

L3. REPREZENTAREA INFORMAȚIEI PRIN CODURI. OPERAȚII ARITMETICE.

1. Obiective

Prin parcurgerea acestei ședințe de laborator studenții vor fi capabili:

- Să reprezinte aritmetic numerele naturale;
- Să reprezinte numerele întregi în cod direct;
- Să reprezinte numerele întregi în cod complementar;
- Să reprezinte numerele întregi în cod invers;
- Să reprezinte cifrele zecimale prin coduri zecimal-binare;
- Să efectueze operații cu numere întregi reprezentate în cod invers și în cod complementar.

2. Reprezentarea aritmetică a numerelor naturale

Numerele naturale pot fi reprezentate pe un număr fix de poziții binare (biți), de obicei 8, 16, 32 sau 64. Reprezentarea aritmetică mai este numită și format cu virgulă fixă.

n	$n - 1$	$n - 2$		2	1
			...		
2^{n-1}	2^{n-2}	2^{n-3}		2^1	2^0

Numărul maxim care poate fi reprezentat pe n poziții binare este $2^n - 1$. Astfel, fiind disponibili n biți, se pot reprezenta numerele naturale din intervalul $[0; 2^n - 1]$.

3. Reprezentarea numerelor întregi în semn-mărime (cod direct)

Reprezentarea numerelor întregi în semn-mărime este similară reprezentării aritmetice a numerelor naturale, cu specificația că prima poziție din stânga este ocupată de semnul numărului întreg S care este egal cu 0 dacă numărul este pozitiv sau cu 1 dacă numărul este strict negativ.

n	$n - 1$	$n - 2$		2	1
			...		
S	2^{n-2}	2^{n-3}		2^1	2^0

În acest caz, numărul N poate fi scris

$$N = (-1)^{b_{n-1}} \sum_{i=0}^{n-2} b_i \cdot 2^i$$

Cunoscând numărul de biți necesari reprezentării, atunci intervalul maxim admis de reprezentarea cod direct este $[-2^{n-1} + 1; 2^{n-1} - 1]$.

4. Reprezentarea numerelor în complement față de 2 (cod complementar)

În cazul numerelor pozitive, reprezentarea în cod complementar este aceeași cu reprezentarea în semn-mărime. În cazul numerelor negative,

$$N = -b_{n-2}b_{n-3} \dots b_1b_0 = - \sum_{i=0}^{n-2} b_i \cdot 2^i$$

complementul său față de 2 se poate scrie

$$\begin{aligned} N_c &= 2^n - \sum_{i=0}^{n-2} b_i \cdot 2^i = 2^n - 1 + 1 - \sum_{i=0}^{n-2} b_i \cdot 2^i = 2^{n-1} + \sum_{i=0}^{n-2} 2^i + 1 - \sum_{i=0}^{n-2} b_i \cdot 2^i \\ &= 2^{n-1} + \sum_{i=0}^{n-2} (1 - b_i) \cdot 2^i + 1 = 2^{n-1} + \sum_{i=0}^{n-2} \bar{b}_i \cdot 2^i = 1\bar{b}_{n-2}\bar{b}_{n-3} \dots \bar{b}_1\bar{b}_0 + 1. \end{aligned}$$

Regula 1. Se reprezintă numărul în valoare absolută, apoi se inversează bit cu bit inclusiv bitul de semn și se adună 1 la cel mai puțin semnificativ bit.

Regula 2. Se consideră numărul N reprezentat pe n biți, atunci numărul n reprezentat în cod complementar va fi $N_c = 2^n - |N|$.

Regula 3. Codul complementar al unui număr negativ N se obține prin inversarea biților reprezentării cu semn a valorii absolute, plecând de la dreapta spre stânga, cu excepția primului bit de 1 și a tuturor biților de 0 dinaintea sa.

Considerând că se folosesc pentru reprezentare n biți, domeniul de valori reprezentabile este $[-2^{n-1}; 2^{n-1} - 1]$.

Operații aritmetice cu numere cu semn în cod complementar

4.1. Adunarea a două numere pozitive

Cazul 1
(nu se depășește domeniul de reprezentare)
 $x = 30; y = 50; x + y = 80.$

$$\begin{array}{r} 00011110 \\ + 00110010 \\ \hline 01010000 = 80. \end{array}$$

Cazul 2
(se depășește domeniul de reprezentare)
 $x = 30; y = 100; x + y = 130.$

$$\begin{array}{r} 00011110 \\ + 1100100 \\ \hline 10000010 = -126. \end{array}$$

$$\begin{array}{r}
 10000010 \quad - \\
 \hline
 1 \\
 10000001 \\
 01111110 \quad = 126.
 \end{array}$$

4.2. Adunarea a două numere negative

Cazul 1

(nu se depășește domeniul de reprezentare)

$$x = -30; y = -50; x + y = -80.$$

$$\begin{array}{r}
 30 = 00011110 \quad 50 = 00110010 \\
 11100001 \quad + \quad 11001101 \quad + \\
 \hline
 1 \quad 1 \\
 -30 = 11100010 \quad -50 = 11001110
 \end{array}$$

$$\begin{array}{r}
 11100010 \quad + \\
 11001110 \\
 \hline
 1 \leftarrow 10110000 \\
 \text{se neglijează} \\
 10110000 = -80
 \end{array}$$

$$\begin{array}{r}
 10110000 \quad - \\
 \hline
 1 \\
 1010111 \\
 01010000 \quad = 80.
 \end{array}$$

Cazul 2

(se depășește domeniul de reprezentare)

$$x = -30; y = -100; x + y = -130.$$

$$\begin{array}{r}
 30 = 00011110 \quad 100 = 01100100 \\
 11100001 \quad + \quad 10011011 \quad + \\
 \hline
 1 \quad 1 \\
 -30 = 11100010 \quad -100 = 10011100
 \end{array}$$

$$\begin{array}{r}
 11100010 \quad + \\
 10011100 \\
 \hline
 1 \leftarrow 01111110 \\
 \text{se neglijează} \\
 01111110 \quad = \\
 126.
 \end{array}$$

4.3. Adunarea a două numere cu semne diferite

$$x = -30; y = 50; x + y = 20.$$

$$\begin{array}{r}
 30 = 00011110 \quad 50 = 00110010 \\
 11100001 \quad + \\
 \hline
 1 \\
 -30 = 11100010 \quad 20 = 00010100 \\
 11100010 \quad + \\
 00110010 \\
 \hline
 1 \leftarrow 00010100 \quad = 20 \\
 \text{se neglijează}
 \end{array}$$

5. Reprezentarea numerelor în complement față de 1 (cod invers)

Pentru numerele pozitive, reprezentarea în cod invers este identică cu reprezentarea în semn-mărime. În cazul numerelor negative, numărul în cod invers are forma $N_i = 1\overline{b_{n-2}b_{n-3} \dots b_1b_0}$.

Regula 1. Se reprezintă valoarea absolută a numărului în cod direct și se inversează toți biții, inclusiv semnul.

Regula 2. Se folosește definiția $N_i = 2^n - 1 - |N|$.

Domeniul de valori reprezentabile în cod invers este $[-2^{n-1} + 1; 2^{n-1} - 1]$.

Operații aritmetice cu numere cu semn în cod invers

5.1. Adunarea a două numere pozitive

Adunarea a două numere pozitive este similară cu adunarea în cod complementar.

5.2. Adunarea a două numere negative

Cazul 1

(nu se depășește domeniul de reprezentare)

$$x = -30; y = -50; x + y = -80.$$

$$-30 = 11100001$$

$$-50 = 11001101$$

$$\begin{array}{r} 11100001 + \\ 11001101 \\ \hline 1 \leftarrow 10101110 + \\ | \text{-----} > 1 \\ \hline 10101111 = -80 \end{array}$$

Cazul 2

(se depășește domeniul de reprezentare)

$$x = -30; y = -100; x + y = -130.$$

$$-30 = 11100001$$

$$-100 = 10011011$$

$$\begin{array}{r} 11100001 + \\ 10011011 \\ \hline 1 \leftarrow 01111100 + \\ | \text{-----} > 1 \\ \hline 01111101 = 125? \end{array}$$

5.3. Adunarea a două numere cu semne diferite

$$x = 50; y = -30; x + y = 20.$$

$$50 = 00110010$$

$$-30 = 11100001$$

$$\begin{array}{r} 00110010 + \\ 11001101 \\ \hline 1 \leftarrow 00010011 + \\ | \text{-----} > 1 \\ \hline 00010100 = 20 \end{array}$$

6. Reprezentarea cifrelor zecimale prin coduri zecimal-binare

Se consideră mulțimea cifrelor zecimale $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Mulțimea cuvintelor de cod va conține cel puțin 10 cuvinte distincte, fiecare cuvânt având cel puțin 4 biți. Se vor obține în total A_{16}^{10} posibilități de codificare. Variantele uzuale de codificări pot fi divizate în coduri ponderate și coduri neponderate.

Un cod ponderat asociază fiecărei cifre zecimale o tetradă binară, iar ponderea fiecărui bit din tetradă este egală cu valoarea cifrei din denumirea codului. Cifra zecimală codificată se obține prin suma biților din cuvântul de cod, ponderați cu valoarea corespunzătoare din denumirea codului.

Coduri zecimal-binare ponderate				
Codul zecimal-binar				
Cifra zecimală	8421 BCD	2421 Codul Aiken	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

Dintre codurile neponderate, cele mai utilizate sunt codul Exces 3 și codul Gray. Codul Exces 3 se obține din cuvântul de cod 8421 al cifrei zecimale respective la care se adaugă 3 (0011 în binar). Codul Gray are proprietatea de adiacență, în sensul că trecerea de la o cifră zecimală la următoarea se face prin modificarea unui singur bit din cuvântul de cod. Dacă se notează cu $a_8 a_4 a_2 a_1$ cifrele binare ale a secvențelor codului 8421 și cu $b_4 b_3 b_2 b_1$ cifrele binare ale secvențelor codului Gray, acestea din urmă pot fi calculate cu ajutorul relațiilor $b_4 = a_8$; $b_3 = a_8 \oplus a_4$; $b_2 = a_4 \oplus a_2$ și $b_1 = a_2 \oplus a_1$.

Operații cu numere reprezentate în BCD

6.1. Adunarea în zecimal codificat binar

Fiecare cifră zecimală se codifică printr-o tetradă binară. Adunarea se efectuează poziție cu poziție de la dreapta spre stânga.

- 1) Dacă rezultatul este în mulțimea $\{0, \dots, 9\}$ acesta nu va necesita corecții.
- 2) Dacă rezultatul este în mulțimea $\{10, \dots, 15\}$ se adună 6 (0110) la suma din poziția respectivă pentru a forța transportul la poziția următoare și se adună transportul la suma de pe poziția următoare.
- 3) Dacă rezultatul este în mulțimea $\{16, \dots, 19\}$, transportul generat se adună la poziția următoare după care se adună 6 (0110) la suma din poziția respectivă.

Exemplu:

5683	+				
2794					
8477					

0101	0110	1000	0011	+
0010	0111	1001	0100	
	1	<--	0001	0111
	1110		0110	7
	0110		0111	
1	<--	0100	7	
1000	4			
8				

6.2. Scăderea în zecimal codificat binar

Fiecare cifră zecimală este codificată printr-o tetradă binară. Se efectuează scăderea poziție cu poziție, dacă nu este necesar împrumut, rezultatul nu necesită corecții. Dacă este necesar împrumutul, se scade 1 de la poziția următoare și se adună 16 la descăzutul poziției curente. Se efectuează scăderea și se scade 6 (0110) din rezultat pentru corecție.

539	-			
275				
264				

0101	-->	10011	1001	-
0010		0111	0101	
0100		1100	0100	
0010		0100	4	
0010		0110		
2		6		

Coduri zecimal-binare neponderate

Cifra zecimală	Exces 3	Gray
0	0011	0000
1	0100	0001
2	0101	0011
3	0110	0010
4	0111	0110
5	1000	0111
6	1001	0101
7	1010	0100
8	1011	1100
9	1100	1101

7. Aplicații propuse

7.1. Să se scrie reprezentarea aritmetică a următoarelor numere naturale pe numărul de biți specificat:

- i) Pe 8 biți
- a) 243 b) 189 c) 38 d) 254
- ii) Pe 16 biți
- a) 3284 b) 7654 c) 5894 d) 65535

7.2. Să se reprezinte în cod direct următoarele numere:

- a) 72 b) -89 c) 124 d) -23
- e) -87 f) -110 g) 109 h) -56

7.3. Să se reprezinte în cod complementar următoarele numere:

- a) 72 b) -89 c) 124 d) -23
- e) -87 f) -110 g) 109 h) -56

7.4. Să se reprezinte în cod invers următoarele numere:

- a) 72 b) -89 c) 124 d) -23
- e) -87 f) -110 g) 109 h) -56

7.5. Se dau numerele zecimale din tabelul de mai jos. Să se reprezinte:

- i) în cod BCD
- | | | | |
|---------|---------|---------|---------|
| a) 1294 | b) 4765 | c) 3762 | d) 8573 |
|---------|---------|---------|---------|
- ii) în codul Aiken
- | | | | |
|---------|---------|---------|---------|
| a) 7651 | b) 5386 | c) 3289 | d) 7248 |
|---------|---------|---------|---------|
- iii) în cod Exces 3
- | | | | |
|---------|---------|---------|---------|
| a) 1539 | b) 3421 | c) 6593 | d) 4618 |
|---------|---------|---------|---------|
- iv) în cod Gray
- | | | | |
|---------|---------|---------|---------|
| a) 5492 | b) 3286 | c) 5693 | d) 5901 |
|---------|---------|---------|---------|

7.6. Să se calculeze folosind reprezentarea în cod complementar și în cod invers următoarele operații:

- | | |
|-----------------|-----------------|
| a) $52 + 63 =$ | b) $-45 + 82 =$ |
| c) $89 - 54 =$ | d) $-88 + 44 =$ |
| e) $-93 + 27 =$ | |

7.7. Să se calculeze folosind reprezentarea BCD:

- | | |
|--------------------|-------------------|
| a) $567 + 893 =$ | d) $1563 - 875 =$ |
| b) $4915 + 3296 =$ | e) $932 - 876 =$ |
| c) $765 + 386 =$ | |

8. Referințe bibliografice

[1] Manta V., Ungureanu F., *Introducere în știința sistemelor și a calculatoarelor*, Volumul I, Editura "Gh.Asachi", Iași, 2002