Unsigned Floating-Point Number Base Conversions

Unsigned floating point integers are represented by placing a decimal point (or binary point, hexadecimal point, etc) to the right of the digit representing the radix raised to the zero power. The integer portion is evaluated the same as any unsigned integer. A different algorithm is used to convert the fractional portion of the number.

Significant Digits.

It might be useful at this point to introduce the concept of significant digits, since when working with floating point fractions it is always necessary to know how many digits to compute. This is done by comparing the ratio of the logarithm (radix power) of the bases. For example:

**Binary/Octal:**

\[ \log_2(8) = 3 \]

Source radix = binary = 2, 
Destination Radix = octal = 8, Result = 3

*It takes three binary digits to represent an octal digit.*

**Binary/Hexadecimal:**

\[ \log_2(16) = 4 \]

Source radix = binary = 2, 
Destination Radix = hexadecimal = 16, Result = 4

*It takes four binary digits to represent an hex digit.*

**Binary/Decimal:**

\[ \log_2(10) = 3.322 \]

Source radix = binary = 2, 
Destination Radix = decimal = 10, Result = 3.322

*It takes 3.322 binary digits to represent a decimal digit.*

Converting Unsigned Floating-Point Binary Numbers to Decimal

As in the preceding sub-topic, when converting between octal or hex and decimal, use binary as an intermediate step. The conversion from an unsigned floating point binary number to octal or hex, or from an unsigned floating point octal or hex to binary is a straight-forward process and will be shown below. The concept of conversions between unsigned floating point binary and decimal values is that same as it was for unsigned integer conversions. An example will make the process clear.
Convert Unsigned Floating-Point Binary 1011101.10111011₂ to Decimal

Solution:

The integer portion is evaluated as an unsigned integer:

\[
\begin{align*}
1011101₂ & \quad \text{Original Integer} \\
\text{decimal} &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
\text{decimal} &= 128 + 0 + 32 + 16 + 8 + 4 + 0 + 1 = 189 \\
189₁₀ & \quad \text{Decimal Equivalent}
\end{align*}
\]

The fractional portion is evaluated by radix expansion, as was done with the unsigned integer, though the powers of the radix are negative. Just as the decimal numbers to the right of a decimal point are negative powers of ten, so are they in any other number system.

For example: \(0.125 = 1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3} = 0.1 + 0.02 + 0.005\)

The radix expansion of the fraction portion of the binary number above results in the following:

\[
\begin{align*}
\text{decimal} &= 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 0 \times 2^{-6} + 1 \times 2^{-7} + 1 \times 2^{-8} \\
\text{decimal} &= 0.5 + 0.125 + 0.0625 + 0.03125 + 0.0078125 + 0.00390625 = 0.731₁₀
\end{align*}
\]

So, the final result is: \(1011101.10111011₂ = 189.731₁₀\)

Convert Unsigned Floating-Point Decimal 52.475₁₀ to Binary

Solution:

The integer portion is evaluated as an unsigned integer:

\[
\begin{align*}
52₁₀ & \quad \text{Original decimal integer}
\end{align*}
\]
The fractional portion of the decimal value is evaluated by repeated multiplication of the decimal fraction by the target radix (2), keeping the integer portion as the next binary significant digit. The fractional portion is kept for the next multiplication. Therefore, the integer portion of the result is not used in the multiplication. The procedure continues until either the overall result of the multiplication is zero, or enough significant digits have been obtained.

\[
\begin{align*}
\frac{52}{2} & = 26 \text{ remainder } = 0 \\
\frac{26}{2} & = 13 \text{ remainder } = 0 \\
\frac{13}{2} & = 6 \text{ remainder } = 1 \\
\frac{6}{2} & = 3 \text{ remainder } = 0 \\
\frac{3}{2} & = 1 \text{ remainder } = 1 \\
\frac{1}{2} & = 0 \text{ remainder } = 1 \\
\end{align*}
\]

110100₂  Binary equivalent.
The final result is the sum of the two calculations:

110100

+ .0111110011

= 110100.0111110011

Convert Floating-Point Binary to Hexadecimal or Octal or Convert Floating-Point Hexadecimal or Octal to Binary

Solution:

The conversion method is identical to that of an unsigned integer as previously discussed. However, note that zero's may have to be added to the least significant end and/or the most significant end of the original number to simplify the grouping of binary digits. For example, consider the conversion of a binary floating-point number to hexadecimal:

110100.0111110011  Original binary number

0011 0100.0111 1100 1100  Hexadecimal Grouping

3 4 . 7 C C  Hexadecimal Solution
To convert back to binary, the method utilizes the above example by reversing the process, from the last line to the first. Note that the extraneous binary zeros are removed.