Universitatea din Pitești
Facultatea de Matematică-Informatică

Bazele aritmetice ale sistemelor de calcul

Cuprins

Descriere generală

Ideea de a reprezenta informația într-un sistem de calcul sub formă logică prin cifrele 0 și 1 aparține cercetătorului C. Shannon.

Definiție

Fie $k \in \mathbb{N}, k \geq 2$. Se numește **bază de numerație** *o mulțime de k simboluri corespunzătoare primelor k numere naturale, $\{0, 1, ..., k-1\}$.*

Definiție

Mulțimea regulilor de reprezentare a unor numere utilizând o bază de numerație se numește **sistem de numerație**.

Descriere generală

Notație

$B_k = \{i \mid 0 \leq i < k\}$

Exemplu

$B_2 = \{0, 1\}$*(sistemul binar)*
$B_8 = \{0, 1, ...6, 7\}$*(sistemul octal)*
$B_{16} = \{0, 1, ...9, A, B, C, D, E, F\}$, *unde*
$A = 10,\ B = 11,\ C = 12,\ D = 13,\ E = 14,\ F = 15$
(sistemul hexazecimal)

Definiție

Trecerea unui număr dintr-un sistem de numerație în altul se numește ***conversie****.*

Reprezentarea unui număr într-o bază de numerație

Definiție

Un număr $x \in \mathbb{Z}_+$ se reprezintă într-o bază b sub forma:

$$x = a_n b^n + a_{n-1} b^{n-1} + ... + a_1 b^1 + a_0 b^0 = \sum_{i=0}^{n} a_i b^i \quad (1)$$

sau $x = (a_n a_{n-1} ... a_1 a_0)_b$ unde $a_n, a_{n-1}, ... a_1, a_0 \in B_b$.

Reprezentarea unui număr într-o bază de numerație

Conversia unui număr $x \in \mathbb{Z}_+$ într-o bază b se realizează prin împărțiri succesive de forma:

$$\begin{aligned} x &= bq_0 + \mathbf{r_0}, 0 \le r_0 < b, x > q_0 \\ q_0 &= bq_1 + \mathbf{r_1}, 0 \le r_1 < b, q_0 > q_1 \\ \ldots \\ q_{n-1} &= bq_n + \mathbf{r_n}, 0 \le r_n < b, q_{n-1} > q_n, \end{aligned}$$

Împărțirile se efectuează până când $\mathbf{q_n = 0}$.
În aceste condiții, numărul x se poate reprezenta astfel:

$$x_{10} = (r_n r_{n-1} ... r_1 r_0)_b \tag{2}$$

Exemplu

Să se reprezinte în baza 2 numărul 153.

$$153 = 2 \cdot 76 + 1$$

$$76 = 2 \cdot 38 + 0$$

$$38 = 2 \cdot 19 + 0$$

$$19 = 2 \cdot 9 + 1$$

$$9 = 2 \cdot 4 + 1$$

$$4 = 2 \cdot 2 + 0$$

$$2 = 2 \cdot 1 + 0$$

$$1 = 2 \cdot 0 + 1$$

Deci $153_{10} = 10011001_2$.

Exemplu

Să se reprezinte în baza 8 *numerele* 452 *și* 104*.*

$$452 = 8 \cdot 56 + 4$$

$$56 = 8 \cdot 7 + 0$$

$$7 = 8 \cdot 0 + 7$$

Deci $452 = 704_8$*.*

$$104 = 64 + 40 = 1 \cdot 8^2 + 5 \cdot 8^1 + 0 \cdot 8^0 = 150_8$$

Exemplu

Să se reprezinte în baza 16 numerele 30 și 684.
$30 = 16 + 14 = 1 \cdot 16^1 + 14 \cdot 16^0$, *deci* $30_{10} = 1E_{16}$

$$684 = 16 \cdot 42 + 12$$

$$42 = 16 \cdot 2 + 10$$

$$2 = 16 \cdot 0 + 2$$

Deci $684_{10} = 2AC_{16}$.

Reprezentarea unui număr într-o bază de numerație

Pentru reprezentarea unui număr dintr-o bază de numerație b în baza 10 se poate folosi formula (1).

Exemplu

Să se reprezinte în baza 10 numărul 12035_8.

$$12035_8 = 1 \cdot 8^4 + 2 \cdot 8^3 + 0 \cdot 8^2 + 3 \cdot 8^1 + 5 \cdot 8^0 = 4096 + 1024 + 24 + 5 = 5149_{10}$$

Exemplu

Să se reprezinte în baza 10 numărul BAC_{16}.

$$BAC_{16} = B \cdot 16^2 + A \cdot 16^1 + C \cdot 16^0 = 11 \cdot 256 + 10 \cdot 16 + 12 \cdot 1 = 2988_{10}$$

Reprezentarea unui număr într-o bază de numerație

- Pentru conversia unui număr dintr-o bază de numerație $b_1 \neq 10$ într-o bază $b_2 \neq 10$ se utilizează, de obicei, o transformare intermediară în baza 10.

Exemplu

Să se realizeze conversia numărul 325_7 *în baza* 3.

$$325_7 = 3 \cdot 7^2 + 2 \cdot 7^1 + 5 \cdot 7^0 = 3 \cdot 49 + 2 \cdot 7 + 5 = 166_{10}$$

$166 = 3 \cdot 55 + 1$, $55 = 3 \cdot 18 + 1$, $18 = 3 \cdot 6 + 0$, $6 = 3 \cdot 2 + 0$, $2 = 3 \cdot 0 + 2$

Deci $166_{10} = 20011_3$, *de unde* $325_7 = 20011_3$.

- O altă metodă este aceea de a utiliza tabele de transcriere.

De exemplu, pentru $b_1 = 2, b_2 = 16$ se poate utiliza următorul tabel:

0: 0000	8: 1000
1: 0001	9: 1001
2: 0010	A: 1010
3: 0011	B: 1011
4: 0100	C: 1100
5: 0101	D: 1101
6: 0110	E: 1110
7: 0111	F: 1111

Reprezentarea unui număr într-o bază de numerație

Exemplu

Să se scrie în baza 16 numerele:

$$1011\ 1001\ 1111\ 0010_2 = B9F2_{16}$$

$$101\ 1010\ 1111_2 = 0101\ 1010\ 1111_2 = 5AF_{16}$$

Exemplu

Să se scrie în baza 2 numerele:

$$1FB_{16} = 0001\ 1111\ 1011_2 = 111111011_2$$

$$DEC6_{16} = 1101\ 1110\ 1100\ 0110_2$$

Reprezentarea unui număr într-o bază de numerație

Fie $x \in \mathbb{R}$. Numărul x se reprezintă în baza b astfel:

$$\begin{aligned} x &= a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + \ldots + a_{-m} b^{-m} \\ &= \sum_{i=-m}^{n} a_i b^i \end{aligned}$$

sau $x = (a_n a_{n-1} \ldots a_1 a_0, a_{-1} \ldots a_{-m})_b$ unde $a_n, a_{n-1}, \ldots a_1, a_0, \ldots a_{-m} \in B_b$.

- Un număr real se poate scrie $x = [x] + \{x\}$, unde $[x]$ este partea întreagă a lui x, iar $\{x\}$ este partea fracționară a lui x.
- Conversia lui x dintr-o bază b_1 într-o bază b_2 se realizează separat pentru partea întreagă și partea fracționară a lui x.

Reprezentarea unui număr într-o bază de numerație

Fie $x \in \mathbb{R}$. Atunci, pentru a converti $\{x\}$ în baza b se pot realiza succesiv operațiile:

$$\begin{aligned} b \cdot \{x\} &= x_1 = & [x_1] + \{x_1\} = \mathbf{r_{-1}} + \{x_1\}, 0 \le r_{-1} < b \\ b \cdot \{x_1\} &= x_2 = & [x_2] + \{x_2\} = \mathbf{r_{-2}} + \{x_2\}, 0 \le r_{-2} < b \\ \ldots & & \\ b \cdot \{x_{n-1}\} &= x_n = & [x_n] + \{x_n\} = \mathbf{r_{-n}} + \{x_n\}, 0 \le r_{-n} < b \\ \ldots & & \end{aligned}$$

Deci, $\{x\} = b^{-1}r_{-1} + b^{-2}r_{-2} + \ldots + b^{-n}r_{-n} + \ldots$, adică $\{x\}$ se poate reprezenta astfel:

$$\{x\}_{10} = (0, r_{-1} \ldots r_{-n} \ldots)_b \tag{3}$$

Reprezentarea unui număr într-o bază de numerație

Sunt posibile două situații:

1. $\{x_n\} = 0$: procedeul se încheie și $\{x\}$ se reprezintă în baza b exact prin expresia de mai sus.
2. $\{x_n\} \neq 0$:
   - se poate observa o periodicitate între r_i, r_{i+k} și se va utiliza pentru reprezentare aceasta perioadă;
   - nu există o regulă de repetare și deci, $\{x\}$ se va reprezenta aproximativ în baza b.

Reprezentarea unui număr într-o bază de numerație

Exemplu

Să se reprezinte în baza 2 numărul $x_{10} = 14,125 = 14 + 0,125$.
$14 = 1110_2$

$$2 \cdot 0,125 = 0,25 = 0 + 0,25$$

$$2 \cdot 0,25 = 0,5 = 0 + 0,5$$

$$2 \cdot 0,5 = 1 = 1 + 0$$

Deci $0,125 = 0,001_2$ *și* $x_{10} = 1110,001_2$.

Exemplu

Să se reprezinte în baza 2 numărul $x_{10} = 0,45$.

$$2 \cdot 0,45 = 0,9 = 0 + 0,9$$

$$2 \cdot 0,9 = 1,8 = 1 + 0,8$$

$$2 \cdot 0,8 = 1,6 = 1 + 0,6$$

$$2 \cdot 0,6 = 1,2 = 1 + 0,2$$

$$2 \cdot 0,2 = 0,4 = 0 + 0,4$$

$$2 \cdot 0,4 = 0,8 = 0 + 0,8$$

$$2 \cdot 0,8 = 1,6 = 1 + 0,6$$

$$\ldots$$

Deci, $x_{10} = 0,01\,1100\,1100_2 = 0,01(1100)_2$.

Exemplu

Să se reprezinte în baza 16 numărul $x_{10} = 28,12 = 28 + 0,12$.

$$16 \cdot 0,12 = 1,92 = 1 + 0,92$$

$$16 \cdot 0,92 = 14,72 = 14 + 0,72 = E + 0,72$$

$$16 \cdot 0,72 = 11,52 = 11 + 0,52 = B + 0,52$$

$$16 \cdot 0,52 = 8,32 = 8 + 0,32$$

$$16 \cdot 0,32 = 5,12 = 5 + 0,12$$

Deci,
$x_{10} = 28_{10} + 0,12_{10} = 1C_{16} + 0,1EB85_{16} = 1C,(1EB85)_{16}$.

Adunare și scădere

Operația	Baza 10	Baza 2	Baza 5	Baza 16
Adunare	179 18 ——— 197	11011 1101 ——— 101000	134 43 ——— 232	*A2D* 1*B*5 ——— *BE*2
Scădere	37 19 ——— 18	110101 1111 ——— 100110	341 143 ——— 143	3*EC* 1*FA* ——— 1*F*2

Tabla adunării hexazecimale:

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	(10)
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	(10)	(11)
3	3	4	5	6	7	8	9	A	B	C	D	E	F	(10)	(11)	(12)
4	4	5	6	7	8	9	A	B	C	D	E	F	(10)	(11)	(12)	(13)
5	5	6	7	8	9	A	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)
6	6	7	8	9	A	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)
7	7	8	9	A	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)
8	8	9	A	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
9	9	A	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)
A	A	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
B	B	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(1A)
C	C	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(1A)	(1B)
D	D	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(1A)	(1B)	(1C)
E	E	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(1A)	(1B)	(1C)	(1D)
F	F	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(1A)	(1B)	(1C)	(1D)	(1E)

Înmulțire

Înmulțire	123	10111	213	5*FD*
	504	101	24	2*A*
	492	10111	1412	3*BE*2
	615	10111	431	*BFA*
	61992	1110011	11222	*FB*82

Tabla înmulțirii hexazecimale:

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	(10)	(12)	(14)	(16)	(18)	(1A)	(1C)	(1E)
3	0	3	6	9	C	F	(12)	(15)	(18)	(1B)	(1E)	(21)	(24)	(27)	(2A)	(2D)
4	0	4	8	C	(10)	(14)	(18)	(1C)	(20)	(24)	(28)	(2C)	(30)	(34)	(38)	(3C)
5	0	5	A	F	(14)	(19)	(1E)	(23)	(28)	(2D)	(32)	(37)	(3C)	(41)	(46)	(4B)
6	0	6	C	(12)	(18)	(1E)	(24)	(2A)	(30)	(36)	(3C)	(42)	(48)	(4E)	(54)	(5A)
7	0	7	E	(15)	(1C)	(23)	(2A)	(31)	(38)	(3F)	(46)	(4D)	(54)	(5B)	(62)	(69)
8	0	8	(10)	(18)	(20)	(28)	(30)	(38)	(40)	(48)	(50)	(58)	(60)	(68)	(70)	(78)
9	0	9	(12)	(1B)	(24)	(2D)	(36)	(3F)	(48)	(51)	(5A)	(63)	(6C)	(75)	(7E)	(87)
A	0	A	(14)	(1E)	(28)	(32)	(3C)	(46)	(50)	(5A)	(64)	(6E)	(78)	(82)	(8C)	(96)
B	0	B	(16)	(21)	(2C)	(37)	(42)	(4D)	(58)	(63)	(6E)	(79)	(84)	(8F)	(9A)	(A5)
C	0	C	(18)	(24)	(30)	(3C)	(48)	(54)	(60)	(6C)	(78)	(84)	(90)	(9C)	(A8)	(B4)
D	0	D	(1A)	(27)	(34)	(41)	(4E)	(5B)	(68)	(75)	(82)	(8F)	(9C)	(A9)	(B6)	(C3)
E	0	E	(1C)	(2A)	(38)	(46)	(54)	(62)	(70)	(7E)	(8C)	(9A)	(A8)	(B6)	(C4)	(D2)
F	0	F	(1E)	(2D)	(3C)	(4B)	(5A)	(69)	(78)	(87)	(96)	(A5)	(B4)	(C3)	(D2)	(E1)

Împărțire

Împărțire	183 : 61 = 3	1011011:1101= 111 1101 10011 1101 − 1101 1101 =	13313:212 = 34 1141 − 1403 1403 =	5E36 : 30A=1F 30A − 2D96 2D96 − =

Reprezentarea internă a numerelor

- Primele mașini de calcul foloseau pentru reprezentarea numerelor sistemul zecimal.
- Ulterior, cercetările întreprinse în domeniul teoriei informației au arătat că sistemul de numerație potrivit calculatoarelor numerice este cel binar (baza 2).
- Orice informație, indiferent de complexitatea ei, poate fi reprezentată prin informații elementare numite biți.
- Un bit este reprezentat de una din cifrele binare: 0, 1.
- Biții se pot grupa în secvențe cu denumiri speciale de 2, 4 și 8 biți, astfel: 2 biți formează un nyp, 4 biți formează un niblle, 8 biți formează un byte(octet).
- Astfel, un calculator numeric prelucrează șiruri de biți.

- Reprezentarea internă a numerelor se realizează pe o zonă de memorie cuprinsă între 2 octeți(tipul *short int*) și 12 octeți (tipul *long double*).
- Reprezentarea numerelor întregi se poate realiza în două moduri principale: cod direct sau cod complementar.
- Codul direct, se folosește, în general, pentru numere întregi fără semn (numere naturale), dar poate fi folosit și pentru numere întregi cu semn, caz în care reprezentarea include valoarea absolută și semnul.
- Primul bit din șirul de reprezentare este rezervat semnului. El are valoarea 1 dacă numărul este negativ și 0 dacă numărul este pozitiv.
- Numărul maxim care poate fi reprezentat pe n biți este $2^{n-1} - 1$.

Reprezentarea numerelor întregi în cod direct

Exemplu

$x_{10} = 13_{10} = 1101_2$

0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*0*	*1*

Exemplu

$y_{10} = -8_{10} = -1000_2$

1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*

Operații aritmetice în cod direct

Pentru a realiza operații aritmetice cu numere reprezentate în cod direct, se folosește operatorul binar $\sim$ (SAU-exclusiv), definit prin:

x	y	x $\sim$ y
0	0	0
1	0	1
0	1	1
1	1	0

Adunare și scădere

Pentru a efectua adunarea sau scăderea a două numere întregi x și y reprezentate în cod direct cu biții de semn notați x_s și y_s se utilizează următorul algoritm:

1. Se determină $r = x_s \sim y_s \sim o$, unde o este semnul operației aritmetice.
2. Dacă $r = 0$ se efectuează adunarea valorilor absolute ale numerelor în baza 2, altfel se efectuează scăderea acestora.
3. Semnul rezultatului este dat de semnul numărului mai mare în modul.

Adunare și scădere

Exemplu

$x = 14, y = -5, z = x - y$
$|x| = 1110_2, x_s = 0, |y| = 101_2, y_s = 1$

1. *Se determină $r = 0 \sim 1 \sim 1 = 0$.*
2. *Deoarece $r = 0$ se efectuează adunarea valorilor absolute ale numerelor.*
3. *Semnul rezultatului este $z_s = 0$.*

x:	**0**	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*0*
y:	**1**	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*
z:	**0**	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*1*	*1*

Deci $z = 19_{10}$.

Adunare și scădere

Exemplu

$x = 10, y = -7, z = x + y$
$|x| = 1010_2, x_s = 0, |y| = 111_2, y_s = 1$

1. *Se determină* $r = 0 \sim 1 \sim 0 = 1$.
2. *Deoarece* $r = 1$ *se efectuează scăderea valorilor absolute ale numerelor.*
3. *Semnul rezultatului este* $z_s = 0$.

x:	**0**	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
y:	**1**	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
z:	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Deci $z = 3_{10}$.

Înmulțire în cod direct

Pentru a efectua înmulțirea a două numere întregi x și y reprezentate în cod direct, cu biții de semn notați x_s și y_s, se utilizează următorul algoritm:

1. Se determină semnul rezultatului $z_s = x_s \sim y_s$.
2. Se efectuează înmulțirea valorilor absolute în baza 2.

Observații

Cifrele care depășesc numărul de biți se pierd.

Înmulțire în cod direct

Exemplu

$x = -2, y = 3, z = x \cdot y$
$|x| = 10_2, x_s = 1, |y| = 11_2, y_s = 0$

1. *Se determină* $z_s = 0 \sim 1 = 1$.
2. *Se efectuează înmulțirea valorilor absolute în baza 2.*

x:	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
y:	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
z:	**1**	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Deci, $z = -6_{10}$.

Înmulțire în cod direct

Exemplu

$x = -8, y = -3, z = x \cdot y$
$|x| = 1000_2, x_s = 1, |y| = 11_2, y_s = 1$

1. *Se determină* $z_s = 1 \sim 1 = 0$.
2. *Se efectuează înmulțirea valorilor absolute în baza 2.*

x:	**1**	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
y:	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
z:	**0**	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Deci, $z = 24_{10}$.

Reprezentarea numerelor întregi în cod complementar

- Numerele întregi (cu semn) se reprezintă, în general, folosind codul complementar.
- Pentru a reprezenta numere întregi cu semn pe n biți se consideră intervalul $[0, 2^n - 1]$ împărțit astfel: $[0, 2^n - 1] = [0, 2^{n-1} - 1] \cup [2^{n-1}, 2^n]$.
- Reprezentarea numerelor întregi pozitive se va realiza pe primul interval, $[0, 2^{n-1} - 1]$, iar numerele întregi negative se vor reprezenta prin operația de complementare pe cel de-al doilea interval.

Exemplu

Pentru reprezentare pe 32 de biți:
$[0, 2^{32} - 1] = [0, 2^{32-1} - 1] \cup [2^{32-1}, 2^{32}]$.

Cod complementar

Operația de complementare presupune transformarea unui număr x în cod complementar față de 2 și implică următorii pași:

1. Reprezentarea în baza 2, pe n biți, a numărului $|x|$.
2. Utilizarea operatorului negație pentru a transforma fiecare bit. Acest operator transformă pe 0 în 1 și invers și se notează pentru bitul curent x_i cu $\overline{x_i}$.
3. Adunarea cifrei 1 la rezultat.

Observații

Pentru orice bit x_i are loc egalitatea $\overline{x_i} = 1 - x_i$.

Cod complementar

Observații

Reprezentarea în cod complementar a numărului întreg x poate fi scrisă ca o relație matematică astfel:

$$x = \sum_{i=0}^{n-1} \overline{x_i} \cdot 2^i + 1$$

Folosind observația de mai sus și efectuând calculele obținem:

$$x = \sum_{i=0}^{n-1} (1 - x_i) \cdot 2^i + 1$$

$$x = \sum_{i=0}^{n-1} 2^i - \sum_{i=0}^{n-1} x_i \cdot 2^i + 1$$

$$x = 2^n - 1 - |x| + 1 = 2^n - |x|.$$

Reprezentarea numerelor întregi în cod complementar

Exemplu

Pentru $x = 1$, reprezentarea pe 32 de biți se realizează astfel:

Bit:	31	30	...	7	6	5	4	3	2	1	0
x:	**0**	0	...	0	0	0	0	0	0	0	1

Exemplu

Pentru $x = 14 = 8 + 4 + 2 = 1110_2$, reprezentarea pe 32 de biți se realizează astfel:

Bit:	31	30	...	7	6	5	4	3	2	1	0
x:	**0**	0	...	0	0	0	0	1	1	1	0

Reprezentarea numerelor întregi în cod complementar

Exemplu

Pentru $x = -1$, reprezentarea pe 32 de biți se realizează astfel:

1.

Bit:	31	30	...	7	6	5	4	3	2	1	0
$\lvert x\rvert$:	**0**	0	...	0	0	0	0	0	0	0	1

2.

Bit:	31	30	...	7	6	5	4	3	2	1	0
$\overline{\lvert x\rvert}$:	**1**	1	...	1	1	1	1	1	1	1	0

3.

Bit:	31	30	...	7	6	5	4	3	2	1	0
$\overline{\lvert x\rvert}+1$:	**1**	1	...	1	1	1	1	1	1	1	1

Reprezentarea numerelor întregi în cod complementar

Exemplu

Pentru $x = -14$, *reprezentarea pe 32 de biți se realizează astfel:*

1.

Bit:	*31*	*30*	...	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
$\lvert x \rvert$:	**0**	*0*	...	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*0*

2.

Bit:	*31*	*30*	...	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
$\overline{\lvert x \rvert}$:	**1**	*1*	...	*1*	*1*	*1*	*1*	*0*	*0*	*0*	*1*

3.

Bit:	*31*	*30*	...	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
$\overline{\lvert x \rvert} + 1$:	**1**	*1*	...	*1*	*1*	*1*	*1*	*0*	*0*	*1*	*0*

Reprezentarea numerelor întregi în cod complementar

Concluzii

- *Pentru o lungime pe n de biți se pot reprezenta numere x între $-2^{n-1} \leq x \leq 2^{n-1} - 1$.*
- *Prima cifră din dreapta reprezintă bitul cel mai puțin semnificativ.*
- *Prima cifră din stânga se numește bitul cel mai semnificativ sau bitul de semn.*

Exemplu

Cea mai mică valoare întreagă care poate fi reprezentată pe 32 de biți este -2^{31}. Reprezentarea acesteia este:

Bit:	*31*	*30*	...	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
x:	**1**	*0*	...	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Reprezentarea numerelor reale

- Numerele reale se reprezintă prin intermediul codificării în virgulă mobilă. Sistemele de calcul actuale utilizează reprezentarea normalizată.

Definiție

Un număr real este scris sub formă normalizată dacă este reprezentat sub forma unui produs dintre un număr subunitar cu prima cifră semnificativă diferită de zero și o putere a bazei.

Definiție

Partea subunitară din forma normalizată se numește mantisă.

Reprezentarea numerelor reale

- Mantisa se obține prin deplasarea delimitatorului zecimal (",") în fața primei cifre semnificative diferite de zero.
- Această operație implică utilizarea unui exponent egal cu numărul de deplasări ale delimitatorului care are semnul + dacă deplasarea s-a efectuat spre stânga, respectiv semnul − dacă deplasarea s-a efectuat spre dreapta.

Astfel, pentru un număr real x reprezentat în baza b sunt posibile situațiile:

- $x = (a_n ... a_1 a_0, a_{-1} ... a_{-m})_b,\ a_n \neq 0,\ \rightarrow$ $x_b = 0, a_n ... a_1 a_0 a_{-1} ... a_{-m} \times b^{n+1}$;
- $x = (0, a_{-1} ... a_{-m})_b,\ a_{-1} \neq 0,\ \rightarrow x_b = 0, a_{-1} ... a_{-m} \times b^0$;
- $x = (0, a_{-1} ... a_{-m})_b,\ a_{-1} = a_{-2} = ... = a_{-k} = 0,\ \rightarrow$ $x_b = 0, a_{-k-1} ... a_{-m} \times b^{-k}$.

Reprezentarea numerelor reale

Exemplu

$x_{10} = 23,45 = 16 + 4 + 2 + 1 + 0,45 = 10111,0111001100...._2 =$
$0,1011101110011000..._2 \times 2^5$ *sau*
$x_{10} = 23,45 = 17,733_{16} = 0,17733733... \times 16^2$

Exemplu

$x_{10} = 237,15 = 128 + 64 + 32 + 8 + 4 + 1 + 0,15 =$
$2^7 + 2^6 + 2^5 + 2^3 + 2^2 + 2^0 + 0,15 = 11101101,00100110011..._2 =$
$0,1110110100100110011..._2 \times 2^8$ *sau*
$x_{10} = 237,15 = ED,266_{16} = 0,ED266 \times 16^2$

Reprezentarea numerelor reale

Exemplu

$x_{10} = 0,45 = 0,45 = 0,0111001100....{}_2 = 0,111001100...{}_2 \times 2^{-1}$
sau $x_{10} = 0,733_{16} = 0,733733... \times 16^0$

Exemplu

$x_{10} = 0,000742 = 0,0000\ 0000\ 0011\ 0000\ 1010\ 0000\ 1011...{}_2 =$
$0,11000010100001011... \times 2^{-10}$ *sau*
$x_{10} = 0,000742 = 0,0030A0B...{}_{16} = 0,30A0B... \times 16^{-2}$

Reprezentarea numerelor reale

- În reprezentarea binară, pentru că prima cifră semnificativă este întotdeauna 1, aceasta nu se mai reprezintă, făcându-se economie de 1 bit.
- În consecință, orice număr real x se poate scrie sub forma: $x = (-1)^S \times 1, mantisa \times 2^{exponent}$ unde S este bitul de semn.
- Reprezentarea numerelor reale în simplă precizie ($n = 4$ octeți) utilizează biții disponibili astfel:

31	30 ... 23	22... 0
bit semn	caracteristica	mantisa

unde caracteristica = exponent+127.

Reprezentarea numerelor reale

Exemplu

$x_{10} = 237,15 = 128 + 64 + 32 + 8 + 4 + 1 + 0,15 =$
$11101101,00100110011001100\ldots_2 =$
$0,111011010010011001\ldots_2 \times 2^8$
$x = (-1)^0 \times 1,11011010010011001100110 \times 2^7 \; S = 0$
caracteristica$= 7 + 127 = 134 = 10000110_2$

31	30	29	28	27	26	25	24	23	22	21	...	1	0
0	1	0	0	0	0	1	1	0	1	1	...	1	0

Reprezentarea numerelor reale

Exemplu

$x_{10} = -29,14 = 11101,00100011110101110000101000111101\ldots_2$

$x = (-1)^1 \times 1,1101001000111101011100001010001111101 \times 2^4$

$S = 1$

caracteristica$= 4 + 127 = 131 = 10000011_2$

31	30	29	28	27	26	25	24	23	22	21	...	1	0
1	1	0	0	0	0	0	1	1	1	1	...	0	1

Reprezentarea numerelor reale

Observații

Există cazuri în care pot apărea ambiguități din cauza primei cifre semnificative 1 care nu se reprezintă în mantisă.
De exemplu, numărul zero s-ar reprezenta la fel ca $0,1_2$.
Din acest motiv, s-a introdus convenția ca numărul zero să fie reprezentat pe un șir de biți în care toate valorile sunt zero.

Reprezentarea numerelor reale

Observații

În operațiile cu numere reprezentate astfel pot apărea erori de reprezentare din cauza depășirii numărului de biți utilizați:

- *depășire flotantă superioară (floating overflow), care apare din cauza depășirii de către caracteristică a valorii 255 (numărul maxim care poate fi reprezentat pe un octet).*
- *depășire flotantă inferioară (floating underflow), care apare din cauza depășirii inferioare de către caracteristică a valorii 0 (exponentul este mai mic decât -127).*
- *O soluție posibilă este mărirea preciziei, adică a numărului de biți alocați pentru reprezentare.*