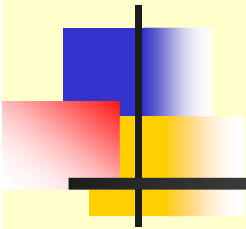


II. REPREZENTAREA NUMERELOR ÎN CALCULATOR. ARITMETICA BINARĂ

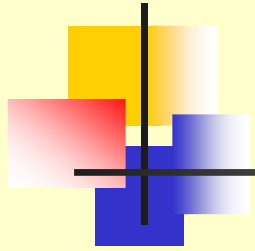




2.1. Reprezentarea numerelor în calculator

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

2.1. Reprezentarea numerelor în calculator



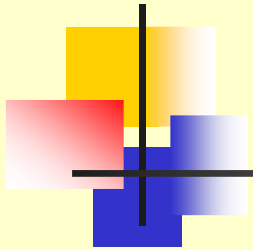
- Numere:
 - întregi
 - fracționare
- **Poziția virgulei** la numere fracționare → determină reprezentarea → poziție fixă sau variabilă a virgulei:
 - reprezentare în virgulă fixă
 - reprezentare în virgulă mobilă (flotantă)

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 poziții binare) determinat de numărul de celule din care sunt realizate registrele utilizate
- Poziția virgulei
 - se stabilește inițial la proiectare
 - nu se mai schimbă
 - nu se realizează fizic, dar localizarea ei trebuie cunoscută

2.1.1. Reprezentarea numerelor în virgulă fixă

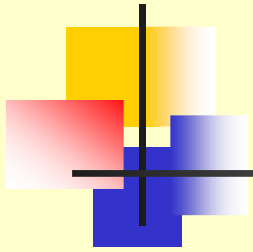
- Blocuri aritmetice ale calculatoarelor care lucrează în virgulă fixă:
 - virgula este în fața cifrei celei mai semnificative
 - numerele sunt deci subunitare (numerele reale sunt transformate în prealabil în acest sens)
- Indicarea semnului → realizată prin mai multe tehnici ⇒ moduri diferite de reprezentare: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



Complement

Definiții:

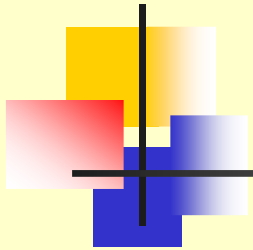
- $\overline{\overline{(N)}}_b = b^n - (N)_b$
- $\overline{(N)}_b = b^n - (N)_b - b^{-m}$
 - $\overline{(N)}_b$ = complement față de baza “b” a numărului $(N)_b$
 - $\overline{\overline{(N)}}_b$ = complement față de “b-1” a numărului $(N)_b$
 - n = nr. de cifre ale părții întregi ale numărului N
 - m = nr. de cifre ale părții fracționare 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 $n = 4, m = 3$
 - $(\overline{N_2})_2 = 2^n - N_2 = 2^4 - 1101,011 = 0010,101$
 - $(\overline{\overline{N_2}})_2 = 2^n - N_2 - 2^{-m} = 2^4 - 1101,011 - 2^{-3} = 0010,100$

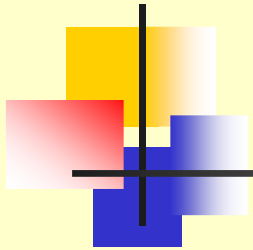


Complement

Determinarea complementului față de 2 \rightarrow 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

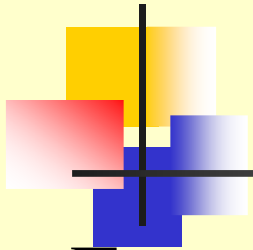


Complement

Determinarea complementului față de 1 \rightarrow 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 față de 2

- Numărul $(N)_2 = 101011$

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

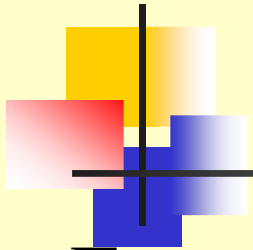
$$\begin{array}{r} 101011 \\ \hline 010101 \end{array}$$

$$2. (\overline{N})_2 = 010100 +$$

$$\begin{array}{r} 000001 \\ \hline 010101 \end{array}$$

$$3. (\overline{N})_2 = 010101$$

$\underbrace{\hspace{2cm}}$ s-au inversat \swarrow rămâne neschimbat



Complement

- Exemplu de determinare a complementului față de 1

- Numărul $(N)_2 = 101011$

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

$$101011 -$$

$$000001 =$$

$$\underline{010100}$$

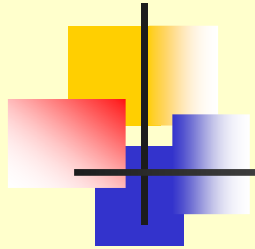
$$2. \overline{\overline{(N)}}_2 = 010101 -$$

$$000001 =$$

$$\underline{010100}$$

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

Reprezentarea prin mărime și semn

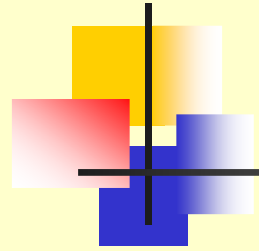


- Relația de reprezentare prin mărime și semn:

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

- 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 și semn



- Avantaj - asemănătoare cu scrierea manuală
- Dezavantaje pentru realizarea calculelor aritmetice:
 - adunarea și scăderea depind și de semnele numerelor
 - este necesară examinarea semnului înaintea operației
 - sunt necesare blocuri diferite pentru adunare și scădere
- Exemplu:

$$\begin{array}{c} \text{■ } +6 = 00110 \\ \swarrow \quad \underbrace{\hspace{1cm}} \\ \text{semn} \quad \text{cifre număr} \end{array}$$

$$\begin{array}{c} \text{■ } -6 = 10110 \\ \swarrow \quad \underbrace{\hspace{1cm}} \\ \text{semn} \quad \text{cifre număr} \end{array}$$

Reprezentarea prin complement față de 2

- Relațiile de reprezentare:

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

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

- $\overline{a_i} = 1 - a_i$ este complementul față de 1 al cifrei a_i

- Exemplu:

- $+6 = 00110$

- $-6 = 11010$ (1010 = complementul față de 2 al lui 6)



Reprezentarea prin complement față de 1

- Relațiile de reprezentare:

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

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

- $\bar{a}_i = 1 - a_i$ este complementul față de 1 al cifrei a_i

- Exemplu:

- $+6 = 00110$

- $-6 = 11001$ (1001 = complementul față 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 semnificațiile:

- mărime și semn $N^* = \sum_{i=1}^n a_i 2^{-i}$

- $N < 0$ în complement față de 2

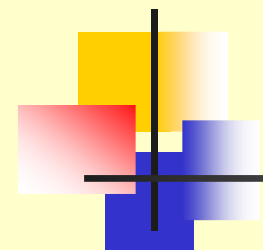
$$N^* = \sum_{i=1}^n \bar{a}_i 2^{-i} + 2^{-n}$$

- $N < 0$ în complement față de 1

$$N^* = \sum_{i=1}^n \bar{a}_i 2^{-i}$$

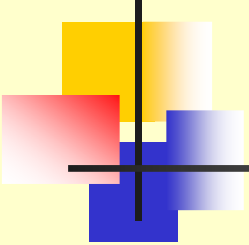
- a_i = cifrele numărului
- n = numărul de cifre din dreapta virgulei
- $\bar{a}_i = 1 - a_i$

2.1.1. Reprezentarea numerelor în virgulă fixă



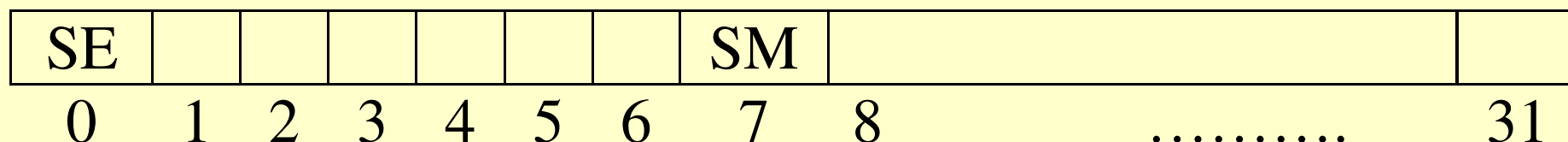
- Avantaje dacă virgula se plasează după prima poziție binară:
 - N^* fiind subunitar poziția virgulei este aceeași după înmulțirea binară
 - înmulțirea nu va duce niciodată la depășirea limitei superioare a gamei de reprezentare a numerelor
 - această plasare poate fi ușor memorată

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 și 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 biți:



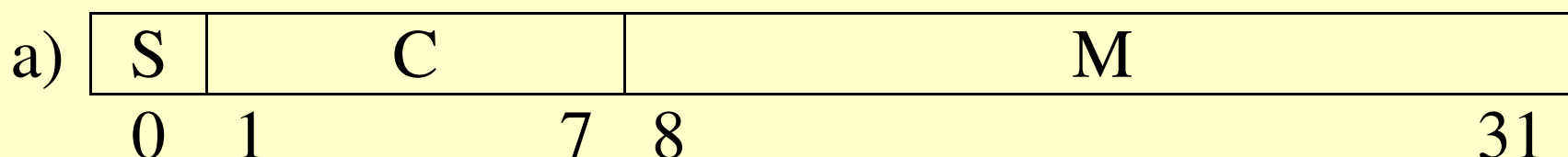
- bitul 0 = SE - semn exponent
 - biții 1- 6 = exponent E
 - bitul 7 = SM - semn mantisă
 - biții 8-31 = mantisă M
- Numărul +12,34 se poate reprezenta în binar:
 - $+1100,01010111$ sau $+0,110001010111 \times 2^{+4}$ sau $+0,00110001010111 \times 2^{+6}$
 - Reprezentarea pe 32 de biți poate fi:
 - 0 000100 0 110001010111000000000000 sau
 - 0 000110 0 001100010101110000000000

2.1.2. Reprezentarea numerelor în virgulă mobilă

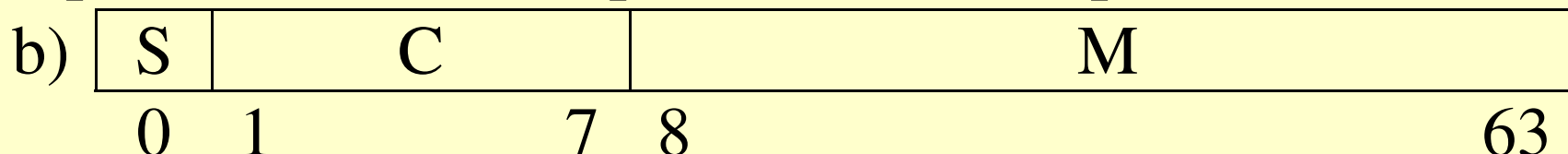
- În calculatoare se utilizează reprezentarea numai cu semn la mantisă, nu și la exponent \Rightarrow se folosește mărimea numită “Caracteristică”
- **Caracteristica** numărului N : $C = E + \text{deplasament}$
 - E = exponentul numărului (putere a lui 16)
 - deplasamentul ales ca să rezulte întotdeauna o valoare pozitivă (ex.: 64 sau 128)
- Valoarea exponentului $E = (C - \text{deplasament})$
 - avantaje
 - este numai pozitiv \Rightarrow operații simplificate
 - cifra 0 reprezentată la fel cu reprezentarea în virgulă fixă \Rightarrow tratare similară
 - dezavantaj - există o operație de scădere în plus!

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Reprezentare în simplă precizie, de exemplu:



- Reprezentare în dublă precizie, de exemplu:



- Exemplu:

- Pentru caracteristica $C = 7$ biți \Rightarrow numerele pot fi între 0 și $2^7-1 \Rightarrow 0 \leq C \leq 127$
- 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 \Rightarrow mantisa câștigă un bit semnificativ în plus
 - simplificare operații
 - creștere precizie
- Probleme cu reprezentarea valorii 0, care nu poate fi normalizată \Rightarrow valoarea 0 are o reprezentare specială

2.1.2. Reprezentarea numerelor în virgulă mobilă

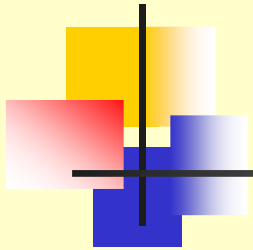
- Exemplu: gama numerelor reprezentate în complement față de 2, în cuvinte de 32 de biți:
 - numere pozitive între: $0,5 \cdot 2^{-128}$ și $(1 - 2^{-24}) \cdot 2^{127}$
 - numere negative între: $-(1 - 2^{-24}) \cdot 2^{127}$ și $-0,5 \cdot 2^{-128}$
 - 5 regiuni necuprinse în aceste domenii → depășire
 - depășire inferioară negativă: $< -(1 - 2^{-24}) \cdot 2^{127}$
 - depășire superioară negativă: $> -0,5 \cdot 2^{-128}$
 - zero
 - depășire inferioară pozitivă: $< 0,5 \cdot 2^{-128}$
 - depășire superioară pozitivă: $> (1 - 2^{-24}) \cdot 2^{127}$
 - coprocesoarele matematice și unitățile de calcul în virgulă mobilă au mecanisme speciale pentru detectarea, semnalizarea și tratarea depășirilor

2.1.2. Reprezentarea numerelor în virgulă mobilă

- Compromis între dimensiunea mantisei și a exponentului
 - mantisa de dimensiune mai mare \Rightarrow crește precizia numerelor care pot fi reprezentate
 - exponentul de dimensiune mai mare \Rightarrow crește domeniul numerelor care pot fi reprezentate
- Creșterea numărului de biți pentru reprezentare:
 - creșterea preciziei
 - creșterea 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
- Operații aritmetice în această reprezentare



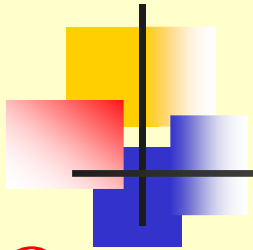
2.2. Aritmetica binară

Operații aritmetice cu numere fără semn

■ Adunarea binară

■ operație modulo 2

- cifra cu valoarea cea mai mare: $2-1 = 1$
- dacă rezultatul adunării a 2 cifre de rang “i” depășește 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 depășirea capacității de reprezentare a rezultatului



2.2. Aritmetica binară

Operații aritmetice cu numere fără semn

■ Adunarea binară

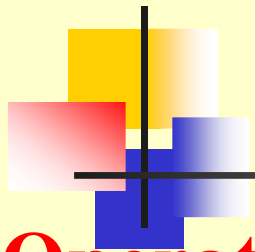
x	y	Transport	Sumă
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

■ Exemplu:

$$22_{10} = 10110_2 +$$

$$19_{10} = 10011_2$$

$$\hline 41_{10} = 101001_2$$



2.2. Aritmetica binară

Operații aritmetice cu numere fără semn

■ Scăderea binară

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

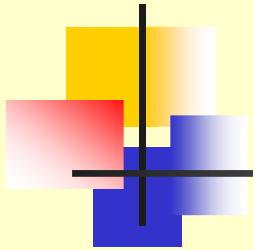
x	y	Împrumut	Diferență
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

■ Exemplu:

$$22_{10} = 10110_2 -$$

$$19_{10} = 10011_2$$

$$\hline 3_{10} = 00011_2$$



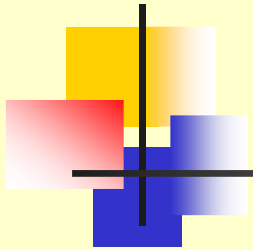
2.2. Aritmetica binară

Operații aritmetice cu numere fără semn

■ Înmulțirea binară

- prin adunarea repetată de produse parțiale
- produsul este 1 doar dacă și deînmulțitul și înmulțitorul sunt 1

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



2.2. Aritmetica binară

Operații aritmetice cu numere fără semn

■ Înmulțirea binară - exemplu:

$$12_{10} = 1100_2 \times$$

$$6_{10} = 0110_2$$

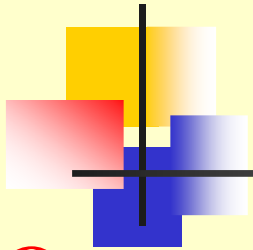
$$\hline 0000$$

$$1100$$

$$1100$$

$$\hline 0000$$

$$1001000_2 \quad (64 + 8 = 72_{10})$$



2.2. Aritmetica binară

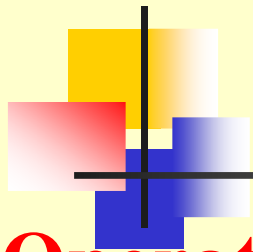
Operații aritmetice cu numere fără semn

■ Împărțirea binară

- nu se poate efectua dacă împărțitorul este egal cu 0!!!
- trebuie să fie satisfăcută relația:

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

- X = deîmpărțit; Y = împărțitor; Q = cât; R = rest
- se fac scăderi ale împărțitorului din resturile parțiale
 - dacă restul parțial este:
 - mai mare ca împărțitorul \Rightarrow cifra câtului este 1
 - dacă restul parțial este mai mic decât împărțitorul \Rightarrow cifra câtului este 0



2.2. Aritmetica binară

Operații aritmetice cu numere fără semn

■ Împărțirea binară - exemplu

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

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

1011

1110

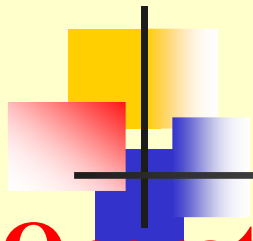
1011

1111

1011

100₂

rest = 4₁₀

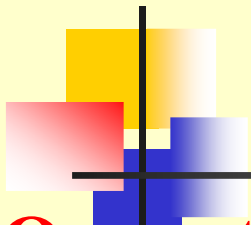


2.2. Aritmetica binară

Operații aritmetice cu numere în virgulă fixă

■ Adunarea numerelor reprezentate în complement față de 2

- se adună numerele bit cu bit, inclusiv biții de semn
- se ignoră transportul de la biții de semn
- dacă rezultatul e negativ apare ca un număr reprezentat în complement față de 2
- Observații:
 - dacă rezultatul în valoare absolută este mai mare decât valoarea maximă care poate fi reprezentată \Rightarrow depășire
 - la adunarea a 2 numere de același semn apare depășire dacă și numai dacă rezultatul are semn contrar semnelor numerelor



2.2. Aritmetica binară

Operații aritmetice cu numere în virgulă fixă

- Adunarea numerelor reprezentate în complement față de 2 - exemple

$+ 9_{10}$	$0\ 1001_2$
$+ 5_{10}$	$0\ 0101_2$
$\hline +14_{10}$	$\hline 0\ 1110_2$

$- 9_{10}$	$1\ 0111_2$
$- 5_{10}$	$1\ 1011_2$
$\hline - 14_{10}$	$\hline 11\ 0010_2$

$+ 7_{10}$	$0\ 0111_2$
$- 4_{10}$	$1\ 1100_2$
$\hline + 3_{10}$	$\hline 10\ 0011_2$

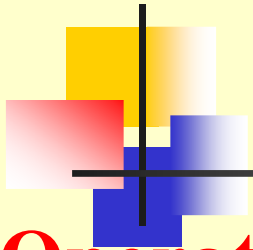
$+ 9_{10}$	$0\ 1001_2$
$+ 11_{10}$	$0\ 1011_2$
$\hline + 20_{10}$	$\hline 1\ 0100_2$

rezultat incorect

$- 9_{10}$	$1\ 0111_2$
$- 11_{10}$	$1\ 0101_2$
$\hline - 20_{10}$	$\hline 10\ 1100_2$

rezultat incorect

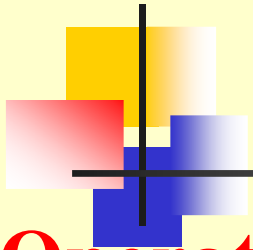
$- 7_{10}$	$1\ 1001_2$
$+ 4_{10}$	$0\ 0100_2$
$\hline - 3_{10}$	$\hline 1\ 1101_2$



2.2. Aritmetica binară

Operații aritmetice cu numere în virgulă fixă

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



2.2. Aritmetica binară

Operații aritmetice cu numere în virgulă fixă

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

$$D: + 7_{10} \quad 0\ 0111_2$$

$$S: - 4_{10} \quad 1\ 1100_2$$

$$\overline{S}: - (-4)_{10} \quad 0\ 0100_2 \quad \text{complement față de 2}$$

Atunci:

$$D: + 7_{10} \quad 0\ 0111_2$$

$$S: \underline{+ 4_{10}} \quad \underline{0\ 0100_2}$$

$$\quad + 11_{10} \quad 0\ 1011_2$$