Representació numèrica

Sencers

- **Sense signe**
- **Amb signe**

Reals

- **Punt fix**
- **Punt flotant**

Sencer sense signe

- Representació (amb n = 8 bits)

  - Valor = $b_0 * 2^0 + b_1 * 2^1 + b_2 * 2^2 + ... + b_{n-1} * 2^{n-1}$
  - Rang: $[0 .. 2^n - 1]$ Per n = 8 → (0.. 255)
  - Exemple: $10_d = 0A_h = 00001010_b$

- Inconvenients
  - No es poden representar valors negatius

Sencer amb signe (Complement a 2)

- Representació (amb n = 8 bits)

  - Valor
    - Positiu (s = 0): $b_0 * 2^0 + b_1 * 2^1 + b_2 * 2^2 + ... + b_{n-2} * 2^{n-2}$
    - Negatiu (s = 1):$-(((1-b_0)2^0 + (1-b_1)2^1 + (1-b_2)2^2 + .. + (1-b_{n-2})2^{n-2}) + 1)$
  - Rang: $[-2^{n-1} .. 2^{n-1}-1]$
  - Exemple: $-10_d = F6_h = 11110110_b$
- Avantatges
  - Es poden representar valor negatius
  - La suma en binari natural permet sumar i restar
  - Representació única pel 0
- Inconvenients
  - Representació asimètrica

Reals: Punt Fix (Poc utilitzat)

- S'utilitza quan:
  - No es disposa d'una unitat de coma flotant.
  - Els càlculs en coma flotant són molt costosos
  - No es necessita una gran precisió o s'utilitza un nombre fix i petit de decimals
- Tècnica
  - Separar la representació en tres blocs fixos:
    - El signe
    - La part sencera
    - La part decimal
- Representació de N amb n bits = $n_s + n_d + 1$
  - n_s bits per la part sencera
  - n_d bits per la part decimal
  - 1 bit de signe
- Exemple: N = - 4,327 amb $n_s = 5$ i $n_d = 6$ (Paraula de 12 bits)
  - $4_{10} = 100_2$
  - $0.327_{10} = 010100_2$

$$
\begin{aligned}
0{,}327 * 2 &= \mathbf{0}{,}654 && (2^{-1}) \\
0{,}654 * 2 &= \mathbf{1}{,}308 && (2^{-2}) \\
0{,}308 * 2 &= \mathbf{0}{,}616 && (2^{-3}) \\
0{,}616 * 2 &= \mathbf{1}{,}232 && (2^{-4}) \\
0{,}232 * 2 &= \mathbf{0}{,}464 && (2^{-5}) \\
0{,}464 * 2 &= \mathbf{0}{,}928 && (2^{-6})
\end{aligned}
$$

El nº en punt fix és:

1 00100 010100

Té un interval de representació i cal tenir en compte l'error:

N = - (4 + **0** * 0,5 + **1** * 0,25 + **0** * 0,125 + **1** * 0,0625 + **0** * 0,03125 + **0** * 0,015625)

N = -4,3125

Reals: Punt Flotant

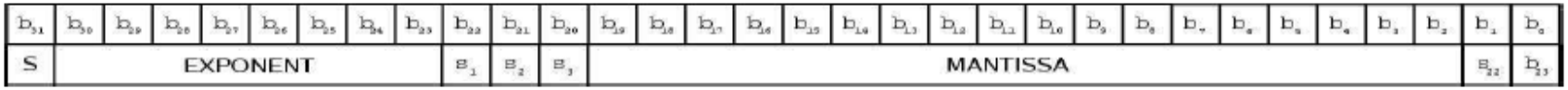
- Avantatges
  - Pot representar valors molt grans i molt petits
  - No utilitza molts bits per la representació
- Inconvenients
  - Pèrdua de precisió
  - L'error augmenta com més s'allunya del interval [-1..1]
- Representació de signe, exponent i mantissa

Valor = $(-1)^{\text{signe}} * \text{Mantissa} * \text{Base}^{\text{Exponent}}$

- Es pot definir la base (2, 4, 8 i 10), els bits dedicats a l'exponent i la mantissa i el seu format.
- L'IEEE 754 defineix un estàndard pels nombres reals de simple precisió (32 bits) anomenat *floats* i de doble precisió (64 bits) anomenats *doubles*.

Reals: Punt Flotant

- Format *float:*



$$Valor = (-1)^S * (1 + Mantissa) * 2^{Exponent}$$

$$Valor = (-1)^S * (1 + (b_1 * 2^{-1}) + (b_2 * 2^{-2}) + .. + (b_{23} * 2^{-23})) * 2^E$$

- Normalització: La mantissa sempre té un 1 implícit (24 bits)
- El camp de l'exponent (8 bits) conté implícitament el signe de l'exponent
- Utilitza una notació polaritzada per l'exponent (desplaçada 127)

$$Valor = (-1)^S * (1 + Mantissa) * 2^{(Exponent + 127)}$$

  - Amb aquesta notació es poden ordenar els valors pel pes dels seus bits sense haver d'interpretar el valor que representen.

- Exemple: -0,75 en un real $-0{,}75_d = 0.11_b$

  Notació científica: $-0.11_b * 2^0$

  Normalització (el primer bit ha de ser 1): $-1.1_b * 2^{-1}$

  $126_{10} = 0111\ 1110_2$

- El valor del real serà:

$$(-1)^1 * (1 + .1000\ 0000\ 0000\ 0000\ 0000\ 000_b) * 2^{126}$$

- Que es representa:

$$1\ 0111\ 1110\ 1000\ 0000\ 0000\ 0000\ 0000\ 000_b = BF40\ 0000_h$$

- Valor = $(-1)^S * (1 + \text{Mantissa}) * 2^q$
  - Q ha de ser un sencer en l'interval:
    - $[1-e_{max} \leq q + p - 1 \leq e_{max}]$
    - p nº de bits de mantissa

Nom	Nom comú	Base	Dígits	Dígits decimals	Bits de l'exponent	$E_{màx}$ Decimal	Biaix del Exponent 7	E_{min}	$E_{màx}$
binary16	Mitja precisió	2	11	3,31	5	4,51	$2^4 - 1 = 15$	−14	+15
binary32	Simple precisió	2	24	7,22	8	38,23	$2^7 - 1 = 127$	−126	+127
binary64	Doble precisió	2	53	15,95	11	307,95	$2^{10} - 1 = 1023$	−1022	+1023
binary128	Quàdruple precisió	2	113	34,02	15	4931,77	$2^{14} - 1 = 16383$	−16382	+16383
binary256	Òctuple precisió	2	237	71,34	19	78.913,20	$2^{18} - 1 = 262143$	−262142	+262143

Reals: Punt Flotant

- Operacions: Suma
  - Valor = (s) M * B^E ;
  - Es poden sumar les mantisses quan els exponents son iguals
- Exemple:
  - X = 24.0 = $(00011000)_2$ = 1.1 * 2^{100}
  - Y = 80.0 = $(01010000)_2$ = 1.01 * 2^{110}

0	10000011	10000000000000000000000
0	10000101	01000000000000000000000

- Registres:
  - X: E1 = 00000100; M1 = 1.1
  - Y: E2 = 00000110; M2 = 1.01
- Pas 1: E = E1-E2 = $100_2 - 110_2 = -10_2$ En C'2: E = 00000100-00000110 = 11111110 (-2) $_{C'2}$
  - E < 0 → M1 = 0.11 i E = E + 1 = -10 + 1 = -1 : 11111110 + 1 = 11111111
  - E < 0 → M1 = 0.011 i E = E + 1 = -1 + 1 = 0 : 11111111 + 1 = 00000000
  - E = 0 → res
- Pas 2: E = Max (E1, E2) = 110_2
  - S'ha de sumar el desplaçament de l'exponent: 110_2 (6) + 01111111_2 (127) = **10000101_2**
- Pas 3: A = M1 + M2
  - A: 0.011 + 1.01 = 1.101
  - Ja està normalitzat (1....)
  - No hi ha desbordament
  - És diferent de zero

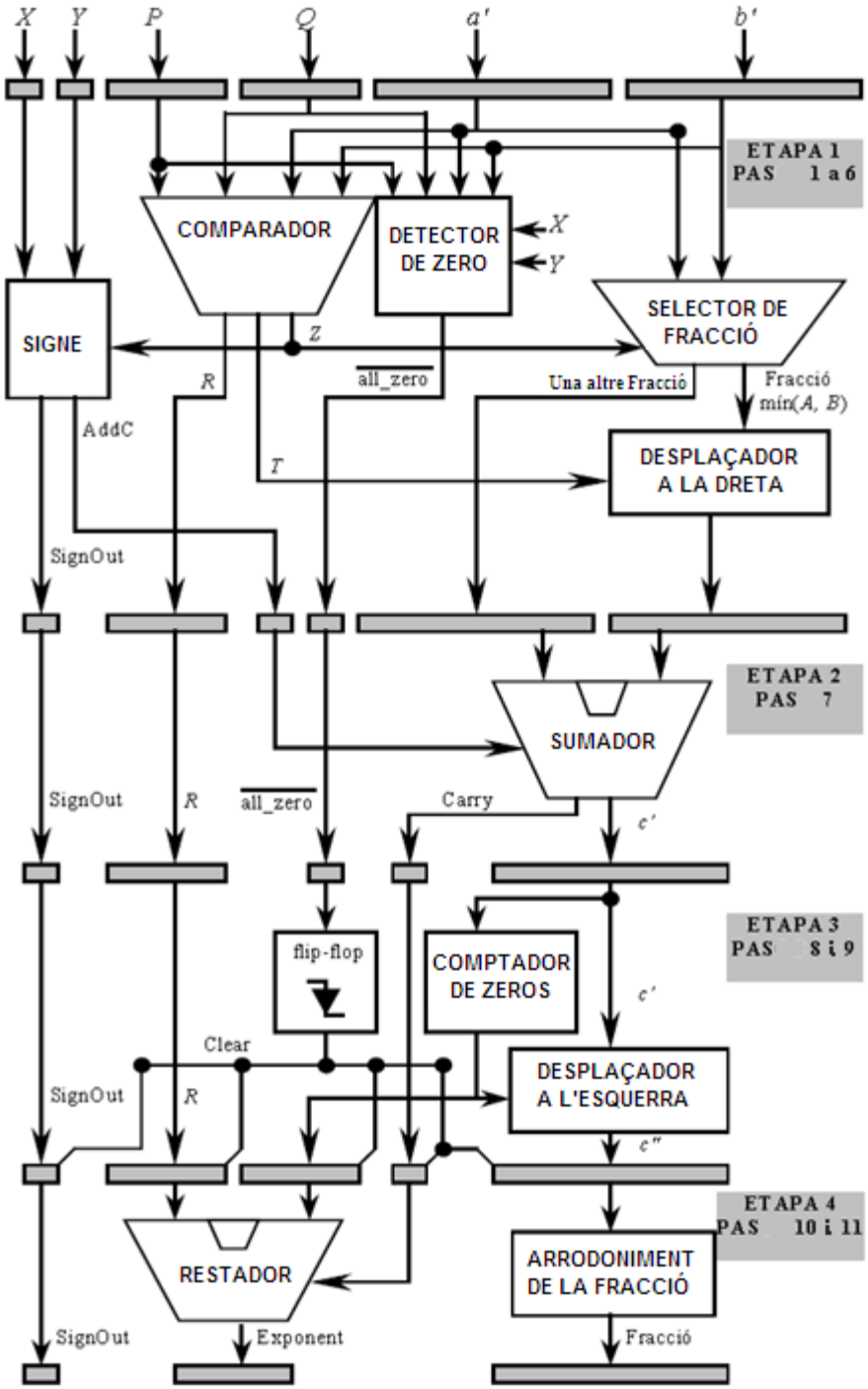
0	10000101	10100000000000000000000

X + Y = 1.101 * 2^6 = 1101000_2 = 64 + 32 + 8 = 104

Disseny d'un sumador / restador de quatre etapes

$A = (-1)^X\ 1.a\ 2^P$ i $B = (-1)^Y\ 1.b\ 2^Q$ $\quad C = A \pm B = (-1)^Z\ 1.c\ 2^R$

1. Afegir el bit implícit de la fracció juntament amb els bits de guarda a les mantisses a i b, obtenint les fraccions amb què s'operarà a‘ i b'.
2. Calcular $R = màx(P, Q)$, determinar la diferència $T = |P - Q|$, i determinar el més gran dels nombres d'entrada: Z (= *0* si $A<B$; = *1* en cas contrari).
3. Determinar el signe del resultat i l'operació binària a realitzar. El signe del resultat està determinat per Z (el signe de la suma dels dos nombres serà el signe del nombre més gran). Per determinar l'operació binària (suma o resta) n'hi haurà prou amb comparar els signes dels respectius números: X i Y.
4. Detectar els casos especials en què s'ha de tractar el zero: els dos números d'entrada són zero o són iguals i l'operació a realitzar és la resta.
5. Seleccionar la fracció (a‘ o b') que es correspon al nombre menor.
6. Desplaçar a la dreta la fracció associada amb el nombre menor T llocs (bits) per igualar els exponents abans de sumar les fraccions.
7. Sumar les fraccions per produir la fracció de la suma no normalitzada *c*'. L'operació pot donar un bit de ròssec que es tradueix en un increment del exponent del resultat, resultat conegut com desbordament de mantissa.
8. Determinar la quantitat u de zeros inicials en la fracció *c*'.
9. Desplaçar *c*‘ a l'esquerra u llocs per produir la fracció normalitzada *c*".
10. Modificar l'exponent resultat restant els u llocs que s'ha desplaçat la fracció per normalitzar i després se li suma l'eventual ròssec produït en el punt 7.
11. Arrodonir-lo d'acord el que indica la norma IEEE 754.


X
Y
P
Q
a'
b'
ETAPA 1
PAS 1 a 6
COMPARADOR
DETECTOR DE ZERO
X
Y
SELECTOR DE FRACCIÓ
SIGNE
Z
R
all_zero
Una altre Fracció
Fracció mín(A, B)
AddC
T
DESPLAÇADOR A LA DRETA
SignOut
ETAPA 2
PAS 7
SUMADOR
SignOut
R
all_zero
Carry
c'
ETAPA 3
PAS 8 i 9
flip-flop
COMPTADOR DE ZEROS
c'
Clear
SignOut
R
DESPLAÇADOR A L'ESQUERRA
c''
ETAPA 4
PAS 10 i 11
RESTADOR
ARRODONIMENT DE LA FRACCIÓ
SignOut
Exponent
Fracció

Reals: Punt Flotant

- Operacions: Producte
  - El signe del resultat és igual a la XOR dels signes dels operants.
  - La mantissa és el resultat normalitzat del producte de les dues mantisses.
  - L'exponent del resultat es correspon a la suma dels exponents
    - E = E1 + E2 – 127 (Es sumen els dos desplaçaments -> s'ha d'eliminar-ne un)

Ex:

$X = (-1)^{s1} * 1.mant1 * 2^{E1}$

$Y = (-1)^{s2} * 1.mant2 * 2^{E2}$ $Z = (-1)^{(s1 \oplus s2)} * (1.mant1 * 1.mant2) * 2^{(E1+E2-127)}$

- Operacions: Divisió
  - El signe del resultat és igual a la XOR dels signes dels operants.
  - La mantissa és el resultat normalitzat de la divisió de les dues mantisses.
  - L'exponent del resultat es correspon a la resta dels exponents
    - E = E1 – E2 + 127 (S'eliminen els dos desplaçaments -> s'ha de sumar desplaçament)

Ex:

$X = (-1)^{s1} * 1.mant1 * 2^{E1}$

$Y = (-1)^{s2} * 1.mant2 * 2^{E2}$ $Z = (-1)^{(s1 \oplus s2)} * (1.mant1 / 1.mant2) * 2^{(E1-E2+127)}$