IV.4.8.
Reprezentarea numerelor reale: aritmetica în virgulă mobilă

Reprezentarea în virgulă mobilă

- Reprezentările în virgulă fixă nu sunt potrivite pentru numere reale
- Dualitatea magnitudine / precizie
  - lungimea n+m fixată
  - creşterea magnitudinii înrăutăţeşte precizia şi reciproc
- Virgulă mobilă: un număr se reprezintă printr-o pereche de reprezentări în virgulă fixă
  - permiţând reprezentarea simultană de numere cu magnitudini şi precizii diferite

Notaţia ştiinţifică; normalizare

- Exemple
    - $7.015791043\ldots_{zece}$
        - » scriere poziţională
    - $55312.1784_{zece} \times 10^{-105}$
        - » ar necesita 110 cifre în scriere poziţională, doar 14 aici.
    - $0.00000083_{zece} \times 10^{4}$
        - » Se poate scrie şi eliminând zerourile semnificative
  - – Numeroase scrieri posibile pentru acelaşi număr
- Notaţia ştiinţifică:
  - – o singură cifră la stânga virgulei
  - – $0.031056_{zece} \times 10^{7}$
  - – încă numeroase scrieri posibile pentru acelaşi număr
        - » $m_1 \times B^{e1} = m_2 \times B^{e1+k}$, unde $m_2 = m_1 / B^{-k}$, pt. oricare $k \in \mathbf{Z}$.
- Scriere normalizată: reprezentare în notaţie ştiinţifică *fără prefix de zerouri semnificative*
    - $6.15_{zece} \times 10^{-75}$
        - » poziţional → 78 de cifre semnificative, dintre care 75 zerouri
  - – unică pentru un număr dat

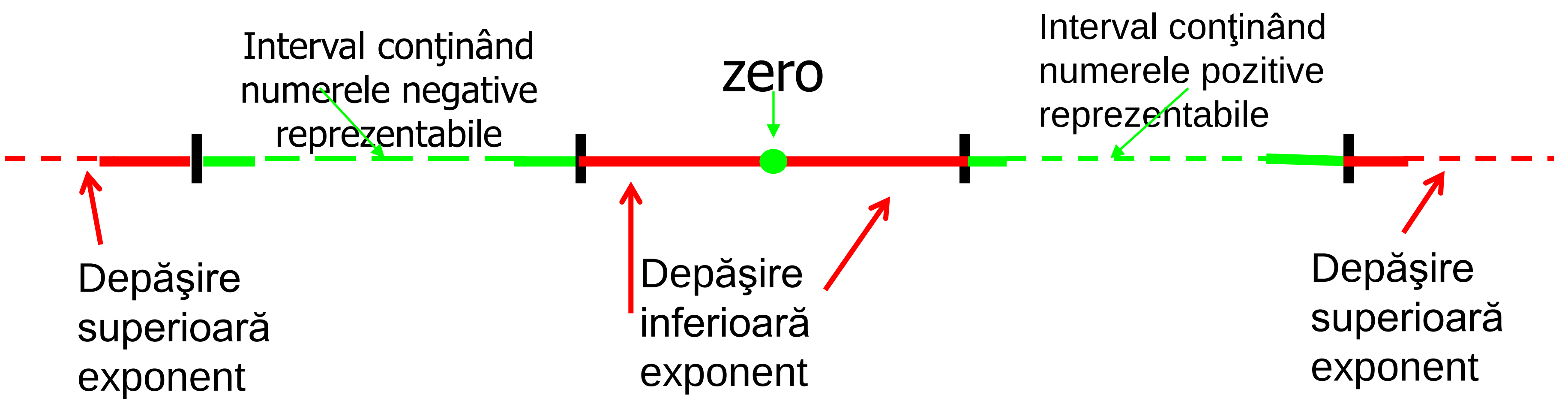
Notaţia ştiinţifică în binar

- $1.101_{doi} \times 2^{2}$
  - $6{,}5_{zece}$

- $1.xxx\ldots x_{doi} \times 2^{yyy}$
  - în binar, doar numărul 0 nu conţine nici o cifră 1
    - definire specială a reprezentării lui 0
  - pentru oricare alt număr reprezentabil, cel mai semnificativ 1 trebuie să devină singura cifră de la partea întreagă
    - ceea ce, în general, alterează exponentul lui 2

Reprezentări în virgulă mobilă

- Semnul (S): 0 sau 1
  - 1 bit
- Partea fracţionară (f); mantisa este 1+f
  - mantisa are 1+23 (sau 1+52) biţi
- Caracteristica (C)
  - k=8 sau k=11 biţi **exces = $2^{k-1} - 1$**
  - **C = exponent + exces**
- **x = (+/-) 1.f × B $^{C - exces}$**
- Primul 1 şi baza 2 se subînţeleg/nu se reprezintă
  - » O figură similară pentru numerele reprezentabile în virgulă fixă ar indica un singur interval conţinând numere reprezentabile, simetric faţă de 0 şi cu cele două capete aflate fiecare într-unul din intervalele figurate aici cu verde.



Standardizare

- Esenţială pentru portabilitate şi pentru ataşarea unei semantici generale a reprezentării în virgulă mobilă.
- Proces început în 1977, încheiat parţial în 1985.
- W. Kahan (University of Toronto).
- Prima implementare comercială a standardului IEEE (pe atunci, în curs de elaborare): 1981 - Intel 8087.

Standardul IEEE 754 / 1985

- Precizie simplă: *float* în C/C++ (32 biţi)

0	1 8	9 31
s	c = exponent + 127	f = partea fracţionară a mantisei

Margini în baza 10: $\mathbf{1.2 \times 10^{-38} \rightarrow 3.4 \times 10^{38}}$

Interesează nu doar structura reprezentării, ci mai ales *operaţiile* ce se pot efectua cu reprezentări.

Exemple:

Compararea: reprezentarea în XS are avantajul inducerii ordinii naturale pe mulţimea reprezentării exponenţilor

Înmulţirea: adunarea exponenţilor la înmulţire – cu scăderea excesului

Precizie dublă

- Precizie dublă: *double* în C/C++ (64 biţi)

0 … 31

1	11	20
s	exponent + 1023	f = partea fracţionară a mantisei
continuarea părţii fracţionare a mantisei		

0 … 31

Margini în baza 10: $\mathbf{1.7 \times 10^{-308} \rightarrow 1.7 \times 10^{308}}$

Faţă de precizia simplă:

- Creşte intervalul pentru exponent
- La exponent egal, creşte acurateţea (precizia) reprezentării datorită lungimii mai mari a părţii fracţionare a mantisei

Două reprezentări în virgulă fixă = o reprezentare în virgulă mobilă

- Mantisa: reprezentare *modul şi semn* (A+S) a coeficientului puterii bazei
- Caracteristica: reprezentare *în exces* a exponentului

- Există deci o valoare minimă e_{min} şi una maximă e_{max} pentru exponent
  - de unde structura mulţimii numerelor reprezentabile
- Ordinea din reprezentare (S C f) facilitează compararea reprezentărilor
  - ordinea numerelor reprezentate coincide astfel cu ordinea lexicografică a reprezentărilor

Reprezentări în norma IEEE 754

	Precizie simplă	Precizie dublă
Biţi "precizie"	24	53
Exponent maxim	128 (pentru numere: 127)	1024 (pentru numere: 1023)
Exponent minim	-127 (pentru numere normalizate: -126)	-1023 (pentru numere normalizate: -1022)
Exces (exponent)	127	1023

Un exemplu

- Cum se reprezintă numărul -7 în virgulă mobilă simplă precizie (IEEE 754) ?

1. **Semnul**: minus, deci 1
2. **Trecere în baza 2:** $7_{zece} = 111_{doi}$
3. **Normalizare:** $111_{doi} = 1.11_{doi} \times 2^2{}_{zece}$
4. **Calculul caracteristicii** (pe 8 biţi) **:** $(2 + 127)_{zece} = 129_{zece} = 10000001_{doi}$
5. **Reprezentarea:**

$1\ 100.0000.1\ 110.0000.0000.0000.0000.0000_{doi}$

$C0E00000_{hexazecimal}$

Încă un exemplu

- Cum se scrie poziţional în baza zece numărul reprezentat în simplă precizie IEEE 754 prin $C1F00000_{\text{hexazecimal}}$?

1. **Scriere în binar:** $C1F00000_{\text{hexazecimal}} =$
   $1100\ 0001\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000_{\text{doi}}$
2. **Semn:** 1, deci – (număr negativ)
3. **Caracteristica:** $10000011_{\text{doi}} = 131_{\text{zece}}$
4. **Exponentul:** $131 – 127 = 4$
5. **Mantisa:** $(1 + 0{,}111)_{\text{doi}} = 1{,}111_{\text{doi}}$
6. **Valoarea:** $-1{,}111 \times 2^4 = -11110_{\text{doi}} = -30_{\text{zece}}$

Aritmetica extinsă (principii)

- Aritmetica reală uzuală
  - proiectată pe mulţimea numerelor reale reprezentabile
  - cu operaţiile uzuale
- La care se adaugă:
  - reprezentare pentru ∞ şi reguli elementare de calcul cu acesta (a / ∞, ∞ + ∞)
  - reprezentări pentru rezultatul operaţiilor nedefinite (NaN) şi reguli de propagare a acestuia (NaN ***op*** x = NaN)

Aritmetica extinsă - exemplu

- Calculul lui arccos cu formula:

  $\arccos(x) = 2 \arctan(\mathrm{sqrt}((1-x) / (1+x)))$
- $\arccos(-1) = ?$
- $1+x \rightarrow 0 \Rightarrow 2 / (1+x) \rightarrow \infty \Rightarrow \arctan((1-x) / (1+x)) \rightarrow \pi / 2$
- Aceste relaţii fac parte din aritmetica în virgulă mobilă IEEE 754
- Rezultat - corect: $\arccos(-1) = \pi$

Tipuri de valori în virgulă mobilă

- În fiecare caz de mai jos, S este +1 sau -1 după cum bitul semn este 0 sau 1

Tip valoare	**e**	**f**	**Valoare**
normalizată	$e_{min}<e<e_{max}$	f oarecare	$(-1)^s \times 1.f \times 2^e$
denormalizată	$e = e_{min}$	$f \neq 0$	$(-1)^s \times 0.f \times 2^e$
zero	$e = 0$	$f = 0$	S 0
infinit	$e = e_{max}$	$f = 0$	S ∞
NaN	$e = e_{max}$	$f \neq 0$	NaN

Depăşiri

- **Depăşire inferioară**: în forma normalizată a numărului, exponentul **negativ** nu poate fi reprezentat în câmpul caracteristicii
  - numărul va fi considerat 0
- **Depăşire superioară**: în forma normalizată a numărului, exponentul **pozitiv** este prea mare pentru a putea fi reprezentat în câmpul caracteristicii
  - numărul va fi considerat $\pm\infty$

Reprezentări denormalizate

- număr mai mic în modul decât cea mai mică reprezentare normalizată
  - se renunţă la normalizare
  - mantisa va fi 0.f, în loc de 1.f
  - iar exponentul va avea valoarea minimă
    - -127 pentru simplă precizie
    - -1023 pentru dublă precizie
  - astfel se pot reprezenta numere mai mici

Aritmetica în virgulă mobilă

- În general: fie $x = x_m \times 2^{xe}$ şi $y = y_m \times 2^{ye}$
  - Relaţiile de mai jos se referă la notaţia ştiinţifică, nu la reprezentarea în standard IEEE
    - » x_m şi y_m sunt mantisele, iar xe şi ye exponenţii (nu caracteristicile)
  - $x + y = (x_m \times 2^{xe-ye} + y_m) \times 2^{ye}$, dacă $xe \le ye$
  - $x - y = (x_m \times 2^{xe-ye} - y_m) \times 2^{ye}$, dacă $xe \le ye$
  - $x \times y = (x_m \times y_m) \times 2^{xe+ye}$
  - $x : y = (x_m : y_m) \times 2^{xe-ye}$
- Operaţiile sunt însă mai complicate decât o arată formulele

Adunarea în virgulă mobilă

1. Se *compară exponenţii* termenilor adunării
   - dacă aceştia sunt diferiţi, atunci partea fracţionară a numărului mai mic se deplasează spre dreapta (denormalizare!) până când exponentul său devine egal cu exponentul mai mare
   - evident, unitatea hardware care face calculele operează în interior cu mai mulţi biţi decât reprezentarea standard
2. Se adună mantisele
   - aici se decide şi **semnul** sumei
   - adunarea mantiselor se efectuează în complement faţă de 2
     - deci pentru numere negative, se face complementarea corespunzătoare, iar rezultatul – dacă este negativ – se complementează pentru a reveni la A+S
3. Dacă este nevoie, se normalizează suma
   - fie decalând rezultatul spre dreapta şi incrementând exponentul, fie decalând spre stânga şi decrementând exponentul
4. Dacă se poduce depăşire → excepţie → stop
5. Rotunjeşte mantisa la numărul permis de biţi
   - dacă astfel s-a produs denormalizare, reia de la 3

Temă

- Să se urmărească paşii adunării în virgulă mobilă pentru reprezentările numerelor scrise în baza zece ca 0,75 şi -0,375.

  Se va considera că semnul, exponentul şi mantisa sunt obţinute ca pentru IEEE 754, dar se reprezintă pe 1 bit, 8 biţi, respectiv 4 biţi.

Înmulţirea în virgulă mobilă

1. Se calculează exponentul rezultatului adunând exponenţii celor doi factori
   - se adună caracteristicile şi se scade excesul
2. Se înmulţesc mantisele
3. Se normalizează rezultatul
   - dacă se produce depăşire →excepţie → stop
4. Se fac rotunjirile necesare
   - dacă se produce denormalizare, reia de la 3
5. Se determină semnul rezultatului

Temă

- Să se urmărească paşii înmulţirii în virgulă mobilă pentru reprezentările numerelor scrise în baza zece ca 0,75 şi 0,375.

  Reprezentările sunt cele din standardul IEEE 754.

Capitolul al V-lea

ARHITECTURA ŞI ORGANIZAREA CALCULATORULUI

V.1.
CALCULATOARE VON NEUMANN

Calculatoare von Neumann

- program memorat
  - memorie infinită (ideal), timp de acces egal
    - realizată practic prin *ierarhii de memorie*
- la execuţie, după o instrucţiune **i** urmează
  - instrucţiunea memorată imediat după ea (regula)
  - eventual instrucţiunea indicată de **i** (dacă **i** este instrucţiune **de control**)
- adresa instrucţiunii următoare se află într-un registru – PC
- în fiecare moment, o singură instrucţiune este încărcată pentru execuţie

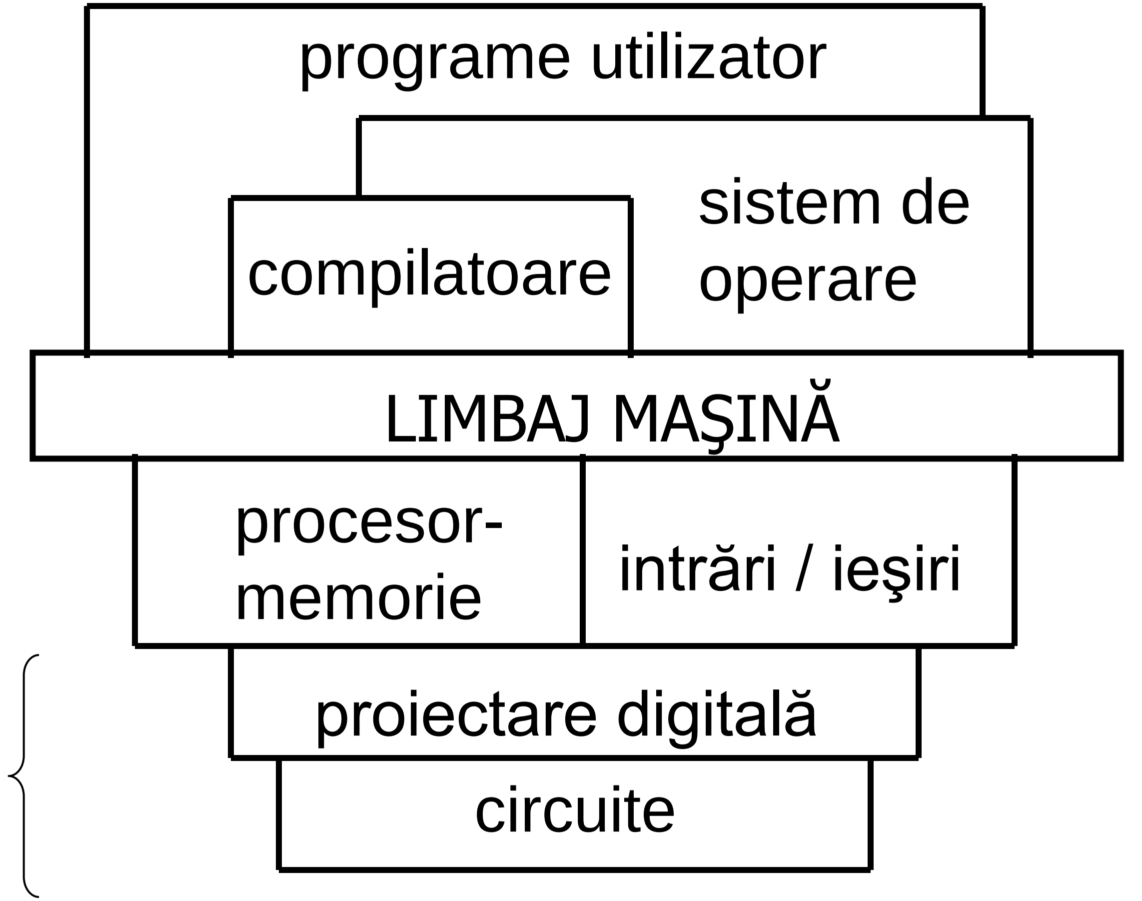
Calculatoare von Neumann

- Anterior:
  - conceptul de automatizare a operaţiilor luate separat (Pascal, Leibniz)
  - conceptul de program – exterior (Babbage)
  - conceptul de calcul ramificat / control
- Conceptul de program memorat
  - John von Neumann et al (1946)
  - concept arhitectural fundamental - procesoarele moderne

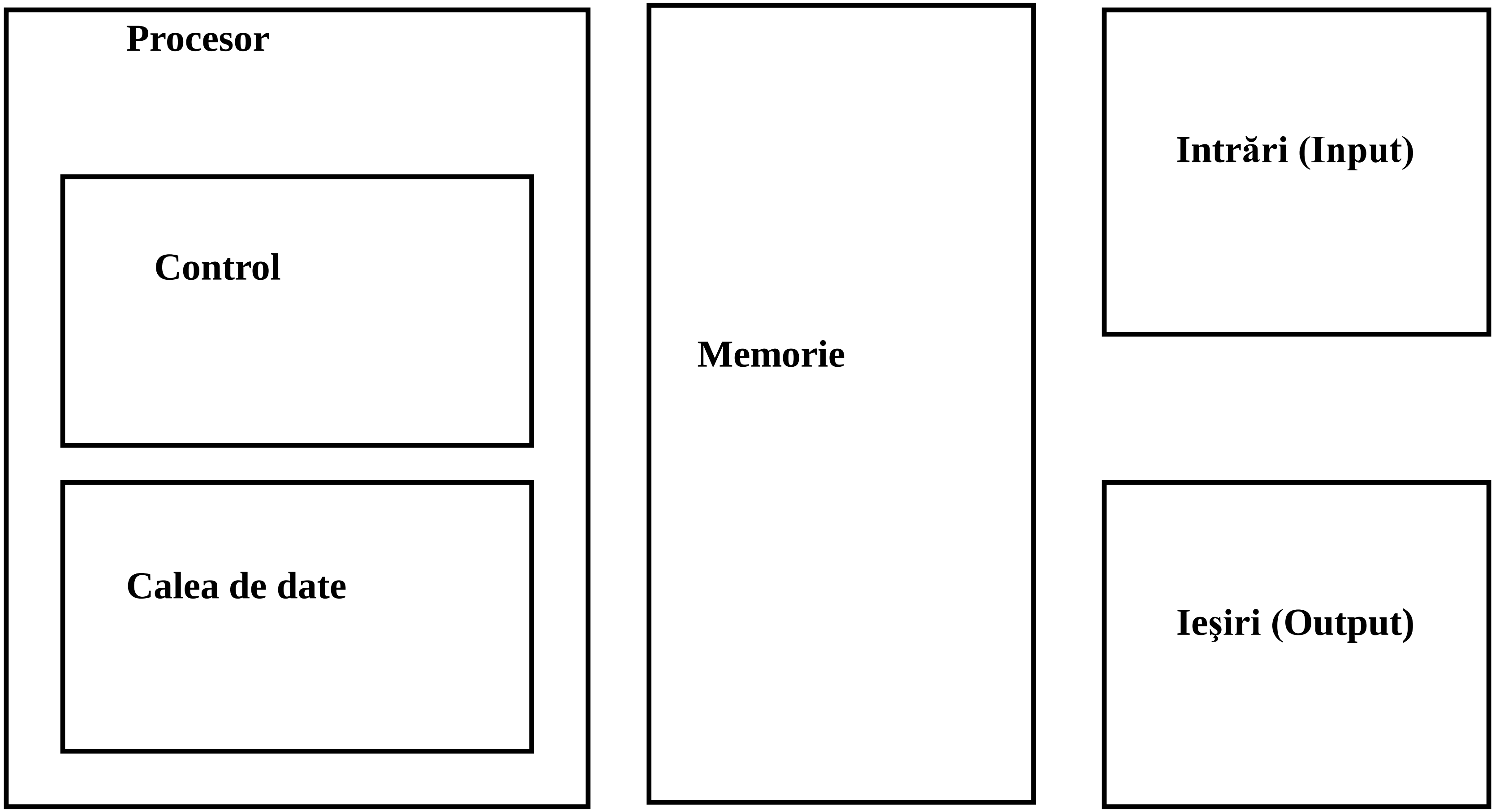
Calculatoare von Neumann

- Programul şi datele - stocate în (aceeaşi) memorie
  - ideal: infinită, omogenă (locaţii la fel de rapid accesabile)
  - practic: o ierarhie de memorii, fiecare fiind omogenă
- Program counter (PC) indică locul din memorie al instrucţiunii de executat
  - conţinutul PC este actualizat la execuţia fiecărei instrucţiuni
  - o dată sau de două ori
- Instrucţiunile programului sunt aduse pe rând din locaţii de memorie în procesor
  - regula: locaţii succesive → incrementare PC
  - excepţia: instrucţiuni de salt
  - ordine fizică şi ordine logică

Arhitectura unui sistem de calcul



Componentele hardware ale unui calculator



Organizarea unui calculator