I. SISTEME DE NUMERAŢIE ŞI CODURI

1.1. Sisteme de numeraţie

- Sistemele numerice prelucrează informaţie
- Informaţia este **codificată** → un anumit tip de reprezentare
- **Sistemul de numeraţie = totalitatea regulilor de reprezentare a numerelor cu ajutorul unor simboluri numite cifre**
- Sistemele de numeraţie:
  - poziţionale - valoarea unei cifre - determinată de poziţia sa în cadrul numărului
  - nepoziţionale

1.1. Sisteme de numeraţie

- Numărul “N”, în sistem poziţional, în baza de numeraţie “b” se reprezintă:

$$N = a_{q-1}b^{q-1} + \ldots + a_0 b^0 + \ldots + a_{-p}b^{-p} = \sum_{i=-p}^{q-1} a_i b^i$$

  - baza “b” este număr întreg, > 1
  - a_i sunt întregi în gama: $0 \le a_i \le b-1$
  - notaţia $(N)_b$ = 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 numeraţie

- **Sistem de numeraţie binar** - baza b=2
  - în sistemul numeric binar avem 2 cifre binare ("biţi"), 0 şi 1
  - complementele cifrelor binare: $\overline{0} = 1$; $\overline{1} = 0$
- Sistemele numerice folosesc pentru reprezentarea informaţiei şi alte sisteme de numeraţie → mai uzuale:
  - octal
  - hexazecimal

1.1. Sisteme de numeraţie

- Sistem de numeraţie octal - baza b = 8
  - 8 cifre: 0 - 7
  - reprezentarea pentru echivalentul în binar - pe 3 biţi
    - 0 = 000; 1 = 001; 2 = 010; 3 = 011; 4 = 100; 5 = 101; 6 = 110; 7 = 111
- Sistem de numeraţie hexazecimal - baza b = 16
  - 16 semne: 0 - 9 şi A - F
  - reprezentarea pentru echivalentul în binar - pe 4 biţi
    - 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 biţi - reprezentare cu 2 semne
  - utilizat pentru reprezentare restrânsă

1.2. Conversia bazei de numeraţie

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

1.2. Conversia bazei de numeraţie

- a) $b_1 < b_2$
  - $(N)b_1$ se exprimă ca un polinom în puterile lui "b_1" şi se evaluează polinomul folosind aritmetica în baza "b_2"
  - Exemplu:
    - $b_1 = 3$
    - $b_2 = 10$
    - $(N)_3 = 2120,1$
    - $(N)_{10} = 2 \bullet 3^3 + 1 \bullet 3^2 + 2 \bullet 3^1 + 0 \bullet 3^0 + 1 \bullet 3^{-1} =$

    $= 54+9+6+0+0,3 = 69,3$

1.2. Conversia bazei de numeraţie

- b) $b_1 > b_2$
  - se utilizează aritmetica în baza “b_1”
  - se face conversia separată a părţii întregi şi a părţii fracţionare
  - conversia părţii întregi:
    - numărul se împarte la baza “b_2”, se obţine un cât şi un rest
    - se reţine restul şi se continuă cu împărţirea câtului la baza “b_2”
    - algoritmul se opreşte când câtul = 0
    - partea întreagă se obţine prin scrierea resturilor obţinute, în ordine inversă generării lor

1.2. Conversia bazei de numeraţie

- b) $b_1 > b_2$
  - conversia părţii fracţionare:
    - numărul de înmulţeşte cu baza “b_2”, se obţine un număr format dintr-o parte întreagă şi una fracţionară
    - se reţine partea întreagă şi se continuă cu înmulţirea cu baza “b_2” a părţii fracţionare obţinute
    - algoritmul se continuă până la obţinerea **preciziei** dorite
    - partea fracţionară se obţine prin scrierea părţilor întregi obţinute, în ordinea generării lor

1.2. Conversia bazei de numeraţie

- b) $b_1 > b_2$
  - Exemplu:
    - $b_1 = 10$
    - $b_2 = 4$
    - $(N)_{10} = 347,4$
    - conversia părţii întregi:
      - 347:4 = 86 rest 3
      - 86:4 = 21 rest 2
      - 21:4 = 5 rest 1
      - 5:4 = 1 rest 1
      - 1:4 = 0 rest 1
      - Partea întreagă = $(11123)_4$
- conversia părţii fracţionare:

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

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

  Partea fracţionară = $(1212...)_4$

Numărul $(N)_4 = (11123,1212\ldots)_4$

1.2. Conversia bazei de numeraţie

- Conversia numerelor din octal şi hexazecimal în binar şi invers - se bazează pe exprimarea prin 3, respectiv 4 biţi a cifrelor din octal şi hexazecimal
- Exemple:
  - din octal şi 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 şi 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 interfaţa 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 biţi
- 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** şi cel cu **pondere negativă 642-3**, sunt coduri ponderate din categoria codurilor autocomplementare → complementul lui “N” este 9-N
- Condiţia 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 conţine combinaţia 0000, care ar putea fi confundată cu lipsa de informaţie

1.3.2. Coduri neponderate

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

1.3.2. Coduri neponderate

- Codul Grey pe 3 biţi se obţine prin reflectarea celui pe 2 biţi
- Codul Grey pe 4 biţi se obţine prin reflectarea celui pe 3 biţi

Gray 2 biţi		Gray 3 biţi		
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 şi 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 şi corecţia erorilor

- În sistemele numerice informaţia se poate altera în procesul de transmitere
- Corectitudinea informaţiei recepţionate → se utilizează coduri detectoare şi corectoare de erori

1.4.1. Coduri detectoare de erori

- Apariţia unei singure erori transformă un cuvânt de cod valid în cuvânt de cod invalid
- Metode de detecţie 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 biţi 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 $\rightarrow$ 0 1011
      - pentru paritate pară - se adaugă înainte de cuvânt cifra 1 $\rightarrow$ 1 1011

1.4.1. Coduri detectoare de erori

- Metode de detecţie a erorilor
  - Codul detector de erori “**2 din 5**”
    - are ponderile 01247, cu excepţia 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 detecţie a erorii de “n” biţi nu se pot folosi mai mult de jumătate din cele 2^n combinaţii posibile ale cifrelor binare
  - Pentru 10 cifre sunt necesare cel puţin 5 cifre binare!
- Distanţa minimă a unui cod = cel mai mic număr de biţi prin care diferă 2 coduri
- Un cod este detector de erori dacă distanţa 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 corecţie - **codurile Hamming**
- Codurile Hamming corectoare de erori singulare (o sigură eroare!) au distanţa minimă = 3
- Relaţia lui Hamming - număr minim de biţi de control necesari pentru corectarea erorilor singulare:

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

  - m = număr de biţi de informaţie utilă
  - k = număr de biţi de control

1.4.2. Coduri corectoare de erori

- Exemplu de cod Hamming pentru m=4; mesaj original în cod BCD (4 biţi de informaţie utilă)
  - Din relaţia lui Hamming $\Rightarrow$ k = 3
  - Număr total de biţi: 4 (utili) + 3 (control) = 7
  - Biţii de control apar pe poziţiile corespunzătoare puterilor lui 2

1	2	3	4	5	6	7
c_1	c_2	b_1	c_3	b_2	b_3	b_4

  - Biţii de control se calculează cu relaţiile:

$$c_1 = b_1 \oplus b_2 \oplus b_4$$
$$c_2 = b_1 \oplus b_3 \oplus b_4$$
$$c_3 = b_2 \oplus b_3 \oplus b_4$$

1.4.2. Coduri corectoare de erori

- Exemplu:
  - $b_1b_2b_3b_4$ = 0100 (cifra 4 din zecimal exprimată în cod BCD)
  - biţii de control calculaţi: $c_1 = 1$, $c_2 = 0$, $c_3 = 1$
  - secvenţa de cod de corecţie va fi: 1001100