Weighted vs nonweighted fixed point codes

weighted

- Sign-magnitude
  - integer

    sign $n-2$ 2^i 2^0

$n-1$	$n-2$	...	i	...	0

  - fraction

    sign 2^{-1} 2^{-1+i-n} 2^{1-n}

$n-1$	$n-2$	...	i	...	0

- Binary coded decimal

  $10^i \cdot 2^j$

  sign 10^{n-2} 10^i 10^0

$n-1$	$n-2$	...	i	...	0

  sign 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0

$n-1$				...				...			

nonweighted

- One's complement
- Two's complement
- Three excess
- Two-out-of-five

Binary vs Decimal

$n = 10 \rightarrow 2^{10} = 1024 \cong 1000 = 10$

n bits $\rightarrow 2^n$ codes (words)

$\frac{n}{4}$ decimal digits $\rightarrow 2^{0{,}83n}$ codes

$$\left.\begin{array}{l} 10 \text{ ---- } 3 \\ x \text{ ---- } \frac{n}{4} \end{array}\right\} \rightarrow x = \frac{10n}{12} \cong 0{,}83n$$

* n bits $\longrightarrow 2^n$ unsigned binary numbers

n bits $\longrightarrow 10^{\frac{n}{5}}$ unsigned 2-out-of-5 decimal numbers

$$10^3 = 1000 \cong 1024 = 2^{10} \rightarrow \left.\begin{matrix} 3 \text{ --- } 10 \\ \frac{n}{5} \text{ --- } x \end{matrix}\right\} \Rightarrow x = \frac{\frac{n}{5} \times 10}{3} = \frac{2}{3} n \cong 0{,}66n$$

$\Rightarrow n$ bits $\longrightarrow x\ 2^{0{,}66n}$ unsigned 2-out-of-5 decimal numbers

Binary floating point numbers

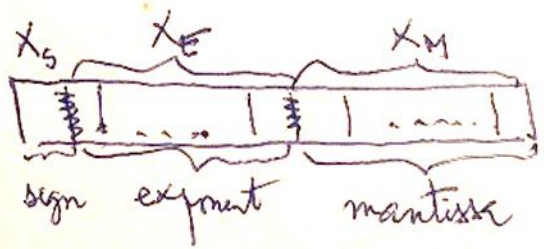
Number notations

- weighted notation: $N = \sum_{i=0}^{n-1} x_i r^i$ where $0 \leq x_i < r$
- scientific notation: $N = M \cdot B^E$

E ← exponent

B ← base

M ← mantissa



$$X = (-1)^{X_S} \cdot X_B^{X_E} \cdot X_M$$

sign, exponent, base, mantissa

Input

normalized floating point number

SM → 0, 1 ..... (sign, magnitude; mantissa)

C2 → $x_s, x_{M\,msb}$ ..... (sign; mantissa)

Mantissa problem

Computer

Floating point number → unnormalized / denormalized

Output normalized floating point number

Bias (Excess)

SM → $-2^{k-1}+1$

C2 → -2^{k-1}

$X_E - k$ bits

Exponent problem

SM → $2^{k-1}-1$

C2 → 2^{k-1}

$X = 100110,101 = 0,100110101 \times 2^7$

sign sign

0	0	0	1	1	1	1	0	0	1	1	1	0	1	0	1

$Y = 0,001011 = 0,1011 \times 2^{-2}$

0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	0

Normalization
- Point leftshift → Increasing exponent
- Point Rightshift → Decreasing exponent

Input
↓ Decimal to binary conversion & Normalization

Mantissa Problem

normalized number [unpack]

Computer
- unnormalized number
- denormalized number

↓ Normalization [pack]
↓ Binary to decimal
Output Conversion

Floating Point Addition

$X + Y = (X_M + Y_M^{Y_E - X_E}) X_E^E$ where $X_E \geq Y_E$

mantissa

1. Exponent comparison
   bx $7 - (-2) \Rightarrow$
   $X_E - Y_E = 7 - (-2) = 9$
2. Rightshift for alignment with $(X_E - Y_E)$ positions
3. Addition

$$\begin{array}{r} 0,100111010101 \\ +\ 0,000000001011 \\ \hline 0,100111010100 11 \end{array}$$

4. Normalization

$0,10011101001 1 \times 2^7$

Normalization rules → Mantissa

SM →

X_S X_M

0/1	----	1	---	

for normalization, point leftshift → increasing exponent or point rightshift & decreasing exponent

C2 →

0/1	1/0	---	

for normalization, point leftshift → increasing exponent or point rightshift & decreasing exponent, but WARNING! for negative mantisse rightshift introducing 1s!!

* Observation:

- through normalization operation

$$\frac{1}{2} \leq |X_M| < 1$$

absolute value

Exponent Problem

$X = +0{,}2_{10} = +0{,}001\,0001,\ldots_2$

X'

0	0	0	1	0	0	1	0

0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2

$$X' = 2^{-3} + 2^{-6} = \frac{9}{64} = 0{,}140\,625_{10}$$

$$\Sigma_{error} = 2 - 0{,}140\,625 = 0{,}059\,375_{10}$$

* from conversion algorithm → truncation error

Normalization rules → Mantissa

SM →

X_S X_M

0/1	----	1	---	

for normalization, point leftshift → increasing exponent or point rightshift & decreasing exponent

C2 →

0/1	1/0	---	

for normalization, point leftshift → increasing exponent or point rightshift & decreasing exponent, but WARNING! for negative mantisse rightshift introducing 1s!!

* Observation:

- through normalization operation

$$\frac{1}{2} \leq |X_M| < 1$$

absolute value

Exponent Problem

$X = +0{,}2_{10} = +0{,}001\,0001,\ldots_2$

X'

0	0	0	1	0	0	1	0

0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2

$$X' = 2^{-3} + 2^{-6} = \frac{9}{64} = 0{,}140\,625_{10}$$

$$\Sigma_{error} = 2 - 0{,}140\,625 = 0{,}059\,375_{10}$$

* from conversion algorithm → truncation error

Errors → Truncation errors

Errors → Rounding errors → Rounding process → Time penalty!

* By floating point operation → Partial result o mante

because errors → $Z = (-1)^{x_n - 1} \cdot X_M \cdot 2^{X_E}$ → significant erro

X_M: very small

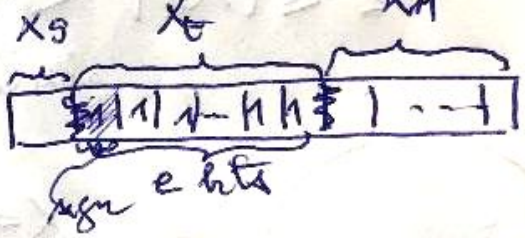
X_E: large

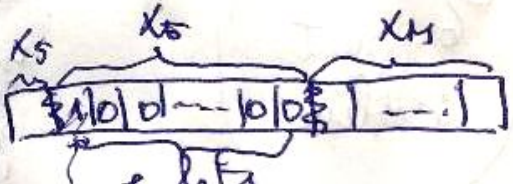
X_E smallest number

S1: $-2^{e-1}+1$

C2: -2^{e-1}

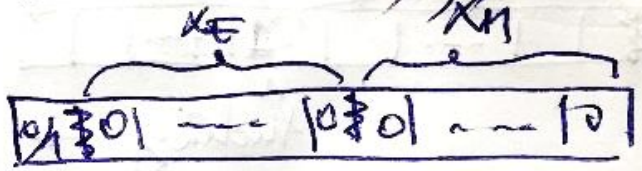
C2's aromaly



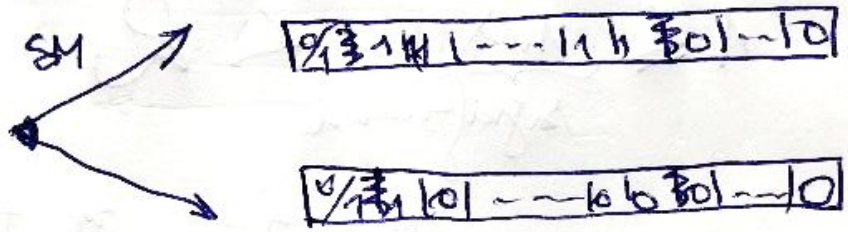


* But, for comparisons reasons (implementing of jumps or branch instructions), we want

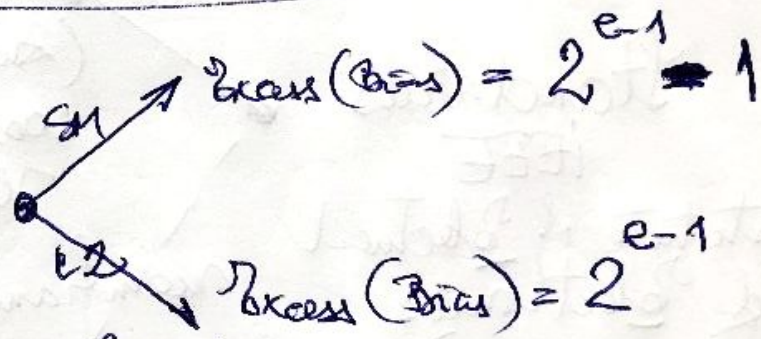
0 representation as



and we have



Excess (Biased) Representation



→ Exponent from signed integer number $\xrightarrow{\text{through biased representation}}$ unsigned binary number