ARHITECTURA CALCULATOARELOR
Lucrarea de laborator nr. 4

A.Coduri binare

Un cod binar este un grup de n biți care formează 2^n combinații distincte de cifre 1 şi 0, fiecare combinație reprezentând codificarea unei singure componente din setul de elemente care se doreşte a fi codificat. Spre exemplu, un set de 4 elemente poate fi codificat folosind un cod pe doi biți; un set de 8 elemente necesită un cod pe 3 biți, respectiv un set de 16 elemente necesită un cod pe 4 biți.

A.1. Codul BCD

Un cod binar poate avea neasignate anumite combiații de biți dacă numărul de elemente care se doreşte a fi codificat nu este o putere a lui 2. De exemplu, cele zece cifre zecimale pot fi codificate folosind un cod pe minim patru biți, adică 2^4=16 combinații, din care 6 vor rămâne nefolosite.

Pot fi obținute multe coduri prin simpla aranjare a celor 4 biți în diverse combinații. Unul dintre cele mai folosite coduri pentru cele zece cifre zecimale se numeşte **codul BCD (binary-coded decimal)**. Codificarea cifrelor 0-9 folosind codul BCD este dată mai jos:

Cifrele zecimale	**Codificarea folosind codul BCD**
0	**0000**
1	**0001**
2	**0010**
3	**0011**
4	**0100**
5	**0101**
6	**0110**
7	**0111**
8	**1000**
9	**1001**
10	0001 0000
20	0010 0000
50	0101 0000
99	1001 1001
248	0010 0100 1000

Este important de înțeles diferența între conversia în binar şi codificarea în binar a numerelor zecimale. De exemplu numărul 99 este reprezentat în binar prin şirul de biți 1100011, şi codificat în cod BCD prin şirul de biți 1001 1001.

Singura diferență între un număr zecimal scris cu cifrele de la 1 la 9 şi acelaşi număr scris în cod BCD sunt simbolurile, numărul în sine este exact acelaşi.

În consecință, aritmetica ce guvernează reprezentarea în zecimal, va fi aplicabilă şi asupra numerelor reprezentate în cod BCD.

A.2. Codul Exces-3

Este tot un cod pe 4 biți. Codificarea fiecărei cifre zecimale se obține prin adunarea valorii 3 la codificarea în cod BCD corespunzătoare. Se obține următoarea codificare:

Cifrele zecimale	**Codificarea folosind EXCES-3**
0	**0011**
1	**0100**
2	**0101**
3	**0110**
4	**0111**
5	**1000**
6	**1001**
7	**1010**
8	**1011**
9	**1100**
10	**0100 0011**
20	**0101 0011**
50	**1000 0011**
99	**1100 1100**
248	**0101 0111 1011**

Acest cod simplifică mult implementarea hardware a aritmeticii BCD. Spre exemplu, complementul față de 9 al fiecărei cifre poate fi obținut simplu (este complementul față de 1 al codului).

Să presupunem operația de adunare între două numere zecimale. În BCD, cînd suma depăşeşte 9, este adăugat un 6 pentru a aduce rezultatul în gama 0-9. Deci este generat un carry în acest proces care se foloseşte la însumarea cifrelor de pe rangurile superioare. În codificarea Exces-3, nu mai este necesară verificarea dacă suma depăşeşte 9, deoarece corecția poate fi realizată prin simpla verificare dacă a rezultat carry sau nu.

Exemple de adunare:

a.

În zecimal:		în BCD:		în excess-3:	
3	+	0011	+	0110	+
2		0010		0101	
5		0101		1011	
		Suma<9: fără corecție		Carry=0: se face scădere cu 3:	-
				0011	
				1000	

b.

În zecimal:		în BCD:		în excess-3:	
5	+	0101	+	1000	+
6		0110		1001	
11		1011		10011	
		Suma>9: se adaugă 6	+	Carry=1: se face adunare cu 3:	+
		0110		0011	
		¬ 1| 0001		**¬ 1| 0110**	

¬ : carry la operația următoare

c.

În zecimal:		în BCD:			în excess-3:		
35	+	0011	0101	+	0110	1000	+
18		0001	1000		0100	1011	
53		0100	1101		**Carry**: 0001	10011	
					1011		
		Se adună transportul 1 de la rangul inferior	Suma>9: se adaugă 6	+	Carry=0: se face scădere cu 3:	Carry=1: se face adunare cu 3:	+
		0001	0110		0011	0011	
		0101	**¬ 1| 0011**		**1000**	**carry ¬ 1| 0110**	

A.3. Coduri alfanumerice

Un set de caractere alfanumerice este un set de elemente care include cele 10 cifre zecimale, cele 26 de litere al alfabetului şi un număr de caractere speciale, cum ar fi $,&,#,+, @ și altele. Un astfel de set de caractere conține între 32 şi 64 elemente (dacă sunt incluse numai caractere majuscule) sau între 64 şi 128 caractere dacă sunt incluse atât caractere majuscule, cât şi minuscule. În primul caz, codul binar va necesita 6 biți, iar în cel de-al doilea 7 biți. **Codul standard pentru reprezentări ale caracterelor alfanumerice este denumit ASCII (abrevierea de la American Standard Code for Information Interchange)** care foloseşte 7 biți pentru a codifica 128 de caractere. Mai jos se oferă o listă a caracterelor codificate în cod ASCII.

Caracterele 0-9 din codul ASCII pot fi convertite în cod BCD prin îndepărtarea biților ce corespund primelor trei ranguri superioare (adică 011).

În prezent, codul ASCII se reprezintă în calculatoarele numerice pe un octet (8 biți) prin adăugarea unui 0 pe poziția celui mai semnificativ bit al octetului; acest lucru permite şi o extensie a codului ASCII, aşa-numitele caractere ASCII extinse, în număr de 128.

Exemplu de mesaj codificat în ACII:

01001000	01100101	01101100	01101100	01101111	00101110
H	e	l	l	o	.

B. Coduri detectoare şi corectoare de erori

În fluxul prelucrării automate a datelor, operația de transfer a informațiilor prin intermediul unui canal de comunicație este extrem de frecventă. Cu ocazia acestui transfer de informație, la nivelul canalului de comunicație pot apare perturbații. Pentru a nu se altera conținutul informațional este necesară protejarea informațiilor împotriva acestor perturbații. Procedura presupune adăugarea unor informații suplimentare, necesare în primul rând detectării erorilor şi ulterior corectării acestora.

B.1. Coduri detectoare de erori

Una din modalitățile practice cel mai frecvent utilizate în detectarea erorilor este reprezentată de **codurile pentru controlul parității**. La emisia unei secvențe binare de n biți se ataşează o cifră binară suplimentară, numită cifră de control, astfel încât:

- în cazul parității pare, numărul pozițiilor binare din şir care au valoarea 1 să fie par (inclusiv cifra de control)
- în cazul parității impare, numărul pozițiilor binare din şir (inclusiv cifra de control) care au valoarea 1 să fie impar

La recepția unui caracter se efectuează suma cifrelor binare care au fost primite. Dacă această sumă respectă convenția de paritate stabilită, mesajul este considerat corect recepționat. În caz contrar se semnalează eroarea, solicitându-se reluarea transmisiei. Acest tip de cod este implementat de regulă hardware la nivelul dispozitivelor fizice ale sistemului de calcul. El are două forme de reprezentare:

- VRC – control de paritate vertical
- LRC – control de paritate orizontal (longitudinal)

Distanța Hamming definește distanta logică între două cuvinte de cod valide si este măsurată prin numărul de biți prin care diferă acestea. Pentru codul ASCII aceasta distanță este egală cu 1. Adăugând un singur bit redundant, la codul ASCII al fiecărui caracter alfanumeric, se poate detecta o singură eroare, întrucât codul eronat se va plasa intre două coduri valide ASCII. O metodă de recodificare a codului ASCII, pentru a obține o distanță Hamming egala cu 2, constă în introducerea unui bit de paritate, plasat la stânga codului normal ASCII. Acest bit va fi calculat pe baza sumei modulo 2 a biților egali cu 1 din codul ASCII. In cazul convenției de paritate pară, bitul de paritate va fi egal cu rezultatul sumei modulo 2, amintita mai sus, iar in cazul parității impare acesta va fi egal cu valoarea negata a acesteia.

P	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Caracterul
1	1	1	0	0	1	0	0	d
0	1	1	0	0	1	0	1	e
0	1	1	0	0	1	1	0	f
1	1	1	0	0	1	1	1	g
0	1	0	0	0	1	0	0	D

Bit de paritate Codul ASCII

B.2. Coduri corectoare

În anumite situații, simpla detectare a erorilor este ineficientă. În vederea eliminării, fie şi numai parțiale, a acestui neajuns, au fost proiectate coduri care, pe lângă detectarea erorilor, oferă şi posibilitatea corectării cifrelor binare eronate. Cele mai semnificative sisteme de coduri din această categorie sunt:

- codurile de paritate încrucişată
- codul HAMMING

La codurile de paritate încrucişată succesiunea de cifre binare care se transmit se divide într-un număr de secvențe binare de aceeaşi lungime n, astfel încât succesiunea va fi formată dintr-un număr variabil de linii şi un număr fix de coloane. La emisie fiecărei linii şi fiecărei coloane i se va ataşa o cifră de control de paritate. Practic se aplică LRC pentru linii şi VRC pentru coloane. La recepție, prin controlul parității, pe fiecare linie şi coloană se detectează eventualele erori, având posibilitatea de a corecta o singură eroare detectată la intersecția liniei şi a coloanei care nu respectă paritatea stabilită.

Exemplu:

Paritatea simplă pară:

1 0 0 1 0 0 1 **1** - par
bitul de control

Paritatea încrucişată pară:

VRC

1 0 0 1 0 0 1 1 ¬ LRC
1 1 0 1 0 1 1 1
1 0 0 0 1 0 0 0
1 1 0 0 1 1 0 0 ¬ VRC

Codul Hamming este un cod autocorector bazat pe teste de paritate. Versiunea cea mai simplă permite corectarea unui bit eronat. Celor m biți de informație li se adaugă k biți de control al parității. Deci avem n = m + k biți necesari pentru transmiterea informației.

Deoarece trebuie indicate n + 1 posibilități de eroare (inclusiv absența erorii) prin cei k biți de control, trebuie ca $2^k \geq n + 1$. Cele 2^k posibilități de de codificare pe k biți servesc la determinarea poziției erorii, apoi se poate corecta bitul eronat.

Dacă se numerotează biții de la dreapta spre stânga pornind de la 1, biții de control (sau de paritate) sunt plasați pe poziția puterilor lui 2 (biții cu numărul 1, 2, 4, 8, 16, ...). Fiecare bit de control efectuează control de paritate (pară sau impară) asupra unui anumit număr de biți de date. Se determină astfel cei n biți de transmis sau de stocat.

Exemplu:

Dacă m = 4 se poate construi un cod Hamming pe 7 biți (n = 7), adăugând 3 biți de control (k = 3).

7	6	5	4	3	2	1
m_4	m_3	m_2	k_3	m_1	k_2	k_1

Cei trei biți de control sunt plasați pe poziția puterilor lui 2:

$k_1 \rightarrow 1$;
$k_2 \rightarrow 2$;
$k_3 \rightarrow 4$.

Vom vedea acum, pentru fiecare bit al mesajului care sunt biții de control care permit verificarea parității sale:

$7 = (0111)_2 = 4 + 2 + 1$ → 7 este controlat de k_3, k_2, k_1;
$6 = (0110)_2 = 4 + 2$ → 6 este controlat de k_3, k_2;
$5 = (0101)_2 = 4 + 1$ → 5 este controlat de k_3, k_1;
$4 = (0100)_2 = 4$ → 4 este controlat de k_3;
$3 = (0011)_2 = 2 + 1$ → 3 este controlat de k_2, k_1;
$2 = (0010)_2 = 2$ → 2 este controlat de k_2;
$1 = (0001)_2 = 1$ → 1 este controlat de k_1;

Problema se pune şi invers: care sunt pozițiile binare controlate de către fiecare cod?

k_1 controlează biții cu numerele 1, 3, 5, 7;
k_2 controlează biții cu numerele 2, 3, 6, 7;

k_3 controlează biții cu numerele 4, 5, 6, 7.

Când se recepționează informația, se efectuează controlul de paritate. Pentru fiecare bit de control se compară valoarea transmisă cu cea recalculată. Dacă cele două valori sunt identice, se atribuie valoarea 0 unei variabile binare Ai asociată bitului de control k_i, altfel, Ai primește valoarea 1.

Valoarea zecimală a configurației binare formată din variabilele A_k, A_{k-1}, ..., A_1 furnizează poziția bitului eronat, care se poate corecta.

Exemplificare:

Presupunem că pentru k_1, $A_1 = 1$, pentru k_2, $A_2 = 1$, iar pentru k_3, $A_3 = 0$. Eroarea se găseşte în poziția $(A_3A_2A_1)_2 = (011)_2 = 3$.

Într-adevăr, k1 poate detecta o eroare în pozițiile 1, 3, 5, 7, k_2 poate detecta o eroare pe pozițiile 2, 3, 6, 7, iar k_3 poate specifica o eroare pe pozițiile 4, 5, 6, 7. O eroare detectată de k_1 şi k_2 dar nu şi de k_3 nu poate proveni decât din bitul 3. de exemplu:

$(A_3A_2A_1)_2 = (000)_2$ → indică absența unei erori;
$(A_3A_2A_1)_2 = (001)_2$ → indică eroare pe bitul 1;
$(A_3A_2A_1)_2 = (110)_2$ → indică eroare pe bitul 6.

Exemplu de recepționare a unui mesaj: $(1011100)_2$.

Ştiind că s-a utilizat un CH cu paritate pară, să se reconstituie mesajul inițial. n = 7, deci k = 3, m = 4.

Număr	7	6	5	4	3	2	1
Tip	m_4	m_3	m_2	k_3	m_1	k_2	k_1
Valoare	1	0	1	1	1	0	0

$k_1 = 0$ controlează pozițiile 1, 3, 5, 7, nu se verifică, deci $A_1 = 1$;
$k_2 = 0$ controlează pozițiile 2, 3, 6, 7, se verifică, deci $A_2 = 0$;
$k_3 = 0$ controlează pozițiile 4, 5, 6, 7, nu se verifică, deci $A_3 = 1$;

Adresa binară a erorii $(A_3A_2A_1)_2 = (101)_2 = 5$. Bitul cu numărul 5, care este egal cu 1 este eronat. Mesajul inițial corectat şi fără biții de control este:

$(1001)_2$

ARHITECTURA CALCULATOARELOR
Lucrarea de laborator nr. 4

Exerciții 1:

1. Să se codifice în cod BCD următoarele valori zecimale:
   a. 23 b. 589 c. 341 d. 452 e. 290 f. 256
2. Ce valori zecimale au fost codificate folosind codul BCD?
   a. 1001.1000 b. 0001.0100.0110 c. 0100.0101.0110.1001
   d. 0011.0000.0111 e. 0011.0010.1001 f. 0011.1000
3. Să se codifice în cod BCD şi Exces-3 următoarele valori zecimale:
   a. 45 b. 89 c. 673 d. 561 e. 231 f.725
4. Să se codifice în BCD şi Exces-3 operanzii expresiilor de mai jos şi să se efectueze operațiile de adunare respective:

a. 3 +	b. 42 +	c. 77 +	d. 69 +
5	7	13	26

5. Iată un mesaj codificat în ASCII utilizându-se opt biți pe simbol. Care este conținutul mesajului ?
   01000011 01101111 01101101 01110000 01110101 01110100
   01100101 01110010 00100000 01010011 01100011 01101001
   01100101 01100011 01100101
6. Codificați următoarele propoziții în codul ASCII:
   a. Unde este el ?
   b. “cum ? “ spuse el.
   c. 2+3=5.
7. În cazul codului ASCII, care este relația dintre codurile pentru o literă mare şi aceeaşi literă din alfabet mică ?
8. Codificați următoarele expresii în cod ASCII utilizând un octet pe caracter. Folosiți cel mai semnificativ bit al fiecărui octet de paritate(paritate impară).
   a. 100/5=20
   b. To be or not to be?
   c. Costul total este de $7.25.
9. Următorii octeți au fost codificați la origine utilizând paritate impară. În care dintre ele detectați producerea unei erori?
   a. 10101101 b.10000001 c.00000000 d.11100000 e.11111111
10. Următorul mesaj a fost transmis inițial cu paritate impară pentru fiecare cuvânt binar. În care dintre cuvinte au apărut în mod precis erori?
11011 01011 10110 00000 11111 10101 10001 00100 01110

11. E posibil ca într-unul dintre octeții de la întrebarea 10 să fi apărut erori pe care să nu le puteți detecta? Justificați-vă răspunsul.
12. Cum se modifică răspunsurile pe care le-ați dat la întrebările 10 si 11 dacă s-ar utiliza paritatea pară în locul parității impare?

13. Să presupunem că este generat un cod de 24 biți prin reprezentarea fiecărui simbol prin trei copii consecutive ale codului său ASCII (de exemplu, simbolul A este reprezentat de şirul de biți 010000010100000101000001). Care sunt proprietățile acestui nou cod în ceea ce priveşte corecția erorilor?
14. Să se precizeze ce distanțe Hamming sunt între cuvintele din codul prezentat mai jos ; se pot corecta erorile dintr-un mesaj codificat folosind acest cod ?

Simbol	Cod
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

15. Utilizând codul corector de erori prezentat în tabelul de mai sus, decodificați următoarele mesaje:

a. 001111 100100 001100
b. 010001 000000 001011
c. 011010 110110 100000 011100

16. Construiți un cod pentru caracterele A, B, C şi D utilizând cuvinte cu lungimea de cinci biți astfel încât distanța Hamming dintre două cuvinte de cod să fie minimum trei.
17. Utilizând codul corector de erori de la problema 14, decodificați următoarele cuvinte:
a. 111010 110110
b. 101000 100110 001100
c. 011101 000110 000000 010100
d. 010010 001000 001110 101111 000000 110111 100110
e. 010011 000000 101001 100110

ARHITECTURA CALCULATOARELOR
Lucrarea de laborator nr. 4

Exerciții 2:

1. Să se codifice în cod BCD următoarele valori zecimale:
   a. 34 b538 c. 421 d. 259 e. 702 f. 512
2. Ce valori zecimale au fost codificate folosind codul BCD?
   a. 1000.1001 b. 0011.0100.0110 c. 0010.0100.0100.1001
   d. 0011.0001.0101 e. 0011.0011.0001 f. 0011.1001
3. Să se codifice în cod BCD şi Exces-3 următoarele valori zecimale:
   a. 45 b. 97 c. 534 d. 522 e. 371 f.792
4. Să se codifice în BCD şi Exces-3 operanzii expresiilor de mai jos şi să se efectueze operațiile de adunare respective:

a. 4 +	b. 22 +	c. 17 +	d. 79 +
1	7	13	56

5. Iată un mesaj codificat în ASCII utilizându-se opt biți pe simbol. Care este conținutul mesajului ?
   01000011 01101111 01101101 01110000 01110101 01110100
   01100101 01110010 00100000 01010011 01100011 01101001
   01100101 01100011 01100101
6. Codificați următoarele propoziții în codul ASCII:
   a. Unde esti tu ?
   b. “ce ? “ spuse ea.
   c. 2+6=8.
7. În cazul codului ASCII, care este relația dintre codurile pentru o literă mare şi aceeaşi literă din alfabet mică ?
8. Codificați următoarele expresii în cod ASCII utilizând un octet pe caracter. Folosiți cel mai semnificativ bit al fiecărui octet de paritate (paritate impară).
   a. 50/5=10
   b. To be or not to be?
   c. Costul real este de $8.99.
9. Următorii octeți au fost codificați la origine utilizând paritate impară. În care dintre ele detectați producerea unei erori?
   a. 10101101 b.10000001 c.00000000 d.11100000 e.11111111
10. Următorul mesaj a fost transmis inițial cu paritate impară pentru fiecare cuvânt binar. În care dintre cuvinte au apărut în mod precis erori?
11011 01011 10110 00000 11111 10101 10001 00100 01110
11. E posibil ca într-unul dintre octeții de la întrebarea 10 să fi apărut erori pe care să nu le puteți detecta? Justificați-vă răspunsul.
12. Cum se modifică răspunsurile pe care le-ați dat la întrebările 10 si 11 dacă s-ar utiliza paritatea pară în locul parității impare?
13. Să presupunem că este generat un cod de 24 biți prin reprezentarea fiecărui simbol prin trei copii consecutive ale codului său ASCII (de exemplu, simbolul A

este reprezentat de şirul de biți 010000010100000101000001). Care sunt proprietățile acestui nou cod în ceea ce priveşte corecția erorilor?

14. Să se precizeze ce distanțe Hamming sunt între cuvintele din codul prezentat mai jos ; se pot corecta erorile dintr-un mesaj codificat folosind acest cod ?

Simbol	Cod
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

15. Utilizând codul corector de erori prezentat în tabelul de mai sus, decodificați următoarele mesaje:

a. 001111 100100 001100
b. 010001 000000 001011
c. 011010 110110 100000 011100

16. Construiți un cod pentru caracterele A, B, C şi D utilizând cuvinte cu lungimea de cinci biți astfel încât distanța Hamming dintre două cuvinte de cod să fie minimum trei.

17. Utilizând codul corector de erori de la problema 14, decodificați următoarele cuvinte:

a. 111010 110110
b. 101000 100110 001100
c. 011101 000110 000000 010100
d. 010010 001000 001110 101111 000000 110111 100110
e. 010011 000000 101001 100110