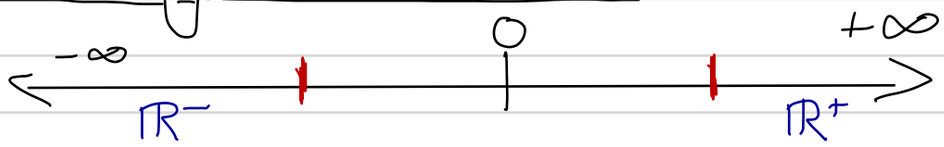
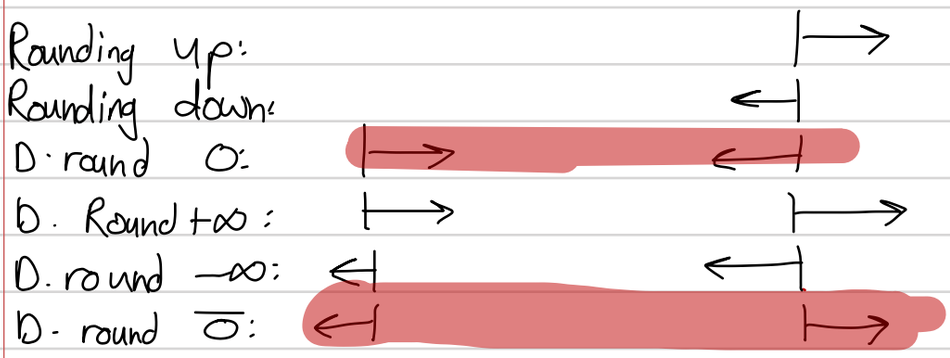
Rounding and guard bits





In Q0.m subtraction and addition adding a guard bit after the LSB (Q0.(m+1)) improves precision.

Add guard bit <u>before</u> normalisation step. Round to Q0.m format towards zero.

$0.101_e001 - 0.011_e000$

<u>Without</u> guard bit:

$$\begin{array}{r} 0.101_e001 \\ -\ 0.001_e001 \\ \hline 0.100_e001 \end{array}$$

← information lost during justification

→ 1.00

With guard bit:

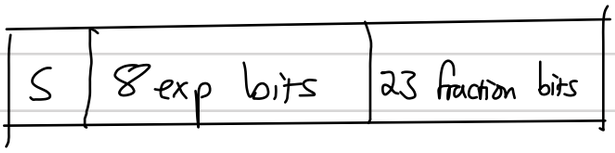
$$\begin{array}{r} 0.1010_e001 \\ -\ 0.0011_e001 \\ \hline 0.0111_e001 \end{array}$$

$= 0.011_e001$ → 0.75

$= 0.111_e000$ → 0.875 ✓

IEEE floating point format

First proposed in 1985: now present in <u>every</u> mobile device regardless of architecture.
<u>THE</u> format used to store data on fixed or removable media.



1 sign bit
8 exponent bits (Q7.0 signed)
23 fractional bits Q0.23 (unsigned)

Exponent has bias of -127
$E_{min} = 1 - 127 = -126$
$E_{max} = 254 - 127 = +127$
(offset of 127 subtracted from stored value)
Stored exponent of 0x00 and 0xFF are interpreted differently.

Exponent	Mantissa = 0	Mantissa ≠ 0	Equation
0x00	+0/-0	sub-normal	$(-1)^{sb} \times 2^{-126} \times 0.FB$
0x01 → 0xFE	Normalised	Normalised	$(-1)^{sb} \times 2^{E-127} \times 1.FB$
0xFF	+∞/-∞	NaN	Quiet signalling

$+1_{10} \Rightarrow$ 0 (sb) 01111111 0000…000

$\Rightarrow$ 3F80 0000

$-2 \Rightarrow$ 1000 0000
1100000 00…000

DEAD BEEF = 1101 1110 1010 1101
IEEE 1011 1110 1110 1111

SB = 1

Exp = 101 1101 = 189 − 127 = 62

1.01011011011110111011111 = 1.35789330 9

DEADBEEF = −6.259853398707798016 × 10^{18}

$y = (-1)^{SB} \times 2^{E-127} \times 1.FB$

DEADBEEF = $(-1)^{1} \times 2^{189-127} \times 1.35789330917584377$