen, ranging from $\pm\frac{1}{2}(r_0+1)$ to $\mp\frac{1}{2}(r_0-1)$. Only r_0+1 values of s_i' are required, which are the same as the values of w_i'. However, the principal advantage of two-transfer addition is lost because one more value of a digit must be introduced to provide the additive inverse of $\pm\frac{1}{2}(r_0+1)$, and r_0+2 values of a subtrahend digit y_i' are required at the input of the adder.

The potential advantages of two-transfer addition and of the resulting modified signed-digit representations are the use of radix $r=2$ (with digit values -1, 0, 1) and the decreased redundancy requirement (only $r+1$ values of a sum digit are required), especially for even radices. The disadvantages are the more complicated process of addition and the greater length of transfer digit propagation. It is interesting to observe that the procedure of two-transfer addition corresponds to the method of addition used for the "coincident carry-borrow storage" representation.[4]

Generally, it may be concluded that the lower limit on the required redundancy of one digit is a function of the number of digital positions over which a signal is allowed to propagate. If no redundancy exists and each sum digit assumes only r values, a sum digit s_i is the function of all the addend digits z_i and augend digits y_i to the right, *i.e.*, $s_i = f(z_i, y_i, z_{i+1}, \cdots, z_m, y_m)$. If each sum digit assumes $r+1$ values, we have $s_i = f(z_i, y_i, z_{i+1}, y_{i+1}, z_{i+2}, y_{i+2})$, and the operation is two-transfer addition. If each sum digit assumes $r+2$ values or more, we have $s_i = f(z_i, y_i, z_{i+1}, y_{i+1})$ with the restriction $r \geq 3$, and the addition is totally-parallel.

III. Signed-Digit Arithmetic

The preceding section presented the properties of a class of number representations which satisfy the four requirements 1) to 4) postulated in Section II-A. Given the class of signed-digit representations described by Definition 3, this section will describe the execution of addition, subtraction, shifting, multiplication and division for signed-digit numbers.

A. Elementary Arithmetical Operations

Addition, subtraction, left shifting and right shifting are the elementary operations for a signed-digit arithmetic unit. Multiplication and division are executed as sequences of additions or subtractions and shifts.

The addition of two signed-digit numbers is performed as the totally-parallel addition of all corresponding digits z_i and y_i according to (1) and (2). The block diagram of a signed-digit adder is shown in Fig. 1. To perform subtraction, first the sign of the subtrahend is changed by inverting the signs of all nonzero digits, then totally-parallel addition of corresponding digits is performed. Given the allowed values of w_i as the sequence: $w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}$; the rules for forming w_i, t_{i-1} and s_i are as follows:

$$w_i = (z_i + y_i) - rt_{i-1}, \tag{29}$$

where

$$\begin{aligned} t_{i-1} &= 0 \text{ if } w_{\min} \leq z_i + y_i \leq w_{\max}, \\ t_{i-1} &= 1 \text{ if } z_i + y_i > w_{\max}, \\ t_{i-1} &= -1 \text{ if } z_i + y_i < w_{\min}; \\ s_i &= w_i + t_i. \end{aligned} \tag{30}$$

It is interesting to note that serial addition or subtraction of signed-digit numbers may be initiated with the most significant digits. Example 1 demonstrates the addition of two radix 10 signed-digit numbers.

Example 1—Signed-Digit Addition (Radix 10): The allowed digit values are: for w_i: 5, 4, 3, 2, 1, 0, $\bar{1}$, $\bar{2}$, $\bar{3}$, $\bar{4}$, $\bar{5}$;
for t_i: 1, 0, $\bar{1}$;
for s_i, z_i, y_i: 6, 5, 4, 3, 2, 1, 0, $\bar{1}$, $\bar{2}$, $\bar{3}$, $\bar{4}$, $\bar{5}$, $\bar{6}$.

Negative values are identified by a bar above the integer. The radix 10 signed-digit operands are:

augend z: $1.\bar{3}65\bar{1}\bar{4}$, algebraic value $Z=0.76486$
addend y: $0.\bar{4}053\bar{1}$, algebraic value $Y=-0.39471$.

The procedure of addition is as follows:

augend z:	1.	$\bar{3}$	6	5	$\bar{1}$	$\bar{4}$
addend y:	0.	$\bar{4}$	0	5	3	$\bar{1}$
Step 1):	0+1.	$\bar{1}$0+3	10+$\bar{4}$	10+0	0+2	0+$\bar{5}$
Step 2):	$\bar{1}$	1	1	0	0	
sum s:	0.	4	$\bar{3}$	0	2	$\bar{5}$

The sum is s: $0.4\bar{3}02\bar{5}$, algebraic value $S=0.37015$.

For the absolute range limits of (15), the limits for the sum or difference of two numbers $S = Z \pm Y$ are

$$|S| \leq 2 + 2\frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \tag{31}$$

The overflow detection rules (16) and (17) assure that $|s_0| \leq 2$ will always hold and that $|t_{-1}| = 1$ may exist only for radix $r=3$, since $|w_0| = 2$ is allowed for all

radices $r>3$. The interim sum digits w_0, w_1 and transfer digits t_{-1}, t_0, t_1 contain all the necessary overflow information before s_0 and s_1 have been formed, and the overflow indication may be generated simultaneously with the sum digits. Immediate standardization may be performed in floating-point addition.

Shifting (left and right) of signed-digit numbers is performed in the conventional manner. If the digits z_i of a number pass through the adder logic during the shifting operation, transfer digits t_{i-1} will be generated when $z_i>w_{\max}$ or $z_i<w_{\min}$ and added to the interim sum w_{i-1} at the left according to (29) and (30). During the right shift of one position, $z_0=0$ is inserted as the most significant digit of the shifted number. If this shift includes transfer generation, the least significant digit z_m in the shifting register will retain the property $|z_m| \leq |z_i|_{\max}-1$, which is useful in multiple precision operations. Before the left shift of one position, the inspection of digits z_1 and z_2 will predict overflow after the shift. If the left shift includes transfer generation, digit z_3 must be inspected for a potential transfer digit generation.

B. Multiplication

Signed-digit multiplication is performed as a sequence of additions or subtractions and right shifts. Given the multiplicand z and the $m+1$ digits-long-multiplier y

$$\left(\text{algebraic value } Y = \sum_{i=0}^{m} y_i r^{-i}\right)$$

in radix $r\geq 3$ signed-digit representations, the product $p_{m+1}=zy+p_0r^{-m}$ is formed by the following recursive process:

$$p_{m+1} = p_m + zy_0 \tag{32}$$

$$p_{j+1} = \frac{1}{r}(p_j + zy_{m-j}) \quad \text{for } j = 0, 1, 2, \cdots, m-1. \tag{33}$$

Here the p_{j+1} are partial products, p_0 is the initial number in the accumulator register and y_{m-j} is the multiplier digit sensed during the jth step of multiplication.

Temporary overflow may occur in the sum p_j+zy_{m-j} before the right shift which forms p_{j+1}. If p_j and z are in range, ($|p_j| \leq |z|_{\max}$, $|z| \leq |z|_{\max}$), we have

$$|p_j + zy_{m-j}| \leq |z|_{\max}(|y_{m-j}|_{\max} + 1), \tag{34}$$

where $|z|_{\max}$ is the maximum allowed magnitude of a number. Since $|y_{m-j}|_{\max}\leq r-1$ always holds, p_{j+1} will also be in range. To store the temporary sum (before right shift) one more storage position (s_{-1}) must be allowed.

It is necessary to note that although both z and y do not indicate overflow, the product zy may indicate overflow if either z or y or both were in the potential overflow indication range given by (19). A test for overflow is necessary at the end of multiplication. If the temporary overflow storage position s_{-1} is available, the correct value of the product is retained in case of overflow.

Example 2 demonstrates signed-digit multiplication for radix 10 numbers.

Example 2—Signed-Digit Multiplication (Radix 10): The allowed digit values are given in Example 1. The radix 10 signed-digit operands are

multiplicand z: $0.\bar{4}62$, algebraic value $Z=-0.338$
multiplier y: $1.\bar{3}1\bar{5}$, algebraic value $Y=0.705$.

The procedure of multiplication is as follows:

step	digit y_{m-j}	quantity	representation	next adder operation
$j=0$	$y_3=\bar{5}$	p_0	0.000	
		zy_3	2.3$\bar{1}$0	add
		p_0+zy_3	2.3$\bar{1}$0	shift right
		p_1	0.23$\bar{1}$0	
$j=1$	$y_2=1$	zy_2	0.$\bar{4}$62	add
		p_1+zy_2	0.2$\bar{3}$10	shift right
		p_2	0.02$\bar{3}$10	
$j=2$	$y_1=\bar{3}$	zy_1	1.014	add
		p_2+zy_1	1.00$\bar{3}$10	shift right
		p_3	0.100$\bar{3}$10	
$j=3$	$y_0=1$	zy_0	0.$\bar{4}$62	add
		p_4	0.$\bar{2}\bar{4}$2$\bar{3}$10	end

The product $zy=p_4$ is $0.\bar{2}\bar{4}2\bar{3}10$, algebraic value $ZY=-0.238290$.

It should be noted that the greatest absolute value of the multiplier digit y_{m-j} in (33) is $\frac{1}{2}r_e+1$ for even radices and $\frac{1}{2}(r_0+1)$ for odd radices when the multiplier y is in the minimal redundancy representation. By serial recoding of y (before sensing of y_{m-j}) to the minimal canonical (nonredundant) form, the greatest absolute value of y_{m-j} is reduced to $\frac{1}{2}r_e$ for even radices and $\frac{1}{2}(r_0-1)$ for odd radices. In this case, for the radix 3, the values of y_{m-j} are -1, 0, 1; for radix 4 they are -1, 0, 1 and 2 or -2; for radix 10 the ten required values are -4 to 4 and 5 or -5. Since y_{m-j} has values from 0 to $r-1$ in conventional representation, fewer additions (or fewer multiple-generating circuits) will be necessary on the average for one multiplication in signed-digit representation.

Eq. (24) shows that $n\leq\frac{1}{2}(r_0+1)$ or $n\leq\frac{1}{2}r_e$ digits can be simultaneously added for signed-digit numbers. If the multiplier y is recoded to the minimal canonical form, only one addition per step of (33) is sufficient for any odd radix $r_0\geq 3$ when each stage of the adder is designed to accept up to $\frac{1}{2}(r_0-1)$ inputs of the multiplicand digit z_i. The resulting product will be in the maximal redundancy form. For any even radix $r_e\geq 4$, two additions per step of (33) will be needed when $|y_{m-j}| = \frac{1}{2}r_e$. For all smaller magnitudes of y_{m-j} one addition per step is sufficient. This shows that the use of higher redundancy in the representation and of greater adder complexity permits very fast signed-digit multiplication.

C. Division

Signed-digit division is performed as a sequence of additions or subtractions and left shifts. The method of division which is most readily applicable to signed-digit representations is due to Robertson.[11] The representation of the quotient digits in this method may be redundant; however, the redundancy allows an inexact selection of quotient digits. In a signed-digit representation the sign of the dividend or a partial remainder is not readily available for inspection if several most significant digits are zero. The magnitude, however, may be estimated, with a limited uncertainty, from the inspection of a few most significant digits. The Robertson division method which requires such an estimate is therefore preferable to restoring or nonrestoring division, both of which require knowledge of the sign for exact selection of quotient digits.

Given the dividend z and the divisor d, the quotient digits q_j are generated by the following recursive process:

$$p_{j+1} = rp_j - dq_{j+1} \quad \text{for} \quad j = 0, 1, 2, \cdots, m-1 \tag{35}$$

$$p_0 = z - dq_0. \tag{36}$$

Here the p_j are partial remainders, p_m is the remainder and q is the quotient ($m+1$ digits long) with the algebraic value

$$Q = \sum_{i=0}^{m} q_i r^{-i}.$$

During each step of division the quotient digit q_{j+1} must be chosen which has a value such that the next partial remainder p_{j+1} in (35) is within the same allowed range as p_j. This range is a function of the magnitude of divisor d. If $|z|$ is not in this range, the choice of $|q_0| = 1$ in (36) must bring p_0 into allowed range; otherwise the quotient overflows. Every quotient digit q_{j+1} in (35) is assigned the value which satisfies the condition

$$|rp_j - dq_{j+1}| \leq c|d|, \tag{37}$$

where c is the *range test constant* whose allowed range is determined by the choice of the allowed values of the quotient digit. A *comparison circuit* must be constructed which performs (37); the value $c = \frac{1}{2}$ is the most practical choice within the allowed range.[12] If the representation of the quotient digits is redundant (q_{j+1} assumes more than r values), the comparison (37) may be inexact, that is, it is sufficient to perform the comparison between truncated values of $|rp_j - dq_{j+1}|$ and $|d|$. To facilitate the comparison (37) the divisor d must be standardized before division; otherwise, very great precision of comparison is required when $|d|$ is small.

The choice of allowed values for the quotient digits q_j is governed by two considerations. The least number of additions (or multiple-generating circuits) will be required for one division when the digits q_j assume the least possible number of values. However, at least minimal redundancy in representation of q_j is necessary to allow an inexact selection of quotient digits. In case of minimal redundancy the values of q_j range from $-\frac{1}{2}r_e$ to $\frac{1}{2}r_e$ (a total of $r+1$ values) for even radices $r_e \geq 4$, and from $-\frac{1}{2}(r_0+1)$ to $\frac{1}{2}(r_0+1)$ (a total of $r+2$ values) for odd radices $r_0 \geq 3$. For these values of q_j, it is sufficient to compare the first four digits of the test quantities in (37).

In the simplest mechanization of (35) and (36), the standardized divisor d is repetitively added to (if signs of d and rp_j disagree) or subtracted from (if signs of d and rp_j agree) the shifted partial remainder rp_j, until the condition

$$\left| \sum_{i=0}^{3} z_i r^{-i} \right| \leq \left| \frac{1}{2} \sum_{i=0}^{3} d_i r^{-i} \right| \tag{38}$$

is detected by the comparison circuit, indicating that p_{j+1} has been generated. Here d_i ($i=0, 1, 2, 3$) are the first four digits of d and z_i ($i=0, 1, 2, 3$) are the first four digits in the accumulator register, which contains the dividend z at the start of division, and later—the partial remainders and incomplete partial remainders. The value of q_{j+1} is equal to the number of additions (sign of q_{j+1} is minus) or the number of subtractions (sign of q_{j+1} is plus) required to generate p_{j+1}. If (38) is satisfied immediately after the left shift of p_j, $q_{j+1}=0$ is the required value. In any case, p_{j+1} is now shifted left and the procedure repeated to generate q_{j+2} and p_{j+2} until $m+1$ quotient digits have been generated. A numerical example of this process of division is given in Example 3.

The above-presented (binary) version of Robertson division requires the least amount of special circuitry. Quotient digits q_j may be generated during one addition cycle if more comparator circuits and circuits which generate multiples of the divisor are added to the arithmetic unit.[11]

Example 3—Signed-Digit Division (Radix 10): The allowed digit values are given in Example 1. The values of quotient digits q_j are 5 to $\bar{5}$ inclusive. The radix 10 signed-digit operands are

dividend z: $0.\bar{2}\bar{3}6\bar{4}15$, algebraic value $Z = -0.224385$
divisor d: $1.\bar{3}1\bar{5}$, algebraic value $D = 0.705$.

The test quantity is

$$\left| \frac{1}{2} \sum_{i=0}^{3} d_i r^{-i} \right|,$$

algebraic value $T = 0.3525$.

[11] J. E. Robertson, "A new class of digital division methods," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 218–222; September, 1958.

[12] Derivations of the range test constant, the precision requirement in comparison, and other results presented here appear in the Appendix.

The procedure of division is as follows:

step	quantity	representation	test	next adder operation	q_{j+1}
$j=-1$	z	$0.\bar{2}36\bar{4}15$	$0.224<T$	shift	$q_0=0$
$j=0$	rp_0	$\bar{2}.36\bar{4}150$	$2.244>T$	add d	
	d	$1.3\bar{1}\bar{5}$			
	rp_0+d	$\bar{2}.5\bar{4}1$	$1.539>T$	add d	
	d	$1.3\bar{1}\bar{5}$			
	rp_0+2d	$\bar{1}.2\bar{3}\bar{4}$	$0.834>T$	add d	
	d	$1.3\bar{1}\bar{5}$			
	p_1	$0.\bar{1}31$	$0.129<T$	shift	$q_1=\bar{3}$
$j=1$	rp_1	$\bar{1}.3\bar{1}1500$	$1.289>T$	add d	
	d	$1.3\bar{1}\bar{5}$			
	rp_1+d	$\bar{1}.4\bar{2}\bar{4}$	$0.584>T$	add d	
	d	$1.3\bar{1}\bar{5}$			
	p_2	0.121	$0.121<T$	shift	$q_2=\bar{2}$
$j=2$	rp_2	1.215000	$1.215>T$	add $-d$	
	$-d$	$\bar{1}.3\bar{1}\bar{5}$			
	rp_2-d	0.510	$0.510>T$	add $-d$	
	$-d$	$\bar{1}.3\bar{1}\bar{5}$			
	p_3	$0.\bar{2}05000$	$0.195<T$	end	$q_3=2$

The quotient q is $0.\bar{3}\bar{2}2$, algebraic value $Q=-0.318$; the remainder p_3 is $0.\bar{2}05$, algebraic value $P_3=-0.195\times10^{-3}$.

D. Roundoff and Multiple Precision Operations

Addition and subtraction of numbers whose length is a multiple k of the $m+1$ digits long adder is performed as a sequence of k ordinary additions or subtractions in signed-digit arithmetic. The least significant parts are brought up first, and if a transfer digit t_{-1} is generated out of the most significant adder stage, it is applied as an input transfer digit t_m during the following operation on the two next higher-significance parts of the operands. This is always possible because $t_m=0$ for any two single-length operands. It should, furthermore, be noted that a single-length number may be added to or subtracted from the more significant half of a double-length number without affecting the less significant half in any manner. This is due to the general property of signed-digit representations that every digit contains its own sign information and therefore any section of a representation is a complete signed-digit representation in its own right.

The operation of roundoff is often required to reduce a number to the standard length. Roundoff for signed-digit numbers is performed by truncation. If the allowed values of a digit z_i are chosen symmetrically around zero $(-a, \cdots, -1, 0, 1, \cdots, a)$ and we assume that every one of these values will occur with the same probability, then the average error which is introduced by truncation is zero and the roundoff is without bias.

The knowledge of the number of significant digits in a fixed-length representation may be desirable in digital computer arithmetic, especially in floating-point operations. Because roundoff is performed by truncation for signed-digit numbers, it is possible to maintain a continuous indication of nonsignificant digits during addition, subtraction and left shifts. A new digit value $z_i=0'$ called the *space-zero* (distinct from the *value-zero* $z_i=0$) may be introduced to identify nonsignificant positions in a signed-digit representation. The following rule applies to the space-zero value $0'$:

$$z_i \pm 0' = 0' \text{ for all values of } z_i. \tag{39}$$

In the first step of totally-parallel digit addition $(z_i\pm0'=rt_{i-1}+w_i)$, $t_{i-1}=0$ and $w_i=0'$ are formed; the second step yields the sum digit $s_i=w_i+t_i=0'+t_i=0'$. Space-zero value digits may appear only as a continuous string of digits extending to the left from the least significant position of a register.

The nonsignificant positions in input numbers are represented by space-zero digits. During left shifts and during standardization in floating-point arithmetic space-zero digits are placed in the least significant position of the shifting register. When two signed-digit numbers are added or subtracted, the space-zero digits in the number with fewer significant digits will round off the sum or difference to the lesser precision. The space-zero digits will maintain a continuous indication of precision for all signed-digit numbers in the machine.

IV. Logical Design of Signed-Digit Adders

One problem of practical interest is the logical design of an adder to perform totally-parallel addition and subtraction with numbers in signed-digit representation. A parallel adder for signed-digit numbers of m-digit length consists of m identical *digit-adders* (also stage-adders[7]) which are linked to their immediate neighbors by transfer digit lines, an output to the left and an input at the right. Fig. 1 shows a block diagram with the required interconnections, inputs and outputs of digit-adders. All transfer digits t_i and interim sum digits w_i are formed simultaneously, thus the addition time for numbers of any length is equal to the addition time for one digit-adder. The objective of a digit-adder is to execute the operations described by (29) and (30) for digits of a given signed-digit representation, which is defined by the choice of the radix r and the allowed values of digits z_i, y_i, w_i and t_i. The allowed ranges of choice have been derived in the preceding sections. Minimal-redundancy representations require the least storage capacity for the values of a digit and therefore are preferable to representations with higher redundancy. Furthermore, less complicated digit-adder logic may be expected when the least possible number of digit values is employed.

The choice of the radix r depends on the desired balance between the increase in storage requirements and the logical complexity of one digit-adder. The relative increase in storage capacity requirements diminishes when r is large; however, one digit-adder must accept more values of a digit and the logical circuits become

more complex. Few digit values are also preferable in multiplication and division. For instance, radix 4 requires seven values (-3 to 3) for the digits of a subtrahend at the input to the digit-adder, at least six of these values (-3 or 3 may be avoided) for the sum digit s_i, and in storage, five values of the quotient digits (-2 to 2) and at least four of these values (-2 or 2 may be avoided) for the multiplier digits. Three binary storage elements per digit are required, compared to two for conventional radix 4 representations. Three binary storage elements per digit are sufficient for all radices $3 \leq r \leq 6$, four elements for radices $7 \leq r \leq 14$. The radix 10 (with 13 values) requires four elements per digit, the same as for conventional representations; the required values of quotient digits are -5 to 5. The space-zero value and additional redundancy may be introduced for radix 10 and many other radices without increasing the storage requirements.

The choice of the binary representation in storage of the allowed values for one digit affects the complexity of logical circuits which perform (29) and (30). A one digit-adder may be treated as a multiple-input and multiple-output switching circuit which is specified by these rules of addition. Alternatively, a digit-adder may be designed as a small conventional adder which consists of basic building blocks (such as half-adders) and has appropriate modifications to generate t_{i-1} and w_i according to the specifications of (29) and (30). Digit values are represented by binary digits corresponding to the storage elements, and conventional methods are used to represent negative values. The minimization of the logic for a digit-adder as a function of the specific storage representation of digit values, of the redundancy of digit representation, and of the choice of basic logic circuits remains as an interesting unsolved program for further investigation.

As an example, logical design has been performed for a radix 4 digit-adder.[13] The "building-block" approach was used in this design and conventional adder design techniques were employed. The seven values of a radix 4 digit (-3, -2, -1, 0, 1, 2, 3), were represented by three binary digits weighted -4, 2, and 1. The radix 4 digit-adder required the equivalent logical circuitry of approximately 12 half-adders of the conventional type; this is 2.5 to 3 times more than required by one position of a conventional radix 4 adder with serial carry propagation. The comparison is more favorable when the auxiliary circuitry of the conventional adders (carry acceleration, carry-completion sensing, conditional corrections) is considered. An investigation of a radix 10 digit-adder showed that it was about two times more complex than one stage of a conventional excess-three code adder. The experimental designs were intended to serve as an existence demonstration and an indication of the order of complexity of a digit-adder. No claim is made for minimal complexity of these designs; furthermore, design problems which are not directly affected by the totally-parallel addition requirements, such as gating, synchronization or timing, and arithmetic control were not considered.

[13] Avizienis, *op. cit.*, pp. 63–70.

V. Conclusion

A class of redundant number representations which permit the elimination of carry-propagation chains in addition and subtraction has been demonstrated. Unique representation exists for the zero value of a number, and all sign information is contained by the individual digits. No special sign digit is required; this eliminates preliminary and terminal corrections caused by sign digits.

Procedures for arithmetical operations have been described for the class of signed-digit representations. The time required for addition or subtraction is independent of the length of the operands; also, simultaneous addition of several numbers is possible. Convenient procedures exist for overflow detection, roundoff, and multiple-precision operations. Multiplier and quotient digits assume values which require fewer adder operations than an equivalent multiplication or division in conventional representation. A number in signed-digit representation requires more allowed values and more storage capacity per digit than the same number in conventional representation. The logical circuits are more complex for a signed-digit adder. The minimization of digit-adders remains a problem for further investigation.

A signed-digit number of arbitrary length consists of a positionally-ordered collection of complete one-digit long representations; this property leads to useful simplifications in arithmetical operations. Partition of a signed-digit number at any position yields two complete signed-digit numbers whose sum is equal to the original number. In a conventional parallel adder all iterative adder circuits are bound together by the requirement for carry propagation. In a signed-digit parallel adder, each iterative digit-adder depends only on its own inputs and the information generated by its immediate neighbor to complete one cycle of addition. It may be concluded that signed-digit representations offer the means for a completely parallel execution of arithmetical operations in a digital computer.

Appendix

The Division Comparison

The maximum allowed magnitude of a partial remainder p_j in (35) and (36) is a function of $|d|$, *i.e.*, $|p_j|_{\max} = k|d|$.

Two requirements are to be satisfied:

Requirement 1: The greatest partial remainder should be returnable into range after a shift by means of the greatest quotient digit $|q_j|_{\max}$

$$rk|d| - |d|\,|q_j|_{\max} \leq k|d|. \tag{40}$$

Eq. (40) yields the upper bound of k as

$$k \leq \frac{|q_j|_{\max}}{r-1}.$$

Requirement 2: If the shifted partial remainder is outside the range ($|rp_j| = k|d| + \delta > k|d|$) by the least detectable difference δ, a single reduction of rp_j by d should yield a partial remainder within range, *i.e.*,

$$k|d| + \delta - |d| \geq -k|d|. \tag{41}$$

Eq. (41) yields the lower limit of k as

$$k \geq \frac{1}{2} - \frac{1}{2}\frac{\delta}{|d|} \approx \frac{1}{2},$$

where the approximation $k = \frac{1}{2}$ is satisfactory since $|d|_{\max} \geq 1$ always holds.

The division requirements are satisfied for any value of k in the range limited by the above requirements, that is, for

$$\frac{1}{2} \leq k \leq \frac{|q_j|_{\max}}{r-1}. \tag{42}$$

The value of k may be allowed to vary within the range of (42); therefore, an *inexact* comparison is possible in (37). The allowed error of truncation must be determined. The quantities which are compared in (37) are the divisor $d = d' + d''$ and the quantity $p = p' + p''$. Here p is first the dividend z, then a shifted partial remainder rp_j or a diminished shifted partial remainder.

The test is to detect the condition $|p'| \leq c|d'|^*$ where c is the *range test constant.* An uncertainty of comparison occurs because of the uncompared lower-significance parts d'' and p''. When both p and d are truncated behind the position $i = h$, we denote $|p''|_{\max} = |d''|_{\max} = \Delta$, and determine the maximum value of Δ which satisfies $|p_j|_{\max} = k|d|$. In the test $|p'| \leq c|d'|^*$ the precision of $c|d'|^*$ is the same as of $|p'|$, that is, $h+1$ digits. Therefore $c|d'| = c|d'|^* + e$, where $0 \leq e < r^{-h}$.

Two conditions must be satisfied:

Condition 1: When $|p'| \leq c|d'|^*$ is indicated, $|p| = k_{\max}|d|$ must hold under the worst conditions, which occur when $|p| = |p'| + \Delta$, $|d| = |d'| - \Delta$ and $|p'| = c|d'|^*$ with $e = 0$. This yields

$$\Delta_1 = |d| \frac{k_{\max} - c}{1 + c} \tag{43}$$

as the maximum allowable error of truncation.

Condition 2: When $|p'| > c|d'|^*$ is indicated, one more reduction of $|p|$ by $|d|$ should retain $|p| - |d|$ in range, that is, $|p| - |d| = -k_{\max}|d|$ must hold under the worst conditions which occur when $|p| = |p'| - \Delta$, $|d| = |d'| + \Delta$ and $|p'| = c|d'|^* + r^{-h}$ with $e = e_{\max}$. This yields

$$\Delta = |d| \frac{k_{\max} + c - 1}{1 + c} + \frac{r^{-h} - e_{\max}}{1 + c}, \tag{44}$$

and since $(r^{-h} - e_{\max})$ is a small positive quantity, a good approximation is

$$\Delta_2 = |d| \frac{k_{\max} + c - 1}{1 + c}. \tag{45}$$

From the above results, the condition $\Delta \geq 0$ applied to (43) and (45) yields the range

$$1 - k_{\max} \leq c \leq k_{\max}, \tag{46}$$

in which no truncation uncertainty is allowed ($\Delta = 0$) for $c = k_{\max}$ or $c = 1 - k_{\max}$, and an uncertainty given by (43) and (45) may exist for $1 - k_{\max} < c < k_{\max}$, where

$$k_{\max} = \frac{|q_j|_{\max}}{r-1} \text{ from (42).}$$

Setting $\Delta_1 = \Delta_2$, we get $c = \frac{1}{2}$ and

$$\Delta_{\max} = \tfrac{1}{3}(2k_{\max} - 1)|d| \tag{47}$$

as the greatest value of Δ as a function of c, since Δ_1 decreases and Δ_2 increases as c ranges from $1 - k_{\max}$ to $k_{\max}$.

When the divisor d is standardized, its least value is

$$|d|_{\min} = \frac{1}{r} + r^{-2}\left(1 - \frac{|z_i|_{\max}}{r-1}\right) + r^{-m}\frac{|z_i|_{\max}}{r-1} \tag{48}$$

for $r \geq 3$, the range of (15) and overflow rules (16) and (17). Condition (48) occurs when $d_0 = 0$, $d_1 = d_2 = 1$ and $d_i = -|z_i|_{\max}$ for $i = 3, 4, \cdots, m$. Therefore $|d|_{\min} \geq 1/r$ holds for all signed-digit representations, and for standardized divisors we have

$$\Delta_{\max} = \frac{1}{3r}\left(2\frac{|q_j|_{\max}}{r-1} - 1\right). \tag{49}$$

This equation indicates the required precision of comparison for any choice of $|q_j|_{\max}$. For example, if $|q_j|_{\max} = r - 1$ (maximum redundancy case), we have $\Delta_{\max} = 1/3r$, and it is sufficient to compare positions $i = 0, 1, 2$. For minimal redundancy and even radices

$$r_e \geq 4, \quad |q_j|_{\max} = \tfrac{1}{2}r_e \quad \text{and} \quad \Delta_{\max} = \frac{1}{3r_e(r_e - 1)};$$

here the positions $i = 0, 1, 2, 3$ must be compared. For minimal redundancy and odd radices

$$r_0 \geq 3, \quad |q_j|_{\max} = \tfrac{1}{2}(r_0 + 1) \quad \text{and} \quad \Delta_{\max} = \frac{2}{3r_0(r_0 - 1)};$$

here the positions $i = 0, 1, 2, 3$ must be compared except for $r = 3$, where $i = 0, 1, 2$ are sufficient.

Derivation of Allowed and Required Digit Values

The conditions (1)–(4) establish the restrictions

$$2|z_i|_{\max} \leq r|t_i|_{\max} + |w_i|_{\max} \tag{50}$$

$$|t_i|_{\max} + |w_i|_{\max} \leq |z_i|_{\max} \tag{51}$$

on the maximum absolute values of t_i, w_i and z_i (or s_i and y_i). Given $|z_i|_{\max}=r-1$, we have the limiting conditions

$$2(r-1) = r|t_i|_{\max} + |w_i|_{\max} \quad (52)$$

$$r-1 = |t_i|_{\max} + |w_i|_{\max} \quad (53)$$

which are simultaneously satisfied by $|t_i|_{\max}=1$ and $|w_i|_{\max}=r-2$, leading to (7)–(9).

In the conversion procedure described by (5) and (6), the $2r-1$ values of x_i (from $-r+1$ to $r-1$) are to be recoded as $x_i=rt_{i-1}+w_i$. The values $x_i=-1$, 0, 1 require corresponding values $w_i=-1$, 0, 1, because (8) does not allow their recoding with $|t_i|>0$. At least r values of w_i are needed to recode all $2r-1$ values of x_i. In this case, $w_i=a$ is generated for each pair of values $x_i=a$ and $x_i=-(r-a)$, except for $w_i=0$ which is generated for $x_i=0$ only. Furthermore, in this case $w_{\max}+1$ is recoded as $r+w_{\min}$, where $w_{\max}$ is the greatest allowed value and $w_{\min}$ is the least allowed value of w_i, giving

$$w_{\max} - w_{\min} = r-1, \quad (54)$$

which is the limiting case of (10). The value $|w_i|_{\max}$ is least when we choose $w_{\max}=|w_i|_{\max}=\frac{1}{2}(r_0-1)$ for odd radices r_0 and $w_{\max}=|w_i|_{\max}=\frac{1}{2}r_e$ for even radices r_e.

In simultaneous totally-parallel addition of n digits, the restrictions are

$$|z_i|_{\max} = |w_i|_{\max} + |t_i|_{\max} \quad (55)$$

$$n|z_i|_{\max} \leq r|t_i|_{\max} + |w_i|_{\max}, \quad (56)$$

which give the lower limit of $|t_i|_{\max}$ as

$$|t_i|_{\max} \geq \frac{n-1}{r-1}|z_i|_{\max}. \quad (57)$$

The upper limit of $|t_i|_{\max}$ is given by (55) as

$$|t_i|_{\max} \leq |z_i|_{\max} - |w_i|_{\max}. \quad (58)$$

The greatest range of $|t_i|_{\max}$ is obtained when the greatest allowed value of $|z_i|_{\max}=r-1$ and the least allowed value of $|w_i|_{\max}=\frac{1}{2}r_e$ or $|w_i|_{\max}=\frac{1}{2}(r_0-1)$ is used in (22). These choices also lead to the maximum values of n given in (24).

ACKNOWLEDGMENT

The author wishes to thank Prof. J. E. Robertson of the Digital Computer Laboratory, University of Illinois, Urbana, Ill., for his comments on this paper and for guidance and critical advice during the writing of the doctoral dissertation[7] on which this paper is based.

Computing Machine Aids to a Development Project*

C. W. ROSENTHAL†, MEMBER, IRE

Summary—A system o integrated computer programs which provide useful machine assistance to the development of a digital system will be described. The individual programs are capable of such separate engineering tasks as verifying design data, optimally locating electronic logic packages on a chassis, routing interconnecting wires and preparing documents like wiring diagrams and wire running lists.

The engineer using the system need not be a computer programmer, because the machine aids programs are called into operation with a simple mnemonic language keyed to the engineer's traditional tasks. This aspect of the work, as well as the accomplishment of the individual engineering tasks, is emphasized in this paper.

INTRODUCTION

AUTOMATION is generally understood to be a substitution of machine controllers for human controllers in industrial or commercial processes and procedures. Frequently the controllers are electronic computers actuating other machines or aiding human beings in their work. The machine controllers are introduced because they provide some combination of improvements in the economy, dependability, productivity, accuracy, or adaptivity to environment of a system.

A group of technical people developing a new system themselves form a system requiring the fulfillment of the roles of designers, draftsmen, manufacturers and operators. These people cooperate to originate, transcribe, distribute and interpret information. Now they, too, may work with the aid of computers. This latest type of automation has been called "design automation"[1] and for our purposes may be described as the use of an integrated set of computer programs to perform a sequence

* Received by the PGEC, August 17, 1960.
† Bell Telephone Labs., Inc., Murray Hill, N. J.

[1] M. Kloomok, P. W. Case, and H. H. Graff, "The recording, checking and printing of logic diagrams," *Proc. Eastern Joint Computer Conf.*, pp. 108–118; December, 1958.