Floating Point Binary

L.O.: To understand how Floating Point representation works.

Floating point numbers are made of two parts – a mantissa and an exponent:

|  |  |
| --- | --- |
| Mantissa | Exponent |
| 2.3423958 | 105 |

Floating point is often uses 12 bits

0000 1100 0011

|  |  |  |
| --- | --- | --- |
| **+/-** | **Mantissa** | **Exponent** |
| 0 | 000 1100 | 0011 |
| 0 | 12 | 23 |

*Convert these to denary:*

1. *0101 1101 0001*

= (64 + 16 + 8 + 4 + 1) x 21 = 93 x 21 = **186**

1. *0101 1101 0011*

= (64 + 16 + 8 + 4 + 2 + 1) x 2(2 + 1) = 93 x 23 = **744**

1. *0000 0111 0100*

= (4 + 2 + 1) x 24 = 7 x 24 = **112**

* With Floating Point, a much wider range of numbers can be produced with the same number of bits as the fixed point system
* Consequently, floating point lends itself to applications where a wide range of values may need to be represented.

The exponent can use Two’s Compliment for negative exponents.

*Convert these to denary:*

1. *0101 1101 1111*

= (64 + 16 + 8 + 4 + 1) x 2(-8 + 4 + 2 + 1) = 93 x 2-1 = 46.5

1. *0101 1101 1101*

= (64 + 16 + 8 + 4 + 1) x 2(-8 + 4 + 1) = 93 x 2-3 = 11.625

1. *0000 0111 1010*

= (4 + 2 + 1) x 2(-8 + 4) = 7 x 2-4 = 0.4375