Recapitulare

1. Clasificarea Informației

bit -> binary digit

- Info
  - -> Instructions
  - -> Data
    - -> Numbers
      - Fixed
      - Float
    - -> Non-numeric
      - ASCII
      - UNICODE
      - EBCDIC

---

bit -> byte (8 biți)
-> cuvinte (mult. de 8)

UNICODE -> local
-> international

↓

UTF-8 / 16 / 32

Nr. virgulă fixă

-> întregi $0 \to 2^{32}-1$ (uint32)

-> fracționare $2^{31} \to 2^{31}-1$ (int32)

Nr. virgulă mobilă

Numitor -> $\underline{2^x}$

$3.14 + 1e20 - 1e20 = 0$

$3.14 + (1e20 - 1e20) = 3.14000$

-> ~~Asociativitate~~

$3.14 + 1 \cdot 10^{20} = 10^{20}(3.14 \cdot 10^{-20} + 1)$

Reprezentarea nr. cu virg. fixă

$$X = \underbrace{x_{n-1}\, x_{n-2} \ldots x_0}_{\text{Intreg}} \cdot \underbrace{x_{-1}\, x_{-2} \ldots x_{-m}}_{\text{Fractionar}}$$

MSB $\to x_{n-1}$

LSB $\to x_{-m}$

$$X = \sum_{j=-m}^{n-1} x_j \cdot 2^j$$

2^j → ponderea cifrei j

$103_{(10)} = 1\ 1\ 0\ 0\ 1\ 1\ 1$

$2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$

pondere

P. Întreagă $X = \sum_{i=0}^{n-1} x_i \cdot 2^i$

P. Fracționară $X = \sum_{i=0}^{n-1} x_i \cdot 2^{i-m}$

Valoarea unui nr. fract. de virg. fixă se obține împărțind val. întreagă a aceleași configurații la 2 la puterea nr. de poz. binare a părții fracționare.

$$\frac{103}{128}_{(10)} = .11\,00111_{(2)}$$

$2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ \ {}^{-5}\ {}^{-6}\ {}^{-7}$

SEMN MĂRIME

poz. → bit de semn 0

neg. → 1

$$X = \underbrace{x_{n-1}}_{\text{semn}} \underbrace{x_{n-2} \ldots x_1 x_0}_{\text{magnitudine}}$$

$$X - Y = X + (-Y) \quad \text{(Nu func.) în SM}$$

MAXINT$_{(SM)}$: $0\,\underbrace{11\ldots111}_{n-1 \text{ bit.}}$

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

$$MAX_{FRA} = \underline{0}.1\ldots1 = \frac{2^{n-1}-1}{2^{n-1}}$$

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

$$2^{n-1} - 1 = 10^{p} \quad / \quad p = \log_{10}(2^{n-1}-1)$$

$$p \le \log_{10} 2^{n} \approx 0{,}3\,(n-1)$$

- Complexitatea Hardware
  - → moderată → favorabilă înmulțirii
  - → dezavantaj → adunare SDAR
    - poz + poz

Complement de 1

$poz_{(SM)} = poz_{(C1)}$

→ nu este repr. pozitională

→ este nevoie de un End Around Carry

Precizie —— || —— SM

→ dezavantaje → Adunare cu EAC

→ Nr Neg. → negarea celor pozitive

~ Nr

Complement de 2

MSB codifică semnul

$N_2 C_2 = N_2 C_1 + 1$

→ ignoră EAC la adunare

→ conversia SM ⟷ C2

- bit semn neschimbat
- negare la toți biții
- adunare 1 (ignoră EAC dacă ajunge la MSB)

$-4 = 10100_{SM}$

$= 11100_{C_2}$

→ adunare / scădere mai simplă ca SM / C_1

→ înmulțire mai grea

Avantajele aritmeticii în C_2

-> adunare / scădere
-> indiferent de semn $x - y = x + (-y)$
-> bitul de semn este folosit la fel ca cel de magnitudine
-> se ignoră EAC rezultat

Comparație -> cele pozitive sunt identice

între SM <-> C2

$-1 = -15$ (pe 5 biți)
$-2 = -14$
- - - -

Anomalie $-16 = 1.0000_{C_2}$

-2^{n-1}

-8 și $+8$
C -> aceeași config. binară
unsigned

Overflow

→ la op. fără semn
→ transport de MSB

$$\begin{array}{r|rrrl} & 1 & 1 & 0 & + \\ & 1 & 0 & 1 & \\ \hline 1 & 0 & 1 & 1 & \end{array}$$

$$\underbrace{011} \to \cancel{3}$$

Interpretarea Robertson

→ facilitează înmulțirea op. în C_2

→ val. unui nr. neg. în C_2 este egală cu val. nr. poz. obț.

Reprezentarea zecimală

BCD 8421 | Exces de 3 | Doi-din-cinci

tetradă = nibble = 4 biți consec.

$297_{10} = 0010\_1001\_0111$ BCD

pondere $10^i * 2^j$

$i \to$ poz. cif zecimală

$j \to$ poz. bit în tetradă

$j \in [0, 3]$

0.2	2	0
0.4	2	0
0.8	2	0
1.6	2	1
1.2	2	....

2	9
00 10	1 0 0 1
$10\cdot 2^0$	$2^3\ 2^2\ 2^1\ 2^0$

$10\cdot 2^1$

$10\cdot 2^2$

$10\cdot 2^3$

→ reprezentare pozițională

Exces de 3

Doi - din - cinci — 2 b. de 1
— 3 b. de 0

Repr. nr. în virgulă mobilă

$$X = X_M \cdot B^{X_E}$$

X_M → mantisă

B → baza (de obicei 2)

X_E → exponent

X_M și X_E pot fi repr. în SM/C_2