A New Class of Digital Division Methods*

JAMES E. ROBERTSON†

Summary—This paper describes a class of division methods best suited for use in digital computers with facilities for floating point arithmetic. The division methods may be contrasted with conventional division procedures by considering the nature of each quotient digit as generated during the division process. In restoring division, each quotient digit has one of the values 0, 1, $\cdots$, $r-1$, for an arbitrary integer radix r. In nonrestoring division, each quotient digit has one of the values $-(r-1), \cdots, -1, +1, \cdots, +(r-1)$. For the division methods described here, each quotient digit has one of the values $-n, -(n-1), \cdots, -1, 0, 1, \cdots, n-1, n$, where n is an integer such that $\frac{1}{2}(r-1) \leq n \leq r-1$. A method for serial conversion of the quotient digits to conventional (restoring) form is given. Examples of new division procedures for radix 4 and radix 10 are given.

INTRODUCTION[1]

A DIVISION method can be categorized by listing the permissible values of each quotient digit as generated during the division process. For an arbitrary radix r, each quotient digit generated during a conventional restoring division has one of the values 0, 1, $\cdots$, $r-1$. For nonrestoring division each quotient digit has one of the values $-(r-1), -(r-2), \cdots, -1, 1, 2, \cdots, r-1$, with 0 excluded. The purpose of this paper is to describe a class of division methods in which each quotient digit has one of the values $-n, -(n-1), \cdots, -1, 0, 1, \cdots, n$, where n is an integer such that $\frac{1}{2}(r-1) < n < r-1$. Conversion of the quotient to the conventional restoring form is required for the latter two classes of division methods.

Division, as executed in most digital computers now in use, involves a recursive process which may be preceded by preliminary operations and followed by terminal operations. Most of the time required for a digital division is spent in the repeated execution of the recursive process. For nonrestoring division and for the class of division methods proposed here, the recursive process can be described by

$$x_{j+1} = rx_j - q_{j+1}d \qquad j = 0, 1, \cdots, m-1$$

for which the following notation is employed:

x_j = partial remainder resulting from the jth execution of the recursive process
x_0 = dividend
x_m = remainder
q_j = (for fractions) the jth digit of the quotient to the right of the radix point
m = the number of digits, radix r, used to represent the quotient
d = divisor.

Each $x_j (j=0, 1, \cdots, m)$ satisfies $|x_j| \leq k|d|$, and the sign of each q_{j+1} is chosen so that $|x_{j+1}| = |(r|x_j| - |q_{j+1}||d|)|$. A straightforward but lengthy analysis of all cases that may arise reveals another property of each quotient digit, namely $|q_{j+1}| \leq k(r-1)$. For nonrestoring division, $k=1$; it is shown that division methods exist for certain discrete values of k in the range $\frac{1}{2} \leq k \leq 1$.

From the equation for the recursive process it can be shown readily that the division procedure is correct.

For $j=0$,

$$x_1 = rx_0 - q_1 d.$$

For $j=1$,

$$x_2 = rx_1 - q_2 d = r^2 x_0 - (rq_1 + q_2)d.$$

For $j=m-1$,

$$x_m = r^m x_0 - (r^{m-1}q_1 + r^{m-2}q_2 + \cdots + rq_{m-1} + q_m)d.$$

The shifted remainder $r^{-m}x_m$ is then

$$r^{-m}x_m = x_0 - d\sum_{i=1}^{m} r^{-i}q_i$$

where

$$\sum_{i=1}^{m} r^{-i}q_i$$

represents the quotient Q. It follows that $Qd + r^{-m}x_m = x_0$; *i.e.*, the sum of the shifted remainder and the product of the quotient and divisor is the dividend.

The mechanization of the recursive process requires three distinct steps.

1) The partial remainder x_j is shifted, *i.e.*, multiplied by the radix r.
2) One of several permissible arithmetic procedures is selected, such that the maximum absolute value of $r|x_j|$, namely $kr|d|$, is reduced by the amount $1/r$, so that the result x_{j+1} satisfies $|x_{j+1}| \leq k|d|$. It should be emphasized that the reduction is in the range over which partial remainders may vary, and not necessarily in the absolute values of specific partial remainders.
3) A quotient digit is generated corresponding to the arithmetic procedure selected.

* Manuscript received by the PGEC, March 19, 1958; revised manuscript received, July 1, 1958. The investigations leading to this paper were part of a computer study program by the staff of the Digital Computer Lab., University of Illinois, Urbana, Ill., under the joint support of the Office of Naval Res. and the Atomic Energy Comm.

† University of Illinois, Urbana, Ill.

[1] Descriptions of conventional restoring and nonrestoring methods can be found in a number of texts, such as the following:
R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Inc., New York, N. Y.; 1955.
M. Phister, Jr., "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y.; 1958.

The key to the study of division methods lies in the analysis of arithmetic procedures which reduce the allowable range of absolute values of the shifted partial remainder (rx_j) by the amount $1/r$.

Analysis of Arithmetic Procedures

It is convenient to normalize the partial remainders with respect to the absolute value of the divisor. If the substitution $z_j = x_j/|d|$ is made, the range restrictions become $-k \leq z_{j+1} \leq k$ and $-rk \leq rz_j \leq rk$. Attention then is focused on arithmetic procedures which transform rz_j into z_{j+1}. The easily mechanized procedures involve addition or subtraction of integral multiples of the divisor from rx_j to yield x_{j+1}; after normalization, the procedures involve addition or subtraction of integers from rz_j to yield z_{j+1}. The arithmetic procedures can be represented as a family of straight lines of the form $z_{j+1} = rz_j - i$, where $i = -n, \cdots, -1, 0, 1, 2, \cdots, n$, as shown in Fig. 1.

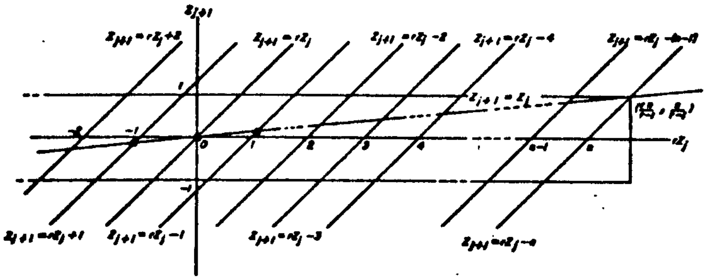

Fig. 1—A normalized graph illustrating arithmetic procedures during division.

In order for one of the proposed class of division methods to exist, it must be possible to superimpose a rectangle on the family of straight lines of Fig. 1 in such a way that the following occur.

1) The rectangle is centered at the origin, with vertices at $(\pm rk, \pm k)$.
2) The projections on the rz_j axis of the line segments within the rectangle cover that portion of the rz_j axis within the rectangle.

It follows from the first condition that the vertices of the rectangle lie on the lines through the origin of slope $\pm 1/r$, whose equations are $z_{j+1} = \pm z_j$. The second condition requires that the rectangle be sufficiently large to insure that $k \geq \frac{1}{2}$. Two additional considerations govern the choice of size of the rectangle, *i.e.*, the choice of k and of n.

3) The number of lines of the form $z_{j+1} = rz_j - i$ necessary to satisfy condition 2) should be minimized. The number of multiples of the divisor which must be formed is proportional to the number of lines employed.
4) The overlap of projections of the line segments on the rz_j axis should be maximized. The precision necessary in the selection process decreases as the overlap increases.

The two considerations are mutually contradictory since 3) requires that the size of the rectangle should be decreased and 4) requires that the size be increased.

The considerations govern the choice of k to the extent that k should take one of a discrete set of values such that the vertex (rk, k) of the rectangle lies on a line $z_{j+1} = rz_j - n$. Any one division method can then be characterized by the positive integers chosen for r and for n.

The value of k as a function of r and n can be found by solving for the point of intersection of the lines $z_{j+1} = z_j$ and $z_{j+1} = rz_j - n$. The value of z_{j+1} at the point of intersection is k, and is found to be $n/(r-1)$. The requirement that $k \geq \frac{1}{2}$ becomes $n \geq (r-1)/2$.

The choice of n for some given radix r involves a balance between time and equipment costs associated with the selection process, on the one hand, and similar costs in forming multiples of the divisor, on the other. Since the balance is so much a function of design details, the choice of n is discussed further only in connection with specific examples.

Quotient Conversion

The conventional representation of a quotient requires that each digit be one of the positive integers $0, 1, \cdots, r-1$. Since nonrestoring division and the method described here involve negative digits in the quotient, some means of conversion is required. The technique employed in conventional nonrestoring division, except for the special case of the binary system, can be described for a radix complement representation by the following rules.

1) If $q_1 < 0$, replace q_1 by $q_1' = r + q_1$ and set the sign of the quotient negative; if $q_1 > 0$, $q_1' = q_1$, and the quotient is positive.
2) For $j = 1, 2, \cdots, m-1$, inspect q_j' and q_{j+1}. If $q_{j+1} < 0$, replace q_j' by $q_j' - 1$ and replace q_{j+1} by $q_{j+1}' = r + q_{j+1}$. If $q_{j+1} > 0$, q_j' is left unchanged, and $q_{j+1}' = q_{j+1}$.

The rules require a serial inspection of the quotient digits, the most significant digit first. When a negative digit is encountered, it is added to the radix, and a unit is borrowed from the next most significant digit.

For the proposed division method, the conversion is complicated by the fact that 0's are permissible quotient digits. The inspection of the sign of q_{j+1} (or q_1 in rule 1) must be replaced by an inspection of signs of the divisor d and the partial remainder x_j (x_0 in rule 1) to determine the sign of the next *nonzero* quotient digit. Agreement of signs of x_j and d corresponds to $q_{j+1} > 0$ in the above rules; disagreement corresponds to $q_{j+1} < 0$. Alternatively, signs can be determined in the usual way and can be associated with those q_{j+1} which are zero. These modifications provide for a borrow propagation through a sequence of 0's.

Both sets of conversion rules require that no quotient digit q_j be such that $|q_j| > r-1$. In particular, the parameter n is one value that $|q_j|$ can assume, and therefore $n \leq r-1$. Thus, n is restricted to the range $\frac{1}{2}(r-1) \leq n \leq r-1$, since it was established previously that $n \geq \frac{1}{2}(r-1)$.

Preliminary and Terminal Operations

Some of the requirements which necessitate preliminary or terminal operations for division methods are listed below.

1) The requirement of standardizing the quotient, in a floating point division, and the requirement of overflow detection in a fixed point unit.
2) The requirement for a rounded quotient.
3) The requirement that a correct remainder be generated.

Procedures vary in computers now in use. For the division methods described in this paper, the above requirements necessitate comparable preliminary or terminal operations.

An additional requirement imposed by the division methods discussed is that each partial remainder x_j should satisfy $|x_j| \leq k|d|$ where d is the divisor and $\frac{1}{2} \leq k \leq 1$. In particular, the restriction applies to the dividend x_0, and may necessitate additional preliminary operations for division methods for which $k<1$, in contrast to conventional procedures for which $k=1$.

Preliminary standardization of the divisor simplifies the selection of the correct multiple of the divisor during the recursive process, since the precision required for selection increases as the minimum absolute value of the divisor decreases. The proposed methods, therefore, are best suited for use in arithmetic units having facilities for floating point operations. The methods can be used for fixed point division if facilities for simultaneously shifting divisor and dividend are available.

Additional problems are posed when a remainder x_m must be generated such that the division algorithm $Qd+r^{-m}x_m=x_0$ (where Q is the quotient represented by m digits, radix r; d is the divisor, and x_0 the dividend), is satisfied. If d and x_0 are initially shifted left p digital positions so that $d'=r^pd$ and $x_0'=r^px_0$, then the value of x_m' such that $Qd'+r^{-m}x_m'=x_0'$ is found to be $x_m'=r^px_m$. For a fixed point division, it would thus be necessary to shift the remainder x_m' p digital positions right to obtain the correct remainder x_m.

A second difficulty arises when the quotient conversion rules require the least significant digit q_m' of the converted quotient to have the value r. If, as is often the case, facilities for addition are not available for the quotient register, q_m' can be set to the value $r-1$, and the remainder x_m must then be replaced by $x_m''=x_m+d$. The division algorithm becomes

$$(Q-r^{-m})d+r^{-m}(x_m+d)=x_0.$$

Similar results are obtained when conventional nonrestoring division methods are employed.

Example 1—Radix 4 Division

Conventional radix 4 division methods require either two uses of a binary adder (use of one adder sequentially or two adders in parallel) or the formation and storage to full precision of $3d$, where d is the divisor. The division method of the class proposed here with $r=4$ and $n=2$ requires a single binary adder, conditional doubling and complementing circuits, and a selection circuit to compare rx_j with $0.5d$ and $1.5d$ to a precision of 7 binary digits, if $\frac{1}{4} \leq |d| \leq 1$.

The mechanization of the division scheme is indicated diagrammatically in Fig. 2.

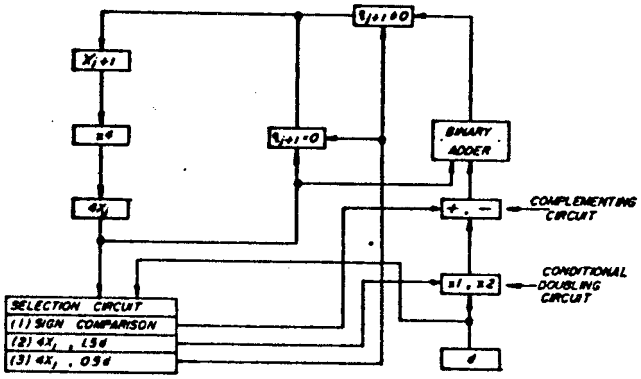

Fig. 2—Radix 4 division.

The selection circuit performs three functions, based upon the seven most significant binary digits of $4x_j$ (or x_j if desired) and of d.

1) Compares signs of x_j and d. If signs agree, the complementing circuit is set to subtract; if signs disagree, the complementing circuit is set to add.
2) Compares absolute values of $4x_j$ and $1.5d$. If $4|x_j| \geq 1\frac{2}{3}|d|$, the conditional doubling circuit must be set to form $2d$. If $4|x_j| \leq 1\frac{1}{3}|d|$, the circuit must be set to form d. If $1\frac{1}{3}|d| < 4|x_j| < 1\frac{2}{3}|d|$, the conditional doubling circuit can be set either way, depending upon the design details of the selection circuit.
3) Makes a similar comparison of absolute values of $4x_j$ and $(\frac{1}{2} \pm \frac{1}{6})d$. For the smaller values of $4|x_j|$, $x_{j+1}=4x_j$; otherwise x_{j+1} is transferred from the binary adder.

After x_{j+1} is formed, $4x_{j+1}$ is formed by a radix 4 left shift to replace $4x_j$. The values selected for the quotient digit q_{j+1} must correspond to the selections made, as summarized in Table I.

TABLE I

Selection 1	Selection 2	Selection 3	q_{j+1}
Subtract	$2d$	$q_{j+1} \neq 0$	+2
Subtract	$1d$	$q_{j+1} = 0$	+0
Subtract	$1d$	$q_{j+1} \neq 0$	+1
Add	$2d$	$\neq 0$	−2
Add	$1d$	$= 0$	−0
Add	$1d$	$\neq 0$	−1

Example 2—Radix 10 Division

From the many possible choices available to the designer, the radix 10 division method of this example is based upon the following.

1) The excess three representation of decimal digits is chosen.
2) A single decimal adder (excess three code) is used sequentially.
3) Storage is provided for the divisor d, but not for any of its multiples.
4) A complementing circuit and a conditional doubling and quintupling circuit are employed, with $\pm d$, $\pm 2d$, and $\pm 5d$ available as inputs to the adder.
5) The division method is characterized by $r=10$, $n=7$, with

$$k = \frac{n}{r-1} = \frac{7}{9}.$$

Doubling and quintupling circuits can be described by the sets of Boolean equations

$$\left.\begin{aligned} q &= b(a \vee c \vee de) \vee ac(d \vee e) \\ r &= \bar{c} \oplus (\bar{a}de \vee a\bar{d}\bar{e}) \\ s &= a \oplus d \oplus e \\ t &= \bar{e} \\ u &= a \end{aligned}\right\}\text{for doubling}$$

and

$$\left.\begin{aligned} v &= e \\ w &= a(\bar{e} \vee b \vee \bar{c}d) \vee \bar{e}b(c \vee d) \\ x &= \bar{b} \oplus (ecd \vee \bar{e}\bar{c}\bar{d}) \\ y &= \bar{e} \oplus c \oplus d \\ z &= \bar{d} \end{aligned}\right\}\text{for quintupling}$$

where, in excess three notation, a, b, c, d; q, r, s, t; and v, w, x, y represent one decimal digit of the divisor, $2x$(divisor), and $5x$(divisor), respectively.

For doubling, e is the binary carry input and u is the binary carry output to the next most significant decimal digit. For quintupling, e and z are most easily described as incoming and outgoing binary borrow signals in a halving circuit, with a wired-in decimal shift of v, w, x, and y converting the halving circuit to a quintupling circuit.

If the permutation $a'=b$, $b'=c$, $c'=d$, $d'=e$, $e'=\bar{a}$ is made in the divisor digits, and the permutation $v=\bar{a}$, $w=q$, $x=r$, $y=s$, $z=t$, at the output of the circuit, the doubling circuit is transformed into a quintupling circuit. Thus the same hardware can be used sequentially for both doubling and quintupling, provided permutation and complementation of input and output signals is correctly arranged.

The arrangement of hardware required for the division scheme is shown in Fig. 3. The two steps required for the generation of each decimal quotient digit can be described as follows.

Each quotient digit q_{j+1} can be decomposed into two digits q_{j+1}', q_{j+1}'', such that $q_{j+1}=q_{j+1}'+q_{j+1}''$, where $q_{j+1}'=-5$, 0, or 5 and $q_{j+1}''=-2$, -1, 0, 1, or 2. Step 1 corresponds to the determination of q_{j+1}' and results in the formation of a quantity x_{j+1}' such that

$$x_{j+1}' = 10x_j \text{ if } q_{j+1}' = 0$$

or

$$|x_{j+1}'| = |(10|x_j| - 5|d|)| \quad \text{if } q_{j+1}' = \pm 5.$$

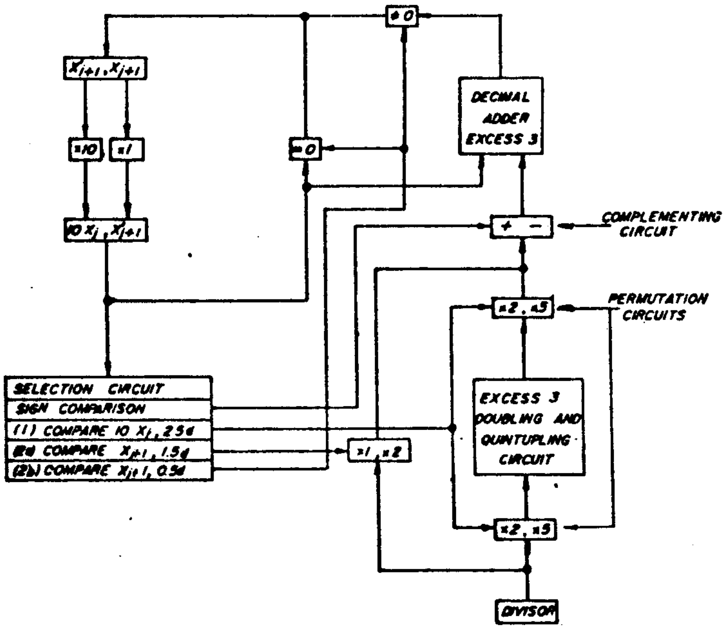

Fig. 3—A decimal division method.

Step 1 reduces the range $0 \leq 10|x_j| < 7\frac{7}{9}|d|$ to $0 \leq |x_{j+1}'| < 2\frac{7}{9}|d|$. The result of step 2 is x_{j+1}, whose range is $0 \leq |x_{j+1}| < \frac{7}{9}|d|$. Step 1 is not required whenever $q_{j+1}'=0$. The details of operations performed for all ranges of $10x_j$ and x_{j+1}' are summarized in Table II. The overlap of $\frac{5}{9}|d|$ in the ranges is such that three decimal digits of d and of $10x_j$ or x_{j+1}' are required for

TABLE II

Range of $10x_j$	Step 1	q_{j+1}'	Range of x_{j+1}'
$-7\frac{7}{9}\|d\| < 10x_j < -2\frac{2}{9}\|d\|$	Add $5\|d\|$	-5	$-2\frac{7}{9}\|d\| < x_{j+1}' < 2\frac{7}{9}\|d\|$
$-2\frac{7}{9}\|d\| < 10x_j < 2\frac{7}{9}\|d\|$	Not required	0	
$2\frac{2}{9}\|d\| < 10x_j < 7\frac{7}{9}\|d\|$	Subtract $5\|d\|$	$+5$	
Range of x_{j+1}'	**Step 2**	q_{j+1}''	**Range of x_{j+1}**
$-2\frac{7}{9}\|d\| < x_{j+1}' < -1\frac{2}{9}\|d\|$	Add $2\|d\|$	-2	$-\frac{7}{9}\|d\| < x_{j+1} < \frac{7}{9}\|d\|$
$-1\frac{7}{9}\|d\| < x_{j+1}' < -\frac{2}{9}\|d\|$	Add $\|d\|$	-1	
$-\frac{7}{9}\|d\| < x_{j+1}' < \frac{7}{9}\|d\|$		0	
$\frac{2}{9}\|d\| < x_{j+1}' < 1\frac{7}{9}\|d\|$	Subtract $\|d\|$	$+1$	
$1\frac{2}{9}\|d\| < x_{j+1}' < 2\frac{7}{9}\|d\|$	Subtract $2\|d\|$	$+2$	

comparison, provided d is standardized to the range $1/10 \leq |d| \leq 1$..

The average number of operations necessary for the division method, assuming all quotient digits $-7, \cdots, +7$ are equally likely, is $1\frac{2}{3}$ operations per digit of the quotient. This figure is to be compared with 3.4 operations per quotient digit for a conventional nonrestoring division method employing doubling and quintupling circuits.[2]

Conclusion

The new methods of division described here, when coupled with two rather obvious comments, show promise of leading to new developments in digital arithmetic. These comments are:

1) Multiplication is the inverse of division.
2) The representation of quotient digits is, at least in the interesting cases, redundant.

This paper is the result of an attempt to find a division inverse to a multiplication method recently described by Lehman.[3] The radix 4 division example is the inverse of the radix 4 equivalent of the binary multiplication described by Lehman, except for the manner in which redundancy is employed in the generation of quoteint digits in the one case and the recoding of multiplier digits in the other. The multiplication method which is the inverse to the radix 10 division is similar to that used on the IBM 602A computing punch;[4] the fact that fewer operations are required per quotient digit for the division method indicates that a better multiplier digit encoding scheme would reduce the number of operations required for multiplication. It is clear that multiplication methods inverse to other division methods of the class described here must exist, and it is felt that further investigation will lead to a broader understanding of digital arithmetic. The interrelationships between quotient digit redundancy, selection procedures, and divisor multiple generation should be studied also, particularly in those cases for which several steps per quotient digit are desirable (*e.g.*, the radix 10 example).

Appendix

Precision Required for Selection

An estimate of the precision required of the divisor d and the shifted partial remainder rx_j for selection of the correct arithmetic procedure can be gained by estimating the variation in the ratio rx_j/d resulting from truncation of rx_j and of d. This variation must be less than the overlap of projections on the rz_j axis (Fig. 1) of two successive lines of the form $z_{j+1} = rz_j - i$ $z_{j+1}' = rz_j - (i+1)$.

The overlap is the difference in values of rz_j for $z_{j+1} = n/(r-1)$, and for $z_{j+1}' = -n/(r-1)$, and is $2n/(r-1) - 1$. Truncation errors in rx_j and d can be expressed by setting $a \leq |x_j| \leq a + \Delta a$, $b \leq |d| \leq b + \Delta b$. The variation in the estimates of the ratio $|rx_j/d|$ is then

$$r\left[\frac{a+\Delta a}{b} - \frac{a}{b+\Delta b}\right].$$

It is required therefore that

$$\frac{a+\Delta a}{b} - \frac{a}{b+\Delta b}$$

$$\simeq \frac{1}{b}\left(\Delta a + \frac{a}{b}\Delta b\right) < \frac{1}{r}\left(\frac{2n}{r-1} - 1\right).$$

Assuming $\Delta a = \Delta b$, one obtains

$$\Delta a < \frac{b[2n - (r-1)]}{(r-1)r(1 + a/b)}.$$

The minimum value of b, assuming standardization, radix r, is $b = 1/r$; the maximum value of $r(a/b)$ is $\frac{1}{2}(2n-1)$. Therefore,

$$\Delta a < \frac{2[2n - (r-1)]}{r(r-1)(2r + 2n - 1)}.$$

For the example $r=4$, $n=2$; $\Delta a < 1/66$, indicating that seven binary digits are required for the comparison. For $r=10$, $n=7$; $\Delta a < 1/297$, indicating that three decimal digits are sufficient for the comparison.

Acknowledgment

The author is grateful for many helpful discussions with staff members of the Digital Computer Laboratory of the University of Illinois.

[2] Richards, *op. cit.*, pp. 274–275.

[3] M. Lehman, "High-speed digital multiplication," IRE Trans. on Electronic Computers, vol. EC-6, pp. 204–205; September, 1957.

[4] Richards, *op. cit.*, pp. 262–263.