Binary Division

321 ÷ 7

D M S B
(Div, Mul, Sub, Bring down)

```
      4 5
7 | 3 2 1
    3 2
   -2 8
    ----
      4 1
```

321 ÷ 7 = 45 r 6.

110101 ÷ 101 = 001010 r 100

```
         0 0 1 0 1 0 1
101 | 1 1 0 1 1 0 1
      1 1 0
     -1 0 1
     ------
        1 1 1
       -1 0 1
       ------
          1 0 0 1
         -1 0 1
         -------
            1 0 0
```

Only on unsigned numbers. resign if necessary.

$10110_1 \div 11 = 01111$

$$
\begin{array}{r}
01111 \\
11 \overline{)\,10110_1} \\
\underline{-\,11} \\
101 \\
\underline{-\,11} \\
100 \\
\underline{-\,11} \\
11 \\
\underline{-\,11} \\
0
\end{array}
$$

$45 \div 3 = 15$ ✓

Fractional components

$11.011 \div 11$

$$
\begin{array}{r}
01.001 \\
11 \overline{)\,11.011} \\
\underline{-\,11} \\
011 \\
\underline{-\,11} \\
011 \\
\underline{-\,11} \\
01
\end{array}
$$

$3.375 \div 3 = 1.125$

$1 \div 11$

$$
\begin{array}{r}
0.0101 \\
11 \overline{)\,1.000000} \\
\underline{-\,11} \\
100 \\
\underline{-\,11} \\
10 \\
\underline{-\,11} \\
10
\end{array}
$$

etc.....

Floating point numbers

$$y = M \times b^E$$

M is the mantissa. in Q0.m format ALWAYS.

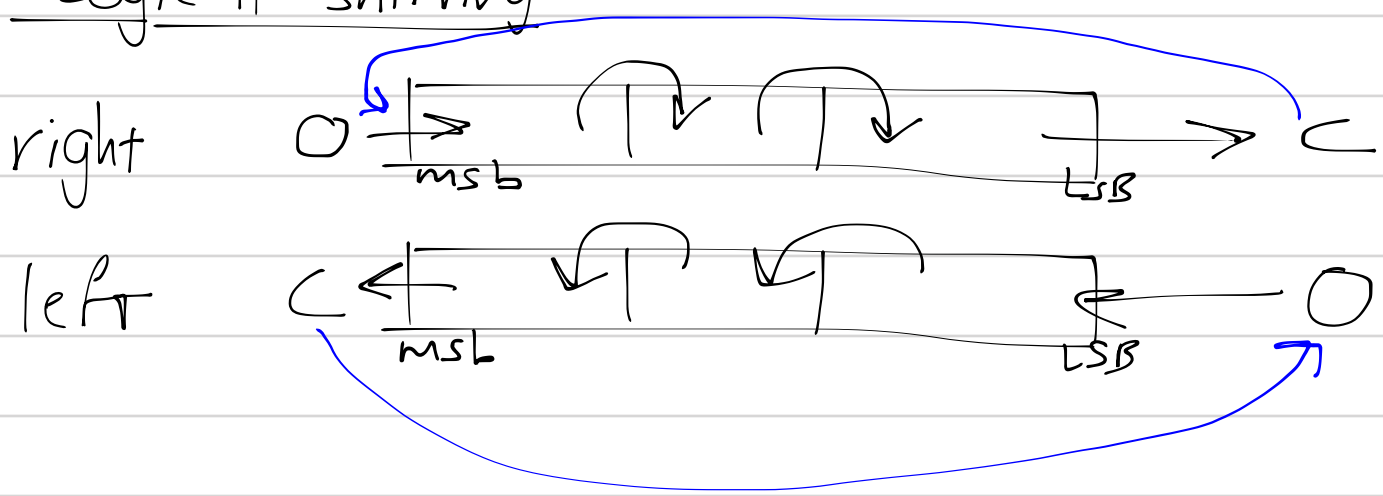
b is the base.

E is the exponent in Qn.0 format ALWAYS

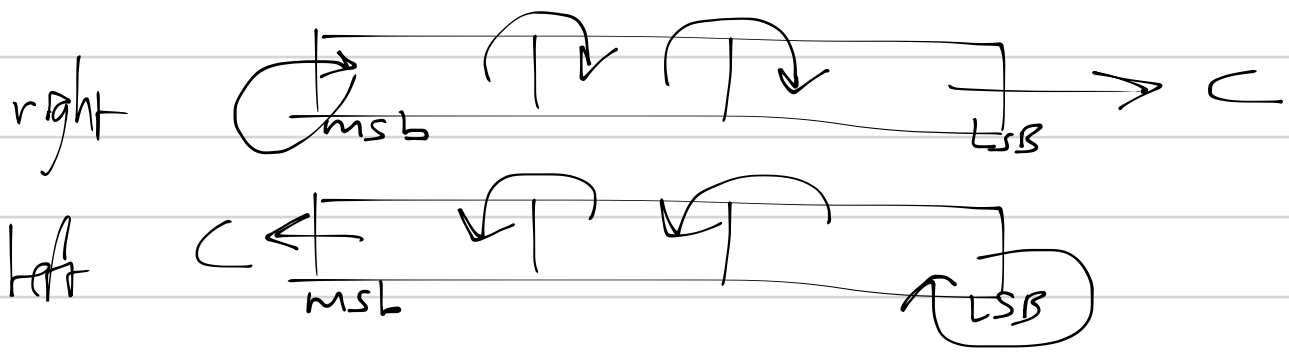
y has format Qmen

Q2e3

Logical shifting



Arithmetic shifting



10110 ras 1 = 11011 [0]

$-16+4+2 = -10$ $\quad -16+8+2+1 = -5$

11011.0 ras 1 = 11101.1

$= -2.5$

01101 lls 1 = 011010 $\approx \times 2$.

Normalisation

Normalisation is the method by which precision is maximised.

$0.11010_e 0000$ → $-5+0 \Rightarrow$ LSB order
$0.01101_e 0001$ → $-5+1 \Rightarrow$ LSB order

$(0+0.5+0.25+0.0625) \times 2^0 = 0.8125$

$(0+0.25+0.125+0.03125) \times 2^1 = 0.8125$

$(35.82 \times 10^3) \div 100 = (35.82 \times 10^{3-2})$.

$1.1101_e 0111$
$= 1.1010_e 0110$ (left logical shift 1)
$= 1.0100_e 0101$ (left logical shift 2)
$= 0.1000_e 0100$? overflow. ✗

To normalise: logical left shift until $MSB = \overline{MSB_{-1}}$
for every shift subtract 1 from exponent.

$0.0010\,01_e 0111$
$0.1001\,00_e 0101$ 2 left shifts.

$2^{-5} \times 2^7 = 2^2$
$2^{-5} \times 2^5 = 2^0$

Justification

Justification is the process by which exponents are matched
NB: Always done on normalised numbers.
Right arithmetic shift until smaller exponent matches bigger one.

Justify $0.110 e 001$ on $1.011 e 011$

$= 0.011 e 010$

$= 0.001 e 011$

NB information lost
Preserve in guard bit

Floating point addition

$0.011101 e 0110 + 1.1101 e 0010$

1. Normalise (both numbers)
2. Justify on largest exponent
3. Do addition
4. Normalise answer.

$= 0.11010 e 0101 + 1.01000 e 0000$ (normalised)

$= 0.11010 e 0101 + 1.11111 e 0101$

$$\begin{array}{r} 0\ 1\ 1\ 0\ 1\ 0 \\ +\ 1\ 1\ 1\ 1\ 1\ 1 \\ \hline [1]\ 0\ 1\ 1\ 0\ 0\ 1 \end{array}$$

normal signed addition.
carry, no overflow

$= 0.11001 e 0101$ (normalised) →