# Computer Multiplication and Division Using Binary Logarithms*

JOHN N. MITCHELL, JR.,† ASSOCIATE, IRE

*Summary*—A method of computer multiplication and division is proposed which uses binary logarithms. The logarithm of a binary number may be determined approximately from the number itself by simple shifting and counting. A simple add or subtract and shift operation is all that is required to multiply or divide. Since the logarithms used are approximate there can be errors in the result. An error analysis is given and a means of reducing the error for the multiply operation is shown.

## I. Introduction

MULTIPLICATION and division operations in computers are usually accomplished by a series of additions and subtractions, and shifts. Consequently, the time required to execute such commands is much longer than the time required to execute an add or subtract command.

Logarithms have long been used as a tool in mathematics to simplify the process of multiplication, division, roots, powers, etc. They reduce multiplication and division problems to addition and subtraction; power and root problems are reduced to multiplication and division.

The storage of a table of logarithms in a computer would be inconvenient considering the memory capacity that would be required. On the other hand, the calculation of logarithms would generally require more machine time than the iterative operations of addition and subtraction.

The purpose of this paper is to describe a scheme of computer arithmetic which utilizes binary logarithms in the operations of multiplication and division. The logarithms used in the arithmetic are approximations to the actual logarithms; because of the approximations, there will be errors in the results of operations using them. It is considered that the simplicity of the method of finding and using these logarithms may make the scheme valuable in some applications.

A method of finding approximate logarithms to the base two will be described and an analysis will be made to determine the maximum errors that may occur as a result of the approximation. A method of reducing the error in multiplication will be shown.

## II. Binary Logarithms

Since computers use binary arithmetic it would seem natural that if logarithms were to be used they would be binary logarithms (to the base two). Since $\log_{10} N$ is usually written $\log N$ and $\log_e N$ is written $\ln N$, to avoid ambiguity and the necessity of writing the subscript a similar notation will be adopted in this paper to imply $\log_2 N$:

$$\lg N \equiv \log_2 N.$$

A table of binary logarithms is shown in Fig. 1, and the familiar logarithmic curve is plotted in Fig. 2. Suppose the points where $\lg N$ is an integer are connected by straight lines. The dashed lines in Fig. 2 describe the resulting curve. If $\lg N$ is approximated by the dashed line curve the data shown in Fig. 3 is obtained. Consider the second and fourth columns which are written in binary form. The characteristic of the logarithm may be determined by inspection; it is the bit position of the most significant "one" bit starting the count at zero. The approximate mantissa is contained in the number itself. The bits to the right of the most significant "one" automatically interpolate between zero and one on a straight line curve.

Hence, to find the binary logarithm of a binary number observe the bit position of the most significant "one," ignore the most significant "one" and interpret the rest of the number as a binary fraction. For example, find the approximate lg 13 which from Fig. 3 should be 3.625.

$$13 = 1101.$$

The most significant "one" bit is in the $2^3$ position, hence the characteristic is three. Considering the bits to the right of the most significant "one" as a binary fraction there results 0.101 which is equivalent to 0.625 in decimal notation. Approximate lg 13 is then 3.625 decimal or 11.101 binary.

One can see how a machine might make use of binary logarithms very easily with no table look-ups. The characteristic can be generated by shifting the machine word until the most significant "one" appears in the left-most position. A counter counts down from $n$ (where $n+1$ is the number of bits in the machine word) one count for each bit shifted. When the most significant "one" appears in the most significant bit position of the word the counter will contain the characteristic, and the mantissa will lie to the right of the most significant bit position.

As an example, consider Fig. 4. Registers $A$ and $B$ contain two numbers. Suppose it is desired to multiply

* Received March 19, 1962.
† Research and Development Division, National Cash Register Company, Dayton, Ohio.

| $N$ | 1g $N$ |
|---|---|
| 1 | 0.00000 |
| 2 | 1.00000 |
| 3 | 2.58496 |
| 4 | 2.00000 |
| 5 | 2.32193 |
| 6 | 2.58496 |
| 7 | 2.80735 |
| 8 | 3.00000 |
| 9 | 3.16992 |
| 10 | 3.32193 |
| 11 | 3.45942 |
| 12 | 3.58496 |
| 13 | 3.70043 |
| 14 | 3.80735 |
| 15 | 3.90689 |
| 16 | 4.00000 |
| 17 | 4.08747 |

Fig. 1—Partial table of binary logarithms.

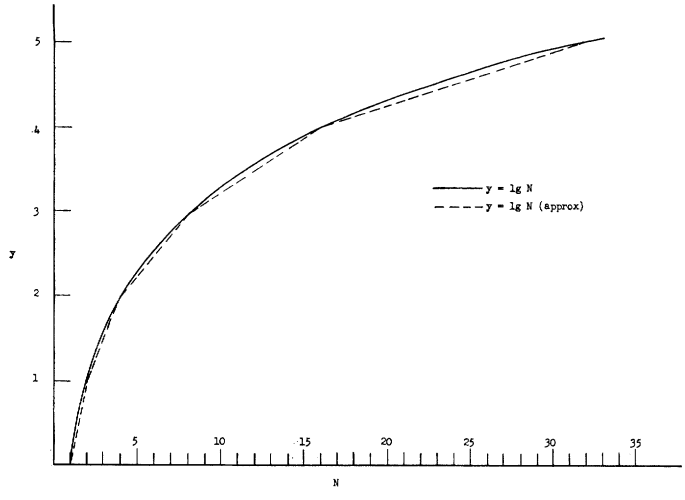


Fig. 2—Logarithmic curve and its straight-line approximation.

| $N$ | $N$ (binary) | Approx 1g $N$ | Approx 1g $N$ (binary) |
|---|---|---|---|
| 1 | 00001 | 0.000 | 000.0000 |
| 2 | 00010 | 1.000 | 001.0000 |
| 3 | 00011 | 1.500 | 001.1000 |
| 4 | 00100 | 2.000 | 010.0000 |
| 5 | 00101 | 2.250 | 010.0100 |
| 6 | 00110 | 2.500 | 010.1000 |
| 7 | 00111 | 2.750 | 010.1100 |
| 8 | 01000 | 3.000 | 011.0000 |
| 9 | 01001 | 3.125 | 011.0010 |
| 10 | 01010 | 3.250 | 011.0100 |
| 11 | 01011 | 3.375 | 011.0110 |
| 12 | 01100 | 3.500 | 011.1000 |
| 13 | 01101 | 3.625 | 011.1010 |
| 14 | 01110 | 3.750 | 011.1100 |
| 15 | 01111 | 3.875 | 011.1110 |
| 16 | 10000 | 4.000 | 100.0000 |
| 17 | 10001 | 4.0625 | 100.0001 |

Fig. 3—Table of binary logarithms (straight-line approximation).

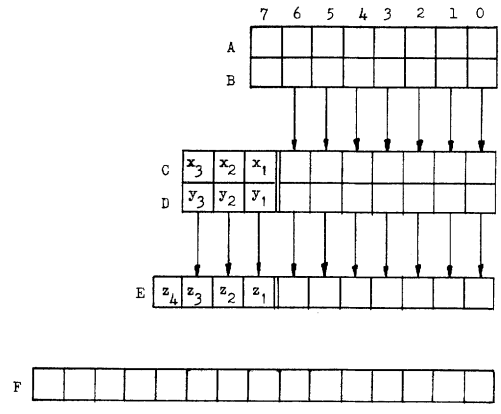


Fig. 4—Example of machine organization to generate and use binary logarithms.

or divide these numbers ($A \times B$ or $A \div B$). The word size in this example is eight bits so the largest possible characteristic will be seven. $x_3x_2x_1$ and $y_3y_2y_1$ each initially contain 111.

Step 1—Shift $A$ and $B$ left until their most significant "one" bits are in the left-most positions and count down counters $x_3x_2x_1$ and $y_3y_2y_1$ during the shifting. After the shifting is completed the counters will contain the characteristics of the logarithms of $A$ and $B$.

Step 2—Shift bits 0–6 of $A$ and $B$ into bit positions 0–6 of $C$ and $D$ as indicated in Fig. 4. $C$ and $D$ now contain the logarithms of the original numbers.

Step 3—Add or subtract $C \pm D \rightarrow E$. This puts the logarithm of the result in $E$.

Step 4—Decode $z_4z_3z_2z_1$ and insert a "one" in appropriate position of $F$. Insert the right-hand portion of $E$ immediately to the right of this "one." $F$ now contains the result.

As an illustration of the use of binary logarithms consider the problem of dividing 3216 by 25. The result should be 128.64.

$$\begin{aligned} 3216 &= 110010010000 \\ 25 &= 000000011001 \\ \lg 3216 &= 1011.1001001 \\ \lg 25 &= 0100.1001 \\ \lg 3216 - \lg 25 &= 0111.0000001 \\ \text{Result} &= 10000001 \\ &= 129. \end{aligned}$$

Not all results will be as close as this however. Consider 15 divided by 3 for example.

$$\begin{aligned} 15 &= 1111 \\ 3 &= 0011 \\ \lg 15 &= 11.111 \\ \lg 3 &= 01.100 \\ \lg 15 - \lg 3 &= 10.011 \\ \text{Result} &= 101.1 \\ &= 5.5 \quad \text{for a 10 per cent error.} \end{aligned}$$

The above discussion shows that approximate binary logarithms are fairly simple to generate in a machine since all the information is contained in the number itself. A simple shift and count scheme is all that is necessary. Multiplication and division operations are reduced to shifting and a single addition or subtraction. The last example given above had an error of 10 per cent in the answer. Section III of this paper will be devoted to an analysis of the logarithm scheme to determine the maximum errors that will occur; a means of reducing the error for multiplication will be developed in Section IV.

## III. Error Analysis

The problem of determining the accuracy of multiplication and division operations will be taken up. The example given in the previous section shows that a 10 per cent error is possible in division. The maximum possible error in the logarithm and the maximum possible errors in multiplication and division will be determined.

Consider a binary number $N$ in the interval $2^{k+1} > N \geq 2^j$, where $j=0, \pm 1, \pm 2, \cdots$; $k=0, \pm 1, \pm 2, \cdots$; and $k \geq j$.

$$n = z_k \cdots z_4 z_3 z_2 z_1 z_0 . z_{-1} z_{-2} z_{-3} \cdots z_j.$$

The number $N$ can be written

$$N = \sum_{i=j}^{k} 2^i z_i$$

where $z_i = 0, 1$. Since $Z_k$ is the most significant bit we may assume $Z_k = 1$ without loss of generality. If $2^k$ is factored out

$$N = 2^k \left(1 + \sum_{i=j}^{k-1} 2^{i-k} z_i\right). \tag{1}$$

Let the term

$$\sum_{i=j}^{k-1} 2^{i-k} z_i = x. \tag{2}$$

Since $k \geq j$, $x$ will fall in the range $0 \leq x < 1$.

$$N = 2^k(1 + x). \tag{3}$$

Note that the term $x$ is that part of the number to the right of the most significant "one" interpreted as a binary fraction in the discussion above.

### A. Error in the logarithm

The approximation to the logarithm $(\lg N)'$ as used in the arithmetic scheme described above is

$$(\lg N)' = k + x. \tag{4}$$

The actual logarithm is

$$\lg N = k + \lg (1 + x). \tag{5}$$

Let $A = \lg N$ and $A' = (\lg N)'$. The error in the approximation of the logarithm is

$$R = A - A'. \tag{6}$$

Substituting (4) and (5) in (6)

$$\begin{aligned} R &= k + \lg (1 + x) - (k + x) \\ &= \lg (1 + x) - x. \end{aligned} \tag{7}$$

Since $x$ can have values only in the range $0 \leq x < 1$, zero error will occur when $x = 0$. The point of maximum error is found by differentiating and setting the derivative equal to zero:

$$\frac{dR}{dx} = \frac{1}{(1 + x) \ln 2} - 1 = 0$$

$$x = \frac{1}{\ln 2} - 1 = 0.44269. \tag{8}$$

The error $R$ will be in the range $0 \leq R \leq [\lg (1.44269) - 0.44269]$ or $0 \leq R \leq 0.08639$.

Note that the number 0.08639 is not a percentage, but the absolute value of the maximum error in the logarithm.

### B. Error in the Product and Quotient

The logarithm of a product is equal to the sum of the logarithms of the multiplier and the multiplicand.

$$\lg P = k_1 + \lg (1 + x_1) + k_2 + \lg (1 + x_2) \tag{9}$$

$$P = 2^{k_1 + k_2}(1 + x_1)(1 + x_2). \tag{10}$$

The subscripts $_1$ and $_2$ refer to the multiplicand and multiplier, respectively.

Similarly, the approximation

$$\lg P' = k_1 + x_1 + k_2 + x_2. \tag{11}$$

This is rewritten as two separate expressions to account for the two possible cases of carrys. One is the case of no carry from the mantissa $x$ to the characteristic $k$ and the other is the case where there is such a carry.

$$\lg P' = k_1 + k_2 + (x_1 + x_2) \qquad x_1 + x_2 < 1 \tag{12}$$

$$\lg P' = 1 + k_1 + k_2 + (x_1 + x_2 - 1) \qquad x_1 + x_2 \geq 1. \tag{13}$$

Taking the antilogarithm

$$P' = 2^{k_1 + k_2}(1 + x_1 + x_2) \qquad x_1 + x_2 < 1 \tag{14}$$

$$P' = 2^{1 + k_1 + k_2}(x_1 + x_2) \qquad x_1 + x_2 \geq 1. \tag{15}$$

The error $E_m$ in multiplication is defined as

$$E_m = \frac{P' - P}{P} = \frac{P'}{P} - 1. \tag{16}$$

Substituting (10) and (14), and (10) and (15) in (16)

$$E_{m1} = \frac{1 + x_1 + x_2}{(1 + x_1)(1 + x_2)} - 1 \qquad x_1 + x_2 < 1 \qquad (17)$$

$$E_{m2} = \frac{2(x_1 + x_2)}{(1 + x_1)(1 + x_2)} - 1 \qquad x_1 + x_2 \geq 1. \qquad (18)$$

The two expressions for $E_m$, (17) and (18), represent the error that may occur when two numbers are multiplied according to the arithmetic scheme proposed. The error for division will now be determined.

The logarithm of a quotient is equal to the difference between the logarithms of the dividend and the divisor.

$$\lg Q = k_1 + \lg (1 + x_1) - k_2 - \lg (1 + x_2) \qquad (19)$$

$$Q = \frac{2^{k_1+k_2}(1 + x_1)}{(1 + x_2)}. \qquad (20)$$

The subscripts 1 and 2 refer to the dividend and the divisor, respectively.

The approximation

$$\lg Q' = k_1 + x_1 - k_2 - x_2. \qquad (21)$$

This expression is also written in two parts to account for cases of the presence or absence of a borrow from the characteristic. Thus

$$\lg Q' = k_1 - k_2 + (x_1 - x_2) \qquad x_1 - x_2 \geq 0 \qquad (22)$$

$$\lg Q' = k_1 - k_2 - 1 + (1 + x_1 - x_2) \qquad x_1 - x_2 < 0 \qquad (23)$$

$$Q' = 2^{k_1-k_2}(1 + x_1 - x_2) \qquad x_1 - x_2 \geq 0 \qquad (24)$$

$$Q' = 2^{k_1-k_2-1}(2 + x_1 - x_2) \qquad x_1 - x_2 < 0. \qquad (25)$$

The division error $E_d$ is defined as

$$E_d = \frac{Q' - Q}{Q} = \frac{Q'}{Q} - 1. \qquad (26)$$

Substituting (20) and (24), and (20) and (25) in (26)

$$E_{d_1} = \frac{(1 + x_1 - x_2)(1 + x_2)}{(1 + x_1)} - 1 \qquad x_1 - x_2 \geq 0 \qquad (27)$$

$$E_{d_2} = \frac{1}{2}\,\frac{(2 + x_1 - x_2)(1 + x_2)}{(1 + x_1)} - 1 \qquad x_1 - x_2 < 0. \qquad (28)$$

The expressions for $E_m$ and $E_d$ will now be examined for their maximum and minimum values and the values of $x_1$ and $x_2$ will be determined at those points.

Consider (17), the first expression for the multiply error.

$$E_{m_1} = \frac{1 + x_1 + x_2}{(1 + x_1)(1 + x_2)} - 1 \qquad x_1 + x_2 < 1. \qquad (17)$$

It should be recalled that $x$ is a number between zero and one.

Let the term

$$\frac{1 + x_1 + x_2}{(1 + x_1)(1 + x_2)} = Z_1 \qquad (29)$$

and

$$x_1 + x_2 = \alpha < 1 \qquad (30)$$

$$Z_1 = \frac{1 + \alpha}{1 + \alpha + x_2(\alpha - x_2)} \qquad (31)$$

$$\frac{dZ_1}{dx_2} = \frac{-(1 + \alpha)(\alpha - 2x_2)}{[1 + \alpha + x_2(\alpha - x_2)]^2} = 0 \qquad (32)$$

$$\alpha = 2x_2$$

$$x_2 = \frac{\alpha}{2} \qquad (33)$$

$$x_2 = x_1.$$

Since $x_1+x_2<1$ for this case, $\alpha$ will range between zero and one. At these extremes of $\alpha$ the following table results:

| | | |
|---|---|---|
| At $\alpha = 0$ | $Z_1 = 1$ | $E_{m_1} = 0$ |
| At $\alpha = 1$ | $Z_1 = 8/9$ | $E_{m_1} = -1/9.$ |

At all other possible values of $\alpha$ the error will lie between zero and $-11.1$ per cent, where the negative sign implies that the answer is low. The values of $x_1$ and $x_2$ when the error is maximum must each be $\frac{1}{2}$ since $\alpha=1$ and $x_1$ and $x_2$ must be equal [from (33)].

Consider next (18), the second expression for the multiply error.

$$E_{m_2} = \frac{2(x_1 + x_2)}{(1 + x_1)(1 + x_2)} - 1 \qquad \begin{matrix} x_1 + x_2 \geq 1 \\ 0 \leq x < 1 \end{matrix} \qquad (18)$$

Let

$$\frac{2(x_1 + x_2)}{(1 + x_1)(1 + x_2)} = Z_2 \qquad (34)$$

and

$$x_1 + x_2 = \alpha. \qquad (35)$$

Since $x_1+x_2\geq 1$ and $0\leq x<1$, $\alpha$ in this case is in the range $1\leq\alpha<2$.

$$Z_2 = \frac{2\alpha}{1 + \alpha + x_2(\alpha - x_2)} \qquad (36)$$

$$\frac{dZ_2}{dx_2} = \frac{-2\alpha(\alpha - 2x_2)}{[1 + \alpha + x_2(\alpha - x_2)]^2} = 0 \qquad (37)$$

$$\alpha = 2x_2$$

$$x_2 = \frac{\alpha}{2} \qquad (38)$$

$$x_1 = x_2.$$

Again considering the extreme values of $\alpha$ the following table is obtained:

| | | |
|---|---|---|
| At $\alpha = 1$ | $Z_2 = 8/9$ | $E_{m_2} = -1/9$ |
| At $\alpha = 2$ | $Z_2 = 1$ | $E_{m_2} = 0$ |

As in the previous case, the error will lie between zero and $-11.1$ per cent and the values of $x_1$ and $x_2$ at the point of maximum error will be $\frac{1}{2}$.

The divide errors $E_{d_1}$ and $E_{d_2}$ will be investigated.

$$E_{d_1} = \frac{(1 + x_1 - x_2)(1 + x_2)}{(1 + x_1)} - 1 \qquad \begin{matrix} x_1 - x_2 \geq 0 \\ 0 \leq x < 1 \end{matrix} \tag{27}$$

Simplifying this expression yields

$$E_{d_1} = \frac{x_2(x_1 - x_2)}{1 + x_1}. \tag{39}$$

Since the $x$'s in this expression are less than unity, $E_{d_1}$ will be maximum when $x_1$ approaches unity. Substituting $x_1 = 1$ in (39)

$$E_{d_1} = \frac{x_2(1 - x_2)}{2} \tag{40}$$

$$\frac{dE_{d_1}}{dx_2} = 0 = 1 - 2x_2 \tag{41}$$

$$x_2 = 1/2.$$

So $E_{d1}$ is maximum when $x_1 = 1$ and $x_2 = \frac{1}{2}$.

Substituting these values in (39) gives $E_{d_1} = \frac{1}{8}$ or 12.5 per cent. The minimum value of $E_{d_1}$ is zero and occurs when $x_2$ is zero or when $x_2 = x_1$.

$$E_{d_2} = \frac{(2 + x_1 + x_2)(1 + x_2)}{2(1 + x_1)} - 1 \qquad \begin{matrix} x_1 - x_2 < 0 \\ 0 \leq x < 1. \end{matrix} \tag{28}$$

Rewriting (28) yields

$$E_{d_2} = \frac{(x_2 - x_1)(1 - x_2)}{2(1 + x_1)}. \tag{42}$$

In the given number range $E_{d_2}$ is maximum when $x_1$ is at its smallest possible value.

At

$$x_1 = 0, \qquad E_{d_2} = \frac{x_2(1 - x_2)}{2} \tag{43}$$

$$\frac{dE_{d_2}}{dx_2} = 0 = 1 - 2x_2$$

$$x_2 = 1/2. \tag{44}$$

$E_{d_2}$ is maximum when $x_1 = 0$ and $x_2 = \frac{1}{2}$. Substituting these values in (42) gives $E_{d_2} = \frac{1}{8}$ or 12.5 per cent.

The minimum value of $E_{d_2}$ is zero and occurs when $x_2$ and $x_1$ are equal or when $x_2 = 1$ (of course $x_2$ can never reach unity since $0 \leq x < 1$).

To summarize, the maximum possible multiply error will be $-11.1$ per cent and will occur at $x_1 = x_2 = \frac{1}{2}$. The maximum possible divide error will be 12.5 per cent and will occur at $x_1 = 1$ and $x_2 = \frac{1}{2}$ for the case where there is no borrow from the characteristic, and at $x_1 = 0$ and $x_2 = \frac{1}{2}$ when there is such a borrow. The minimum multiply error will be zero and will occur when $x_1$ and $x_2$ are zero. The minimum error for divide will also be zero and will occur when $x_2$ is zero or when $x_1$ and $x_2$ are equal.

The foregoing analysis of the method of logarithmic multiplication and division shows that the errors are rather high. Although such errors might be tolerated in some applications, general use in computers requires much better accuracy. It should be noted that the errors are always positive for division and negative for multiplication so that errors may be compounded if successive operations of the same type are made on the data. The following section of this paper will be devoted to a means of reducing the multiply error.

## IV. Reducing the Multiplication Error

Two expressions were written for the approximate product of two numbers. They are repeated here.

$$P' = 2^{k_1+k_2}(1 + x_1 + x_2) \qquad x_1 + x_2 < 1 \tag{14}$$

$$P' = 2^{1+k_1+k_2}(x_1 + x_2) \qquad x_1 + x_2 \geq 1. \tag{15}$$

The actual product was

$$P = 2^{k_1+k_2}(1 + x_1)(1 + x_2). \tag{10}$$

Expanding (10) yields

$$P = 2^{k_1+k_2}(1 + x_1 + x_2 + x_1x_2). \tag{45}$$

The magnitude of the error in (14) is

$$(45) - (14) = 2^{k_1+k_2}(x_1x_2). \tag{46}$$

Similarly, the magnitude of the error in (15) is

$$(45) - (15) = 2^{k_1+k_2}[1 + x_1x_2 - (x_1 + x_2)]. \tag{47}$$

Expression (46) is a simple correction term, however, (47) at first glance seems rather complex. It may be simplified, however, in the following way.

Let $x_1 = 1 - x_1'$ and $x_2 = 1 - x_2'$, where $x_1'$ and $x_2'$ are the two's complements of $x_1$ and $x_2$. The bracketed term in expression (47) in terms of $x_1'$ and $x_2'$ becomes

$$1 + (1 - x_1')(1 - x_2') - (1 - x_1' + 1 - x_2') = x_1'x_2'. \tag{48}$$

The magnitude of the error in (15) reduces to $2^{k_1+k_2}(x_1'x_2')$.

The error in multiplication may be eliminated if

$2^{k_1+k_2}(x_1x_2)$ is added to the result if there was no carry into the characteristic; or if $2^{k_1+k_2}(x_1'x_2')$ is added to the result if there was a carry into the characteristic.

If (14) and (15) are rewritten to account for the error, there results

$$P' = 2^{k_1+k_2}(1 + x_1 + x_2) + 2^{k_1+k_2}(x_1x_2) \quad x_1 + x_2 < 1 \quad (49)$$

$$P' = 2^{1+k_1+k_2}(x_1 + x_2) + 2^{k_1+k_2}(x_1'x_2') \quad x_1 + x_2 \geq 1. \quad (50)$$

A machine could handle the correction easily. Perform the logarithmic multiplication as described in Section II of this paper. Sense whether or not there is a carry from the mantissa into the characteristic. Let $x_1x_2$ or $x_1'x_2'$ as the case may be, be a new product. Generate this product by another similar addition operation. Add this correction (scaled by the factor $2^{k_1+k_2}$) to the previous result.

The extra two additions (one to perform the multiplication and the other to add to the previous result) will serve to reduce the error so that the maximum error that can occur as a result of the double operation cannot exceed 2.8 per cent. This maximum error is demonstrated as follows: If $x_1+x_2<1$, let

$$x_1 = 2^{-j_1}(1 + x_3)$$

$$x_2 = 2^{-j_2}(1 + x_4)$$

$$x_1x_2 \approx 2^{-(j_1+j_2)}(1 + x_3 + x_4).$$

Substituting in the *error term* of (49), the new approximation to $P$ becomes:

$$P' = 2^{k_1+k_2}(1+x_1+x_2)+2^{k_1+k_2}\,2^{-(j_1+j_2)}(1+x_3+x_4)$$

from (10), (14), and (16)

$$E_m = \frac{2^{k_1+k_2}(1+x_1+x_2)+2^{k_1+k_2-j_1-j_2}(1+x_3+x_4)-2^{k_1+k_2}(1+x_1)(1+x_2)}{2^{k_1+k_2}(1+x_1)(1+x_2)} \quad (51)$$

$$E_m = \frac{2^{-(j_1+j_2)}(1+x_3+x_4)-x_1x_2}{(1+x_2)(1+x_2)}. \quad (52)$$

Since $x_1+x_2<1$ and, as before, error is maximum at $x_1=x_2\approx\frac{1}{2}$, we have $j_1=j_2=2$, $x_3=x_4\approx 1$.
Then

$$E_m = \frac{2^{-4}(1 + 1 + 1) - \frac{1}{2} \times \frac{1}{2}}{(1 + \frac{1}{2})(1 + \frac{1}{2})} = -\frac{1}{36}, \text{ or } -2.8 \text{ per cent.}$$

A similar result is obtained for the case where $x_1+x_2\geq 1$ when the substitutions $1-x_1=2^{-j_1}(1+x_3)$ and $1=-x_2=2^{-j_2}(1+x_4)$ are made in the *error term* of (50).

This error might be reduced to an arbitrary value by adding similar correction operations, however, if too many such operations are required one may as well use conventional iterative multiplication techniques.

A comment should be made with regard to the division error. Unfortunately, no such simple scheme to reduce this error has been developed at this writing. The error term for divide is a series which does not converge fast enough to allow the use of its first term . This disadvantage however, should not discourage the use of binary logarithms for multiplication.

## V. Conclusion

Approximations to binary logarithms are very easy to generate. No table looks-ups are required, and multiplication and division operations are reduced to addition and subtraction operations. The purpose of using logarithms is to gain speed. The accuracy of a computation using logarithms depends on the accuracy of the tables used. The method proposed in this paper does not use tables but a straight line interpolation between the points where the mantissa is zero is utilized. For this reason there can be errors in the results of operations using this approximation. The multiplication error can be as large as $-11.1$ per cent but this may be reduced by a double or higher-order operation. The division error can be as large as 12.5 per cent. Unfortunately, the errors for any particular type of operation are in the same direction so that cancellation of errors is not possible. A means of reducing the multiply error has been suggested. It is desirable that a simple method of reducing the divide error be available. (One method which might be used is to store correction factors for various intervals of $x$. This would require a compare operation and a memory lookup for each operand and the time involved might defeat the purpose of using logarithms.)

Even with the disadvantages it is considered that the ease of generating approximations to binary logarithms makes them valuable for some special applications. Further work in this area might point the way for their use, not necessarily restricting them to the operations of multiplication and division but other functions as well.