1. REPREZENTAREA INFORMAȚIILOR ÎN CALCULATOR

1.1. CONCEPTUL DE DATĂ ȘI INFORMAȚIE

- Datele desemnează elementele primare, provenind din diverse surse, fără o formă organizată care să permită luarea unor decizii;
- Informaţiile sunt date ce au un caracter de noutate, care îmbogăţesc nivelul de cunoştinţe ale celui care primeşte aceste informaţii.

O dată care nu aduce nimic nou nu se poate considera informaţie.

Definiţie. Prelucrarea datelor (numită şi procesare) presupune totalitatea transformărilor, ca formă şi conţinut a datelor.

Dintre transformările cele mai importante:

1. GENERAREA – operaţia prin care se obţin date primare.
2. CULEGEREA ŞI PREGĂTIREA. Forma iniţială a datelor diferă de forma sub care acestea intră în prelucrare. Pregătirea datelor în vederea prelucrării lor presupune etapele:
   - Selectarea – alegerea dintre datele primare a celor necesare pentru obţinerea numitor informaţii;
   - Codificarea – trecerea datelor într-o formă adecvată prelucrării;
   - Conversia – transferarea datelor de pe un suport pe altul;
   - Copierea/multiplicarea – reproducerea pe un mediu identic a datelor, păstrându-se forma, conţinutul şi semnificaţia acestora;
   - Verificarea – analizarea corectitudinii datelor, ca formă şi conţinut, urmărind eliminarea posibilelor alterări datorate culegerii, prezentării sau transmiterii.
3. PRELUCRAREA datelor în scopul obţinerii diferitelor rezultate se face cu ajutorul funcţiilor:
   - Sortarea – aranjarea datelor într-o anumită ordine, conform unor criterii bine precizate;
   - Clasificarea – gruparea datelor după caracteristici comune, fără stabilirea unor anumite priorităţi în cadrul grupelor formate;
   - Compararea – stabilirea asemănărilor şi/sau deosebirilor dintre două sau mai multe date;
   - Analiza – examinarea unui grup de date, urmărindu-se condiţiile îndeplinirii uneia/mai multor cerinţe de către fiecare dată a grupului în parte;
   - Sintetizarea – stabilirea unor caracteristici esenţiale, generale, pe care

le au datele dintr-un grup, formându-se o nouă structură ce surprinde acele caracteristici comune;

- ➢ Calcularea – operaţii aritmetice sau logice efectuate asupra uneia sau mai multor date, simultan.

4. FURNIZAREA datelor este impusă de faptul că forma în care datele sunt disponibile în urma prelucrării nu corespunde, de obicei, cerinţelor beneficiarului. Ca urmare, se impune furnizarea datelor într-o formă clară, astfel încât să nu existe probleme de înţelegere sau interpretare eronată.
5. PĂSTRAREA datelor se face în colecţii de date, alcătuite după reguli bine definite, în vederea unor prelucrări ulterioare. Asupra unei astfel de colecţii se pot face următoarele operaţii:
    - ➢ Validare – precizarea modului în care o dată poate fi introdusă în colecţie;
    - ➢ Regăsire – căutarea şi localizarea unei date;
    - ➢ Modificare – transformarea unei date din colecţie prin schimbarea unor atribute;
    - ➢ Distrugere – eliminarea din colecţie a unor date, cu precizarea condiţiilor în care se poate face acest lucru.
6. TRANSMITEREA (COMUNICAREA) datelor se referă la modul în care datele trec de la o etapă la alta, pe parcursul prelucrării.

1.2. SISTEME DE NUMERAȚIE

Un **sistem de numerație** este alcatuit dintr-o mulțime finită de simboluri și un set de reguli de reprezentare a numerelor cu ajutorul simbolurilor respective.

De exemplu, în:

- **Sistemul zecimal** – sistem de numerație în baza 10; se utilizează 10 simboluri: 0, 1, 2, 3, 4, 5, 6, 7, 8 ,9; b=10.
- **Sistemul binar** – sistem de numerație în baza 2; se utilizează 2 simboluri: 0, 1; b=2.
- **Sistemul octal** – sistem de numerație în baza 8; numărul de cifre utilizat este 8, având cifrele 0,1,2,3,4,5,6,7; b=8.
- **Sistemul hexazecimal** – sistem de numerație în baza 16, se utilizează 16 simboluri: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Semnificația zecimală a simbolurilor A, B, C, D, E, F este, în ordine 10, 11, 12 ,13, 14, 1; b=16.

Pentru a face distincție între numerele din diferite baze de numerație există mai multe metode de notare:

a) La sfârșitul numărului se adaugă o literă corespunzătoare bazei de numerație:
   - B - binar (ex. 10011101B)
   - Q - octal (ex. 23701Q)
   - D - zecimal (ex. 5429D)
   - H - hexazecimal (ex. FD37BH)

b) La sfârșitul numărului se adaugă, în paranteze, baza căreia îi aparține numărul:
   - (2) - binar (ex. 10011011(2))
   - (8) - octal (ex. 24673(8))
   - (10) - zecimal (ex. 9546(10)
   - (16) - hexazecimal (ex. 34A4D(16))

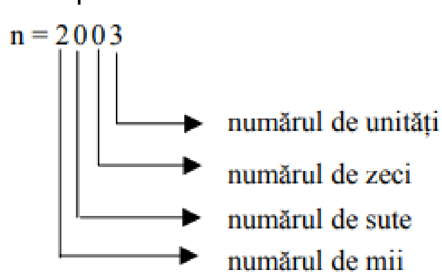
c) La sfârșitul numărului se adaugă ca și indice, în paranteze, baza căreia îi aparține numărul:
   - Număr$_{(2)}$ - binar (ex. 101101(2))
   - Număr$_{(8)}$ - octal (ex. 5572(8))
   - Număr$_{(10)}$ - zecimal (ex. 9334(10))
   - Număr$_{(16)}$ - hexazecimal (ex. 53FD1(16))

Observație: Deoarece baza 10 este considerată o bază implicită, numerele din această bază pot să nu fie urmate de simbolul corespunzător bazei.

După valoarea echivalentă asociată unei cifre, sistemele de numeraţie pot fi:

- poziţionale;
- nepoziţionale.

Un sistem de numeraţie se numeşte poziţional, dacă valoarea unei cifre este dată de poziţia pe care aceasta o ocupă în cadrul numărului.
Exemplu: considerăm numărul 2003 scris în baza 10.



Se observă că, în funcţie de poziţia pe care o ocupă, cifra 0 are valori diferite.

Un sistem de numeraţie se numeşte nepoziţional, dacă valoarea unei cifre este aceeași indiferent de poziţia pe care aceasta o ocupă în cadrul numărului. Reprezentarea prin sistemele de numeraţie nepoziţionale, necesită utilizarea unor reguli dificile, greu de implementat şi din această cauză nu este folosită.

Un sistem de numeraţie poziţional folosește un număr numit **baza sistemului** respectiv. Numărul de simboluri ale alfabetului unui sistem de numerație trebuie să fie egal cu baza sistemului de numerație respectiv.

În cazul sistemului **zecimal**, b=10.

În cazul sistemului **binar**, b=2.

În cazul sistemului **octal**, b=8.

În cazul sistemului **hexazecimal**, b=16.

- ✓ Baza sistemului de numeraţie se notează cu *b* şi satisface condiţia *b>1*.
- ✓ Numerele pot fi reprezentate în baza *b* folosindu-se cifrele cuprinse în intervalul [0, b-1].
- ✓ In conformitate cu locul lor în reprezentarea unui număr în baza b, fiecare cifră se înmulțește cu puteri ale bazei b începând cu puterea 0 pentru cifra cea mai puțin semnificativă și continuând crescător din unu în unu până la puterea maximă egală cu numărul de cifre ale numărului minus unu.

$$N_{(b)} = a_{n-1}*b^{n-1}+ a_{n-2}*b^{n-2}+........+a_1*b_1*a_0*b_0 \qquad (1)$$

Ex: $43793 = 4x10^4+3x10^3+7x10^2+9x10^1+3x10^0$

Un calculator poate fi prevăzut să funcţioneze în orice sistem de numeraţie. Cel mai avantajos este sistemul binar.

Sistemul binar a fost preferat din următoarele motive:

- ✓ simplitatea regulilor pentru operaţiile aritmetice şi logice;
- ✓ materializarea fizică relativ simplă a cifrelor.

Sistemele de calcul nu operează de fapt cu numerele 0 şi 1, ci cu două stări asociate lor, respectiv semnalele Low şi High, contact închis/contact deschis, prezenţa sau absenţa de curent etc.

Circuitele care trebuie să diferenţieze numai două stări sunt mai sigure în funcţionare decât cele care trebuie să diferenţieze 10 stări.

Fizic sunt folosite elemente bistabile cu două stări asociate cu valorile 0 şi respectiv 1:

- ✓ conducţie (+5V)
- ✓ blocat (0V)

Un astfel de element reprezintă unitatea de memorie denumită bit (binary digit - “cifră binară”).

Sistemele octal şi hexazecimal sunt notaţii folosite de către programatori pentru scurtarea notaţiilor prea lungi care ar rezulta în cazul reprezentării în binar a numerelor mari.

1.3. CONCEPTELE: BIT, OCTET, CUVÂNT

Deoarece calculatorul recunoaşte numai două stări, datele supuse prelucrării sunt transformate în şiruri de cifre binare 0 şi 1.

O cifră binară se mai numeşte **bit**[1] (prescurtat, **b,** Binary Digit) şi reprezintă cea mai mică unitate de măsură a cantităţii de informaţie.
Concret, bitul nu este decât starea de „închis”- „deschis” sau „sus”-„jos” dintr-un circuit.
Noţiunea de bit a fost utilizată pentru prima dată în teza de doctorat a matematicianului Claude Shannon, care a „inventat”, prin teza sa, un nou domeniu, numit teoria informaţiei.

În 1964, proiectanţii calculatorului mainframe IBM System/360 au stabilit, ca şi convenţie, folosirea grupurilor de 8 biţi ca unitate de bază a memoriei calculatorului. Astfel a apărut octetul (o) sau byte-ul (B).

O succesiune de 8 biţi se numeşte **byte** (prescurtat, **B**) sau **octet**, („o”), fiind cea mai mică unitate de date ce poate fi reprezentată şi adresată de către memoria unui sistem de calcul.
Deoarece datele reprezentate în memorie ocupă o succesiune de Bytes, acestea sunt exprimate în multipli ai acestuia:

- 1 kiloByte 1 kB = 1024 B = 2^{10} B
- 1 MegaByte 1 MB = 1024 kB = 2^{10} kB
- 1 GigaByte 1GB = 1024 MB = 2^{10} MB
- 1 TerraByte 1 TB= 1024 GB = 2^{10} GB
- 1 PetaByte 1 PB= 1024 TB = 2^{10} TB
- 1 ExaByte 1 EB= 1024 PB = 2^{10} PB

Numărul maxim reprezentabil pe un byte este $\mathbf{11111111_2}$, adică $\mathbf{255_{10}}$, deci o valoare total insuficientă.
Din acest motiv, s-a trecut la reprezentarea datelor/informaţiilor pe:

- cuvinte de memorie: un cuvânt = 2 bytes (16 biţi);
- cuvinte duble: un cuvânt dublu = 4 bytes (32 biţi);
- cuvinte cvadruple: un cuvânt cvadruplu = 8 bytes (64 biţi).

[1] binary digit – cifră binară.

Numerele întregi se reprezintă în calculator pe 1,2,4 sau 8 octeți (8,16, 32 sau 64 biți). Reprezentarea în baza 16 este importantă pentru că un octet poate fi reprezentat prin două cifre hexazecimale.

2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024

Tabel cu puterile lui 2

2^{10} = 1 024	1 Kilo
2^{20} = 1 048 576	1 Mega
2^{30} = 1 073 741 824	1 Giga

Puteri ale lui 2 cu denumiri speciale

1.4. CONVERSIA NUMERELOR ÎNTREGI ÎN DIFERITE BAZE DE NUMERAȚIE

1.4.1. Conversia unui număr din baza *b* în baza 10

Se realizează astfel:

- Se scrie numărul înmulțind fiecare cifră cu baza la puterea corespunzătoare, ca în formula (1);
- Se înlocuiesc cifrele numărului cu valoarea lor în baza 10;
- Se realizează calculele (înmulțiri și adunări) în baza 10.

Ex: Conversia din baza 2 în baza 10

Transformarea numărului $01011100_{(2)}$ în baza 10 se face astfel:

$001011100_{(2)} = 0*2^7+1*2^6+0*2^5+1*2^4+1*2^3+1*2^2+0*2^1+0*2^0=0+64+0+16+8+4+0+0$
$=92$

Ex: Conversia din baza 8 în baza 10

Transformarea unui număr din baza 8 în baza 10 se face astfel

$157_{(8)} = 1*8^2 + 5*8^1 +7*8^0 = 111_{(10)}$

Ex: Conversia din baza 16 în baza 10

Transformarea unui număr din baza 16 în baza 10 se face astfel

$A9B_{(16)} = A*16^2 + 9*16^1 +B*16^0 = 10*16^2 + 9*16^1 +11*16^0 = 2715_{(10)}$

Binar	Zecimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9

Numărarea în sistemul binar este în bună măsură asemănătoare cu cea din sistemul zecimal obișnuit. Diferența constă în faptul că în binar stau la dispoziție doar două cifre anume 0 și 1, în timp ce în sistemul zecimal există zece cifre, cele de la 0 la 9. Regulile pentru toate sistemele, deci și pentru cel binar, sunt următoarele două:

- Numărarea începe cu o singură poziție, care pornește ca valoare (sau conținut) de la cifra 0 și continuă crescător până la cea mai mare cifră din sistem. Această poziție, cea mai din dreapta a numărului, poartă numele de "poziția (cifra) cea mai puțin semnificativă".
- După ce o poziție curentă ajunge la cifra maximă, poziția curentă "sare" înapoi la 0, iar poziția din stânga ei trebuie incrementată cu o unitate. Această situație se numește "depășire". (Dacă cumva poziția din stânga încă nu există, ea se creează și i se dă mai întâi valoarea 1.) Prin acest procedeu este posibil ca și poziția din stânga să prezinte o depășire. În acest caz se aplică chiar această regulă din nou, altfel spus, în mod recursiv, din ce în ce mai spre stânga, până se întâlnește un 0, care, fără depășire, devine un 1. Pentru numărarea în binar aceasta înseamnă că, după ce o poziție a devenit 1, ea se repune la 0, iar la poziția din stânga ei trebuie adăugat un 1. Rezultatul arată astfel: 0, 1, 10, 11, 100, 101, 110, 111, 1000 ș.a.m.d.

1.4.2. Conversia unui număr din baza *b* în baza 10

Transformarea din baza 10 în orice bază se face prin împărţiri succesive ale numărului la baza către care se face transformarea, resturile obţinute reprezentând cifrele ce alcătuiesc numărul în noua reprezentare.

Algoritmul este următorul:

1. Se împarte numărul reprezentat în baza 10 la b. Se obține un cât și un rest care se reține
2. Se împarte câtul obținut la b și se obține un alt cât și un rest care se reține
3. Procesul se repetă cu câtul și restul obținut. Algoritmul se oprește atunci când câtul obținut este zero.
4. Resturile se convertesc în baza b și numărul căutat este reprezentat de concaternarea resturilor începând de la cel de pe urmă până la cel dintâi obținut (primul rest obținut este cifra cea mai putin semnificativă).

Exemple:

1. Să se convertească numărul $13_{(10)}$ în baza 2

	câtul	restul	Observații
13 : 2	6	1	
6 : 2	3	0	
3 : 2	1	1	
1 : 2	0 (algoritmul se oprește)	1	bitul cel mai semnificativ

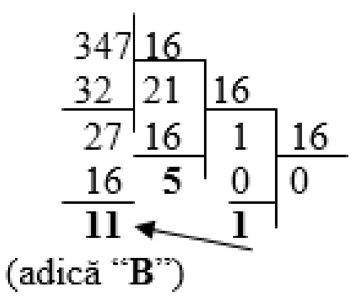
Deci, $13_{(10)} = 1101_{(2)}$

Pentru a completa până la 8 biți se pun în față zerouri:

$13_{(10)} = 00001101_{(2)}$

2. Să se convertească numărul întreg 347 din baza 10 în baza 16, 2 şi 8.

Mai întâi se converteşte în baza 16 pentru că aceasta se realizează prin mai puţine împărţiri decât conversia în baza 2 sau 8.



Astfel, luând resturile în ordine inversă obţinem 15B(H).

347(D) = 15B(H) = 1 0101 1011(B) = 533(O)

1.4.3. Conversia între bazele 2, 8 și 16

Având un număr în baza 2, pentru reprezentarea sa în baza 16 se grupează câte 4 cifre binare care se convertesc într-o cifră hexazecimală.

Exemple:

$(1100)_2 = (C)_{16}$

$(10011110)_2 = (1001\ 1110)_2 = (9E)_{16}$

Având un număr în baza 2, pentru reprezentarea în baza 8 se grupează câte 3 cifre binare care se convertesc într-o cifră octală.

Exemplu

$(11100)_2 = (11\ 100)_2 = (011\ 100)_2 = (34)_8$

Conversia unui număr din baza 16 în baza 2 se face reprezentând fiecare cifră hexazecimală prin 4 cifre binare.

Exemplu

$(A4)_{16}=(1010\ 0100)_2$

Conversia unui număr din baza 8 în baza 2 se face convertind fiecare cifră octală în 3 cifre binare.

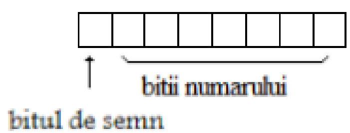
Exemplu

$(63)8 = (110\ 011)_2$

Pentru conversii de numere între bazele 2, 8, 10 şi 16 şi operații cu numere în aceste baze se poate folosi aplicația Calculator a sistemului de operare Windows.

1.4.4. Reprezentarea numerelor binare cu semn

În cazul numerelor binare cu semn, bitul cel mai semnificativ este bitul de semn. El este 0 pentru numere pozitive şi 1 pentru numere negative.



Există trei reprezentări ale numerelor binare cu semn.

1.4.4.1. Reprezentarea în mărime și semn

Un număr pozitiv X se reprezintă în mărime şi semn ca:

$$X = 0 * 2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

Un număr negativ X se reprezintă în mărime şi semn ca:

$$-X = 1 * 2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

Se observă că diferă doar bitul de semn. Biții numărului sunt aceiaşi, pentru număr pozitiv sau negativ, ei reprezentând valoarea absolută a numărului |X|.

Exemple. Vom considera numere întregi reprezentate pe 8 biți, un bit de semn şi 7 biți ai numărului:

Baza 10	Baza 2	Baza 16
13	0000 1101	0D
-13	1000 1101	8D
25	0001 1001	19
-7	1000 0111	87
127	0111 1111	7F
-127	1111 1111	FF

Gama numerelor întregi reprezentabile pe un octet în mărime şi semn este [-127, 127]. Reprezentarea în mărime şi semn este rar folosită în calculatoare.

1.4.4.2. Reprezentarea în complement față de 1

Fie N un număr natural reprezentat în baza b şi fie a_i cifrele numărului. Complementul unei cifre a numărului în baza b – 1 este:

$\overline{a_i}$=b-1-a_i

Reamintim că cifrele bazei r sunt {0, 1, ..., b – 1}.

Numărul N este reprezentat în complement față de b – 1 complementând fiecare cifră a numărului față de b – 1.

De exemplu, un număr binar se complementează față de 1 complementând fiecare cifră binară față de 1, un număr hexazecimal se complementează față de 15 complementând fiecare cifră hexazecimală fața de 15, etc.

Un număr pozitiv X se reprezintă în complement față de 1 ca:

$$X = 0 * 2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

Un număr negativ X se reprezintă în complement față de 1 ca:

$$X = 0 * 2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

Unde $\overline{a_i}$=1-a_i este complementul față de 1 al cifrei a_i .

Reprezentarea unui număr negativ în complement față de 1 se face modificând toate cifrele numărului pozitiv din 1 în 0 şi din 0 în 1, inclusiv bitul de semn. Exemple de numere reprezentate în complement față de 1 pe un octet.

Baza 10	Baza 2	Baza 16
15	0000 1111	0F
-15	1111 0000	F0
-19	1000 1100	8C
19	0001 0011	13
127	0111 1111	7F
-127	1000 0000	80

Reprezentarea în complement față de 1 nu se foloseşte în calculatoare.

1.4.4.3. Reprezentarea în complement față de 2

Pentru reprezentarea internă a numerelor întregi cu semn se folosește complementul față de 2.

Pentru a calcula complementul față de 2 al unui număr negativ pe *n* poziții binare se scrie reprezentarea binară a numărului fără semn (modulul numărului) pe *n* poziții binare, apoi se complementează fiecare bit (transformând toți biții 1 în 0 și invers), pentru ca în final să adunăm 1 la ceea ce se obține.

Exemplu:

Să se calculeze complementul față de 2 al numărului -105 pe n=8 biți.

$105=01101001_{(2)}$

Complementăm toți biții:10010110

Se adună 1 și se obține rezultatul final: 10010111.

1.5. CONVERSIA NUMERELOR REALE

Un număr real are o parte zecimală și o parte fracționară. Pentru a converti numărul care conţine atât parte întreagă cât şi parte zecimală, între diverse baze de numerație, trebuie convertite separat partea întreagă şi cea zecimală și apoi se reunesc.

1.5.1. Conversia părţii zecimale a unui număr din baza 10 în altă bază de numerație

Conversia părții zecimale a unui număr din baza 10 în altă baza de numerație se realizează înmulțind partea zecimală (fracționară) cu baza în care dorim să facem conversia. In continuare se inmulțește succesiv partea fracționară a rezultatului înmulțirii precedente cu baza. Rezultatul în noua bază este reprezentat de partea întreagă a fiecărei înmulțiri.

In cazul ideal, rezultatul final se obține în momentul în care partea fracționară a rezultatului înmulțirii cu baza este zero. De cele mai multe ori însă, partea fracționară nu devine zero niciodată. De aceea este necesară oprirea calculelor prin stabilirea preciziei de reprezentare a părții fracționare rezultate (adică numărul de cifre al părții fracționare rezultate).

Exemplu:

Să se convertească numărul 0,136 din baza 10 în bazele 2, 8, 16. Precizia de reprezentare: 8.

Număr x bază = parte fracționară + parte întreagă

Partea întreagă se reține, partea fracționară continuă să fie inmulțită cu baza.

Baza 2	Baza 8	Baza 16
$0.136 \times 2 = 0.272+0$	$0.136 \times 8 = 0.088+1$	$0.136 \times 16 = 0.176+2$
$0.272 \times 2 = 0.544+0$	$0.088 \times 8 = 0.704+0$	$0.176 \times 16 = 0.816+2$
$0.544 \times 2 = 0.088+1$	$0.704 \times 8 = 0.632+5$	$0.816 \times 16 = 0.056+13$ (D)
$0.088 \times 2 = 0.176+0$	$0.632 \times 8 = 0.056+5$	$0.056 \times 16 = 0.896+0$
$0.176 \times 2 = 0.352+0$	$0.056 \times 8 = 0.448+0$	$0.896 \times 16 = 0.336+14$ (E)
$0.352 \times 2 = 0.704+0$	$0.448 \times 8 = 0.584+3$	$0.336 \times 16 = 0.376+5$
$0.704 \times 2 = 0.408+1$	$0.584 \times 8 = 0.672+4$	$0.376 \times 16 = 0.016+6$
$0.408 \times 2 = 0.816+0$	$0.672 \times 8 = 0.376+5$	$0.016 \times 16 = 0.256+0$
0.00100010(2)	**0.10550345(8)**	**0.22D0E560(16)**

Este uşor să se folosească și schema următoare, care prin cele două linii separă mai clar cifrele reprezentării precum şi indică mai bine poziţia virgulei (cifrele de la prima înmulţire în jos adică de sub linie sunt după virgulă). Trebuie remarcat că se înmulţeşte doar ceea ce este în dreapta virgulei.

Exemplu:

- Să se convertească numărul 0,47(D) în binar, octal şi hexazecimal:

0,	47*2
0	94
1	88
1	76
1	52
1	04
0	08
0	16
0	32
0	64
1	28
0	56
1	12
0	24
0	48
0	96
1	...

Deci 0,47(D) = 0,0111 1000 0101 0001 (B) = 0,7851(H) = 0,3605(O)

1.5.2. Conversia părţii zecimale a unui număr dintr-o bază oarecare în baza 10

Pentru a converti un număr dintr-o bază oarecare în baza 10 se poate folosi formula definită în prima parte a lucrării şi anume dacă se dă un număr scris într-o bază oarecare "b" sub forma parte întreagă şi parte zecimală:

Nr(b) =Cn-1Cn-2 … C2C1C0 , D1D2D2D3…

atunci valoarea sa în baza 10 va fi:

Nr(10) = $C_{n-1}*b^{n-1}+ C_{n-2}*b^{n-2}+\ldots\ldots+C_1*b_1*C_0*b_0, D_1*b^{-1}+ D_2*b^{-2}+ D_3*b^{-3}+\ldots$

Exemple:

- Se dă numărul întreg în hexazecimal 3A8(H) şi se cere valoarea sa în zecimal:
  N = 3*162 + 10*16 1 + 8 = 3*256 + 160 + 8 = 936(D)

- Se dă numărul fracţionar 0,341(Q) scris în baza 8 şi se cere valoarea sa în zecimal
  N = $3*8^{-1} + 4*8^{-2} +1*8^{-3}$ = 3/8 +4/64 + 1/512 = 0.4394(D)

- Se dă numărul în binar 110,11(B) şi se cere valoarea sa în hexazecimal şi în zecimal:
  N = 110,11(B) = 6,C(H) = 6,75(D)