Fixed Point Numbers

→ keeps the decimal point in fixed position

→ faster, simpler calculations, but less precise

→ no. are equally spaced.

Example

$(10.1)_2$, $-(10.1)_2$ etc. $(12.345)_{10}$ this point is fixed

fixed point representation : $x = \pm (d_1 d_2 \cdots d_{k-1} \cdot d_k \cdots d_n)_\beta$

where, $d_1 \cdots d_n \in \{0, 1 \dots \beta-1\}$

converting $(\overset{10-1}{10.1})_2$ to $(?)_{10}$

$= 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1}$

$= (2.5)_{10}$

Example → 64-bit integers

(signed) largest possible no. $= 2^{63} - 1$

$= 2^{n-1} - 1$

(MSB) 1 bit for sign → 0 (+ve), → 1 (-ve)

Range : -2^{63} to $2^{63}-1$

Floating Point Numbers

→ allows the decimal point to move

→ more precision, wider range of values, more complex

→ no. are not equally spaced

Both fixed point and ~~flat~~ floating points are how numbers are stored/represented in a computer.

Floating Point Representation

$$F = \underset{\text{sign}}{\pm}\ (\underbrace{0 \cdot d_1 d_2 \cdots\cdots d_n}_{\text{significand/fraction/mantissa}})_{\beta} \times \beta^{e}$$

exponent

base

where, $\beta, d_i, e \in \mathbb{Z}$ (set of integers)

$$0 \le d_i \le \beta - 1$$

$$e_{min} \le e \le e_{max}$$

Examples

$123 \cdot 45$
$= 12 \cdot 345 \times 10^1$
$= 1 \cdot 2345 \times 10^2$
$= 0 \cdot 12345 \times 10^3$

$1001 \cdot 11$
$= 0 \cdot 100111 \times 2^4$

exponent

base

Conventions

Convention 1 : $\pm (0 \cdot \boxed{d_1} d_2 \cdots d_m)_{\beta} \times \beta^{e}$

$d_1 = 1$ (always)

① Example : $\beta = 2,\ m = 3,\ e = e$

$m=3$ means, after decimal point, there will be 3 digits.

So, highest value $= (0 \cdot d_1 d_2 d_3)_2 \times 2^e$

$= (0 \cdot 1 d_2 d_3)_2 \times 2^e$ [by convention, $d_1 = 1$ always]

$= (0 \cdot 1\,\underline{1}\,\underline{1})_2 \times 2^e$ (Ans)

② Example: $\beta = 2,\ m = 3,\ e_{min} = -1,\ e_{max} = 2$

max. value $= +(0 \cdot 111)_2 \times 2^{2 \rightarrow max} = \boxed{\frac{7}{2}}$

min. value (non-negative) $= +(0 \cdot 100)_2 \times 2^{-1} = \boxed{\frac{1}{4}}$

min. value $= -$ (max. value) $= \boxed{-\frac{7}{2}}$

Calculation

$$(\overset{0}{0}\cdot\overset{-1}{1}\overset{-2}{1}\overset{-3}{1})_2 \times 2^2$$

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

$$= \left(\frac{7}{2}\right)_{10}$$

Convention : 2 $\Rightarrow$ $\boxed{\text{Normalized form.}}$

$$\pm\ (1\cdot d_1 d_2 \ldots d_m)_\beta \times \beta^e$$

Note:

⊞ if $e = (0, 2]$

$\Downarrow$

$1, 2$

Example: $\beta = 2$, $m = 3$, $e = [-1, 2]$

$\Downarrow$

$-1, 0, 1, 2$

$$\text{Highest value} = (\overset{0}{1}.\overset{-1}{1}\overset{-2}{1}\overset{-3}{1})_2 \times \cancel{2}\,2^2$$

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

$$= (7\cdot 5)_{10}$$

Convention : 3 $\Rightarrow$ $\boxed{\text{Denormalized form}}$

$$\pm\ (0\cdot 1\, d_1 d_2 \ldots\ldots d_m)_\beta \times \beta^e$$

Example: $\beta = 2$, $m = 3$, $e = [-1, 2]$

$$\text{Highest value} = (0\cdot 1 d_1 d_2 d_3)_2 \times 2^2$$

$$= (0\cdot 1111)_2 \times 2^2$$

$$= (3\cdot 75)_{10}$$

Total combinations

Convention: 1

$\beta = 2,\ m = 3,\ e = [-1, 2] = \{-1, 0, 1, 2\}$

emin   emax

$(0.1\,\underline{d_2}\,\underline{d_3})_2 \times 2^{-1}$

$$\left.\begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix}\right] \rightarrow \text{4 combinations. for } e = -1$$

same goes for $e = 0,\ e = 1,\ e = 2$

$\Downarrow \quad \Downarrow \quad \Downarrow$

$4 \quad 4 \quad 4$

total $= 4 + 4 + 4 + 4 = 16$ combinations / floating numbers that can be represented.

Convention: 2

$1.\,\underset{0/1}{\underline{d_1}}\ \underset{0/1}{\underline{d_2}}\ \underset{0/1}{\underline{d_3}} = 2^3 = 8$ combinations. for each exponent.

So, total $= 8 \times 4 = 32$

Convention: 3

$0.1\,\underline{d_1}\,\underline{d_2}\,\underline{d_3}$

total $= 8 \times 4 = 32$

Floating Point numbers are not equally spaced

$e = [-1, 1]$ $\beta = 2$, $m = 3$ → convention 1

$e = -1$

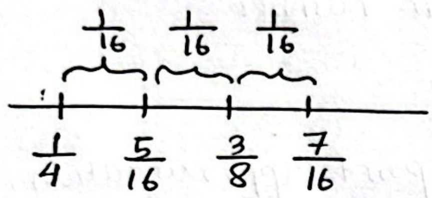
Smallest non-negative number combinations

1st smallest $= (0\cdot100) \times 2^{-1} = \frac{1}{4}$

2nd " $= (0\cdot101) \times 2^{-1} = \frac{5}{16}$

3rd " $= (0\cdot110) \times 2^{-1} = \frac{3}{8}$

4th " $= (0\cdot111) \times 2^{-1} = \frac{7}{16}$



☐ when exponent is constant, numbers are equally spaced.

$e = 0$

$(0\cdot100) \times 2^0 = \frac{1}{2}$] diff. $= \frac{1}{8}$

$(0\cdot101) \times 2^0 = \frac{5}{8}$] diff $= \frac{1}{8}$

$(0\cdot110) \times 2^0 = \frac{3}{4}$] diff $= \frac{1}{8}$

$(0\cdot111) \times 2^0 = \frac{7}{8}$

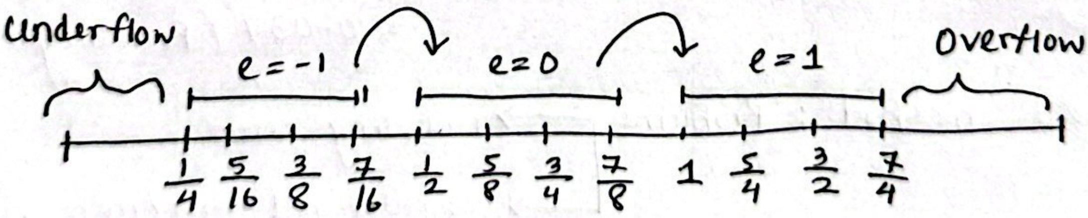
$e = 1$

$(0\cdot100) \times 2^1 = 1$] diff $= \frac{1}{4}$

$(0\cdot101) \times 2^1 = \frac{5}{4}$] diff $= \frac{1}{4}$

$(0\cdot110) \times 2^1 = \frac{3}{2}$] diff $= \frac{1}{4}$

$(0\cdot111) \times 2^1 = \frac{7}{4}$



☐ with change in e, the gap/interval between numbers changes.

For convention 1 & denormalised form, we don't have zero in our number system.

Others

$$\left.\begin{aligned} x + 2y &= 0 \\ 2x - \pi y &= 1 \end{aligned}\right\}$$ we can solve this simultaneous equation easily (direct method)

Problem: $\pi \rightarrow$ transcendental ~~π~~ number

= computer has finite memory

= hence, it uses a floating-point approximation, which incurs $\boxed{\text{rounding error.}}$

Example : $\pi = 3.14159\ldots$
$\Downarrow$
$\pi = 3.142$ } floating point approximation.

Example: 2.5763
2.58 } FP approximation

difference between the numbers $= 2.58 - 2.5763$
$= 0.037$ (Rounding error)

direct method : Problems → rounding error
→ computational time/memory

Example : $\sin(1.2)$ [Iterative method]
↳ adding extra terms to improve approximation.

$$\sin(1.2) = 1.2 - \frac{(1.2)^3}{3!} + \frac{(1.2)^5}{5!} - \frac{(1.2)^7}{7!} + \ldots \text{ (infinite series)}$$

Even if a computer could do real arithmetic, there would be an error due to stopping the iterative process at some finite point. This is called $\boxed{\text{truncation error}}$

$$\sin(1.2) = \underbrace{\boxed{1.2 - \frac{(1.2)^3}{3!} + \frac{(1.2)^5}{5!}}}_{\text{approximate}} \underbrace{- \frac{(1.2)^7}{7!} + \cdots}_{\cancel{\text{trans}}\ \text{truncation error.}}$$

- for limited resources, we are taking first 3 terms, eliminating the rest. The error caused by eliminating those terms are known as truncation error. //

IEEE standard for double precision (1985)

64 bit Architecture / Arithmetic

$\beta = 2$, 52 bits $\rightarrow$ fraction $\rightarrow$ precision

11 bits $\rightarrow$ exponent $\rightarrow$ range

1 bit $\rightarrow$ sign (0 means +ve, 1 means -ve)

normalised form $= (1.d_1 d_2 \ldots d_{52}) \times 2^e$

sign	exponent	fraction
1 bit	11 bits	52 bits

For exponent, 2^{11} combinations $= 2048$ combination.

possible values $= [0, 2047]$ $\quad \therefore 2047 = 2^{11} - 1$

$\Downarrow$

total 2048 values

$$\left.\begin{aligned} e_{min} &= 0 \\ e_{max} &= 2047 \end{aligned}\right\}$$ However, this excludes negative values of e. (ie. e^{-1} etc)

largest possible non-negative no. $= (1.11\cdots1)_2 \times 2^{2047}$

smallest possible non-negative no. $= (1.00\cdots0)_2 \times 2^0$

$= 1$ [not enough small, cannot express smaller values]

Therefore, we do $\boxed{\text{exponent biasing}}$

$2048/2 = 1024$

1024 values to left

1024 values to right.

$$\Rightarrow \underbrace{1023 \mid 0 \mid 1024}_{2048 \text{ numbers.}}$$

] exponent biasing (done to represent smaller number < 1)

new, $e_{min} = -1023$

$e_{max} = 1024$

$\boxed{2^{e-1023}}$

$0-1023 = -1023$

$2047-1023 = 1024$

Now,

$$(1.d_1d_2\cdots d_{52})_2 \times 2^{-1023}$$

$$= (0.1d_1d_2\cdots d_{52})_2 \times 2^{-1022}$$

$$(1.d_1d_2\cdots d_{52})_2 \times 2^{1024}$$

$$= (0.1d_1d_2\cdots d_{52})_2 \times 2^{1025}$$

Before, $e \in [0, 2047]$

After, $e \in [-1022, 1025]$

Now, after exponent biasing :

$$\text{highest no.} = (0.111111\ldots1)_2 \times 2^{1025} \approx \infty$$

$\boxed{\text{Important}}$

$$\text{lowest no.} = (0.100\ldots\ldots0)_2 \times 2^{-1022} \approx 0$$

In IEEE standard, 2 bits from exponent is used/reserved to store ∞ and 0.

Now,

$$\text{Highest possible num (except infinity)} = (0.11\ldots1)_2 \times 2^{1024} \approx 1.798 \times 10^{308}$$

$$\text{Lowest possible num (except 0)} = (0.10\ldots0)_2 \times 2^{-1021} \approx 2.225 \times 10^{-308}$$

For exponent,

$$2^{1025} \rightarrow \pm\infty$$

$$2^{-1022} \rightarrow \pm 0$$