### Sisteme de numere positionale

☐ **Sistem pozițional** - un număr este reprezentat printrun şir de cifre, unde fiecare poziție a unei cifre este asociată o anumită contributie (pondere).

$$D = d_{m-1} d_{m-2} ... d_1 d_0 .d_{-1} d_{-2} ... d_{-n}$$

MSD

Virgula fixă

LSD

Siteme de numere binare

Ex.:  $N=11001.011_2$  $N=1*2^4+1*2^3+0*2^2+0*2^1+1*2^0+0*2^1+1*2^2+1*2^3=25.375_{10}$ 

- $\square$  Baza 8 corespunde sistemului octal. cifre  $\{0,1,2,3,4,5,6,7\}$
- □ Baza 16 corespunde sistemului hexazecimal.

cifre {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

- □ Conversia din binar în hexazecimal Ex: 010011110111.101001010
  - Partiţionarea numărului binar în grupuri de 4 pornind de la virgulă şi inaintand spre dreapta sau stanga :

0100\_1111\_0111 • 1010\_0101\_0000

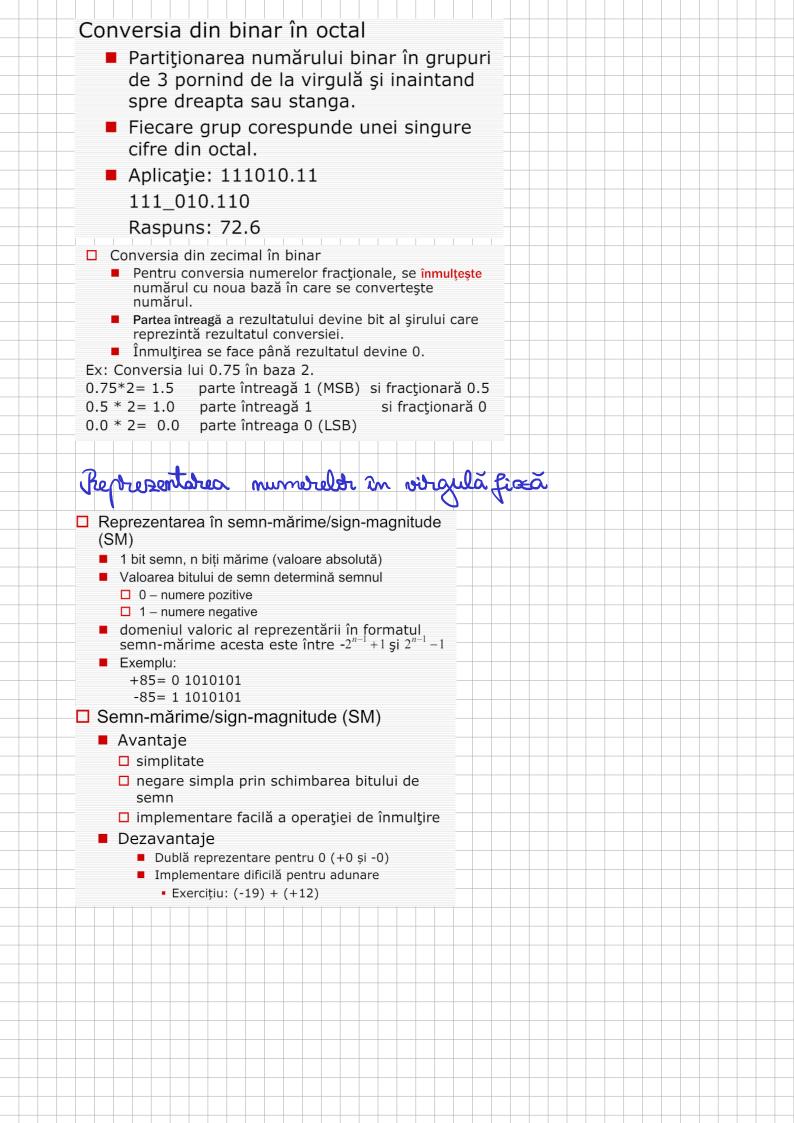
4F7.A50

Fiecare grup corespunde unei singure cifre hexazecimale. Folosind Tabelul ant. obţinem:

Aplicație: Convertiți numărul din binar în hexazecimal:

111 1100 1010.0111 1111

Raspuns: 7CA.7F



## Complement de 1

- 1 bit semn, *n* biţi pentru mărime
- Numerele pozitive identic cu SM
- Numerele negative complementarea/ negarea marimii
- Exemplu:

- ☐ Dezavantaje:
  - C1 nu este un format ponderat în conformitate cu notația pozițională
  - există două reprezentări pentru numărul zero (pentru un numar reprezentat pe 6 biţi avem 0 00000, respectiv 1 11111), deci testarea pentru zero se va face de două ori
- Avantaje:
  - o implementare mai facilă a operaţiei de adunare comparativ cu SM
- □ domeniul valoric pentru numere întregi:

$$-(2^{n-1}-1)$$
 și  $2^{n-1}-1$ 

. d

## Complement de 2

- Numerele pozitive identic cu SM
- Numerele negative negarea valorii pozitive la care se adaugă 1
- Întregi:  $1\overline{b_{n-2}}...\overline{b_1}\overline{b_0} + 0.0...01$
- Fracționare:  $1.\overline{b_{-1}}...\overline{b_{-n+1}}\overline{b_{-n}} + 0.0...01$
- Exemplu:

#### Dezavantaje:

- ☐ Mai dificil de obținut decât SM și C1;
- □ nu este un format ponderat în conformitate cu notația pozițională
- □ anomalia complementului de doi

#### Avantaje:

- □ O singura reprezentare pentru O!
  - **0000000**
- □ Implementarea facilă a operației de adunare
  - Exerciţiu (-19) + (+12)

Număr zecimal	Format SM	Format C1	Format C2
+3	0 11	0 11	0 11
+2	0 10	0 10	0 10
+1	0 01	0 01	0 01
+0	0 00	0 00	0 00
-0	1 00	1 11	
-1	1 01	1 10	1 11
-2	1 10	1 01	1 10
-3	1 11	1 00	1 01
-4			1 00

Domeniul valoric pentru numere întregi:  $-2^{n-1}$  si  $2^{n-1}-1$ 

- □ Se dau următoarele perechi de numere întregi: +23 şi +18, +23 şi -18, -23 şi +18, respectiv -23 şi -18. Se cere:
  - Să se convertească numerele în formatele semn-mărime, complement de 1, respectiv complement de 2.
  - Să se efectueze adunarea celor două numere.

nur	mere.			
Mr	SM	CA	C2.	
23	010114	010114	010111	
( 8	010010	010010	010010	-
- 18	110010	101101	101110	
- 23	110111	101000	101001	•
+ 13 + 18	S 10111 0 10010		SM-de obic	ei m ee paate
41	0 1001	- 4	dacă e M	nai mult de 1 wrco
+ 23 - 48	11111 0 10111 1 0110 1	Cl C <sub>2</sub>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
5	0 00100		000,01	= 5
	0 00 10 1	-5 =>	-5-111010	
- <u>2</u> 3 + 18	101000	101007	0 02	
-5	111010	1,101		
- 18 -23	101101	191110	Cz	
_ \ \	1010101	1010111		
	1010110	- 1010111		

Preparentarea numerales en virgula flotanta

- □ reprezentate folosind notaţia ştiinţifică (care nu este poziţională) → un domeniu valoric foarte mare.
- □ Pentru a reprezenta un număr in virgulă flotantă folosim trei numere conform relaţiei:

$$N = M * B^E$$

M - mantisa. (M poate fi reprezentată în SM sau C2)

B - baza (de obicei e 2 sau o putere a lui 2)

E - exponent. (E este reprezentat în SM sau cod exces)

Sbebm	
-------	--

E M

*M* - mantisa. (*M* poate fi reprezentată în SM sau C2)

B - baza (de obicei e 2 sau o putere a lui 2)

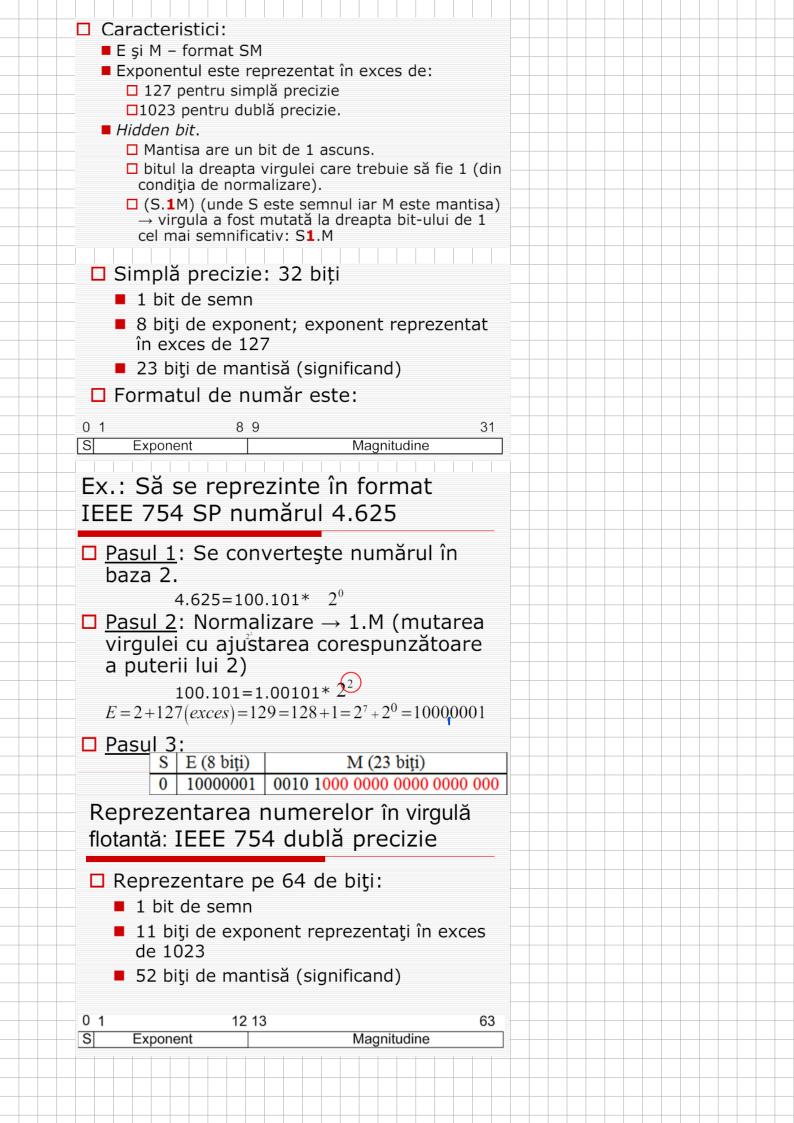
E - exponent. (E este reprezentat în SM sau cod exces)

- □ Reprezentarea mantisei:
  - Reprezentarea lui 18:

 $18*10^0 = 1.8*10^1 = 0.18*10^2 = \dots = 0.0 \dots 018*10^n$ 

- Obs.: M, B şi E au o infinitate de valori posibile
- Pentru o tratare unitară şi eficientă prin prisma procesării în sistemele de calcul → o reprezentare unică → normalizarea mantisei M
- ☐ M în SM primul bit din dreapta virgulei trebuie să fie 1.
- ☐ M în C2 şi M corespunde la o:
  - valoare negativă atunci primul bit din dreapta virgulei trebuie să fie 0.
  - valoare pozitivă, atunci folosim regula de la SM.

		ării lui 0 în vii	<b>J</b>			
<b>0</b> = 0 *	$B^{E}$					
■ mai r	multe varian	te de repreze	entare			
		ât mai uşor d iţi de `0'				
■ Dar in	ı calcule reci posibil ca în i	urgem la apro	oximari			
dator	ită acestor a	proximari suc	cule (FP), cessive, să parte mic (M≠0			
☐ Pentru a aferent li	minimiza	eroarea→	exponentul			
□ valoarea		•				
■ Toate valo biti sunt	denlasate	(devin po	zitive) prin	IN		
adunarea	a unui bias	(unui surr	zitive) prin olus) = i mic număr			
valoarea	absolută	a celui mai	mic număr			
exponen		numărul d	le biçi			
□ Pentru ex		enrezentat	în:			
		oarea bias-u				
		oarea bias-ul				
		Valoare	cu semn			
Reprezentare	Valoare	(reală-cea a	a numărului			
binară	fără semn	repreze				
11111111	255	Bias = 127 +128	Bias = 128 +127			
11111111	255	+128	+127			
11111110	237	1121	1120			
			:			
10000001	129	+2	+1			
10000000 01111111	128 127	+1 0	0 -1			
01111111	127	-1	-1			
OIIIIII	120	-1	-2			
			:			
00000001	1	-126	-127			
0000000	0	-127	-128			
Gele M	wi henr	Rentative	2 stando	ide pentru	virgula	moleila
					0	
	IEEE 4	54				
1 2	IBM 4	360/370				
		'				
Standa	rdul IEE	E 754/20	008- forn	nate:		
■Half pr						
	precision					
•	•					
■ Double	e precision					
	extended					



# Reprezentarea numerelor în virgulă flotantă: IEEE 754 valori speciale

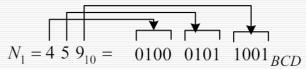
Nr.	Exponent	Mantisa (M)	Valoare speciala
	(E)		
1.	0	0	±0
2.	0	<b>≠</b> 0	Denormalized
			numbers
3.	255	0	±∞
4.	255	<b>≠</b> 0	NaN

- Nr. denormalizate: rezultat care este mai mic decât valoarea minimă reprezentabilă
- Infinit: situația în care rezultatul intermediar este infinit sau avem overflow
- □ 0/0 sau radical din nr. negativ

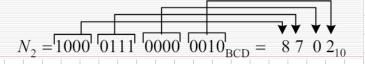
(!) Cod (code),	O colecție de șiruri diferite pe $n$ biți, iar fiecare dintre aceste șiruri are o semnificatie (reprezintă un număr, caracter, etc.) poartă denumirea de $cod$ (code). Numărul maxim de cuvinte ale unui cod pe n biți este $2^n$ . Nu întotdeauna însă, toate aceste combinații posibile pe $n$ biți sunt folosite (fac parte din colecția de șiruri care alcătuiesc codul).
Cuvânt al unui cod (code word)	Un şir al colecției care reprezintă o combinație de $n$ valori de $\theta$ sau $I$ se numește cuvânt de cod (code word).

### Coduri linare pentru numbre secimale - BCS

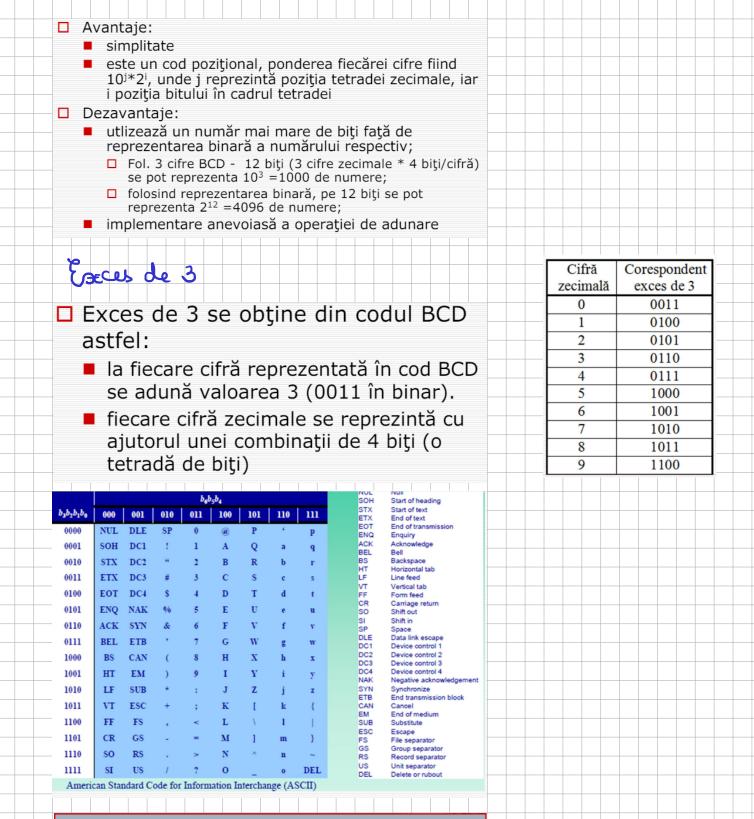
- există situații cand se dorește afișarea rezultatelor de către interfețele externe ale dispozitivului de calcul într-un format ușor de înțeles (decodificat) de către utilizator – și anume mult întrebuințatul format zecimal;
- cel mai la îndemână cod zecimal este BCD (binary-code decimal):
  - reprezentarea unei cifre BCD → înlocuirea cu reprezentarea în binar care îi corespunde→ cu un nr. pe 4 biţi
- conversia unui număr zecimal în BCD prin înlocuirea succesivă a cifrelor zecimale cu tetradele corespunzătoare



 operaţia inversă de transformare a unui număr reprezentat în BCD în omologul zecimal



Cifră	Corespondent
zecimală	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001



litera **A** are primele 3 pozitii (765) secventa 100, iar pe urmatoarele 4 pozitii (4321) secventa 0001. Deci A=(1000001)!