Fixed Point Binary Numbers

So far, you have only looked at ways of representing binary integers. To understand how binary fractions work, first consider how decimal fractions work.

1000	100	10	Units		1/10	1/100	1/100
0	5	4	9	.	3	6	7

In binary, the column headings are powers of 2 rather than 10.

8	4	2	Units		1/2	1/4	1/8	1/16
1	0	1	1	.	1	0	1	1

This would store the number 8 + 2 + 1 + 0.5 +0.125 + 0.0625 = 11.6875

When storing a fraction in binary digits the binary point is not stored. We need to assume a certain number of bits before the binary point and a number after the binary point. This representation of binary numbers is called **fixed point binary**.

Assuming 8 bits either side of the binary point and we get the following headings,

128	64	32	16	8	4	2	1		0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125	0.00390625
128	64	32	16	8	4	2	1		1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------
0	0	1	0	1	0	1	1		0	1	0	0	1	0	1	1

In the above example the number stored would be :

32 + 8 + 2 + 1 + 0.25 + 0.03125 + 0.0078125 + 0.00390625 = 43.2869375

Unless a number is an exact power of 2 it is impossible to store it exactly using this method. Errors in conversion of decimal numbers arise unless a large number of bits are allocated to store them.

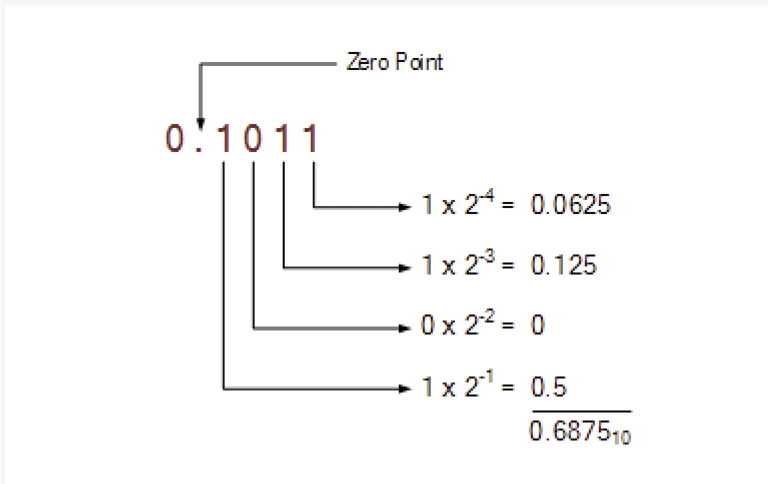
You can convert any decimal fraction to a binary fraction. Multiply the fractional part of the number by 2. Take the integer part of the result (1 or 0) as the first bit. Repeat this process with the result until you run out of patience. For example, to convert 0.3568 into fixed point binary with 8 bits to the right of the binary point,

0.3568	x	2	=	0.7136	:**0**
0.7136	x	2	=	1.4272	:**1**
0.4272	x	2	=	0.8544	:**0**
0.8544	x	2	=	1.7088	:**1**
0.7088	x	2	=	1.4176	:**1**
0.4176	x	2	=	0.8352	:**0**
0.8352	x	2	=	1.6704	:**1**

0.6704 x 2 = 1.3408 :**1**

0.3568 is .01011011

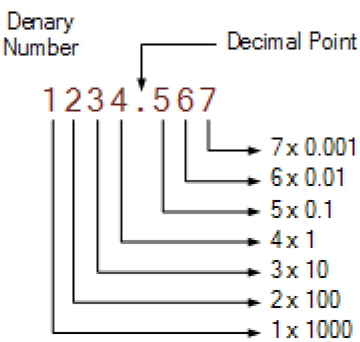
When we convert the binary result back to denary, we get 0.35546875. This isn't too far away - with 16 bits we could get, 0.356796264648437. The precision increases the more bits we use.



Binary Fractions

Binary Fractions use the same weighting principle as decimal numbers except that each binary digit uses the base-2 numbering system

A Typical Fractional Number



Here in this decimal (or denary) number example, the digit immediately to the right of the decimal point (number 5) is worth one tenth (1/10 or 0.1) of the digit immediately to the left of the decimal point (number 4) which as a multiplication value of one (1).

Binary Fractions

The binary numbering system is a base-2 numbering system which contains only two digits, a “0” or a “1”. Thus each digit of a binary number can take the “0” or the “1” value with the position of the 0 or 1 indicating its value or weighting. But we can also have binary weighting for values of less than 1 producing what are called unsigned fractional binary numbers.

Similar to decimal fractions, binary numbers can also be represented as unsigned fractional numbers by placing the binary digits to the right of the decimal point or in this case, binary point. Thus all the fractional digits to the right of the binary point have respective weightings which are negative powers of two, creating a binary fraction. In other words, the powers of 2 are negative.

So for the fractional binary numbers to the right of the binary point, the weight of each digit becomes more negative giving: 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, and so on as shown.

Binary Fractions

$$2^0 = 1$$

$$2^{-1} = \frac{1}{2^1} = \frac{1}{2} = 0.5$$

$$2^{-2} = \frac{1}{2^2} = \frac{1}{4} = 0.25$$

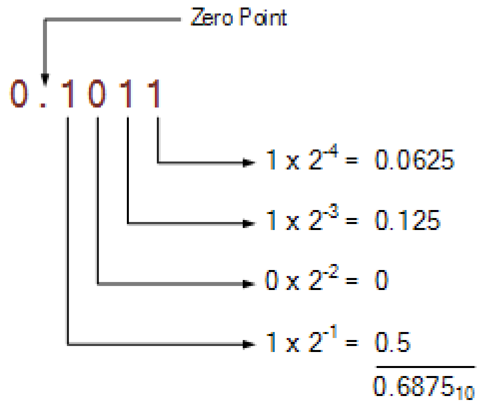
$$2^{-3} = \frac{1}{2^3} = \frac{1}{8} = 0.125$$

$$2^{-4} = \frac{1}{2^4} = \frac{1}{16} = 0.0625$$

$$2^{-5} = \frac{1}{2^5} = \frac{1}{32} = 0.03125$$

etc, etc.

Thus if we take the binary fraction of 0.1011_2 then the positional weights for each of the digits is taken into account giving its decimal equivalent of:



For this example, the decimal fraction conversion of the binary number 0.1011_2 is 0.6875_{10}.

Binary Fractions Example No1

Now lets suppose we have the following binary number of: 1101.0111_2, what will be its decimal number equivalent.

$1101.0111 = (1\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0) + (0\times2^{-1}) + (1\times2^{-2}) + (1\times2^{-3}) + (1\times2^{-4})$

$= 8 + 4 + 0 + 1 + 0 + 1/4 + 1/8 + 1/16$

$= 8 + 4 + 0 + 1 + 0 + 0.25 + 0.125 + 0.0625 = 13.4375_{10}$

Hence the decimal equivalent number of 1101.0111_2 is given as: 13.4375_{10}

So we can see that fractional binary numbers, that is binary numbers that have a weighting of less than 1 (2^0), can be converted into their decimal number equivalent by successively dividing the binary weighting factor by the value of two for each decrease in the power of 2, remembering also that 2^0 is equal to 1, and not zero.

Other Binary Fraction Examples

$0.11 = (1\times2^{-1}) + (1\times2^{-2}) = 0.5 + 0.25 = 0.75_{10}$

$11.001 = (1\times2^1) + (1\times2^0) + (1\times2^{-3}) = 2 + 1 + 0.125 = 3.125_{10}$

$1011.111 = (1\times2^3) + (1\times2^1) + (1\times2^0) (1\times2^{-1}) + (1\times2^{-2}) + (1\times2^{-3})$

$= 8 + 2 + 1 + 0.5 + 0.25 + 0.125 = 11.875_{10}$