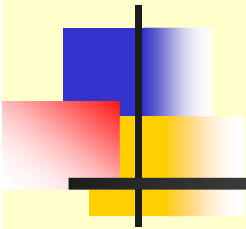
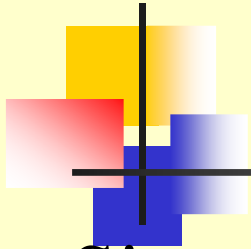


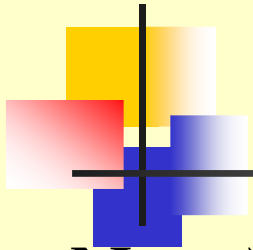
I. SISTEME DE NUMERAȚIE ȘI CODURI





1.1. Sisteme de numerație

- Sistemele numerice prelucrează informație
- Informația este **codificată** → un anumit tip de reprezentare
- **Sistemul de numerație = totalitatea regulilor de reprezentare a numerelor cu ajutorul unor simboluri numite cifre**
- Sistemele de numerație:
 - poziționale - valoarea unei cifre - determinată de poziția sa în cadrul numărului
 - nepoziționale

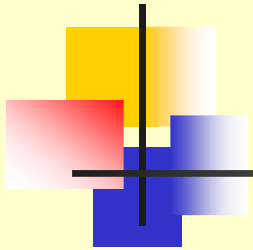


1.1. Sisteme de numerație

- Numărul “N”, în sistem pozițional, în baza de numerație “b” se reprezintă:

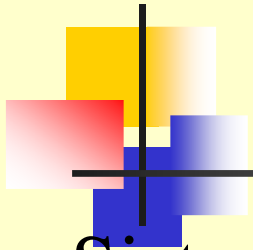
$$N = a_{q-1}b^{q-1} + \dots + a_0b^0 + \dots + a_{-p}b^{-p} = \sum_{i=-p}^{q-1} a_i b^i$$

- baza “b” este număr întreg, > 1
- a_i sunt întregi în gama: $0 \leq a_i \leq b-1$
- notația $(N)_b$ = numărul “N” în baza “b”
- dacă baza “b” nu se specifică - este implicit 10
- complementul unei cifre “a”, notat “ \bar{a} ” în baza “b” este definit: $\bar{a} = (b-1) - a$



1.1. Sisteme de numerație

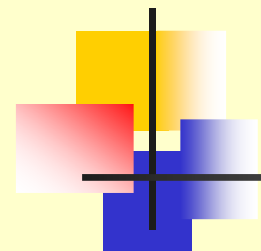
- **Sistem de numerație binar** - baza $b=2$
 - în sistemul numeric binar avem 2 cifre binare (“biți”), 0 și 1
 - complementele cifrelor binare: $\overline{0} = 1$; $\overline{1} = 0$
- Sistemele numerice folosesc pentru reprezentarea informației și alte sisteme de numerație → mai uzuale:
 - octal
 - hexazecimal



1.1. Sisteme de numerație

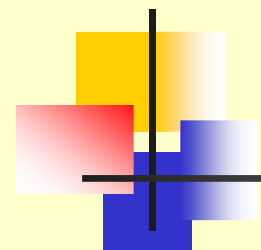
- Sistem de numerație octal - baza $b = 8$
 - 8 cifre: 0 - 7
 - reprezentarea pentru echivalentul în binar - pe 3 biți
 - 0 = 000; 1 = 001; 2 = 010; 3 = 011; 4 = 100; 5 = 101; 6 = 110; 7 = 111
- Sistem de numerație hexazecimal - baza $b = 16$
 - 16 semne: 0 - 9 și A - F
 - reprezentarea pentru echivalentul în binar - pe 4 biți
 - 0 = 0000; 1 = 0001; 2 = 0010; 3 = 0011; 4 = 0100; 5 = 0101; 6 = 0110; 7 = 0111; 8 = 1000; 9 = 1001; A = 1010; B = 1011; C = 1100; D = 1101; E = 1110; F = 1111
 - 1 byte (octet) = 8 biți - reprezentare cu 2 semne
 - utilizat pentru reprezentare restrânsă

1.2. Conversia bazei de numerație



- Conversia se face din baza “ b_1 ” în baza “ b_2 ”
- În sistemele poziționale conversia se face prin înmulțiri sau împărțiri repetate
- Se disting 2 cazuri de conversie:
 - a) $b_1 < b_2$
 - b) $b_1 > b_2$

1.2. Conversia bazei de numerație



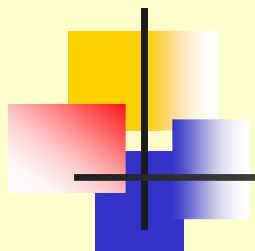
- a) $b_1 < b_2$
 - $(N)_{b_1}$ se exprimă ca un polinom în puterile lui “ b_1 ” și se evaluează polinomul folosind aritmetica în baza “ b_2 ”
 - Exemplu:
 - $b_1 = 3$
 - $b_2 = 10$
 - $(N)_3 = 2120,1$
 - $(N)_{10} = 2 \cdot 3^3 + 1 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0 + 1 \cdot 3^{-1} =$
 $= 54 + 9 + 6 + 0 + 0,3 = 69,3$

1.2. Conversia bazei de numerație

■ b) $b_1 > b_2$

- se utilizează aritmetica în baza “ b_1 ”
- se face conversia separată a părții întregi și a părții fracționare
- conversia părții întregi:
 - numărul se împarte la baza “ b_2 ”, se obține un cât și un rest
 - se reține restul și se continuă cu împărțirea câtului la baza “ b_2 ”
 - algoritmul se oprește când câtul = 0
 - partea întreagă se obține prin scrierea resturilor obținute, în ordine inversă generării lor

1.2. Conversia bazei de numerație



- b) $b_1 > b_2$
 - conversia părții fracționare:
 - numărul de înmulțește cu baza “ b_2 ”, se obține un număr format dintr-o parte întreagă și una fracționară
 - se reține partea întreagă și se continuă cu înmulțirea cu baza “ b_2 ” a părții fracționare obținute
 - algoritmul se continuă până la obținerea **preciziei** dorite
 - partea fracționară se obține prin scrierea părților întregi obținute, în ordinea generării lor

1.2. Conversia bazei de numerație

■ b) $b_1 > b_2$

■ Exemplu:

■ $b_1 = 10$

■ $b_2 = 4$

■ $(N)_{10} = 347,4$

■ conversia părții întregi:

■ $347:4 = 86 \text{ rest } 3$

■ $86:4 = 21 \text{ rest } 2$

■ $21:4 = 5 \text{ rest } 1$

■ $5:4 = 1 \text{ rest } 1$

■ $1:4 = 0 \text{ rest } 1$

■ Partea întreagă = $(11123)_4$

■ conversia părții fracționare:

$$0,4 \cdot 4 = 1,6 \Rightarrow 1$$

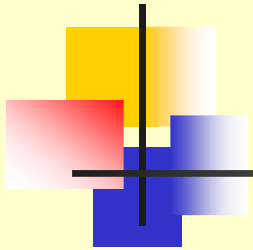
$$0,6 \cdot 4 = 2,4 \Rightarrow 2$$

$$\text{Partea fracționară} = (1212\dots)_4$$

$$\text{Numărul } (N)_4 = (11123,1212\dots)_4$$

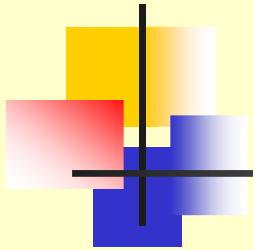
1.2. Conversia bazei de numerație

- Conversia numerelor din octal și hexazecimal în binar și invers - se bazează pe exprimarea prin 3, respectiv 4 biți a cifrelor din octal și hexazecimal
- Exemple:
 - din octal și hexazecimal în binar
 - $(123,4)_8 = (001\ 010\ 011, 100)_2 = (1010011,1)_2$
 - $(2C5F,8)_{16} = (0010\ 1100\ 0101\ 1111, 1000)_2 = (10110001011111,1)_2$
 - din binar în octal și hexazecimal
 - $(1010110,0101)_2 = (001\ 010\ 110, 010\ 100)_2 = (126,24)_8$
 - $(10111001101010,10011)_2 = (0010\ 1110\ 0110\ 1010, 1001\ 1000)_2 = (2E6A,98)_{16}$



1.3. Coduri binare

- Sistemul zecimal este preferat în interfața om - sistem de calcul
- **Cifrele zecimale se reprezintă prin succesiuni de cifre binare → coduri binare (binar-zecimale)**
- 10 cifre zecimale, 0 - 9, se pot reprezenta pe 4 biți
- Codurile binare:
 - ponderate
 - neponderate



1.3.1. Coduri ponderate

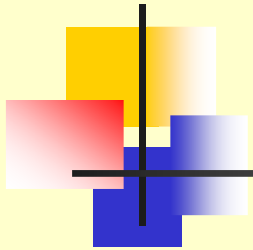
- La fiecare cifră binară - se asociază o pondere
- Pentru fiecare grup de 4 cifre binare din cod - suma ponderilor acelor cifre binare care au valoarea 1 = cifra zecimală pe care o reprezintă
- O cifră dintr-un cod ponderat, unde a_i poate lua valoarea 0 sau 1, se scrie:

$$N = \sum_{i=0}^3 a_i b_i$$

1.3.1. Coduri ponderate

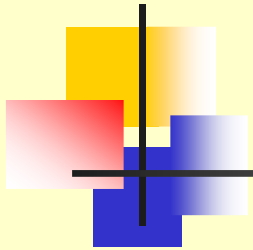
■ Exemple de coduri binare ponderate

Cifră zecimală	b ₃	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	Pondere negativă			
	8	4	2	1	2	4	2	1	6	4	2	-3
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	1	0	1
2	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1	1	0	0	1
4	0	1	0	0	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1	1	0	1	1
6	0	1	1	0	1	1	0	0	0	1	1	0
7	0	1	1	1	1	1	0	1	1	1	0	1
8	1	0	0	0	1	1	1	0	1	0	1	0
9	1	0	0	1	1	1	1	1	1	1	1	1



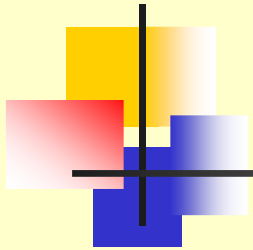
1.3.1. Coduri ponderate

- Codul **BCD** (**B**inary **C**oded **D**ecimal) = codul binar natural - ponderea **8421** (puterile lui 2!) - fiecare cifră zecimală este convertită în binar
- Codurile **2421** și cel cu **pondere negativă 642-3**, sunt coduri ponderate din categoria codurilor autocomplementare → complementul lui “N” este $9-N$
- Condiția de cod autocomplementar la codurile ponderate - suma ponderilor să fie egală cu 9



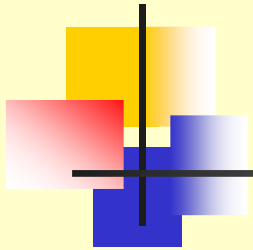
1.3.2. Coduri neponderate

- Au alte reguli de formare
- Codul **EXCES 3**:
 - format prin adăugarea lui 0011 (reprezentarea în binar a cifrei 3) la fiecare cuvânt de cod din codul ponderat BCD
 - este un cod autocomplementar
 - nu conține combinația 0000, care ar putea fi confundată cu lipsa de informație



1.3.2. Coduri neponderate

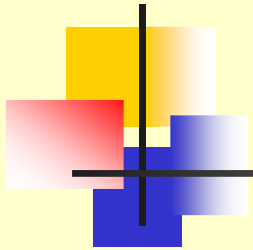
- Coduri ciclice - cuvintele de cod succesive diferă doar printr-o cifră binară
- Coduri reflectate - cuvântul de cod de “n” biți se generează prin reflectarea cuvântului de cod de “n-1” biți
- Codul **GRAY**:
 - ciclic
 - reflectat



1.3.2. Coduri neponderate

- Codul Grey pe 3 biți se obține prin reflectarea celui pe 2 biți
- Codul Grey pe 4 biți se obține prin reflectarea celui pe 3 biți

Gray 2 biți		Gray 3 biți		
0	0	0	0	0
0	1	0	0	1
1	1	0	1	1
1	0	0	1	0
		1	1	0
		1	1	1
		1	0	1
		1	0	0

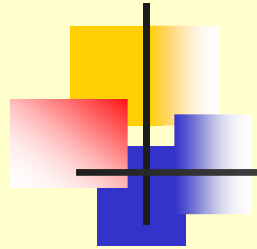


1.3.2. Coduri neponderate

- Codurile binar-zecimale EXCES 3 și Gray

Cifră zecimală	Exces 3	Gray
0	0 0 1 1	0 0 0 0
1	0 1 0 0	0 0 0 1
2	0 1 0 1	0 0 1 1
3	0 1 1 0	0 0 1 0
4	0 1 1 1	0 1 1 0
5	1 0 0 0	0 1 1 1
6	1 0 0 1	0 1 0 1
7	1 0 1 0	0 1 0 0
8	1 0 1 1	1 1 0 0
9	1 1 0 0	1 1 0 1

1.4. Detectarea și corecția erorilor



- În sistemele numerice informația se poate altera în procesul de transmitere
- Corectitudinea informației recepționate → se utilizează coduri detectoare și corectoare de erori

1.4.1. Coduri detectoare de erori

- Apariția unei singure erori transformă un cuvânt de cod valid în cuvânt de cod invalid
- Metode de detecție a erorilor
 - **Metoda bitului de paritate**
 - se adaugă o cifră binară în plus la fiecare cuvânt de cod al unui cod dat, pentru a face ca numărul de biți de 1 din fiecare cuvânt să fie impar sau par
 - Exemplu - se transmite cuvântul de cod 1011
 - pentru paritate impară - se adaugă înainte de cuvânt cifra 0 → 0 1011
 - pentru paritate pară - se adaugă înainte de cuvânt cifra 1 → 1 1011

1.4.1. Coduri detectoare de erori

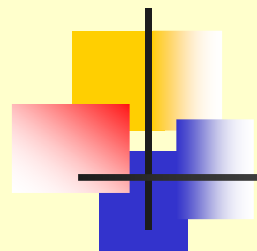
■ Metode de detecție a erorilor

■ Codul detector de erori “2 din 5”

- are ponderile 01247, cu excepția cuvântului de cod 0 zecimal

Codul detector de erori “2 din 5”					
Cifra zecimală	0	1	2	4	7
0	0	0	0	1	1
1	1	1	0	0	0
2	1	0	1	0	0
3	0	1	1	0	0
4	1	0	0	1	0
5	0	1	0	1	0
6	0	0	1	1	0
7	1	0	0	0	1
8	0	1	0	0	1
9	0	0	1	0	1

1.4.1. Coduri detectoare de erori



- Pentru un cod de detecție a erorii de “n” biți nu se pot folosi mai mult de jumătate din cele 2^n combinații posibile ale cifrelor binare
 - Pentru 10 cifre sunt necesare cel puțin 5 cifre binare!
- Distanța minimă a unui cod = cel mai mic număr de biți prin care diferă 2 coduri
- Un cod este detector de erori dacă distanța sa minimă este ≥ 2

1.4.2. Coduri corectoare de erori

- Cod corector de eroare - dacă întotdeauna cuvântul de cod corect poate fi dedus din cuvântul eronat
- Cele mai cunoscute coduri de corecție - **codurile Hamming**
- Codurile Hamming corectoare de erori singulare (o sigură eroare!) au distanța minimă = 3
- Relația lui Hamming - număr minim de biți de control necesari pentru corectarea erorilor singulare:

$$2^k \geq m + k + 1$$

- m = număr de biți de informație utilă
- k = număr de biți de control

1.4.2. Coduri corectoare de erori

- Exemplu de cod Hamming pentru $m=4$; mesaj original în cod BCD (4 biți de informație utilă)
 - Din relația lui Hamming $\Rightarrow k = 3$
 - Număr total de biți: 4 (utili) + 3 (control) = 7
 - Biții de control apar pe pozițiile corespunzătoare puterilor lui 2

1	2	3	4	5	6	7
c1	c2	b1	c3	b2	b3	b4

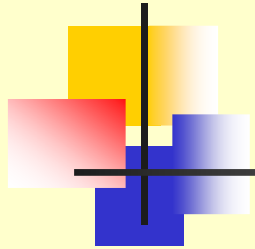
- Biții de control se calculează cu relațiile:

$$c_1 = b_1 \oplus b_2 \oplus b_4$$

$$c_2 = b_1 \oplus b_3 \oplus b_4$$

$$c_3 = b_2 \oplus b_3 \oplus b_4$$

1.4.2. Coduri corectoare de erori



■ Exemplu:

- $b_1b_2b_3b_4 = 0100$ (cifra 4 din zecimal exprimată în cod BCD)
- biții de control calculați: $c_1 = 1$, $c_2 = 0$, $c_3 = 1$
- secvența de cod de corecție va fi: 1001100