

2. Unitatea aritmetică și logică

- Adunarea
- Înmulțirea
- Împărțirea
- Numere și operații în virgulă mobilă

Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

Reprezentarea numerelor în virgulă mobilă (1)

- Mai multe posibilități pentru reprezentarea valorilor care nu sunt întregi
- Reprezentare în virgulă fixă
 - Virgula binară plasată într-o poziție predefinită
 - Sunt necesare operații de scalare și deplasare
- Reprezentare în virgulă mobilă
 - Factorul de scală devine o parte a cuvântului din calculator
 - Poziția virgulei binare variază în mod automat

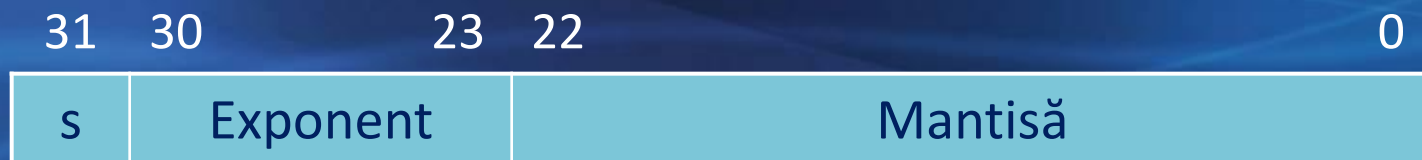
Reprezentarea numerelor în virgulă mobilă (2)

- Reprezentarea în virgulă mobilă (VM) a numărului N :

$$N = (-1)^s \times b^e \times m$$

- s – semn (0 sau 1)
- b – bază
- e – exponent: ordinul de mărime al numărului
- m – mantisă: valoarea exactă a numărului într-un anumit domeniu
- Codificarea reprezentării: un șir de biți
- Exemplu de codificare →

Reprezentarea numerelor în virgulă mobilă (3)



- Reprezentare în mărime și semn
- Nu există un câmp rezervat pentru baza b
- De obicei, nu se reprezintă exponentul real
→ **exponent deplasat** (caracteristică)

$$E = e + \text{deplasament}$$

- Exponentul deplasat este întotdeauna pozitiv

Reprezentarea numerelor în virgulă mobilă (4)

● Exemplu:

- Exponent deplasat de 8 biți: $E \in [0, 255]$
- Deplasament = 128 (80_{16})
- Exponentul real: $e \in [-128, +127]$
- Exponentul real este:
 - Negativ dacă $E < 128$
 - Pozitiv dacă $E > 128$
 - Zero dacă $E = 128$
- Exponentul este reprezentat în exces 128

Reprezentarea numerelor în virgulă mobilă (5)

- **Avantaje** ale exponentului deplasat:
 - Simplificarea operațiilor cu exponentul
 - Reprezentarea numărului zero
 - Mantisa: cifre de 0 în toate pozițiile
 - Exponentul: poate avea, teoretic, orice valoare → “zero impur” sau “zero pur”
 - “Zero pur”: reprezentarea în VM a numărului zero este aceeași cu reprezentarea numerelor întregi → se pot utiliza aceleași circuite

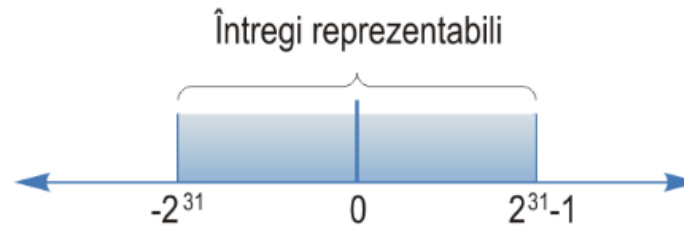
Reprezentarea numerelor în virgulă mobilă (6)

- Ordonarea numerelor
 - Numerele pozitive în VM sunt ordonate în același fel ca și numerele întregi
 - Mărimea numerelor în VM poate fi comparată cu un comparator pentru numere întregi
- Dezavantaj al exponenților deplasați:
 - Adunarea exponenților este mai complicată
→ deplasamentul trebuie scăzut din suma exponenților

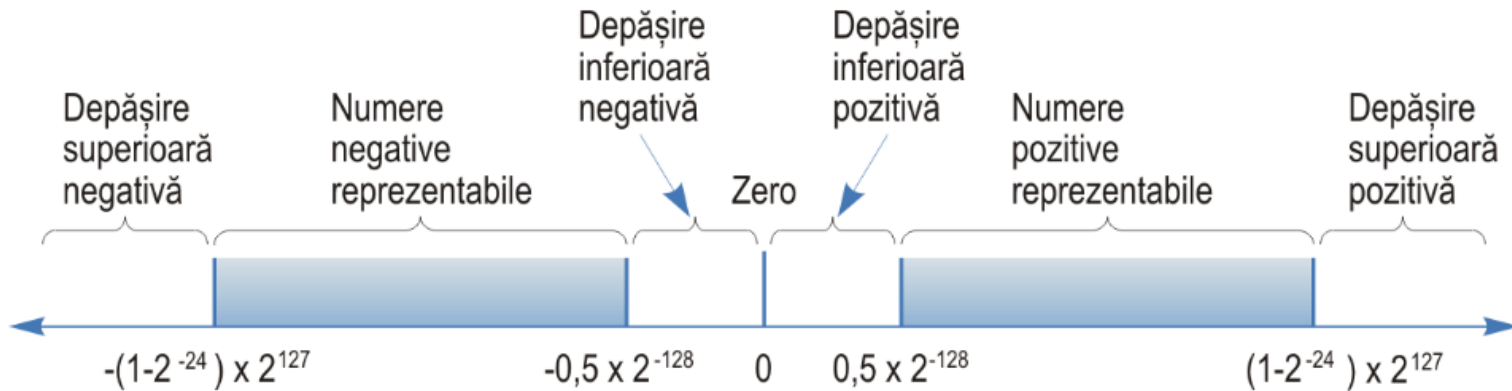
Reprezentarea numerelor în virgulă mobilă (7)

- Reprezentările în VM nu sunt unice: $0,1 \times 10^0$; 1×10^{-1} ; $0,01 \times 10^1$
- Reprezentare sub formă normalizată:
 - Bitul c.m.s. al mantisei este 1
 - **Avantaje:** operațiile sunt simplificate și precizia este crescută
- Deoarece bitul c.m.s. este 1, acest bit nu este memorat → **bit ascuns**
- Gama numerelor care pot fi reprezentate pe 32 de biți, întregi și în VM →

Reprezentarea numerelor în virgulă mobilă (8)



(a) Întregi în C2



(b) Numere în virgulă mobilă

Reprezentarea numerelor în virgulă mobilă (9)

- În unele cazuri, bitul ascuns se presupune poziționat la stânga virgulei binare $\rightarrow 1, m$
- **Depășire superioară**: exponentul depășește valoarea maximă (ex., $e > 127$)
- **Depășire inferioară**: exponentul are o valoare negativă prea mică (ex., $e < -128$)
- Unitățile de calcul în VM au mecanisme pentru detectarea, semnalarea și tratarea depășirilor

Reprezentarea numerelor în virgulă mobilă (10)

- Alegerea unui format în VM: compromis între dimensiunea mantisei și cea a exponentului
 - Creșterea dimensiunii mantisei → creșterea preciziei
 - Creșterea dimensiunii exponentului → creșterea domeniului
- Pentru a crește atât precizia, cât și domeniul: utilizarea unui număr mai mare de biți
 - Formate în precizie simplă și precizie dublă

Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (1)

- Au existat diferențe în modul de execuție a operațiilor în VM la diferite familii de calculatoare
- Societatea de Calculatoare a IEEE a elaborat un standard pentru reprezentarea numerelor în VM → IEEE 754-1985
 - Revizia curentă: IEEE 754-2008
- Majoritatea unităților de calcul în VM se conformează acestui standard

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (2)

- Specifică **formate** și **metode** pentru operații în VM la sistemele de calcul
- Definește **condiții de excepție** și tratarea acestor condiții
- Scopul: calcule în VM cu aceleași rezultate
 - Procesarea prin hardware, software sau o combinație a acestora
 - Erori raportate într-un mod consecvent

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (3)

- Standardul specifică:
 - Formate pentru operații aritmetice cu date binare și zecimale în VM
 - Formate pentru interschimbarea datelor
 - Operații de bază în VM
 - Conversii între: diferite formate în VM; formate întregi și în VM; formate în VM și reprezentări externe (șiruri de caractere)
 - Reguli de rotunjire
 - Excepții și tratarea acestora

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (4)

- Reprezentarea mantisei este numită *significand* în standardul IEEE
- Formate de bază
 - Trei formate **binare** (32, 64 și 128 biți)
 - Două formate **zecimale** (64 și 128 biți)
 - O implementare conformă trebuie să implementeze cel puțin unul din formatele de bază
 - Inițializare, operații, conversii, citire și scriere utilizând o codificare definită

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (5)

- Formate cu precizie extinsă
 - Extind formatele de bază: mai multe cifre ale mantisei, domeniu mai larg al exponentului
- Formate cu precizie extensibilă
 - Lungimea mantisei și domeniul exponentului sunt definite de utilizator
- Implementarea formatelor cu precizie extinsă și extensibilă este opțională
- Codificarea: dependentă de implementare

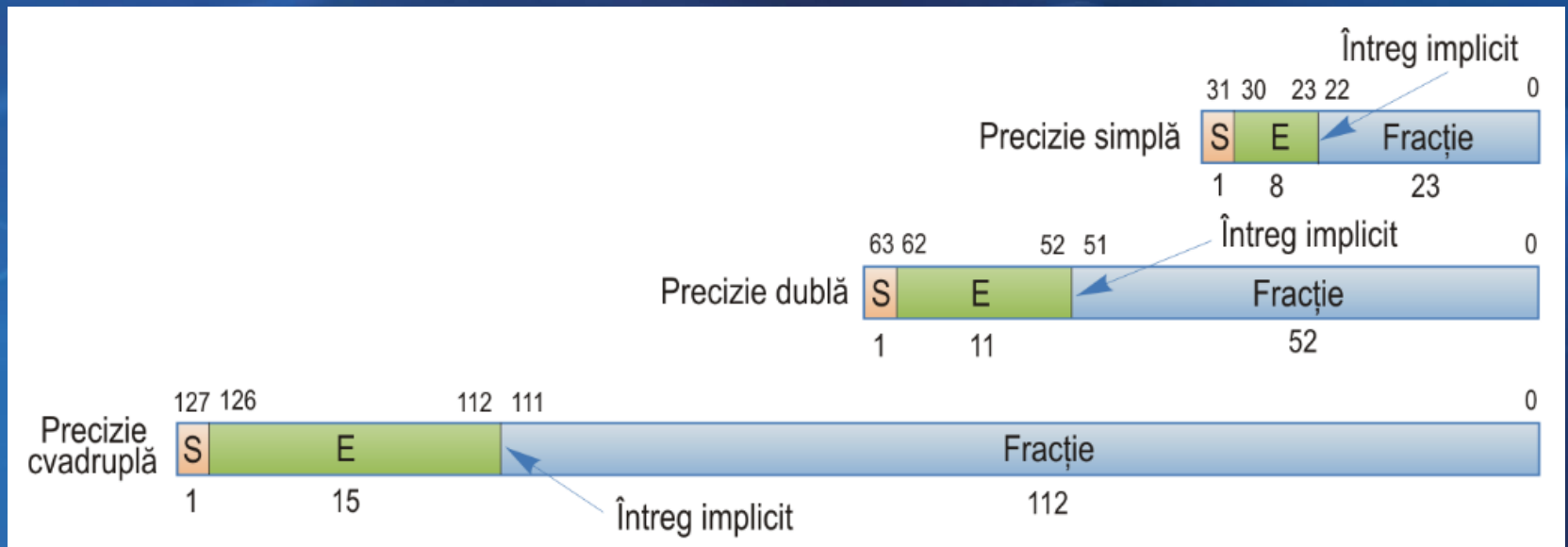
Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (6)

- Parametrii formatelor în VM
 - Determină setul finit de numere în VM care poate fi reprezentat cu un format
 - b – baza (2 sau 10)
 - p – precizia (numărul de cifre ale mantisei)
 - $emax$ – exponentul real maxim
 - $emin$ – exponentul real minim
 - $emin = 1 - emax$ pentru toate formatele

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (7)

Format →	binary32	binary64	binary128	decimal64	decimal128
Nume	Precizie simplă	Precizie dublă	Precizie cvadruplă		
b	2	2	2	10	10
p	23+1	52+1	112+1	16	34
emin	-126	-1022	-16382	-383	-6143
emax	127	1023	16383	384	6144
Deplasament	127	1023	16383	398	6176

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (8)



Formatele binare cu precizie simplă, precizie dublă și precizie cvadruplă

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (9)

- Pentru formatele binare:
 $m = 1, \text{Fracție} \ (1,0 \leq m < 2,0)$
- Valoarea unui număr într-un format binar:
 - $N_S = (-1)^s \times 2^{E-127} \times m$
 - $N_D = (-1)^s \times 2^{E-1023} \times m$
 - $N_Q = (-1)^s \times 2^{E-16383} \times m$
- Gama numerelor reprezentabile
 - Precizie dublă: $2,23 \times 10^{-308} \dots 1,79 \times 10^{308}$
 - Precizie cvadruplă: $3,37 \times 10^{-4932} \dots 1,18 \times 10^{4932}$

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (10)

Exemplul 2.5

- Reprezentarea binară a numărului $-0,75$ în precizie simplă
- $-0,75 = -0,11_2 = -0,11 \times 2^0 = -1,1 \times 2^{-1}$
- Exponentul real: $e = -1$
- Exponentul deplasat: $E = e + 127 = -1 + 127 = 126 = 7E_{16} = 0111\ 1110_2$
- Reprezentarea numărului $-0,75$:

31	30	23	22	0
1	0111 1110	1000 0000 0000 0000 0000 000		

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (11)

● Exemplul 2.6

- Numărul zecimal corespunzător cuvântului următor în formatul cu precizie simplă:

31	30	23	22	0
1	1000 0001	0100 0000 0000 0000 0000 000		

- $s = 1$
- $E = 81_{16} = 129$; $e = E - 127 = 129 - 127 = 2$
- $\text{Fracție} = 1 \times 2^{-2} = 0,25$; $m = 1,25$
- $N = (-1)^1 \times 1,25 \times 2^2 = -1,25 \times 4 = -5,0$

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (12)

- Standardul permite reprezentarea unor valori speciale
 - Pentru acestea sunt rezervate valoarea minimă și cea maximă a exponentului
- Valoarea zero
 - $E = 0$, Frație = 0
 - Două reprezentări pentru valoarea 0: +0, -0
 - Bitul ascuns de la stânga virgulei binare este implicit 0 în loc de 1

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (13)

- Numere nenormalizate
 - “*Subnormal*” (“*denormal*”)
 - $E = 0$; Frație $\neq 0$; bitul ascuns este 0
 - Un număr nenormalizat este generat prin tehnica numită depășire inferioară graduală
 - Exemplu:
 - Format cu precizie simplă $\rightarrow$ exponentul real minim este -126
 - Rezultatul necesită un exponent egal cu -129 pentru a se obține un număr normalizat

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (14)

- Mantisa este deplasată la dreapta și exponentul este incrementat

Operație	s	Exponent	Mantisă
Rezultat normalizat	0	-129	1,01011100000...0
Deplasare mantisă, e++	0	-128	0,10101110000...0
Deplasare mantisă, e++	0	-127	0,01010111000...0
Deplasare mantisă, e++	0	-126	0,00101011100...0
Rezultat nenormalizat	0	-126	0,00101011100...0

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (15)

● Valoarea infinit

- E = valoarea maximă pentru formatul utilizat, Frație = 0
- Două reprezentări pentru infinit: $+\infty$, $-\infty$
- Valoarea infinit se poate utiliza ca operand:
$$\infty + n = \infty ; n / \infty = 0 ; n / 0 = \infty$$
- Utilizatorul poate decide dacă va trata depășirea superioară ca o condiție de eroare

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (16)

- **NaN** (*Not a Number*)
 - S-a prevăzut pentru indicarea unor excepții
 - Operații nedefinite: ∞/∞ , $\infty - \infty$, $\infty * 0$, $0/\infty$, $0/0$
 - Extragerea rădăcinii pătrate dintr-un număr negativ
 - E = valoarea maximă, Frație $\neq 0$
 - Două tipuri de valori **NaN**:
 - **qNaN**: nu se semnalează o excepție (“quiet” NaN)
 - **sNaN**: este semnalată o excepție (“signaling” NaN)
 - Atunci când argumentul unei operații este **NaN**, rezultatul trebuie să fie **NaN**

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (17)

- Formate zecimale
 - Prevăzute pentru aplicații comerciale
 - Un număr poate avea reprezentări multiple
→ sunt admise zerouri în pozițiile c.m.s. ale mantisei
 - **Cohortă**: set de reprezentări ale unui număr
 - Membrii unei cohorte se pot deosebi între ei
 - Operațiile au exponenți preferați pentru rezultatele lor
 - Se păstrează alinierea virgulei zecimale

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (18)

● Excepții

- Depășire inferioară
- Depășire superioară
- Împărțire la zero
- Rezultat inexact: rezultatul trebuie rotunjit
- Operație invalidă: apare în cazul unor operații ca $0/0$ sau $\infty - \infty$
- Implicit, la apariția unei excepții este setat un indicator și calculele continuă

Standardul IEEE 754 pentru reprezentarea în virgulă mobilă (19)

- **Avantaje** ale standardului IEEE 754:
 - Facilitează scrierea unor **programe portabile**
 - Definește condiții pentru scrierea **programelor reproductibile** → produc același rezultat cu diferite implementări ale unui limbaj
- **Dezavantaje:**
 - Conține specificații opționale
 - Implementare lentă a depășirii inferioare graduale comparativ cu setarea la zero
 - Nu se specifică aritmetica pentru valori complexe

Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

Operații cu numere în VM

- Considerăm numerele în VM:
 - $X = b^{E_x} \times m_x$
 - $Y = b^{E_y} \times m_y$
- Operațiile aritmetice elementare

Operație	Rezultat
$X + Y$	$b^{E_y} \times (b^{E_x-E_y} \times m_x + m_y), E_x < E_y$
$X - Y$	$b^{E_y} \times (b^{E_x-E_y} \times m_x - m_y), E_x < E_y$
$X \times Y$	$b^{E_x+E_y} \times (m_x \times m_y)$
$X \div Y$	$b^{E_x-E_y} \times (m_x \div m_y)$

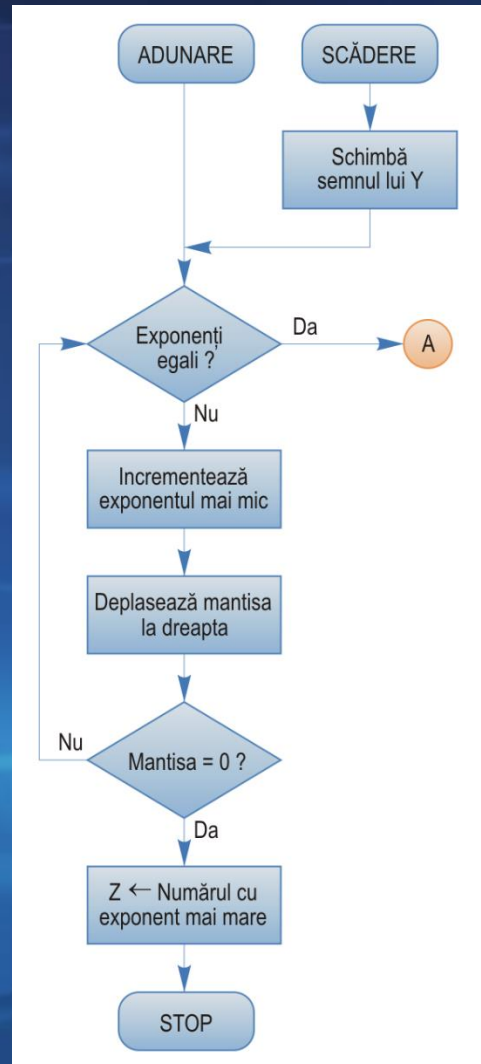
Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

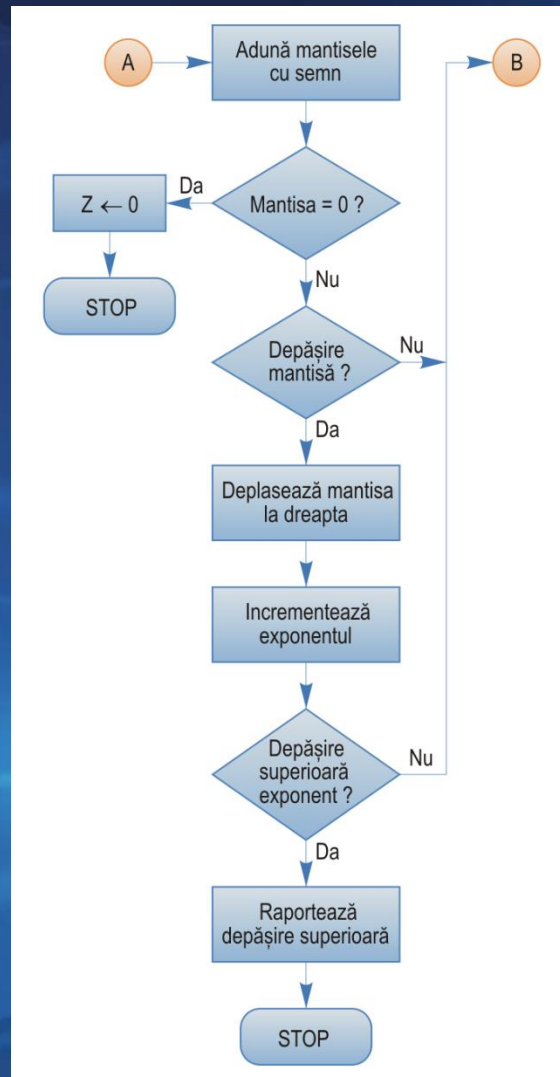
Adunarea și scăderea în VM (1)

- Pentru adunare sau scădere, trebuie să se realizeze **egalizarea exponenților** numerelor
 - Compararea mărimii exponenților
 - Alinierea mantisei numărului cu exponentul mai mic
- **Etapele algoritmului**
 - Alinierea mantiselor
 - Adunarea sau scăderea mantiselor
 - Normalizarea rezultatului
 - Rotunjirea rezultatului

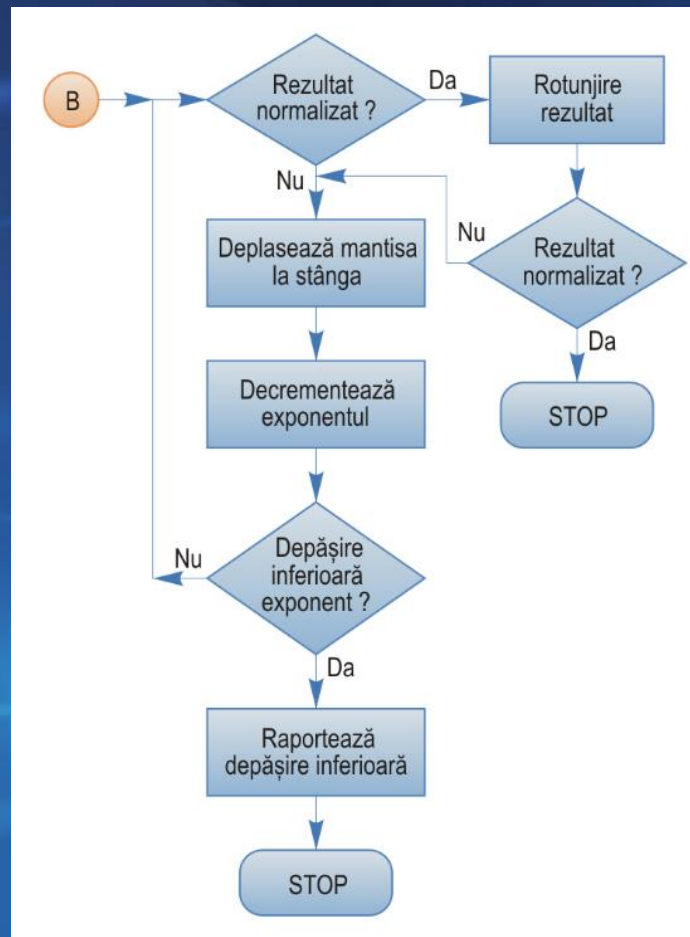
Adunarea și scăderea în VM (2)



Adunarea și scăderea în VM (3)



Adunarea și scăderea în VM (4)



Adunarea și scăderea în VM (5)

● Exemplul 2.7

- $X = 0,625_{10}$; $Y = -0,4375_{10}$
- Adunarea în binar, precizia mantisei $p = 4$
- Exponentul real $e \in [-126, +127]$
- Conversia în binar
 - $X = 0,625_{10} = 0,101 \times 2^0 = 1,010 \times 2^{-1}$
 - $Y = -0,4375_{10} = -0,0111 \times 2^0 = -1,110 \times 2^{-2}$
- Alinierea mantiselor
 - $Y = -1,110 \times 2^{-2} = -0,111 \times 2^{-1}$

Adunarea și scăderea în VM (6)

- Adunarea mantiselor (cu semn)

$$1,010 \times 2^{-1} -$$

$$\underline{0,111 \times 2^{-1}}$$

$$0,011 \times 2^{-1}$$

- Normalizarea rezultatului

- $Z = 0,011 \times 2^{-1} = 1,100 \times 2^{-3}$

- $e_Z = -3, -126 \leq -3 \leq +127$

- Rotunjirea rezultatului → nu este necesară

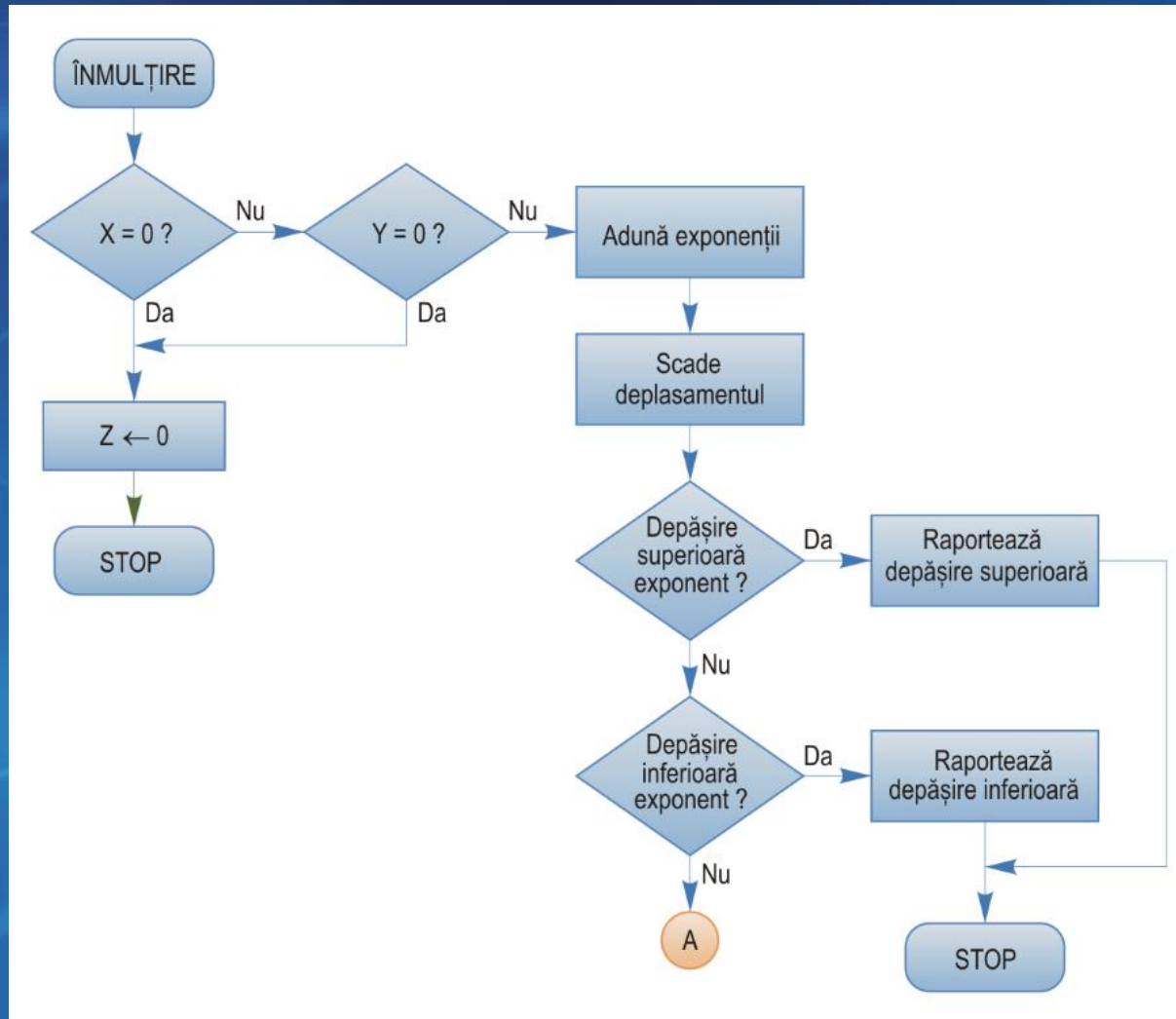
- $Z = 1,100 \times 2^{-3} = 0,0011 = 3/16_{10} = 0,1875_{10}$

- $X + Y = 0,625_{10} - 0,4375_{10} = 0,1875_{10}$

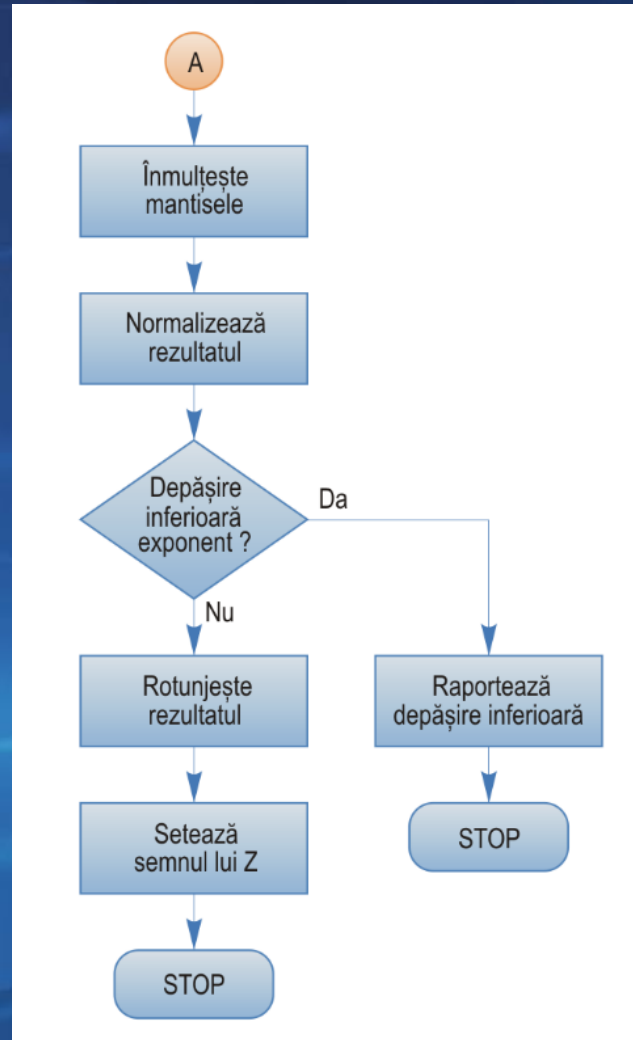
Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

Înmulțirea și împărțirea în VM (1)



Înmulțirea și împărțirea în VM (2)



Înmulțirea și împărțirea în VM (3)

- **Normalizarea** rezultatului după înmulțire
 - Presupunem că produsul se obține în registrele A (partea c.m.s) și Q , de câte p biți
 - **Cazul 1:** Bitul A_{p-1} este 0
 - Se deplasează registrul A la stânga cu o poziție
 - Se introduce bitul Q_{p-1} în poziția A_0
 - **Cazul 2:** Bitul A_{p-1} este 1
 - Nu este necesară deplasarea registrului A
 - Se incrementează exponentul produsului cu 1

Înmulțirea și împărțirea în VM (4)

● Exemplul 2.8

- $X = 0,75_{10}$; $Y = -5,0_{10}$
- Înmulțirea în binar, precizia mantisei $p = 4$
- Exponentul real $e \in [-126, +127]$
- Conversia în binar
 - $X = 0,75_{10} = 0,110 \times 2^0 = 1,100 \times 2^{-1}$
 - $Y = -5,0_{10} = -101,0 \times 2^0 = -1,010 \times 2^2$
- Adunarea exponenților (reali)
 - $e_z = -1 + 2 = +1, -126 \leq +1 \leq +127$

Înmulțirea și împărțirea în VM (5)

- Înmulțirea mantiselor
 - $1100 \times 1010 = 0111\ 1000$
 - $A = 0111$, $Q = 1000$
- Normalizarea produsului
 - $A_3 = 0$, cazul 1: $A = 1111$
 - $Z = 1,111 \times 2^1$
- Rotunjirea produsului nu modifică rezultatul
- Stabilirea semnului produsului
 - $Z = -1,111 \times 2^1 = -11,11 = -(3 + \frac{3}{4})_{10} = -3,75_{10}$
 - $X * Y = 0,75_{10} \times (-5,0_{10}) = -3,75_{10}$

Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

Rotunjirea rezultatelor în VM (1)

- Numerele în VM sunt **aproximări** ale unor **numere reale** pe care nu le pot reprezenta exact
 - Există o infinitate de numere reale într-un anumit interval
- Este necesară o **metodă de rotunjire**
- Pentru executarea cu acuratețe a operațiilor aritmetice, circuitele trebuie să utilizeze **cifre suplimentare**

Rotunjirea rezultatelor în VM (2)

- Fie p numărul corespunzător de cifre ale mantisei pentru o precizie dată
- Cifra de gardă g
 - Este plasată la dreapta primelor p cifre ale mantisei pentru a evita pierderea cifrelor
- Cifra de rotunjire r
 - Dacă rezultatul este normalizat, dar există cifre diferite de zero la dreapta mantisei, numărul trebuie rotunjit

Rotunjirea rezultatelor în VM (3)

- Este plasată la dreapta cifrei de gardă
- După deplasarea la stânga a rezultatului pentru normalizare, acesta poate fi rotunjit
- **Moduri de rotunjire** specificate de standardul IEEE 754
 - Rotunjire **spre $+\infty$** (în sus)
 - Rotunjire **spre $-\infty$** (în jos)
 - Rotunjire **spre zero** (trunchiere)
 - Rotunjire **la cel mai apropiat număr par**

Rotunjirea rezultatelor în VM (4)

- Ultimul mod: prevăzut pentru situațiile în care numărul real se află exact la jumătatea intervalului dintre două reprezentări în VM
 - Dacă bitul c.m.p.s. este impar, se adună 1
 - Dacă bitul este par, numărul este trunchiat
 - Pentru a se obține rotunjirea corectă în ultimul caz, este necesar un alt tip de cifră suplimentară → **bit de colectare s** (*sticky bit*)
 - Bitul **s** este setat dacă există biți $\neq 0$ la dreapta bitului **r**

Rotunjirea rezultatelor în VM (5)

- Amplasarea cifrelor suplimentare
 - Considerăm înmulțirea mantiselor, cu registrele A și Q de câte 4 biți



- $g := Q_3$
- $r := Q_2$
- $s := Q_1 + Q_0$



Rotunjirea rezultatelor în VM (6)

- Actualizarea cifrelor g, r, s la adunare
 - Alinierea mantiselor
 - $g :=$ bitul c.m.s. deplasat în afara registrului
 - $r :=$ următorul bit deplasat în afara registrului
 - $s :=$ SAU logic între ceilalți biți
 - Adunarea mantiselor, dacă apare depășire
 - $r :=$ bitul 0 al mantisei înainte de deplasare
 - $s := g + r + s$
 - Dacă rezultatul este deja normalizat
 - $r := g, s := r + s$

Rotunjirea rezultatelor în VM (7)

- Normalizarea rezultatului
 - La prima deplasare la stânga, se introduce bitul g în poziția bitului 0, apoi se introduc biți de 0
 - La o singură deplasare, r și s nu se modifică
 - La mai multe deplasări, $r := 0$, $s := 0$

• Reguli pentru modurile de rotunjire

Mod de rotunjire	Rezultat ≥ 0	Rezultat < 0
Spre $-\infty$		+1 dacă $r + s = 1$
Spre $+\infty$	+1 dacă $r + s = 1$	
Spre 0		
La cel mai apropiat nr. par	+1 dacă $r \cdot Z_{p-1} = 1$ sau $r \cdot s = 1$	+1 dacă $r \cdot Z_{p-1} = 1$ sau $r \cdot s = 1$

Rotunjirea rezultatelor în VM (8)

● Exemplul 2.9

- $X = 0,28125_{10}$; $Y = 1,875_{10}$
- Adunarea în binar, precizia mantisei $p = 4$
- Conversia în binar
 - $X = 0,28125_{10} = 1,001 \times 2^{-2}$
 - $Y = 1,875_{10} = 1,111 \times 2^0$
- Alinierea mantiselor
 - $X = 1,001 \times 2^{-2} = 0,010\ 0100 \times 2^0$
 - $g = 0$, $r = 1$, $s = 0$

Rotunjirea rezultatelor în VM (9)

- Adunarea mantiselor

$$\begin{array}{r} 0,010 \times 2^0 + \\ \underline{1,111 \times 2^0} \\ 10,001 \times 2^0 \end{array}$$

- Depășire: deplasare la dreapta, e_z++

- $Z = 1,000 \times 2^1$ ($e_z = 0 + 1 = 1$)

- $r = 1, s = g + r + s = 0 + 1 + 0 = 1$

- Rotunjirea rezultatului în sus

- $Z = 1,00\underline{1} \times 2^1 = 10,01 = 2,25_{10}$

- $X + Y = 0,28125_{10} + 1,875_{10} = 2,15625_{10}$

Rotunjirea rezultatelor în VM (10)

- Actualizarea cifrelor g, r, s la înmulțire
 - Înmulțirea mantiselor
 - După înmulțire, se setează biții g, r și s în funcție de conținutul registrului Q
 - Cazul 1: Bitul A_{p-1} este 0
 - Se deplasează registrul A la stânga cu o poziție și se introduce bitul g din registrul Q în poziția A_0
 - Cazul 2: Bitul A_{p-1} este 1
 - $s := s + r$
 - $r := g$

Rotunjirea rezultatelor în VM (11)

● Exemplul 2.10

- $X = 14,5_{10}$; $Y = 5,0_{10}$
- Înmulțirea în binar, precizia mantisei $p = 4$
- Conversia în binar
 - $X = 14,5_{10} = 1110,1 \times 2^0 = 1,110 \times 2^3$
 - $Y = 5,0_{10} = 101,0 \times 2^0 = 1,010 \times 2^2$
- Adunarea exponenților (reali)
 - $e_z = 3 + 2 = 5$

Rotunjirea rezultatelor în VM (12)

- Înmulțirea mantiselor
 - $1110 \times 1010 = 1000\ 1100$
 - $A = 1000$, $g = 1$, $r = 1$, $s = 0$
- Normalizarea produsului
 - $A_3 = 1$, cazul 2: $s = s + r = 0 + 1 = 1$; $r = g = 1$
 - $e_z = e_z + 1 = 5 + 1 = 6$
 - $Z = 1,000 \times 2^6$
- Rotunjirea produsului în sus
 - $Z = 1,00\underline{1} \times 2^6 = 100\ 1000 = (4 \times 16 + 8)_{10} = 72_{10}$
 - $X * Y = 14,5_{10} \times 5,0_{10} = 72,5_{10}$

Rotunjirea rezultatelor în VM (13)

- Pentru operațiile zecimale este specificat un mod de rotunjire suplimentar
 - Necesar pentru a efectua rotunjirea în același mod ca și la calculele financiare
- Rotunjire la cel mai apropiat număr, cu îndepărtare de la zero
 - Dacă numărul real se află exact la jumătatea intervalului dintre două reprezentări în VM, este rotunjit la valoarea cea mai apropiată cu mărimea mai mare

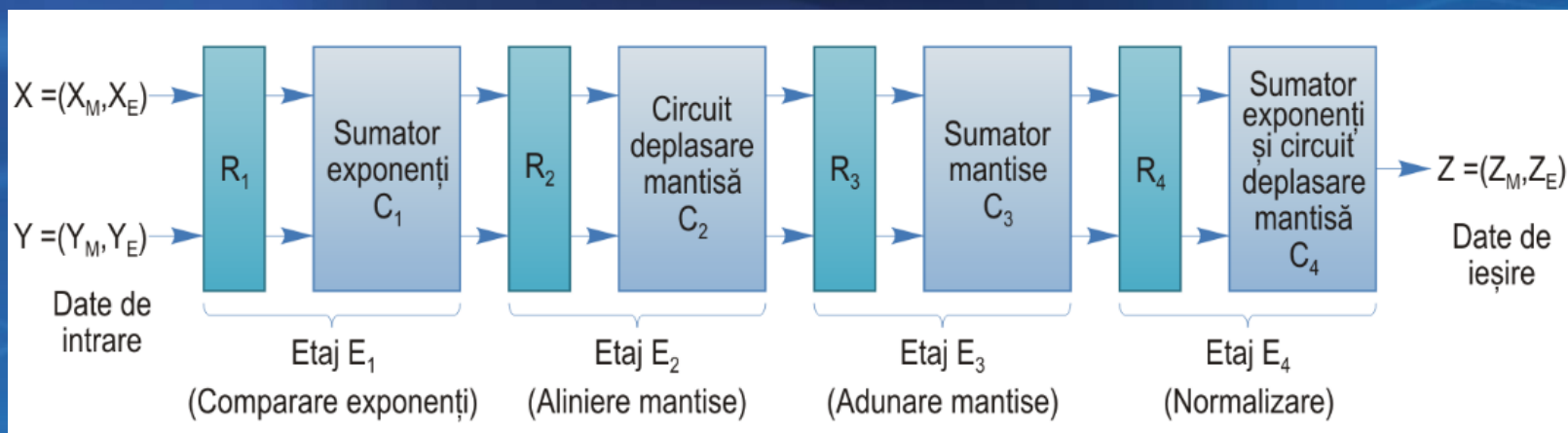
Numere și operații în virgulă mobilă

- Numere și operații în virgulă mobilă
 - Reprezentarea numerelor în virgulă mobilă
 - Standardul IEEE 754 pentru reprezentarea în virgulă mobilă
 - Operații cu numere în virgulă mobilă
 - Adunarea și scăderea în virgulă mobilă
 - Înmulțirea și împărțirea în virgulă mobilă
 - Rotunjirea rezultatelor în virgulă mobilă
 - Circuite *pipeline* pentru operații în virgulă mobilă

Circuite *pipeline* pentru operații în VM (1)

- Operațiile în VM pot fi implementate printr-un circuit *pipeline*
 - Operațiile pot fi descompuse
- **Exemplu:** Adunarea a două numere normalizate în virgulă mobilă, X și Y
 - Compararea exponenților
 - Alinierea mantiselor
 - Adunarea mantiselor
 - Normalizarea rezultatului

Circuite *pipeline* pentru operații în VM (2)



Sumator în VM cu patru etaje

- E_1 : Se compară X_E și Y_E prin scădere
- E_2 : Mantisa numărului cu exponentul mai mic (X_M) se deplasează la dreapta $\rightarrow (X'_M, Y_E) = (X_M, X_E)$

Circuite *pipeline* pentru operații în VM (3)

- E_3 : Se adună mantisele X'_M și Y_M
- E_4 : Se normalizează rezultatul prin contorizarea numărului cifrelor de 0 de la începutul mantisei și deplasarea la stânga
- Etape de proiectare pentru un circuit aritmetic *pipeline*:
 - Găsirea unui **algoritm secvențial** cu mai **multe etape**
 - Etapele trebuie să fie echilibrate
 - Plasarea unor registre buffer între etaje

Rezumat (1)

- Pentru numere fracționare, este avantajoasă reprezentarea în **virgulă mobilă (VM)**
- Reprezentarea în VM cu **exponent deplasat** prezintă mai multe avantaje
- Standardul **IEEE 754** a fost elaborat pentru a facilita portabilitatea programelor
 - Permite obținerea acelorași rezultate, indiferent de implementare
 - Definește **formate de bază** (binare, zecimale), cu **precizie extinsă** și cu **precizie extensibilă**

Rezumat (2)

- Definește reprezentarea unor **valori speciale**: zero, infinit, NaN, numere nenormalizate
- Adunarea și scăderea în VM necesită **alinierea mantiselor (egalizarea exponenților)**
- Pentru creșterea preciziei, circuitele aritmetice în VM trebuie să utilizeze **cifre suplimentare**: de gardă, de rotunjire, de colectare
- De obicei, pentru implementarea circuitelor aritmetice în VM se utilizează **tehnica pipeline**
 - Sunt plasate registre buffer între etajele circuitului

Noțiuni, cunoștințe (1)

- Reprezentare în VM cu exponent deplasat
- Reprezentare normalizată în VM
- Formate specificate de standardul IEEE 754 pentru numere în VM
- Valori speciale specificate de standardul IEEE 754 pentru numere în VM
- Numere nenormalizate specificate de standardul IEEE 754 pentru numere în VM
- Tehnica de depășire inferioară graduală
- Valoarea specială NaN

Noțiuni, cunoștințe (2)

- Excepții specificate de standardul IEEE 754 pentru numere în VM
- Avantaje/dezavantaje ale standardului IEEE 754
- Algoritmul de adunare/scădere în VM
- Algoritmul de înmulțire/împărțire în VM
- Cifre de gardă și de rotunjire
- Moduri de rotunjire specificate de standardul IEEE 754 pentru numere în VM
- Schema bloc a unui sumator *pipeline* în VM