2. CODURI

Prin *codificare* se realizează o schimbare a formei de exprimare a unei informații. Dacă $X = \{ x_1 , \dots , x_p \}$ este mulțimea simbolurilor primare care urmează a fi codificate prin intermediul unor simboluri elementare aparținând unei mulțimi de elemente $B = \{ b_1 , \dots , b_n \}$, prin codificare se asociază fiecărui element $x_i \in X$ o secvență de simboluri $b_j \in B$ astfel încât modelul de codificare va fi reprezentat de *corespondența biunivocă*:

$$\begin{aligned} &x_1 \leftrightarrow b_1b_2b_3 = s_1 \\ &x_2 \leftrightarrow b_2b_5b_6b_7 = s_s \\ &\dots \\ &x_p \leftrightarrow b_1b_3b_1 = s_p \end{aligned} \qquad (2.1)$$

Cuvintele de cod formează o mulțime $S = \{ s_1 ,\dots, s_p \}$. *Codificarea* este o aplicație de forma $f : X \to S$. Codul se numeşte *uniform* dacă toate cuvintele $s_i \in S$ au aceeaşi lungime.

În electronica digitală, $B = \{ 0, 1 \}$, deci cuvintele mulțimii S sunt cuvinte binare de o anumită lungime, în general 8 (octet sau byte), 16, 24, 32 sau 64 de biți.

Informația primară poate fi compusă numai din simboluri numerice, sau atât din simboluri numerice, cât şi simboluri literale şi semne de ortografie. Se obțin astfel două tipuri de coduri: *coduri numerice*, respectiv *coduri alfanumerice*.

Exemple de coduri numerice sunt codurile: binar, octal, hexazecimal, zecimal codat binar (BCD- Binary Coded Decimal) etc. Exemple de coduri alfanumerice sunt: ASCII (American Standard Code for Information Interchange) şi EBCDIC (Extended Binary Coded Decimal Interchange Code).

2.1 Coduri numerice

Prin intermediul cuvintelor binare se pot codifica numere din sistemele de numerație binar, zecimal, octal, hexazecimal etc., rezultând *coduri binare*, *zecimal - binare*, *octal - binare*, *hexazecimal - binare* etc.

2.1.1 Coduri binare

I. Reprezentarea numerelor fără semn

Corespondența între un număr binar şi un cuvânt de cod binar poate fi chiar identitate, deci cuvântul de cod este chiar numărul respectiv.

$$x = b_{n-1} \dots b_1 b_0 , b_{-1} \dots b_{-m} \quad (2.2)$$

Virgula nu se reprezintă fizic, dar utilizatorul trebuie să ştie între ce biți ai cuvântului este localizată. Gama numerelor reprezentate este $x = [\,0 , 2^{n} - 2^{-m}\,]$.

În multe cazuri, numerele din acest domeniu se *scalează* prin împărțire la 2^n; virgula binară va fi poziționată în fața bitului cel mai semnificativ, iar gama reprezentabilă va deveni: $x \in [0 ,1-2^{-n-m}]$. Aceste reprezentări se numesc *numere fracționare în virgulă fixă*.

II. Reprezentarea numerelor cu semn

Prin convenție, „+” se reprezintă prin 0, iar „-” prin 1. Din cei n biți folosiți pentru partea întreagă, primul (bitul b_{n-1}) va fi folosit pentru reprezentarea semnului.

Există trei forme uzuale pentru reprezentarea numerelor cu semn, forme descrise în continuare.

a) *cod direct* (în modul şi semn)

$$x_d = \begin{cases} x, x \geq 0 \\ 2^{n-1} - x, x < 0 \end{cases} \qquad |x| = b_{n-2} \dots b_1 b_0 , b_{-1} \dots b_{-m} \quad (2.3)$$

Gama de reprezentare: $x \in [-(2^{n-1} - 2^{-m}), 2^{n-1} - 2^{-m}]$. Prin *scalare* (împărțire la 2^{n-1}), virgula se va situa imediat după bitul de semn, iar gama de reprezentare va deveni: $x \in [-(1-2^{-(n-1)-m}) , (1-2^{-(n-1)-m})]$.

b) *cod invers* (în complement față de 1)

$$x_i = \begin{cases} x, x \geq 0 \\ 2^{n} + x - 2^{-m}, x < 0 \end{cases} \qquad |x| = b_{n-2} \dots b_1 b_0 , b_{-1} \dots b_{-m} \quad (2.4)$$

Regula de inversare a unui număr negativ este următoarea: se complementează toți biții din reprezentarea în valoare absolută şi se ataşează 1 în rangul semn:

$$x_i = 1\overline{b_{n-2}} \dots \overline{b_1}\,\overline{b_0} , \overline{b_{-1}} \dots \overline{b_{-m}} \quad , \qquad \overline{b_i} = 1 - b_i \text{ pentru } x < 0 \quad (2.5)$$

Gama de reprezentare: $x \in [-(2^{n-1} - 2^{-m}), 2^{n-1} - 2^{-m}]$. Prin scalare gama de reprezentare va fi: $x \in [-(1-2^{-(n-1)-m}), (1-2^{-(n-1)-m})]$.

c) *cod complementar față de 2*

$$x_c = \begin{cases} x, x \geq 0 \\ 2^{n} + x, x < 0 \end{cases} \qquad |x| = b_{n-2} \dots b_1 b_0 , b_{-1} \dots b_{-m} \quad (2.6)$$

Regula de complementare a unui număr negativ este următoarea: se inversează cifrele binare ale numărului în valoare absolută, se sumează valoarea 1 la rangul -m şi se ataşează 1 în rangul semn:

$$x_c = 1\,\overline{b_{n-2}}...\overline{b_1}\,\overline{b_0},\overline{b_{-1}}...\overline{b_{-m}} + \underbrace{0{,}00...1}_{m} \tag{2.7}$$

O altă regulă de complementare este şi cea care urmează: codul complementar al unui număr negativ se obține prin inversarea biților din reprezentarea în valoare absolută începând cu primul bit 1 (exclusiv) întâlnit prin parcurgerea numărului de la dreapta la stânga, ataşându-se apoi 1 în rangul semn.

Gama de reprezentare este $x \in [-2^{n-1}, 2^{n-1} - 2^{-m}]$ iar prin scalare gama de reprezentare va fi: $x \in [-1, 1-2^{-(n-1)-m}]$.

Exemple:

x = -1011,1100
x_d = 11011,1100
x_i = 10100,0011
x_c = 10100,0100

Toate codificările scalate, cu virgula binară situată imediat după bitul de semn, se numesc *reprezentări în virgulă fixă*.

III. Reprezentarea numerelor în virgulă flotantă

Un număr rațional x se reprezintă prin două numere binare:

$$x = M \cdot 2^{E}, \tag{2.8}$$

unde: M - *mantisă* (număr fracționar cu semn; m biți)
E - *exponent* (număr întreg cu semn; n biți)

Dacă *m* are numai parte fracționară, gama de reprezentare este:

$$x \in (-2^{(2^{n-1}-1)}, 2^{(2^{n-1}-1)})$$

unde s-a presupus E reprezentat în complement față de 2.

Pentru mărirea preciziei calculelor, mantisa se normează după fiecare operație aritmetică, astfel încât cifra binară de după virgulă a modulului mantisei să fie nenulă.

Exemplu:

$$0{,}001011 \cdot 2^{10} = 0{,}1011 \cdot 2^{8}.$$

2.1.2 Coduri zecimal - binare

În cadrul acestei clase, X = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }. Mulțimea S trebuie să conțină 10 cuvinte distincte, deci pentru codificare sunt necesari minimum 4 biți ($2^3 < 10 < 2^4$). Cu aceşti 4 biți se pot forma 16 cuvinte de cod distincte, deci există A_{16}^{10} posibilități de codificare. În practică se folosesc anumite variante, cele mai importante fiind prezentate în continuare.

I. Coduri ponderate

Un cod ponderat asociază fiecărei cifre zecimale o *tetradă binară*, ponderea fiecărui bit din tetradă fiind egală cu valoarea cifrei din denumirea codului.

Fie X $\in \{0, 1, \ldots, 9\}$ şi $z_{(x)} = (y_3\, y_2\, y_1\, y_0)_{(2)}$. Fiecărei cifre binare (bit) y_j, $j = \overline{0,3}$, i se ataşează o pondere p_j astfel încât: $x = \sum_{k=0}^{3} y_k p_k$.

Câteva coduri ponderate sunt prezentate în tabelul 2.1.

În codul *8421*, cuvintele de cod sunt numere succesive în sistemul binar natural şi din acest motiv, codul se mai numeşte *cod zecimal-binar natural* (*NBCD*)

Codurile *2421* şi *4221* au pentru primele 4 cifre zecimale aceeaşi exprimare ca şi codul 8421. Codul pentru 5 se obține inversând codul pentru 4; la fel se obține 6 din 3, 7 din 2, 8 din 1 şi 9 din 0. Codurile cu această proprietate se numesc *coduri autocomplementare*.

	8421	2421	4221	7421
0	0000	0000	0000	0000
1	0001	0001	0001	0001
2	0010	0010	0010	0010
3	0011	0011	0011	0011
4	0100	0100	0110	0100
5	0101	1011	1001	0101
6	0110	1100	1100	0110
7	0111	1101	1101	0111
8	1000	1110	1110	1001
9	1001	1111	1111	1010

Figura 2.1 Coduri ponderate

II. Coduri neponderate

Exemple de coduri neponderate sunt:

- codul *binar reflectat*
- codul *8421 cu bit de paritate*
- codul *exces 3*
- codul *2 din 5*

Codul “binar reflectat” se obține se obține prin “reflectări repetate” a codurilor pe n-1 ranguri, adăugând biți 0 într-unul din domenii şi biți 1 în celălalt domeniu (cele 2 domenii sunt separate prin planul de oglindire). Exemple de coduri “binar reflectate” sunt *codul Gray* şi *codul Gray închis*.

Codul *Gray* prezintă proprietatea de *adiacență*: trecerea de la o cifră zecimală la următoarea se face prin modificarea unui singur bit din cuvântul de cod. Acest cod este util în cazul mărimilor ce cresc succesiv.

În mediile puternic influențate de zgomot, verificarea transmiterii corecte a informațiilor se face prin folosirea *codurilor detectoare de erori*. În *codul 8421 cu bit de paritate*, fiecare cuvânt de cod are un număr par sau impar de biți 1. La emisie se adaugă un bit 1 sau un bit 0 astfel încât numărul de biți 1 să fie par sau impar. La recepție se numără biții 1, numărul acestora putând indica dacă au apărut erori constând în modificarea unui număr impar de biți din structura cuvântului binar.

Codul *exces 3* se obține din *codul 8421* la care se adună 3 = 0011. În acest fel se poate face distincție între 0 şi lipsa informației (zero este codificat prin 0011 şi nu prin 0000, fiind numit uneori „zero viu”).

Codul *2 din 5* se caracterizează printr-un cuvânt de cod de 5 biți, din care numai doi biți sunt 1. Se realizează astfel o unicitate a reprezentării deoarece numai 10 din cele 32 de configurații posibile pe 5 biți satisfac această condiție. Prin folosirea acestui cod se pot detecta erorile multiple apărute la transmiterea informației.

Principalele coduri neponderate sunt prezentate în tabelul din figura 2.2.

Alte tipuri de coduri sunt codurile *detectoare de erori* şi codurile *corectoare de erori*, care, pe lângă detecția erorilor asigură şi corectarea lor.

	Binar reflectat	**Gray**	**Gray închis**	**8421** cu paritate impară	**exces3**	**2 din 5**
0	0000	0000	0010	00000	0011	00011
1	0001	0001	0110	10001	0100	00101
2	0011	0011	0111	10010	0101	00110
3	0010	0010	0101	00011	0110	01001
4	0110	0110	0100	00100	0111	01010
5	0111	0111	1100	00101	1000	01100
6	0101	0101	1101	00110	1001	10001
7	0100	0100	1111	10111	1010	10010
8	1100	1100	1110	11000	1011	10100
9	1101	1101	1010	01001	1100	11000
10	1111					
11	1110					
12	1010					
13	1011					
14	1001					
15	1000					

Figura 2.2 Coduri neponderate

2.1.3 Coduri ponderate particulare

Codul ponderat *8421* este cel mai răspândit fiind particularizat pentru reprezentarea cifrelor in diverse baze de numerație. Deoarece fiecare bit are ponderea numărului în binar şi cuvintele de cod sunt chiar numerele succesive în sistemul binar natural, acest cod se mai numeşte *cod zecimal binar natural* (NBCD, *Natural Binary Coded Decimal*). El cuprinde cifrele binare de la 0 la 10. Atunci când codifică toate combinațiile binare pe 4 biți este numit cod BCD (*Binary Coded Decimal*).

În funcție de baza de numerație a numărului care trebuie codificat putem avea şi alte tipuri de coduri (octal-binar, hexazecimal-binar).

Tabelul din figura 2.3 prezintă principalele coduri binare uzuale.

Codul *octal-binar* realizează corespondența biunivocă între cifrele sistemului de numerație în baza *8* şi triadele binare succesive.

Codul *hexazecimal - binar* realizează corespondența biunivocă între cifrele sistemului de numerație în baza 16 şi tetradele binare succesive corespunzătoare.

Număr zecimal	**BCD**	**NBCD**	**octal**	**hexa**
0	0000	0000	0000	0
1	0001	0001	0001	1
2	0010	0010	0010	2
3	0011	0011	0011	3
4	0100	0100	0100	4
5	0101	0101	0101	5
6	0110	0110	0110	6
7	0111	0111	0111	7
8	1000	1000		8
9	1001	1001		9
10	1010			A
11	1011			B
12	1100			C
13	1101			D
14	1110			E
15	1111			F

Figura 2.3 Coduri binare uzuale

2.2 Coduri alfanumerice

În cazul acestor coduri, mulțimea *X* a informațiilor primare este formată din cifre, litere, semne ortografice, comenzi speciale, denumite în general *caractere*. Codificarea datelor alfanumerice este necesară pentru vehicularea diferitelor mesaje. Trebuie codificate minim 88 caractere distincte (2 x 6 litere, 10 cifre, 26 de caractere speciale), deci sunt necesari minimum 7 biți.

Cel mai răspândit cod alfanumeric este codul ASCII (*American Standard Code for Information Interchange*).

Codul ASCII codifică 128 de caractere (cele 52 de litere, majuscule şi minuscule, ale alfabetului englez, cele 10 cifre zecimale, caractere speciale şi caractere de comandă).

Codul EBCDIC (*Extended Binary Coded Decimal Interchange Code*) codifică 136 de caractere. Există caractere ASCII care nu au corespondent EBCDIC şi invers.

Codul ASCII, datorită succesiunii caracterelor majuscule şi minuscule, poate fi utilizat pentru ordonări alfabetice.

Exemplu:
Caracterul „?” în ASCII are codul 3F iar în EBCDIC are codul 6F. Deasemenea, caracterele „0”, „A”, „a” au reprezentările ASCII 30, 41, 61 iar cele EBCDIC F0, C1, 81.