CONVERSION OF RADIX (DECIMAL/BINARY)

It is apparent that a general-purpose calculator should be capable of receiving and putting forth numerical data in standard decimal form. In the case of a machine operating in the binary system a means of decimal-to-binary as well as binary-to-decimal conversion must be provided. Let us consider various methods which might possibly be employed to attain the conversions.

To convert an n-column decimal number into its binary form it is possible to store the binary equivalents of the n powers of ten and perform the requisite number of binary additions dependent upon the decimal digits appearing in the respective columnar positions. A modification of this method requires the storage of 9n binary equivalents -- namely, the equivalents of $k \cdot 10^{p+m}$ for k=1, 2, 3, ..., 9 and m=1, 2, 3, ...,n. Here p is an integer dependent on the disposition of the decimal point.

Although a maximum of only n-1 additions completes the conversion as compared with a maximum of 9n-1 additions in the former scheme, the storage and selection requirements are correspondingly more severe.

Another means of attack involves a preliminary translation of the original decimal number into the binary-coded decimal system. Let the decimal columns be numbered 1, 2, ..., n reading left to right. The binary equivalent of Column 1 is multiplied by 1010 (the binary equivalent of ten) and to this product the binary equivalent of Column 2 is added. This sum is in turn multiplied by 1010, and the process continues until the binary equivalent of Column n is added to the last product. If the original decimal number was an integer, then this final result gives the required binary equivalent. However, if the decimal point lay at the left of Column 1, then one additional multiplication is required to complete the conversion, the multiplier being 10^{-n} expressed in the binary system. The following example shows the step by step procedure applied to the conversion of the fractional decimal 0.71875 to the binary scale of notation. Figures shown in parentheses following a binary number give the corresponding decimal equivalent.

1 1 1	(7)
x 1 0 1 0	(10)
1 0 0 0 1 1 0	(70)
+ 1	(1)
1 0 0 0 1 1 1	(71)
x 1 0 1 0	(10)
1 0 1 1 0 0 0 1 1 0	(710)
+ 1 0 0 0	(8)
1 0 1 1 0 0 1 1 1 0	(718)
x 1 0 1 0	(10)
1 1 1 0 0 0 0 0 0 1 1 0 0	(7,180)
+ 1 1 1	(7)
1 1 1 0 0 0 0 0 1 0 0 1 1	(7,187)
x 1 0 1 0	(10)
1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 1 0	(71,870)
+ 1 0 1	(5)
1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1 1	(71,875)
x 0.1 0 1 0 0 1 1 1 1 1 $\cdot 2^{-16}$	(10^{-5})
0.1 0 1 1 0 1 1 1 1 1 . . .	(~0.71875)

It is possible to effect the conversion without recourse to binary arithmetic processes. Consider an n-column decimal number with all digits to the right of the decimal point. If the digit in the first decimal column (from the left) is 5 or greater, then the first binary digit to the right of the binary point in the binary equivalent is 1; otherwise the binary digit is 0. The decimal number is then successively doubled, the magnitude of the digit appearing in the first decimal column being sensed following each doubling operation. Proceeding in this fashion, the binary equivalent is obtained according to the following rule: If $0 \leq p_k < 5$, then $S_{k+1} = 0$; if $5 \leq p_k \leq 9$, then $S_{k+1} = 1$. Here, p_k is the decimal digit appearing in the first column after the kth doubling has occurred, and S_{k+1} is the k+1st binary digit to the right of the binary point. This process applied to the conversion of 0.71875 is shown below. Obviously, carries progressing into the units column have no effect on the procedure. For this reason only the fractional part of the successive decimal numbers need be recorded.

Binary Digit	
1	0.71875
0	0.43750
1	0.87500
1	0.75000
1	0.50000
0	0.00000
.	
.	----
.	

The binary equivalent is thus 0.10111.

Integral decimal numbers may be converted to the corresponding binary integer by means of sucessive halving operations applied to the original number. The kth binary digit counting from right to left is obtained by sensing the digit in the first column to the right of the decimal point at the completion of the kth halving. The required binary digit is 0 or 1 dependent upon whether the first decimal column is 0 or 5, respectively. Following each halving operation the fractional part of the decimal number is discarded once the above-mentioned sensing has been effected. An example of the conversion is given below.

	Binary Digit
437	
218.5	1
109.0	0
54.5	1
27.0	0
13.5	1
6.5	1
3.0	0
1.5	1
0.5	1

Thus, the binary equivalent of the decimal number 437 is 110110101.

Conversion from the binary scale to the decimal scale of notation can be accomplished by processes which correspond to those discussed above for the inverse conversion. The decimal equivalents of the requisite powers of 2 may be stored. Successive decimal additions are performed for each 1 appearing in the original binary number.

Binary numbers with all digits to the right of the binary point may be converted through repeated multiplication by 1010, the binary equivalent of ten. The integral portions of the successive products are, in fact, the binary equivalents of the decimal digits (left to right) of the required decimal number. Only the fractional part of any given product is involved in the succeeding multiplication, the integral part being discarded. Thus, the conversion of the binary number 0.10111 proceeds as follows:

Decimal Digit	
	0.10111
	x1010
7	111.00110
	0.00110
	x1010
1	1.11100
	0.11100
	x1010
8	1000.11000
	0.11000
	x1010
7	111.10000
	0.10000
	x1010
5	101.00000

We obtain, then 0.71875 which is the required decimal equivalent.

If a fractional binary number is to be converted, a method involving solely decimal computation will be discussed. Suppose the given binary number to have n columns. The digit in the nth column (counting left to right) can be assumed to be 1 without loss of generality. This 1 is recorded decimally as 1.0 and halved to give 0.5. At this point the digit appearing in the n - 1st column of the binary number is introduced into the units column of the decimal value 0.5. The new result is halved and the n - 2nd binary digit inserted into the units column as before. When the nth halving operation is concluded, the conversion is completed and the decimal number thus obtained is the desired equivalent. It is clear that the binary digit insertions involve no actual addition since the units column is continually cleared at the completion of the successive halving processes. Conversion of the binary number 0.10111 using this method follows:

0.10111	1.00000
	1.50000
	1.75000
	0.87500
	1.43750
	0.71875

Thus 0.71875 is the required decimal equivalent.

Conversion of an integral binary number to integral decimal notation may be effected by a procedure involving successive doubling. Each 1 (moving left to right) of the binary number is recorded as a 5 in the first decimal column, following which a decimal doubling takes place. This process continues until the nth (for an n-column binary integer) doubling has been completed at which point the equivalent decimal integer is obtained. Applying this procedure to the binary number 110110101, we have:

0.5	1
1.5	1
3.0	0
6.5	1
13.5	1
27.0	0
54.5	1
109.0	0
218.5	1
437.	

Thus, 437 is the required decimal integer.