Radix Conversion in an Unnormalized Arithmetic System

By N. Metropolis and R. L. Ashenhurst

Introduction. The question of radix conversion of variable-precision binary numbers arises naturally in the context of unnormalized number representation [1], but may be of interest in other situations where it is desired to have number representations carry a reflection of significance. The present paper discusses a method for binary-decimal conversion of unnormalized numbers; this method differs in certain respects from one previously developed, and described elsewhere [2], for use with the MANIAC III computer. The question of decimal-binary conversion, taking into account explicit "uncertainty" in the decimal representation, is also investigated from the significance viewpoint.

Binary-Decimal Conversion. First, consider the task of converting a number in unnormalized floating point binary form to a decimal in such a way that a discrepancy in the lowest-order binary digit corresponds to a discrepancy in the lowest-order decimal digit in the result (the standard MANIAC III routine for doing this has a "guard bit" provision for specifying a position other than the lowest-order one, but this feature can be incorporated by a preliminary transformation and so is neglected in the development here). The question of whether the decimal exponent of the result is represented explicitly or by the insertion of a decimal point character is irrelevant to the present discussion; it will be assumed that the desired output is a string of decimal characters, representing an integer, and a second integer specifying an associated power of 10.

Since $2^{-10} \approx 10^{-3}$, equivalent precision in binary and decimal is given by numbers of digits in roughly the ratio 10 to 3; one could, of course, simply keep a count of decimal characters generated in a standard conversion procedure and stop at some approximately appropriate point. It seems not unreasonable, however, to ask for a conversion procedure which affords the user a more precise statement of the relation between the binary form and the decimal result. The observation that the exact conversion of integers gives also a true estimate of precision (i.e., the 10-for-3 criterion is naturally achieved), suggests that binary-decimal conversion can be accomplished by first transforming the floating point number to an integer expressed with the same significance, which differs from the original number only by a power of 10, and then taking the converted representation of this integer as the desired decimal string.

Both the earlier method [2] and the one here described are based on this notion; the main difference centers around the preliminary transformation. It is believed that certain formal advantages accrue from the present approach, particularly with respect to the straightforward representation of the conversion error.

Description of the Conversion. It is assumed that a sign-and-magnitude decimal representation is desired, so that negative numbers are made positive before con-

Received September 8, 1964. This work was supported by the Atomic Energy Commission under contract no. AT(11-1)-614.

version. Let (e, f) represent an unnormalized floating point number, $x = 2^e f$, with e the integral exponent and f a non-negative (generally unnormalized) coefficient, represented as a p-place binary number according to the usual fixed-point convention which requires $f < 1$. The conversion procedure is to produce an integer d, represented by a sequence $d_0, d_1, \cdots$, of n decimal digits (lowest-order written first), and an integer m which is the associated power of 10; that is, the relation

$$2^e f = 10^m d,$$

where

$$d = \sum_{k=0}^{n-1} 10^k \cdot d_k,$$

is to be satisfied to the appropriate degree of precision.

The approach presently advocated is based on the following definitions:

(1) Define

$$t = p - e$$

and

$$\bar{x} = 2^t x;$$

then

$$\bar{x} \text{ is an integer}, \qquad \bar{x} = 2^p f.$$

(2) Define m as the unique integer satisfying

$$1 \leqq 10^{-m} 2^{-t} < 10.$$

and let

$$w = 10^{-m} 2^{-t}.$$

(3) Define d as the unique integer satisfying

$$d - \tfrac{1}{2} \leqq w\bar{x} < d + \tfrac{1}{2},$$

and r as the remainder

$$r = w\bar{x} - d,$$

which thus satisfies

$$-\tfrac{1}{2} \leqq r < \tfrac{1}{2}.$$

It may now be verified that

$$\begin{aligned} 10^m(d + r) &= 10^m w\bar{x} \\ &= 10^m(10^{-m}2^{-t})(2^t x) \\ &= x; \end{aligned}$$

hence the decimal representation of d may be taken to represent x scaled by 10^{-m}. That the number of digits in d appropriately reflects the significance of the representation of x is seen as follows; let x^{T} be a "true" value of which x is an approximation; then define the error δ by

$$\delta = 2^t(x - x^{\mathrm{T}});$$

effectively, δ is the error measured in units of the pth place of f. Thus

$$x = x^{\mathrm{T}} + 2^{-t}\delta$$

and

$$\begin{aligned} d &= 10^{-m}x - r \\ &= 10^{-m}(x^{\mathrm{T}} + 2^{-t}\delta) - r \\ &= 10^{-m}x^{\mathrm{T}} + w\delta - r. \end{aligned}$$

The value

$$\epsilon = w\delta - r$$

may thus be taken as the error in d, measured in integral units, induced by the error δ in $\bar{x}$; since $-\frac{1}{2} \leqq r < \frac{1}{2}$, it follows that the essential effect is the multiplication of the original error δ by w, where $1 \leqq w < 10$. Thus the error propagation effect in the conversion (some manner of which is inevitable due to the incommensurability of the radices) is well represented by the magnitude of w; a convenient indication of this is provided by making the integer part of w (perhaps rounded) available to the user along with d and m.

Computing Method. The foregoing description is essentially abstract, and serves to indicate the basic features of the conversion procedure. In order to carry it out in practice, however, attention must be paid to questions of number representation, rounding, etc. These can be summed up as follows:

(1) The computation of $\bar{x}$ from x is performed without error; if (e, f) represents x, then (p, f) represents $\bar{x}$. The computation of t is also performed without error, since t is an integer.

(2) The value $w = 10^{-m}2^{-t}$ is, in general, not representable exactly with p binary digits; however, a normalized rounded representation will have a relative error of order 2^{-p-1}, which ordinarily induces an error in the product $w\bar{x}$ which is small compared to the remainder r, and hence does not affect the value of the integer d. To characterize the situation more precisely, suppose that w^* is an n-digit approximation to w, satisfying

$$w^* = w(1 + \sigma), \quad -2^{-p-1} \leqq \sigma < 2^{-p-1};$$

then the product $w^*\bar{x}$ is used to define d^* and r^* satisfying

$$\begin{aligned} d^* - \tfrac{1}{2} \leqq w^*\bar{x} &< d^* + \tfrac{1}{2}, \\ r^* &= w^*\bar{x} - d^*. \end{aligned}$$

The difference $w^*\bar{x} - w\bar{x} = w\bar{x}\sigma$ is bounded by

$$| w\bar{x}\sigma | < 10 \cdot 2^{-p-1}\bar{x},$$

and this is, in general, small unless $\bar{x}$ is very large (which is the case only when x is represented to practically full significance). When $\bar{x}$ is small, it may cause $d^* \neq d$

only when $r \approx \pm\frac{1}{2}$, and, in this case, the values of d^* and d differ by unity, and $r^* \approx \mp \frac{1}{2}$. When $\bar{x}$ is very large, it may cause $d^* \neq d$ by an appreciable amount (since $|\bar{x}| < 2^p$, $|w\bar{x}\sigma| < 5$ is the best absolute bound attainable), but this is the case where x is known very precisely to begin with.

(3) The calculation of d^* and r^* from the product $w^*\bar{x}$ is exact, and in those cases where $d^* = d$, the overall calculation of d may be regarded as exact, despite the error in using w^* instead of w. The exceptional cases are those discussed above under (2).

The actual computational procedure is, of course, conditioned by the instructions available on the computer which is used to carry it out; the MANIAC III vocabulary includes exponent manipulation instructions well suited to performing the computation of t and $\bar{x}$, and a "specified point" multiplication instruction which allows the computation of $w\bar{x}$ and its decomposition into d and r to be accomplished straightforwardly.

The determination of m and w is perhaps that part of the computation which requires most programming attention. A straightforward method is to have a table of m and w as a function of t, and achieve the desired end by a table-lookup operation. The total number of possible t values, however, is of the order of the number of possible exponent values, which, in general, leads to a rather sizable table. If it is noted that the defining relation for p implies

$$0 \leqq -m - t \log_{10} 2 < 1,$$

it is seen that m is determined as the integer part of $-t \log_{10} 2$; by computing this, one can obtain w from a table with one entry for each possible m value (if one is willing to compute $w = 10^{-m}2^{-t}$ from m and t, no table at all is necessary).

One more seeming difficulty occurs in practice; if x requires full p-digit precision to represent it, then the integer d may require more than p digits, since d may be up to 10 times $\bar{x}$. In the actual MANIAC III program this difficulty is neatly avoided by multiplying by $w/10$ instead of w; since

$$\frac{w}{10} x = \frac{d - d_0}{10} + \frac{d_0 + r}{10},$$

the integer part of this product is essentially that which is ordinarily obtained after the first stage of conversion of d to decimal form. This stage is therefore skipped, and the integer d_0 is determined from the fractional part of the initial product.

Decimal-Binary Conversion. The basic task in decimal-binary conversion may be taken to be the reversal of binary-decimal conversion; thus, it is appropriate to consider the problem of obtaining from a given integer d and decimal scale factor m a binary number $x \doteq 10^m d$, represented in a form which would have given d and m on conversion by the procedure of the previous sections. It is evident that this requirement does not uniquely define the value t, since it was only required earlier that $1 \leqq 10^{-m}2^{-t} < 10$; hence, it may be further specified that t satisfy $1 \leqq 10^{-m}2^{-t} < 2$, so that the binary form obtained is that which converts to d and m with the minimum possible value of $w = 10^{-m}2^{-t}$. Once w is defined, $\bar{x}$ is determined as the rounded integer quotient obtained by dividing d by w; the remainder of this division is $r = d - w\bar{x}$, which must satisfy $-\frac{1}{2} \leqq r < \frac{1}{2}$. The final result is then defined as $x = 2^{-t}\bar{x}$, differing in form from $\bar{x}$ only in exponent.

Computing Method. The integer d and normalized 10^{-m} have the following representations:

$$d \sim (p, 2^{-p}d),$$
$$10^{-m} \sim (t + 1, 2^{-t-1}10^{-m}).$$

The value of $d/10^{-m} \doteq x$ is unchanged if both numerator and denominator are multiplied by the same factor; hence, x is approximately obtained by dividing

$$2^{-t}d \sim (p - t, 2^{-p}d)$$

by

$$2^{-t}10^{-m} \sim (1, 2^{-t-1}10^{-m}).$$

Assume d and m have been converted to binary form, and the value 10^{-m} to have been obtained, from a table or otherwise, with exponent $t + 1$; the MANIAC III exponent manipulation instructions then allow the multiplication of d and 10^{-m} by 2^{-t} to be simply effected, and a "specified point" division can be used to obtain a quotient with exponent $p - t$ (agreeing with that of the dividend). If steps are taken to see that the quotient is correctly rounded, the result should be a binary representation of x which, when reconverted by the binary-decimal algorithm, again yields d and m. In practice, the fact that the dividend 10^{-m} can only be represented approximately in the decimal-binary conversion should exactly cancel the effect of the similar approximation used in the binary-decimal conversion.

Extended Decimal-Binary Conversion. Suppose a decimal number $10^m d$ is assumed to be subject to some "uncertainty," expressed in integral units by $\pm u$; thus the number might be represented as $10^m(d \pm u)$. A way of taking this into account in decimal-binary conversion is to define the corresponding binary number as one which converts to decimal with an effective error amplification factor of the order of magnitude of u. Let t be defined as in the preceding section, so that $1 < w < 2$, with $w = 10^{-m}2^{-t}$, and let t' be defined in such a way that $u \approx w'$, with $w' = 10^{-m}2^{-t'}$; adjustment of the binary result to exponent $p - t'$ then gives approximately the effect desired. For example, one might define $t' = t - k$, where k is specified by $2^k \leqq u < 2^{k+1}$; this would be consistent with the basic decimal-binary conversion, for then $t' = t$ for $1 \leqq u < 2$. One further modification of this rule seems desirable, however. By the foregoing rule, $w' = 2^k w$ may lie anywhere in the range $2^k \leqq w' < 2^{k+1}$, and, hence, the ratio u/w' is only determined to within a factor 4, as $\frac{1}{2} < u/w' < 2$. This range can be narrowed by the adoption of a rule of the form

$$t' = t - k - \lambda, \qquad \lambda = \begin{cases} -1, & \frac{1}{2} < u/2^k w \leqq \theta, \\ 0, & \theta < u/2^k w < 2\theta, \\ 1, & 2\theta \leqq u/2^k w < 2, \end{cases}$$

where θ is some number $\frac{1}{2} < \theta < 1$. Although a standard choice for θ would be $\theta = \sqrt{2}/2$, giving $\sqrt{2}/2 \leqq u/w' \leqq \sqrt{2}$ for $w' = 2^{k+\lambda}w$, the choice of the nearby value $\theta = \frac{2}{3}$ has a practical advantage as regards reconvertibility of the result of binary-decimal conversion. To see this, note that, whenever u is the integer value obtained by rounding a value $w = 10^{-m}2^{-t}$ in the range $1 < w < 10$, then $\frac{2}{3} < u/w < \frac{4}{3}$; hence, confining u/w' to this range means that binary-decimal-binary conversion using the rounded w from the first conversion as the u for the second always gives $w' = w$, so the original binary number is obtained again in the reconversion.

The value k is obtained from the exponent of the normalized form of u, which is $k + 1$; thus

$$u \sim (k + 1, 2^{-k-1}u).$$

Furthermore, one has

$$2^{-t+k}d \sim (p - t + k, 2^{-p}d)$$
$$2^{-t+k}10^{-m} \sim (k + 1, 2^{-t-1}10^{-m}).$$

The latter form can be obtained by substituting the exponent of normalized u for the exponent of normalized 10^{-m}; then, since $2^{-t+k}10^{-m} = 2^k w$, the condition on the ratio $u/2^k w$ can be determined by a division and a comparison (if the division is carried out floating point, the condition can be evaluated by a test on the exponent and on the magnitude of the coefficient of the result). The exponent $p - t + k$ can also easily be obtained by subtracting the exponent of normalized 10^{-m} and adding the exponent of normalized u to the original exponent of d, since $p -$

TABLE 1

Sample Decimal-Binary-Decimal Conversions

The first column gives decimal numbers, with power-of-10 exponent and uncertainty u in square brackets. The second column gives, in sexadecimal form, the binary numbers obtained by decimal-binary conversion (the 2-digit exponent is represented excess-128, the 10-digit coefficient is represented true-complement). The third column gives the decimal numbers obtained in turn by binary-decimal conversion, with rounded amplification factor w in square brackets (the representation of w is actually sexadecimal, which is the same as decimal except in the occurrence of "A" for 10).

Decimal			Binary		Decimal		
E+28	0.3332	[1]	F7	0000000AC4	E+28	0.3332	[1]
E+28	0.3332	[2]	F8	0000000562	E+28	0.3332	[2]
E+28	0.3332	[3]	F8	0000000562	E+28	0.3332	[2]
E+28	0.3332	[4]	F9	00000002B1	E+28	0.3332	[5]
E+28	0.3332	[6]	F9	00000002B1	E+28	0.3332	[5]
E+28	0.3332	[7]	FA	0000000159	E+28	0.3337	[A]
E+28	0.3332	[12]	FA	0000000159	E+28	0.3337	[A]
E+28	0.3332	[13]	FB	00000000AC	E+28	0.333	[2]
E+28	0.3332	[20]	FB	00000000AC	E+28	0.333	[2]
E−13	−0.13989018219	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018220	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018221	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018222	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018223	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018224	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018225	[6]	5A	FF81FF8527	E−13	−0.13989018222	[7]
E−13	−0.13989018226	[6]	5A	FF81FF8526	E−13	−0.13989018229	[7]
E−13	−0.13989018227	[6]	5A	FF81FF8526	E−13	−0.13989018229	[7]
E−13	−0.13989018228	[6]	5A	FF81FF8526	E−13	−0.13989018229	[7]
E+03	0.100	[12]	AB	0000000006	E+03	0.10	[2]
E−21	0.0	[5]	60	0000000000	E−21	0.0	[4]
E−13	−0.13989018219	[1]	57	FC0FFC293C	E−13	−0.139890182190	[8]
E+28	0.3352	[20]	FB	00000000AD	E+28	0.335	[2]
E+03	0.115	[14]	AB	0000000007	E+03	0.11	[2]
E+02	0.96	[4]	A9	0000000018	E+02	0.96	[4]

$(t + 1) + (k + 1) = p - t + k$; if this and the divisor exponent are modified by $\lambda = \pm 1$ (if necessary) before the final division, the result will be $10^{-m}d$ at exponent $p - t + k + \lambda$, which can be taken as the binary form corresponding to $10^{-m}(d \pm u)$.

Note that the means described here for insuring that numbers remain invariant under binary-decimal-binary conversion are not applicable to the case of decimal-binary-decimal conversion. This asymmetry is due to the assumption that measures of error such as w and u are carried along explicitly only with decimal numbers; with binary numbers these remain implicit, and, hence, more uncertain.

Experimental Routines. MANIAC III programs for both the binary-decimal and decimal-binary conversion schemes have been prepared. Results of experimental runs demonstrating the properties of the conversions are given in Table 1.

Institute for Computer Research
University of Chicago
Chicago, Illinois

1. R. L. ASHENHURST & N. METROPOLIS, "Unnormalized floating point arithmetic," *J. Assoc. Comput. Mach.*, v. 6, 1959, pp. 415–428. MR **21** #4568.

2. H. KANNER, "Number base conversion in a significant digit arithmetic," *J. Assoc. Comput. Mach.*, v. 12, 1965, pp. 242–246.