

$$-12.5 \equiv -1.25 \times 10^1 \quad \text{or} \quad \begin{matrix} -25 \times 2^{-1} \\ \equiv -11001_2 \times 2^{-1} \\ \equiv -1.1001_2 \times 2^3 \end{matrix} \quad \text{when normalised.}$$

$$0.1 \equiv 1.0 \times 10^{-1} \quad \text{or} \quad \frac{1}{10} \equiv \frac{1}{1010_2} \equiv \frac{0.1_2}{101_2}$$

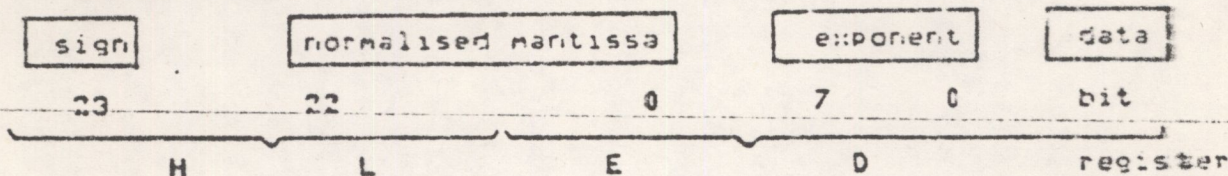
so now we need to do some binary long division...

$$\begin{array}{r} 0.0001100 \\ 101 \overline{) 0.1000000000000000} \\ \underline{101} \\ 110 \\ \underline{101} \\ 1000 \\ \underline{101} \end{array}$$

at this point we see that the fraction recurs

$$= \frac{0.1_2}{101_2} = 0.0001100_2 = 1.1001100_2 \times 2^{-4} \quad \text{answer.}$$

So how do we use the above results to represent these numbers in the computer? Well, firstly we reserve 4 bytes of storage for each real in the following format:



- sign: the sign of the mantissa; 1 = negative, 0 = positive.
- normalised mantissa: the mantissa normalised to the form 1.xxxxxx with the top bit (bit 22) always 1 except when representing zero (HL=0, DE=0).
- exponent: the exponent in binary 2's complement form.

Thus:

2	≡	0	10000000	00010000	00000000	00000001	(E40, E00, E00, E01)
1	≡	0	10000000	00000000	00000000	00000000	(E40, E00, E00, E00)
-12.5	≡	1	1100100	00000000	00000000	00000011	(E24, E00, E00, E03)
0.1	≡	0	1100110	01100110	01100110	11111100	(E66, E66, E66, EFC)