+ GP

3.1. radix $16 = 2^4$: IBM's f.p. format. $X = (-1)^S * 16^{X_E - 64} * (0.X_M^*)$

normalization: at most 3 leading 0.

++, -- X_E => moving the binary point over 4 bits.

0.000 1 ---

0.0000 1

$X_E -= 1$

IBM 32 / 64

1: Sign

7: Exp.

24 / 56 fractal part of mantissa X_M

Shifter 1: accelerates mantisses allignment.

if $k > 0$ => RightShift M2 by $4 \cdot k$ bits.

if $k < 0$ => RightShift M1 by $-4k$ bits.

Adder 2: 56 bits. ML-CLA.

zero digit checker: accelerates normalization of result

- determines value of l = number of groups of 4 leading 0s

0000 0000 1 ---

1 group of 4 leading 0s | 1 group of 4 leading 0s

$e = 2$

-> Shifter 2: LeftShift result by $4 \cdot l$ bits

-> Adder 3: substract l from result's exponent.

Adder 3: choses the result's exponent by inspecting the sign of k

3.2. Rounding: convert higher precision representation into a lower precision representation.

- for storage / data transmission

IEEE 754 4 rounding modes

to 0

towards $-\infty$

towards $+\infty$

to nearest even

Let $X = x_{n-1} x_{n-2} \cdots x_1 x_0 . x_{-1} x_{-2} \cdots x_{-m}$

n bits of integer part — m bits of fractional part

for brevity, consider X^* = rounded value of X.

X^* - integer.

⇒ eliminate the fractional bits

! For IEEE754 f.p. nos, the rounding does not eliminate the fractional bits.

32 bit IEEE

X_S^*

S	x_E	fractional part of x_S^*
<1>	<8>	<23>

$X = (-1)^S * 2^{x_E - 127} * (1.x_S^*)$

what IEEE754 f.p. rounding does:

if X_S^* has more than 23 bits

⇒ in order to be able to store the value ⇒ only preserve 23 bit

$X * Y$

$x_S^* * x_Y$

24 bit * 24 bit

⇒ 48 bit res

of 42 bits of fractional part

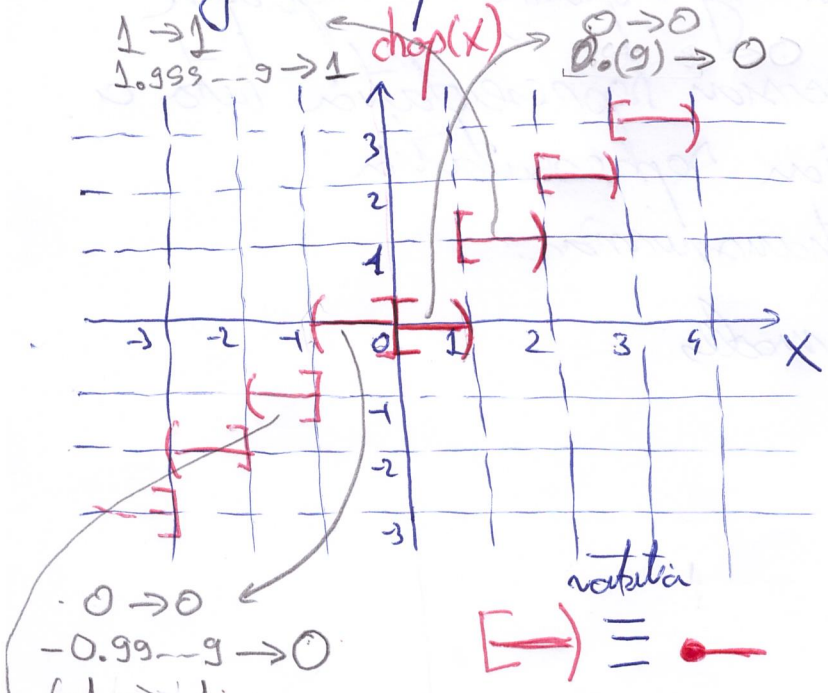
one can only store 23

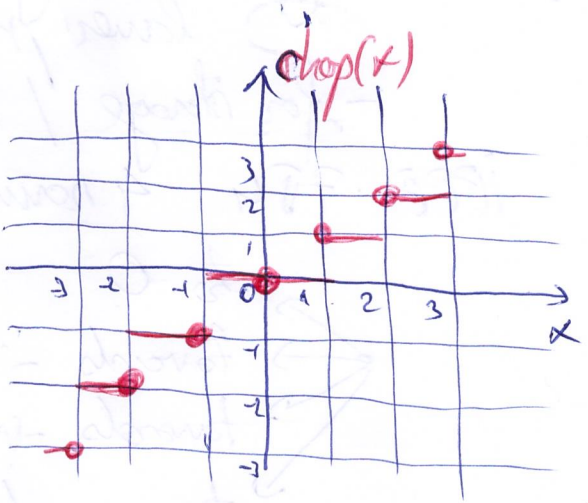
- discard all fr. bits

A) to 0 (inwards rounding)

X^*: largest integer, for which $|X^*| \leq |X|$

if X is represented in SM ⇒ round to 0 ≡ truncate to integer





(3) towards $-\infty$ (downwards rounding)

- X^* is the largest integer for which $X^* \leq X$

! - for positive values rounding to $0 \equiv$ rounding to $-\infty$

- if X is represented in C2 $\Rightarrow$ downward rounding $\equiv$ eliminating all fractal bits.

$+4_{C2} = 0100_{C2}$ (.5 .25 / 2^{-1} 2^{-2})

$X = +4.25_{C2} = 0100.\overline{01}_{C2}$ $\quad 4.25 \to 4$

$X^* = 0100._{C2} = 4$

(downward rounding)

$-4_{C2} = 1100_{C2}$

$-4.25_{C2} = 1011.\overline{11}_{C2}$

$X^* = 1011._{C2} = -5$

(downwards rounding)

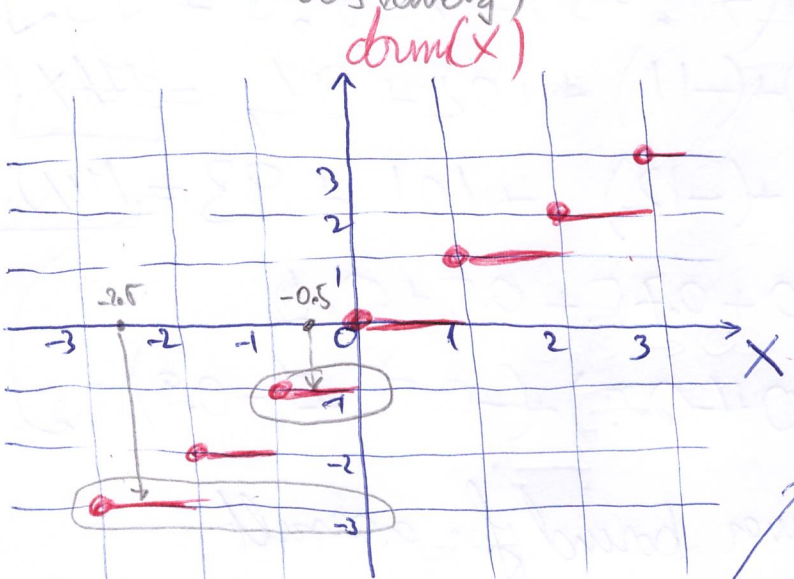
$+4.25_{C2} = 0100.01$

$-4.25_{C2} = 1011.11$

$1011._{C2}$ (2^3)

$= -2^3 + 0.011_{C2}$

$= -8 + 3 = -5$



if X is represented in SM

if $X \geq 0$: discard (eliminate) all fractal bits.

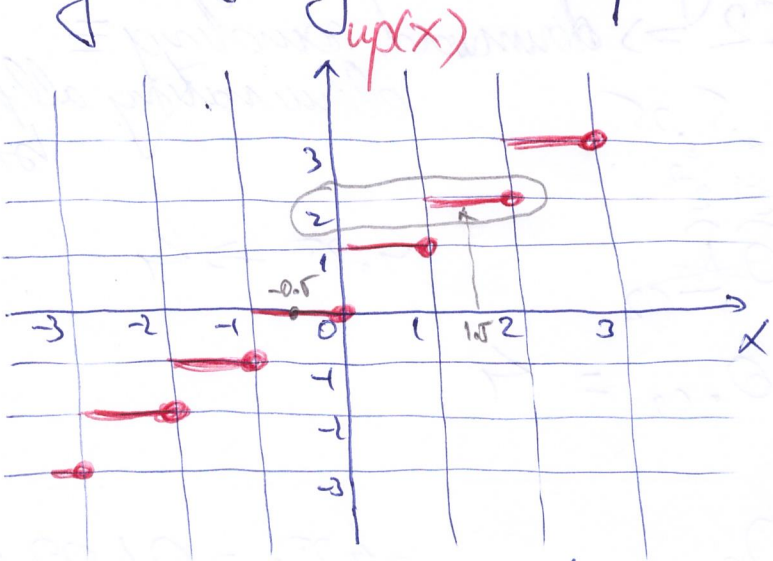
if $X < 0$:

$$X^* = \begin{cases} \overrightarrow{x_{m-1}x_{m-2}\ldots x_1x_0}., & \text{if } .x_{-1}x_{-2}\ldots x_{-m} = 0 \\ x_{m-1}x_{m-2}\ldots x_1x_0. - 1, & \text{if } .x_{-1}x_{-2}\ldots x_{-m} \neq 0 \end{cases}$$

© towards $+\infty$ (upwards rounding)

X^* is the smallest integer for which $X^* \geq X$

if X is negative ⇒ upwards rounding ≡ inwards rounding



error $\varepsilon = X^* - X$

upwards & downwards rounding:

- all errors are in the same direction ↳ either positive, or negative

⇒ errors accumulate faster

$1.1 * 10^0$

$S = 1.1 + 2 + (-11.75) + 101.5 + 93.9$

upwards $2 + 2 + (-11) + 102 + 94 = \underline{MAX}$

downwards $1 + 2 + (-12) + 101 + 93 = \underline{MIN}$

$\varepsilon_{upwards} = 0.9 + 0 + 0.75 + 0.5 + 0.1$

$\varepsilon_{downwards} = (-0.1) + 0 + (-0.25) + (-0.5) + (-0.9)$

- provide upper / lower bound for a result.

$$S \leq MAX \quad \underline{acl}$$

$$S \geq MIN$$

- internal arithmetic

! exponent of the values influence the error !!!

$X_1 = 1.4 * 2^{-32} \rightarrow 2 * 2^{-32} \quad \varepsilon = 0.6 * 2^{-32}$

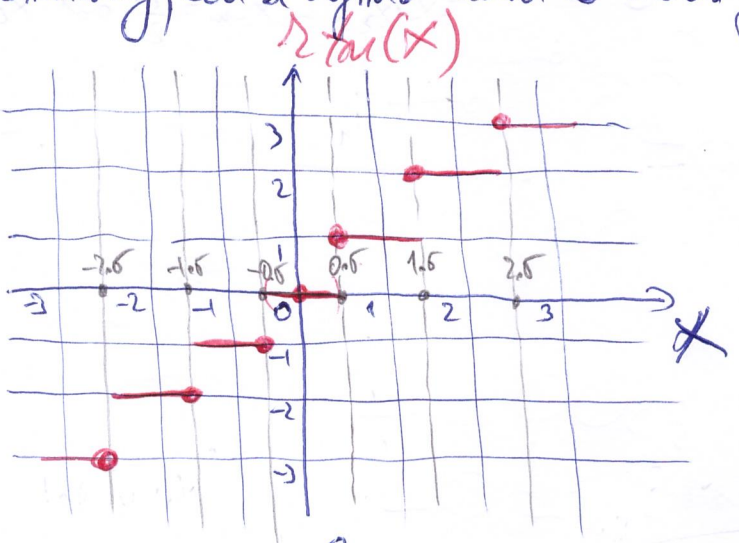
$X_2 = 1.4 * 2^{+32} \rightarrow 2 * 2^{+32} \quad \varepsilon = 0.6 * 2^{+32}$

(D) to nearest even
- derived from Round to nearest.

Round to nearest

Let X is positive $X^* = \begin{cases} x_{m-1}x_{m-2}\dots x_1x_0. , & \text{if } .x_1x_2\dots x_m < \frac{1}{2} \\ x_{m-1}x_{m-2}\dots x_1x_0. + 1, & \text{if } .x_1x_2\dots x_m \geq \frac{1}{2} \end{cases}$

Similarly, can be defined round to nearest for negatives



Error analysis
- for bonus $X_{-3} = X_{-4} = \dots = X_{-m} = 0$
- only X_{-1} and X_{-2} can be $\neq 0$

if $X > 0$

inputs X_{-1}	inputs X_{-2}	Outputs $X^* = rtn(X)$	Outputs $\varepsilon = X^* - X$
0	0	$x_{m-1}x_{m-2}\dots x_1x_0.$	0
0	1	$x_{m-1}x_{m-2}\dots x_1x_0.$	$-\frac{1}{4}$
1	0	$x_{m-1}x_{m-2}\dots x_1x_0. + 1$	$+\frac{1}{2}$
1	1	$x_{m-1}x_{m-2}\dots x_1x_0. + 1$	$+\frac{1}{4}$

$,00_{(2)} = .0_{(10)}$
$01_{(2)} = .25_{(10)}$
$10_{(2)} = .5_{(10)}$
$11_{(2)} = .75_{(10)}$

$+\frac{1}{2}$
$-\frac{1}{2}$
with equal probability

Consider all 4 cases to be equally probable

$$\varepsilon_{mean} = \frac{0 + (-\frac{1}{4}) + \frac{1}{2} + \frac{1}{4}}{4} = \frac{1}{8}$$

after 8 operations ε can become 1

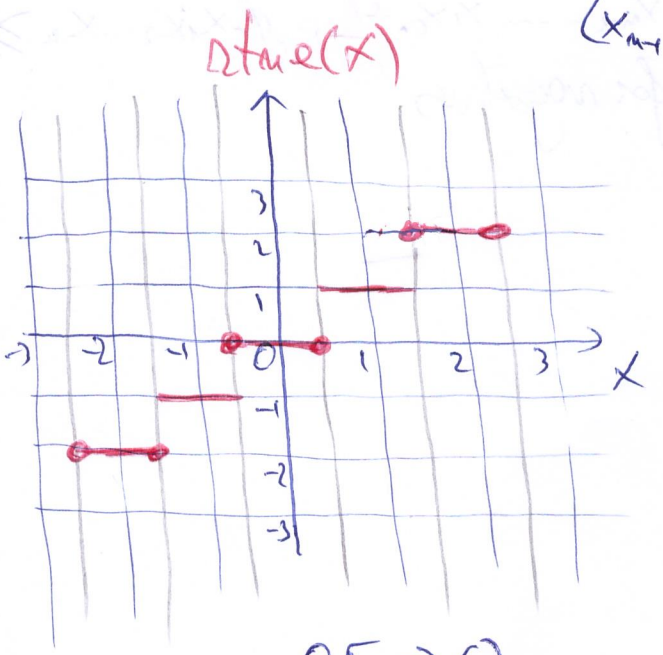
! - if the case $.10_{(2)}$ $(0.5_{(10)})$ if more probable

$\Rightarrow \varepsilon_{mean} > \frac{1}{8}$

- round the case $.1\underline{0}_{(2)}$ with equal probability ↗ upwards $(+\frac{1}{2})$ ↘ downwards $(-\frac{1}{2})$

rounding to nearest even: discard bit X_0

if X is positive $X^* = \begin{cases} x_{m-1}x_{m-2}\dots x_{00}, & \text{if } \begin{cases} \cdot x_{-1}x_{-2}\dots x_{-n} < \frac{1}{2}, \text{ or} \\ \cdot x_{-1}x_{-2}\dots x_{-n} = \frac{1}{2}, \text{ and } x_0 = 0 \end{cases} \\ x_{m-1}x_{m-2}\dots x_{00}+1, & \text{if } \begin{cases} \cdot x_{-1}x_{-2}\dots x_{-n} > \frac{1}{2}, \text{ or} \\ \cdot x_{-1}x_{-2}\dots x_{-n} = \frac{1}{2}, \text{ and } x_0 = 1 \end{cases} \end{cases}$



$0.5 \to 0$

$-0.5 \to 0$

$1.5 \to 2$

$-1.5 \to 2$