II. REPREZENTAREA NUMERELOR ÎN CALCULATOR. ARITMETICA BINARĂ

2.1. Reprezentarea numerelor în calculator

- Reprezentarea numerelor în calculatoarele numerice se face pe baza sistemului de numeraţie binar
- Numere pozitive şi negative → reprezentare semn?
  - numere fără semn - reprezentare în binar sau în cod binar-zecimal
  - numere cu semn - asociat pe poziţia cea mai semnificativă un **bit special de semn**
  - convenţie:
    - semnul plus - cifra 0
    - semnul minus - cifra 1
  - un număr binar de “n” biţi, cu semn are “n+1” biţi

2.1. Reprezentarea numerelor în calculator

- Numere:
  - întregi
  - fracţionare
- Poziţia virgulei la numere fracţionare → determină reprezentarea → poziţie fixă sau variabilă a virgulei:
  - reprezentare în virgulă fixă
  - reprezentare în virgulă mobilă (flotantă)

2.1.1. Reprezentarea numerelor în virgulă fixă

- Calculatoarele pot opera cu numere de lungime fixă
- Numărul de cifre (de exemplu, 32 sau 64 de poziţii binare) determinat de numărul de celule din care sunt realizate registrele utilizate
- Poziţia virgulei
  - se stabileşte iniţial la proiectare
  - nu se mai schimbă
  - nu se realizează fizic, dar localizarea ei trebuie cunoscută

2.1.1. Reprezentarea numerelor în virgulă fixă

- Blocuri aritmetice ale calculatoarelor care lucrează în virgulă fixă:
  - virgula este în faţa cifrei celei mai semnificative
  - numerele sunt deci subunitare (numerele reale sunt transformate în prealabil în acest sens)
- Indicarea semnului → realizată prin mai multe tehnici ⇒ moduri diferite de reprezentare:
  - mărime şi semn
  - complement faţă de 2 (cod complementar)
  - complement faţă de 1 (cod invers)

Complement

Definiţii:

- $\overline{(\mathrm{N})}_b = b^n - (\mathrm{N})_b$
- $\overline{\overline{(\mathrm{N})}}_b = b^n - (\mathrm{N})_b - b^{-m}$
  - $\overline{(\mathrm{N})}_b$ = complement faţă de baza "b" a numărului $(\mathrm{N})_b$
  - $\overline{\overline{(\mathrm{N})}}_b$ = complement faţă de "b-1" a numărului $(\mathrm{N})_b$
  - n = nr. de cifre ale părţii întregi ale numărului N
  - m = nr. de cifre ale părţii fracţionare ale numărului N
- În binar, $N = 2^n - 1$ = cel mai mare număr binar de "n" cifre care poate fi reprezentat $\Rightarrow 2^n$ necesar pentru complement nu se poate reprezenta $\Rightarrow 2^n$ se echivalează cu numărul 0

Complement

Exemple generale:

- $N_1 = (123{,}45)_{10}$ cu $n = 3$, $m = 2$
  - $(\overline{N_1})_{10} = 10^n - N_1 = 10^3 - 123{,}45 = 876{,}55$
  - $(\overline{\overline{N_1}})_{10} = 10^n - N_1 - 10^{-m} = 10^3 - 123{,}45 - 10^{-2} = 876{,}54$
- $N_2 = (1101{,}011)_2$ cu $n = 4$, $m = 3$
  - $(\overline{N_2})_2 = 2^n - N_2 = 2^4 - 1101{,}011 = 0010{,}101$
  - $(\overline{\overline{N_2}})_2 = 2^n - N_2 - 2^{-m} = 2^4 - 1101{,}011 - 2^{-3} = 0010{,}100$

Complement

Determinarea complementului faţă de 2 → există 3 procedee:

- $(\overline{N})_2 = 0 - N$
- $(\overline{N})_2 = \overline{\overline{N}} + 2^{-n}$
- pornind de la dreapta spre stânga se păstrează neschimbate cifrele egale cu 0, inclusiv prima cifră egală cu 1, după care toate celelalte cifre se inversează
  - n = numărul de cifre ale numărului

Complement

Determinarea complementului faţă de 1 → există 3 procedee:

- $\overline{\overline{(N)}}_2 = 0 - N - 2^{-n}$
- $\overline{\overline{(N)}}_2 = \overline{N} - 2^{-n}$
- se inversează fiecare cifră în parte

Complement

- Exemplu de determinare a complementului faţă de 2
- Numărul $(N)_2 = 101011$

1. $$(\overline{N})_2 = \begin{array}{r} 000000\ - \\ \underline{101011}\ = \\ 010101 \end{array}$$

2. $$(\overline{N})_2 = \begin{array}{r} 010100\ + \\ \underline{000001}\ = \\ 010101 \end{array}$$

3. $$(\overline{N})_2 = \underbrace{01010}_{\text{s-au inversat}}1$$

rămâne neschimbat

Complement

- Exemplu de determinare a complementului faţă de 1
- Numărul $(N)_2 = 101011$

1. $(\overline{\overline{N}})_2 = 000000$ -

$$\begin{array}{r} 000000\ - \\ 101011\ - \\ 000001\ = \\ \hline 010100 \end{array}$$

2. $(\overline{\overline{N}})_2 =$

$$\begin{array}{r} 010101\ - \\ 000001\ = \\ \hline 010100 \end{array}$$

3. $(\overline{\overline{N}})_2 = 010100 \rightarrow$ am inversat cifrele (0 cu 1, 1 cu 0)

Reprezentarea prin mărime şi semn

- Relaţia de reprezentare prin mărime şi semn:

$$N = a_n 2^n + \sum_{i=-m}^{n-1} a_i 2^i$$

- a_n = bit de semn
  - N pozitiv $\Rightarrow$ $a_n = 0$
  - N negativ $\Rightarrow$ $a_n = 1$
- a_i = cifrele binare ale numărului N

Reprezentarea prin mărime şi semn

- Avantaj - asemănătoare cu scrierea manuală
- Dezavantaje pentru realizarea calculelor aritmetice:
  - adunarea şi scăderea depind şi de semnele numerelor
  - este necesară examinarea semnului înaintea operaţiei
  - sunt necesare blocuri diferite pentru adunare şi scădere
- Exemplu:
  - $+6 = \underset{\text{semn}}{0}\underbrace{0110}_{\text{cifre număr}}$ $\quad$ $-6 = \underset{\text{semn}}{1}\underbrace{0110}_{\text{cifre număr}}$

Reprezentarea prin complement faţă de 2

- Relaţiile de reprezentare:

$$N = 0 \cdot 2^n + \sum_{i=-m}^{n-1} a_i 2^i \qquad \text{pentru N > 0}$$

$$N = -1 \cdot 2^n + \sum_{i=-m}^{n-1} \overline{a}_i 2^i + 2^{-m} \quad \text{pentru N < 0}$$

- $\overline{a_i}$ = 1 - a_i este complementul faţă de 1 al cifrei a_i
- Exemplu:
  - +6 = 00110
  - - 6 = 11010 (1010 = complementul faţă de 2 al lui 6)

Reprezentarea prin complement faţă de 1

- Relaţiile de reprezentare:

$$N = 0 \cdot 2^n + \sum_{i=-m}^{n-1} a_i 2^i \qquad \text{pentru N > 0}$$

$$N = -1 \cdot 2^n + \sum_{i=-m}^{n-1} \overline{a}_i 2^i \qquad \text{pentru N < 0}$$

- $\overline{a_i}$ = 1 - a_i este complementul faţă de 1 al cifrei a_i
- Exemplu:
  - +6 = 00110
  - \- 6 = 11001 (1001 = complementul faţă de 1 al lui 6)

2.1.1. Reprezentarea numerelor în virgulă fixă

- Un număr N în virgulă fixă se poate scrie:

$$N = a_0 2^0 + N^*$$

  - a_0 = bit de semn
  - N* are semnificaţiile:
    - mărime şi semn $N^* = \sum_{i=1}^{n} a_i 2^{-i}$
    - N < 0 în complement faţă de 2 $N^* = \sum_{i=1}^{n} \overline{a}_i 2^{-i} + 2^{-n}$
    - N < 0 în complement faţă de 1 $N^* = \sum_{i=1}^{n} \overline{a}_i 2^{-i}$
      - a_i = cifrele numărului
      - n = numărul de cifre din dreapta virgulei
      - $\overline{a}_i = 1 - a_i$

2.1.1. Reprezentarea numerelor în virgulă fixă

- Avantaje dacă virgula se plasează după prima poziţie binară:
  - N* fiind subunitar poziţia virgulei este aceeaşi după înmulţirea binară
  - înmulţirea nu va duce niciodată la depăşirea limitei superioare a gamei de reprezentare a numerelor
  - această plasare poate fi uşor memorată

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Pentru numere foarte mari sau foarte mici, cu grad de precizie ridicat
- Reprezentarea unui număr - prin mantisă M şi exponent E
- **Exponent** - indică ordinul de mărime al numărului printr-o putere
- **Mantisa** determină mărimea (valoare) exactă a numărului în cadrul ordinului respectiv

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Exemplu - o reprezentare pe 32 de biţi:

SE							SM		
0	1	2	3	4	5	6	7	8 ……….	31

  - bitul 0 = SE - semn exponent
  - biţii 1- 6 = exponent E
  - bitul 7 = SM - semn mantisă
  - biţii 8-31 = mantisă M
- Numărul +12,34 se poate reprezenta în binar:
  - +1100,01010111 sau +0,110001010111 x 2^{+4} sau +0,001100010101110 x 2^{+6}
  - Reprezentarea pe 32 de biţi poate fi:
    - 0 000100 0 110001010111000000000000 sau
    - 0 000110 0 001100010101110000000000

2.1.2. Reprezentarea numerelor în virgulă mobilă

- În calculatoare se utilizează reprezentarea numai cu semn la mantisă, nu şi la exponent ⇒ se foloseşte mărimea numită "Caracteristică"
- **Caracteristica** numărului N: C = E + deplasament
  - E = exponentul numărului (putere a lui 16)
  - deplasamentul ales ca să rezulte întotdeauna o valoare pozitivă (ex.: 64 sau 128)
- Valoarea exponentului E = (C - deplasament)
  - avantaje
    - este numai pozitiv ⇒ operaţii simplificate
    - cifra 0 reprezentată la fel cu reprezentarea în virgulă fixă ⇒ tratare similară
  - dezavantaj - există o operaţie de scădere în plus!

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Reprezentare în simplă precizie, de exemplu:

a)

S	C	M
0	1 … 7	8 … 31

- Reprezentare în dublă precizie, de exemplu:

b)

S	C	M
0	1 … 7	8 … 63

- Exemplu:
  - Pentru caracteristica C = 7 biţi $\Rightarrow$ numerele pot fi între 0 şi 2^7-1 $\Rightarrow$ $0 \leq C \leq 127$
  - Dacă deplasamentul = 64 $\Rightarrow$ exponentul E = C – 64, deci $-64 \leq E \leq 63$

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Forma normalizată - bitul cel mai semnificativ al mantisei = 1
  - acest bit nu se mai memorează ⇒ mantisa câştigă un bit semnificativ în plus
  - simplificare operaţii
  - creştere precizie
- Probleme cu reprezentarea valorii 0, care nu poate fi normalizată ⇒ valoarea 0 are o reprezentare specială

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Exemplu: gama numerelor reprezentate în complement faţă de 2, în cuvinte de 32 de biţi:
  - numere pozitive între: $0{,}5 \bullet 2^{-128}$ şi $(1 - 2^{-24}) \bullet 2^{127}$
  - numere negative între: $-(1 - 2^{-24}) \bullet 2^{127}$ şi $-0{,}5 \bullet 2^{-128}$
  - 5 regiuni necuprinse în aceste domenii $\rightarrow$ depăşire
    - depăşire inferioară negativă: $< -(1 - 2^{-24}) \bullet 2^{127}$
    - depăşire superioară negativă: $> -0{,}5 \bullet 2^{-128}$
    - zero
    - depăşire inferioară pozitivă: $< 0{,}5 \bullet 2^{-128}$
    - depăşire superioară pozitivă: $> (1 - 2^{-24}) \bullet 2^{127}$
  - coprocesoarele matematice şi unităţile de calcul în virgulă mobilă au mecanisme speciale pentru detectarea, semnalizarea şi tratarea depăşirilor

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Compromis între dimensiunea mantisei şi a exponentului
  - mantisa de dimensiune mai mare $\Rightarrow$ creşte precizia numerelor care pot fi reprezentate
  - exponentul de dimensiune mai mare $\Rightarrow$ creşte domeniul numerelor care pot fi reprezentate
- Creşterea numărului de biţi pentru reprezentare:
  - creşterea preciziei
  - creşterea domeniului numerelor

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Standardul pentru “floating point”:
  - IEEE 754 din 1985, actualizat în august 2008 și apoi în iulie 2019
- Reprezentarea numerelor în virgulă mobilă
- Operaţii aritmetice în această reprezentare

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Adunarea binară**
  - operaţie modulo 2
    - cifra cu valoarea cea mai mare: 2-1 = 1
    - dacă rezultatul adunării a 2 cifre de rang “i” depăşeşte valoarea 1 $\rightarrow$ apare transport către rangul “i+1”, care se adaugă la suma cifrelor de rang “i+1”
    - transportul de la cifra cea mai semnificativă indică depăşirea capacităţii de reprezentare a rezultatului

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Adunarea binară**

x	y	Transport	Sumă
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Exemplu:

$$\begin{array}{r} 22_{10} = \quad 10110_2 + \\ 19_{10} = \quad 10011_2 \\ \hline 41_{10} = 101001_2 \end{array}$$

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Scăderea binară**
  - 2 cifre de rang “i”; poate să apară împrumut de la rangul “i+1”

x	y	Împrumut	Diferenţă
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

- Exemplu:

$$22_{10} = 10110_2 \; -$$
$$19_{10} = 10011_2$$
$$\overline{\;3_{10} = 00011_2\;}$$

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Înmulţirea binară**
  - prin adunarea repetată de produse parţiale
  - produsul este 1 doar dacă şi deînmulţitul şi înmulţitorul sunt 1

x	y	Produsul
0	0	0
0	1	0
1	0	0
1	1	1

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Înmulţirea binară** - exemplu:

$$
\begin{array}{r}
12_{10} = 1100_2 \ \mathrm{x} \\
6_{10} = 0110_2 \\
\hline
0000 \\
1100 \\
1100 \\
0000 \\
\hline
1001000_2
\end{array}
\quad (64 + 8 = 72_{10})
$$

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Împărţirea binară**
  - nu se poate efectua dacă împărţitorul este egal cu 0!!!
  - trebuie să fie satisfăcută relaţia:

$$X = Q \bullet Y + R$$

    - X = deîmpărţit; Y = împărţitor; Q = cât; R = rest
  - se fac scăderi ale împărţitorului din resturile parţiale
    - dacă restul parţial este:
      - mai mare ca împărţitorul $\Rightarrow$ cifra câtului este 1
      - dacă restul parţial este mai mic decât împărţitorul $\Rightarrow$ cifra câtului este 0

2.2. Aritmetica binară

Operaţii aritmetice cu numere fără semn

- **Împărţirea binară** - exemplu

$147_{10} = 10010011_2$; $11_{10} = 1011_2$

$10010011 : 1011 = 1101_2$ cât = 13_{10}

$$\begin{array}{l}
\underline{1011} \\
\ \ 1110 \\
\ \ \underline{1011} \\
\ \ \ \ 1111 \\
\ \ \ \ \underline{1011} \\
\ \ \ \ \ \ 100_2
\end{array}$$

rest = 4_{10}

2.2. Aritmetica binară

Operaţii aritmetice cu numere în virgulă fixă

- **Adunarea numerelor reprezentate în complement faţă de 2**
  - se adună numerele bit cu bit, inclusiv biţii de semn
  - se ignoră transportul de la biţii de semn
  - dacă rezultatul e negativ apare ca un număr reprezentat în complement faţă de 2
  - Observaţii:
    - dacă rezultatul în valoare absolută este mai mare decât valoarea maximă care poate fi reprezentată $\Rightarrow$ depăşire
    - la adunarea a 2 numere de acelaşi semn apare depăşire dacă şi numai dacă rezultatul are semn contrar semnelor numerelor

2.2. Aritmetica binară

Operaţii aritmetice cu numere în virgulă fixă

- **Adunarea numerelor reprezentate în complement faţă de 2** - exemple

$+9_{10}$	$0\,1001_2$	$+\ 9_{10}$	$0\,1001_2$	
$+5_{10}$	$0\,0101_2$	$+11_{10}$	$0\,1011_2$	
$+14_{10}$	$0\,1110_2$	$+20_{10}$	$1\,0100_2$	rezultat incorect
-9_{10}	$1\,0111_2$	$-\ 9_{10}$	$1\,0111_2$	
-5_{10}	$1\,1011_2$	-11_{10}	$1\,0101_2$	
-14_{10}	$\mathit{1}1\,0010_2$	-20_{10}	$\mathit{1}0\,1100_2$	rezultat incorect
$+7_{10}$	$0\,0111_2$	$-\ 7_{10}$	$1\,1001_2$	
-4_{10}	$1\,1100_2$	$+4_{10}$	$0\,0100_2$	
$+3_{10}$	$\mathit{1}0\,0011_2$	-3_{10}	$1\,1101_2$	

2.2. Aritmetica binară

Operaţii aritmetice cu numere în virgulă fixă

- **Scăderea numerelor reprezentate în complement faţă de 2**
  - se poate executa prin 2 metode:
    - prin scădere directă, dacă se dispune de scăzătoare elementare
    - prin adunarea complementului faţă de 2 a scăzătorului, dacă se dispune de sumatoare elementare
  - pot să apară depăşiri, care trebuie detectate
  - la scăderea unor numere de semne contrare pot să apară depăşiri dacă şi numai dacă rezultatul are acelaşi semn cu scăzătorul

2.2. Aritmetica binară

Operaţii aritmetice cu numere în virgulă fixă

- **Scăderea numerelor reprezentate în complement faţă de 2** - exemplu pentru metoda cu adunare cu complementul faţă de 2 al scăzătorului

D: $+\,7_{10}$	$0\,0111_2$	
S: $-\,4_{10}$	$1\,1100_2$	
$\overline{S}$: $-\,(-\,4)_{10}$	$0\,0100_2$	complement faţă de 2

Atunci:

D: $+\,7_{10}$	$0\,0111_2$
S: $\underline{+\,4_{10}}$	$\underline{0\,0100_2}$
$+11_{10}$	$0\,1011_2$