Binary Point Numbers

Up to now, our work with number systems has dealt only with integers – in fact we haven't even dealt with negative numbers yet! Since we know that all types of numbers must be able to be handled by computers, there must still be a few things left for us to discover! Just as in decimal, we can have ***floating point*** numbers which are real numbers, in base two, we can also have an equivalent situation, and refer to these as binary point numbers.

Converting from a binary point number to its decimal equivalent is simply a matter of using the same process as we used before, but moving to the right of the binary point involves using negative exponents, just like in decimal.

Example: Convert from 1101.101_2 to decimal

$$\begin{aligned} 1101.101_2 &= 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-3} \\ &= 8 + 4 + 1 + 0.5 + 0.125 \\ &= 13.625 \end{aligned}$$

Converting in the opposite direction (from decimal point to binary point) could using a method like we first used with integers – simple inspection. However, this gets very difficult very quickly, and an algorithm becomes essential. Would you have been able to figure out what 0.625 was as a sum of fractions of powers of 2?

The Algorithm

Since we divided by 2 repeatedly to convert decimal integers to binary, perhaps you might expect the reverse to be true to convert decimal point numbers to binary – and it is!

Examples: Convert to binary

a) 5.75 b) 13.625 c) 7.7

2 x 0.75 = **1**.50
2 x 0.50 = **1**.00
2 x 0.0 = STOP

Read DOWN to get answer

$5.75 = 101.11_2$

Exercises

1. Convert the following to their decimal equivalents.
   a) 10.01_2 b) 101.101_2 c) 1010.1001_2 d) 101.0011_2
2. Convert to binary
   a) 8.375 b) 13.5625 c) 9.875 d) 8.1875
3. Convert to binary
   a) 1.1 b) 2.2 c) 3.3 d) 4.4