Universitatea Politehnica Timişoara

P-ţa Victoriei nr. 2
RO 300006 - Timişoara
Tel: +4 0256 403000
Fax: +4 0256 403021
rector@rectorat.upt.ro
www.upt.ro

Logică digitală

-Curs 3-

REPREZENTAREA DATELOR ÎN SISTEME DE CALCUL

- 2021-

Reprezentarea numerelor în sistemele de calcul

- Sisteme de numerație poziționale (binar, octal, hexazecimal);
- Reprezentarea numerelor în virgulă fixă (SM, C1, C2);
- Reprezentarea numerelor de virgulă flotantă;
- Coduri binare pentru numere zecimale;

Sisteme de numere poziționale

- **Sistem pozițional** - un număr este reprezentat printr-un şir de cifre, unde pt. fiecare poziţie a unei cifre este asociată o anumită greutate.

- Valoarea numărului este o sumă a cifrelor înmulţită cu greutatea aferentă:

  Ex1: $1734 = 1*10^3+7*10^2+3*10^1 +4*10^0$

- Fiecare greutate e o putere a lui 10 corespunzătoare poziţiei numărului. Pentru numere cu virgulă folosim puteri negative a lui 10.

Ex2: $5186.67=5*10^3+1*10^2+8*10^1 +6*10^0+6*10^{-1}+7*10^{-2}$

Sisteme de numere poziționale

- **Sistem pozițional** - un număr este reprezentat printr-un şir de cifre, unde fiecare poziţie a unei cifre este asociată o anumită contributie (pondere).

$$D = d_{m-1}\, d_{m-2} \ldots d_1\, d_0\,.\,d_{-1}\, d_{-2} \ldots d_{-n}$$

MSD

Virgula fixă

LSD

Sisteme de numere binare

- Formă generală a unui număr binar

$$D = \boldsymbol{b_{m-1}\, b_{m-2} \ldots b_1\, b_0\, .\, b_{-1}\, b_{-2} \ldots b_{-n}}$$

MSB Virgula fixă LSB

Valoarea:

$$D = \sum_{i=-n}^{p-1} b_i * r^i \quad \text{, unde } r = 2 \text{ radix(bază)}$$

Sisteme de numere binare

Ex.: $N=11001.011_2$

$N= 1*2^4+1*2^3+0*2^2+0*2^1+1*2^0+ 0*2^{-1}+1*2^{-2}+1*2^{-3}=25.375_{10}$

- Baza 8 corespunde sistemului octal. cifre {0,1,2,3,4,5,6,7}

- Baza 16 corespunde sistemului hexazecimal.

  cifre {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Binar, octal, hexazecimal ...

Zecimal	Binar	Octal	Hexazecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversia numerelor

- Conversia din binar în hexazecimal

  Ex: 010011110111.101001010

  - Partiţionarea numărului binar în grupuri de 4 pornind de la virgulă şi inaintand spre dreapta sau stanga :

    0100_1111_0111 . 1010_0101_0000

  - Fiecare grup corespunde unei singure cifre hexazecimale. Folosind Tabelul ant. obţinem:

    4F7.A50

Aplicaţie: Convertiţi numărul din binar în hexazecimal:

111 1100 1010.0111 1111

Raspuns : 7CA.7F

Conversia numerelor

Conversia din binar în octal

- Partiţionarea numărului binar în grupuri de 3 pornind de la virgulă şi inaintand spre dreapta sau stanga.
- Fiecare grup corespunde unei singure cifre din octal.
- Aplicaţie: 111010.11

  111_010.110

  Raspuns: 72.6

Conversia numerelor

- Conversia din zecimal în binar

  - Conversia unui număr întreg din zecimal în binar se face prin algoritmul împărţire succesive ale lui N (nr de convertit) la r (noua bază).

  - Procesul se repetă până se obţine câtul **0**.

Conversia numerelor

- Conversia din zecimal în binar

Ex: Conversia nr zecimal 119:

119: 2 = 59 rest 1 (LSB)

59 : 2= 29 rest 1

29 : 2= 14 rest 1

14 : 2= 7 rest 0

7 : 2= 3 rest 1

3 : 2= 1 rest 1

1 : 2= **0** rest 1 (MSB)

$119_{10}=1110111_2$

Condiția de oprire

Conversia numerelor

Conversia numărului 73 din baza 10 în baza 2

$N9 = 73_{10} = 1001001_2$

$$
\begin{array}{llll}
73/2=36 & \text{rest} & 1 & b_0 \\
36/2=18 & \text{rest} & 0 & b_1 \\
18/2=9 & \text{rest} & 0 & b_2 \\
9/2=4 & \text{rest} & 1 & b_3 \\
4/2=2 & \text{rest} & 0 & b_4 \\
2/2=1 & \text{rest} & 0 & b_5 \\
1/2=0 & \text{rest} & 1 & b_6
\end{array}
$$

Conversia numerelor

- Conversia din zecimal în binar
  - Pentru conversia numerelor fracţionale, se **înmulţeşte** numărul cu noua bază în care se converteşte numărul.
  - **Partea întreagă** a rezultatului devine bit al şirului care reprezintă rezultatul conversiei.
  - Înmulţirea se face până rezultatul devine 0.

Ex: Conversia lui 0.75 în baza 2.

0.75*2= 1.5 parte întreagă 1 (MSB) si fracţionară 0.5

0.5 * 2= 1.0 parte întreagă 1 si fracţionară 0

0.0 * 2= 0.0 parte întreaga 0 (LSB)

Reprezentarea numerelor în virgulă fixă

- Sistemele digitale sunt realizate din circuite ce procesează cifrele binare „0” şi „1”
- **Numerele fără semn:**
  - un şir de „0” şi „1”.
  - Fiecare cifră binară poartă denumirea de bit (**b**inary dig**it**).
  - Val. şirului binar $N = b_{n-1}b_{n-2}...b_1b_0.b_{-1}b_{-2}...b_{m-1}$
  - este dată de formula:

$$N = \sum_{i=-m}^{n-1} b_i * 2^i$$

Reprezentarea numerelor în virgulă fixă

- Pentru reprezentarea numerelor cu semn, se alocă bitul cel mai semnificativ (cel mai din stânga – most significant bit - MSB) semnului:
  - „0” - numere pozitive,
  - „1” - numere negative.
- Uzual în sistemele de calcul se operează fie cu numere întregi, fie cu numere fracţionare;
- Din acest motiv, poziţia virgulei este considerată implicit după cum urmează:
  - **numere întregi** poziţia virgulei este la dreapta bitului cel mai puţin semnificativ: $b_{n-1}b_{n-2}...b_1b_0.$
  - **numere fracţionare** poziţia virgulei este la dreapta bitului cel mai semnificativ, care este şi bitul de semn:

$$b_0.b_{-1}...b_{-n+1}b_{-n}$$

Reprezentarea numerelor în virgulă fixă

- Pentru reprezentarea numerelor cu semn există trei formate de reprezentare:
  - semn-mărime - **SM**,
  - complement de 1 – **C1**,
  - complement de 2 – **C2**.

Reprezentarea numerelor în virgulă fixă

- Reprezentarea în semn-mărime/sign-magnitude (SM)
  - 1 bit semn, n biți mărime (valoare absolută)
  - Valoarea bitului de semn determină semnul
    - 0 – numere pozitive
    - 1 – numere negative
  - domeniul valoric al reprezentării în formatul semn-mărime acesta este între $-2^{n-1}+1$ şi $2^{n-1}-1$
  - Exemplu:

    +85= 0 1010101

    -85= 1 1010101

Reprezentarea numerelor în virgulă fixă

- Semn-mărime/sign-magnitude (SM)
  - Avantaje
    - simplitate
    - negare simpla prin schimbarea bitului de semn
    - implementare facilă a operaţiei de înmulţire
  - Dezavantaje
    - Dublă reprezentare pentru 0 (+0 și -0)
    - Implementare dificilă pentru adunare
      - Exercițiu: (-19) + (+12)

Reprezentarea numerelor în virgulă fixă – SM

- Ex.:

			Semn	Marime				
+	25	=	0	1	1	0	0	1
-	25	=	1	1	1	0	0	1

a.

+	2	=	0	0	1	0	+	
+	3	=	0	0	1	1		
			0	1	0	1	=	+5

b.

-	2	=	1	0	1	0	+		
-	3	=	1	0	1	1			
			0	1	0	1	=	+5	(!)

c.

-	2	=	1	0	1	0	+		
+	3	=	0	0	1	1			
			1	1	0	1	=	-5	(!)

d.

+	2	=	0	0	1	0	+		
-	3	=	1	0	1	1			
			1	1	0	1	=	-5	(!)

Reprezentarea numerelor în virgulă fixă

- Complement de 1/One’s complement (C1)
  - 1 bit semn, *n* biți pentru mărime
  - Numerele pozitive – identic cu SM
  - Numerele negative – complementarea/ negarea marimii
  - Exemplu:

    +85= 0 1010101

    -85= 1 0101010

Reprezentarea numerelor în virgulă fixă – **C1**

- Dezavantaje:
  - C1 **nu** este un format ponderat în conformitate cu notaţia poziţională
  - există **două** reprezentări pentru numărul **zero** (pentru un numar reprezentat pe 6 biţi avem 0 00000, respectiv 1 11111), deci testarea pentru zero se va face de două ori
- Avantaje:
  - o implementare mai facilă a operaţiei de adunare comparativ cu SM
- domeniul valoric pentru numere întregi:

$$-(2^{n-1}-1) \text{ și } 2^{n-1}-1$$

Reprezentarea numerelor în virgulă fixă – **C1**

+	2	=		0	0	1	0	+	
+	3	=		0	0	1	1		
				0	1	0	1	=	+5

a.

-	2	=		1	1	0	1	+	
-	3	=		1	1	0	0		
			1	1	0	0	1	+	
			EAC				1		
				1	0	1	0	=	-5

b.

-	2	=		1	1	0	1	+	
+	3	=		0	0	1	1		
			1	0	0	0	0	+	
			EAC				1		
				0	0	0	1	=	+1

c.

+	2	=		0	0	1	0	+	
-	3	=		1	1	0	0		
				1	1	1	0	=	-1

d.

Reprezentarea numerelor în virgulă fixă

- Complement de 2/ Two’s complement (C2)
  - Numerele pozitive – identic cu SM
  - Numerele negative – negarea valorii pozitive la care se adaugă 1
  - Întregi: $1\overline{b_{n-2}}...\overline{b_1}\,\overline{b_0} + 0.0...01$
  - Fracționare: $1.\overline{b_{-1}}...\overline{b_{-n+1}}\,\overline{b_{-n}} + 0.0...01$
  - Exemplu:

    +85= 0 1010101

    -85= 1 0101011

Reprezentarea numerelor în virgulă fixă – **C2**

- Dezavantaje:
  - Mai dificil de obținut decât SM și C1;
  - nu este un format ponderat în conformitate cu notaţia poziţională
  - anomalia complementului de doi
- Avantaje:
  - **O singura reprezentare pentru 0 !**
    - 0000000
  - Implementarea facilă a operației de adunare
    - Exercițiu (-19) + (+12)

Reprezentarea numerelor în virgulă fixă – **C2**

Număr zecimal	Format SM	Format C1	Format C2
+3	0 11	0 11	0 11
+2	0 10	0 10	0 10
+1	0 01	0 01	0 01
+0	0 00	0 00	0 00
-0	1 00	1 11	
-1	1 01	1 10	1 11
-2	1 10	1 01	1 10
-3	1 11	1 00	1 01
-4	----	----	1 00

Reprezentarea numerelor în virgulă fixă – **C2**

+	2	=	0	0	1	0	+			
+	3	=	0	0	1	1				
			0	1	0	1	=	+	5	

a.

-	2	=	1	1	1	0	+		
-	3	=	1	1	0	1			
			1	0	1	1	=	-	5

b.

-	2	=	1	1	1	0	+		
+	3	=	0	0	1	1			
			0	0	0	1	=	+	1

c.

+	2	=	0	0	1	0	+		
-	3	=	1	1	0	1			
			1	1	1	1	=	-	1

d.

- Domeniul valoric pentru numere întregi:
  - 2^{n-1} și $2^{n-1}-1$

Aplicație:

- Se dau următoarele perechi de numere întregi: +23 şi +18, +23 şi -18, -23 şi +18, respectiv -23 şi -18. Se cere:
  - Să se convertească numerele în formatele semn-mărime, complement de 1, respectiv complement de 2.
  - Să se efectueze adunarea celor două numere.