Dan NICULA

ELECTRONICĂ DIGITALĂ
Carte de învăţătură 2.0

Editura Universităţii *TRANSILVANIA* din Braşov
ISBN 978-606-19-0563-8

2015

Lecţia 1

Reprezentarea datelor în sistemele digitale

1.1 Noţiuni teoretice

În sistemele de calcul, numerele sunt reprezentate în baza de numeraţie 2. În această bază de numeraţie, numerele se codifică cu două simboluri: 0 şi 1.

Reprezentarea numerelor în baza 16 necesită 16 simboluri: 10 cifre ($0-9$) şi 6 litere (A, B, C, D, E, F).

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

Numerele reprezentate în diferite baze de numeraţie şi diferite tipuri de codări pot fi convertite dintr-o formă în alta.

- **Conversie din baza 2 în baza 16.** Se *grupează câte 4 cifre binare de la dreapta spre stânga*. Fiecare grup de 4 cifre binare generează o cifră în baza 16.

- **Conversie din baza 2 în baza 10.** Numărul în baza 10 se determină prin *adunarea ponderată a puterilor lui 2*. Puterea 2^i se adună în suma finală ponderată cu bitul corespunzător b_i. Altfel zis, dacă $b_i = 1$, la sumă se adaugă 2^i, altfel nu.
  Matematic, formula este:

$$b_{N-1}b_{N-2}...b_1b_0 = \sum_{i=0}^{N-1}(b_i \cdot 2^i)$$

Exemplu: $1011001|_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 0 + 16 + 8 + 0 + 0 + 1 = 89|_{10}$

Metoda *adunării ponderilor* presupune însumarea puterilor lui 2 ale căror indecşi apar în reprezentarea binară. Exemplu: $1011001|_2 = 2^6 + 2^4 + 2^3 + 2^0 = 64 + 16 + 8 + 1 = 89|_{10}$

O alternativă mai rapidă este de a se converti numărul din baza 2 în baza 16 şi apoi din baza 16 în baza 10.

- **Conversie din baza 10 în baza 2.** Metoda *scăderii ponderate* este o metodă iterativă care presupune listarea puterilor lui 2 mai mici decât numărul considerat. Iteraţiile încep cu considerarea puterii lui 2 celei mai mari (mai mică decât numărul). Se scade numărul putere a lui 2 din numărul considerat. Cu restul obţinut se repetă scăderea următoarei puteri ale lui 2 (mai mică decât cea anterioară). Dacă restul este mai mic decât puterea lui 2, se consideră în forma binară un 0 (zero). Altfel, se consideră un 1 şi se obţine un nou rest prin scăderea puterii lui 2 din valoarea restului. Algoritmul se termină la cea mai mică putere a lui 2 ($2^0 = 1$).

  O alternativă constă în *împărţirea succesivă a numărului zecimal la 2 şi considerarea resturilor* (primul rest obţinut reprezintă bitul cel mai puţin semnificativ, din dreapta).

  Se poate face o conversie indirectă, din baza 10 în baza 16 şi apoi din baza 16 în baza 2.

- **Conversie din baza 10 în baza 16.** O metodă constă în *împărţirea succesivă a numărului zecimal la 16 şi considerarea resturilor* (primul rest obţinut reprezintă cifra cea mai puţin semnificativă a numărului în baza 16, cifra din dreapta).

- **Conversie din baza 16 în baza 2.** *Fiecare cifră hexazecimală generează 4 cifre binare.*

- **Conversie din baza 16 în baza 10.** Numărul în baza 10 se determină prin *suma ponderată a puterilor lui 16.* Ponderea termenului 16^i este cifra corespunzătoare $h_i = 1$. Matematic, formula este:

$$h_{N-1}h_{N-2}...h_1h_0 = \sum_{i=0}^{N-1}(h_i \cdot 16^i)$$

Stocarea informaţiilor în sisteme de calcul necesită codificarea acestora cu coduri binare, pe baza simbolurilor 0 şi 1.

Numerele pozitive sunt codificate prin reprezentarea acestora în baza 2.

Numerele întregi pot fi codificate în mai multe feluri:
- *mărime şi semn*,
- *complement faţă de 1*,
- *complement faţă de 2.*

Cea mai utilizată codificare este codificarea numerelor întregi în *complement faţă de 2.* Această codificare s-a impus datorită avantajelor acesteia în implementarea sistemelor de calcul digitale. În acest capitol, în lipsa altor menţiuni, numerele întregi sunt considerate reprezentate în *complement faţă de 2.*

În sistemele de calcul, numerele sunt reprezentate sub forma unor codificări binare. Operaţiile aritmetice în baza 2 sunt similare operaţiilor în baza 10, par mai grele pentru operatorul uman, însă sunt mai uşor de implementat în sisteme de calcul hardware digitale.

Semnul numerelor întregi (pozitive sau negative) este şi el codat binar (0 = număr pozitiv, 1 = număr negativ). Operaţiile aritmetice binare implementate hardware trebuie să ţină cont de semnul operanzilor şi să îl proceseze corect.

Reprezentarea în complement faţă de 2 a numerelor negative se realizează astfel:

- Se determină reprezentarea binară a modulului numărului considerat (pozitiv).
- Se neagă bit cu bit reprezentarea binară obţinută.
- Se adaugă 1, cu considerarea transportului. Bitul de semn se va obţine 1.

  O alternativă constă în aplicarea următoarei transformări: se consideră biţii numărului pozitiv de la dreapta la stânga, se copiază biţii până la primul 1 inclusiv, ulterior fiecare bit se neagă.

Algoritmul expus poate fi folosit şi pentru determinarea numărului negativ $(-N)$ pornind de la reprezentara binară a numărului pozitiv asociat (N).

Determinarea numărului pozitiv (N) pornind de la reprezentara binară a numărului negativ asociat $(-N)$ se face cu algoritmul:

- Se neagă bit cu bit reprezentarea binară a numărului negativ $(-N)$.
- Se adaugă 1, cu considerarea transportului. Bitul de semn se va obţine 0.

Pe un număr fix de biţi se pot reprezenta un număr finit de numere. În cazul unor operaţii aritmetice, este posibil ca rezultatul să nu mai poate fi reprezentat pe numărul de biţi ai operanzilor. În acest caz, este necesară **extinderea de semn** a operanzilor. Creşterea numărului de biţi pe care este reprezentat un număr codat în complement faţă de 2 se face prin extinderea bitului de semn, aflat pe poziţia cea mai semnificativă (din stânga) pe toate poziţiile superioare ale formatului cu mai mulţi biţi de cod.

Exemple de codare în complement faţă de 2, cu număr minim de biţi, cu 8 biţi şi cu 16 biţi:

$10|_{10} = 1010|_2 = 01010|_{C2} = 0000\_1010|_{C2} = 0000\_0000\_0000\_1010|_{C2}$
$-10|_{C2} = \overline{0\_1010} + 1 = 1\_0101 + 1 = 1111\_0110|_{C2} = 1111\_1111\_1111\_0110|_{C2}$

Adunarea numerelor reprezentate în complement faţă de 2 poate fi uşor implementată în hardware deoarece semnul numerelor nu necesită o procesare diferită faţă de celelalte cifre binare. Adunarea numerelor reprezentate în complement faţă de 2 (fie pozitive, fie negative) se face similar cu adunarea numerelor în baza 10: de la cea mai puţin semnificativă cifră (din dreapta) spre cea mai semnificativă (din stânga), cu considerarea transportului spre bitul superior în caz că suma a 2 biţi (şi transportul de la bitul inferior) este mai mare sau egală cu 2 (baza de numeraţie).

Exemple, operaţii aritmetice cu numere codificate în complement faţă de 2 pe 8 biţi:

- $10|_{10} = 1010|_2 = 0000\_1010|_{C2}$
- $20|_{10} = 10100|_2 = 0001\_0100|_{C2}$
- $30|_{10} = 11110|_2 = 0001\_1110|_{C2}$
- $-10|_{C2} = \overline{0000\_1010} + 1 = 1111\_0101 + 1 = 1111\_0110|_{C2}$
- $-20|_{C2} = \overline{0001\_0100} + 1 = 1110\_1011 + 1 = 1110\_1100|_{C2}$
- $-30|_{10} = \overline{0001\_1110} + 1 = 1110\_0001 + 1 = 1110\_0010|_{C2}$

- $10 + 20 = 30$
```
 10+       0000_1010+
 20        0001_0100
 --        ---------
 30        0001_1110
```
- $10 + (-20) = (-10)$
```
 10+       0000_1010+
-20        1110_1100
---        ---------
-10        1111_0110
```
- $(-10) + 20 = 10$
```
-10+       1111_0110+
 20        0001_0100
---        ---------
 10        0000_1010
```
- $(-10) + (-20) = (-30)$
```
-10+       1111_0110+
-20        1110_1100
---        ---------
-30        1110_0010
```

Caracterele alfa-numerice sunt codate în codul ASCII (Engl. *"American Standard Code for Information Interchange"*). Codul ASCII asociază fiecărui caracter un cod binar de 8 biţi.

1.2 Pentru cei ce vor doar să promoveze examenul

1. Realizaţi următoarele conversii:

$2015\vert_{10} = ?\vert_2$	$2015\vert_{10} = ?\vert_{16}$
$0101\_1100\vert_2 = ?\vert_{10}$	$0101_1100\vert_2 = ?\vert_{16}$do
$DA7C\vert_{16} = ?\vert_{10}$	$DA7C\vert_{16} = ?\vert_2$

2. Reprezentaţi următoarele numere întregi, în complement faţă de 2, pe un număr minim de biţi. Prezentaţi codarea acestora pe 8 şi 16 biţi, în complement faţă de 2: $66, -73, 51, -84, 200, -200$

3. Realizaţi următoarele adunări cu numere întregi exprimate în complement faţă de 2, pe 8 biţi:

$70 + 17$	$88 + (-15)$
$120 + (-122)$	$73 + (-73)$
$(-85) + (-15)$	$(-14) + 14$
$(-68) + 15$	$(-66) + 90$

1.3 Pentru cei ce vor să înveţe

1. Enumeraţi primele 20 de numere care sunt puteri ale lui 2.

   Soluţie

$2^0 = 1$				
$2^1 = 2$	$2^5 = 32$	$2^9 = 512$	$2^{13} = 8.192 = 8K$	$2^{17} = 131.072 = 128K$
$2^2 = 4$	$2^6 = 64$	$2^{10} = 1.024 = 1K$	$2^{14} = 16.384 = 16K$	$2^{18} = 262.144 = 256K$
$2^3 = 8$	$2^7 = 128$	$2^{11} = 2.048 = 2K$	$2^{15} = 32.768 = 32K$	$2^{19} = 524.288 = 512K$
$2^4 = 16$	$2^8 = 256$	$2^{12} = 4.096 = 4K$	$2^{16} = 65.536 = 64K$	$2^{20} = 1.048.576 = 1M$

2. Care este cel mai mare număr binar, exprimat pe 16 biţi?
   Care este echivalentul său în baza 10? Dar în baza 16?

   Soluţie

   Cel mai mare număr binar, exprimat pe 16 biţi este $1111\_1111\_1111\_1111\vert_2$. Echivalentul său în baza 10 este $2^{16} - 1 = 65.535\vert_{10}$, iar în baza 16 este $FFFF\vert_{16}$.

3. Care este cel mai mare număr pozitiv, exprimat în baza 10 care se poate reprezenta pe 8 biţi? Dar pe 4 bytes?

   Soluţie

   Cel mai mare număr pozitiv, exprimat în baza 10, care se poate reprezenta pe 8 biţi este $2^8 - 1 = 255$.
   4 bytes= $4 \times 8 = 32$ biţi. Deci, cel mai are număr pozitiv, exprimat în baza 10, care poate fi reprezentat pe 4 bytes este $2^{32} - 1 = 4.294.967.295$.

4. Realizaţi următoarea conversie şi prezentaţi modul în care aţi obţinut rezultatul: $11010111\vert_2 = ?\vert_{16}$

   Soluţie

   Se grupează cifrele binare câte 4, de la dreapta la stânga. Fiecărui grup de 4 cifre binare îi va corespunde o cifră hexazecimală.
   $11010111\vert_2 = 1101\_0111\vert_2 = D7\vert_{16}$ $(1101\vert_2 = D\vert_{16}$ şi $0111\vert_2 = 7\vert_{16})$.

5. Valoarea $100|_{16}$ este egală cu valoarea exprimată în baza 10:
   a) 100 **b)** 256 **c)** 2^{10} **d)** 400

   Soluţie

   $100|_{16} = 1 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0 = 256|_{10}$. Răspuns corect: **b)**.

6. Realizaţi următoarele conversii:

a) $13|_{10} = ?|_2 = ?|_8 = ?|_{16}$
b) $110\_1101|_2 = ?|_{16} = ?|_8 = ?|_{10}$
c) $11101010101|_2 = ?|_{10}$
d) $130|_{10} = ?|_2 = ?|_8 = ?|_{16}$
e) $11\_1000\_1101|_2 = ?|_{16} = ?|_8 = ?|_{10}$
f) $11010011|_2 = ?|_{10}$
g) $1999|_{10} = ?|_2 = ?|_{16}$
h) $2012|_{10} = ?|_2 = ?|_8 = ?|_{16}$
i) $BEEF|_{16} = ?|_2 = ?|_8 = ?|_{10}$
j) $0101\_1010\_1101|_2 = ?|_{16} = ?|_{10}$
k) $201|_{10} = ?|_2 = ?|_8 = ?|_{16}$
l) $ABCD|_{16} = ?|_2 = ?|_8 = ?|_{10}$
m) $10001000|_2 = ?|_{10}$
n) $DEAD\_BEEF|_{16} = ?|_2 = ?|_{10}$
o) $369|_{10} = ?|_2 = ?|_8 = ?|_{16}$
p) $1101\_0110|_2 = ?|_{10} = ?|_8 = ?|_{16}$
q) $372|_8 = ?|_2 = ?|_{10} = ?|_{16}$
r) $DAC|_{16} = ?|_2 = ?|_{10} = ?|_8$

Soluţie

a) $13|_{10} = (8+4+1)|_{10} = (1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0)|_{10} = 1101|_2$
Pentru a converti numărul din baza 2 în baza 8 se grupează cifrele binare câte 3, din dreapta în stânga. Fiecărui grup de 3 cifre binare îi va corespunde o cifră în baza 8.
$1101|_2 = 1\_101|_2 = 15|_8$
$13|_{10} = D|_{16}$.

b) $1101101|_2 = 110\_1101|_2 = 6D|_{16}$;
$1101101|_2 = 1\_101\_101|_2 = 155|_8$;
$155|_8 = (1 \cdot 8^2 + 5 \cdot 8^1 + 5 \cdot 8^0)|_{10} = (64+40+5)|_{10} = 109|_{10}$

c) $111\_0101\_0101|_2 = (1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0)|_{10} = 1024+512+256+64+16+4+1 = 1877|_{10}$.

d) $130|_{10} = (128+2)|_{10} = (1 \cdot 2^7 + 1 \cdot 2^1) = 1000\_0010|_2$.
$10\_000\_010|_2 = 202|_8$.
$1000\_0010|_2 = 82|_{16}$.

e) $11\_1000\_1101|_2 = 38D|_{16}$.
$1\_110\_001\_101|_2 = 1615|_8$.
$38D|_{16} = (3 \cdot 16^2 + 8 \cdot 16^1 + 13 \cdot 16^0) = 768 + 128 + 13 = 909|_{10}$.

f) $1101\_0011|_2 = (1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^0)|_{10} = 128 + 64 + 16 + 2 + 1 = 211|_{10}$.

g) $1999|_{10} = (1024+512+256+128+64+8+4+2+1)|_{10} = (1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0)|_{10} = 111\_1100\_1111|_2$.

h) $2012|_{10} = (1024+512+256+128+64+16+8+4)|_{10} =$
$= (1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2)|_{10} = 111\_1101\_1100)|_2$.
$11\_111\_011\_100|_2 = 3734|_8$.
$111\_1101\_1100|_2 = 7DC|_{16}$.

i) Se reprezintă fiecare cifră din reprezentarea în baza 16 pe 4 biţi, în baza 2:
$BEEF|_{16} = 1011\_1110\_1110\_1111|_2$.
$1\_011\_111\_011\_101\_111 = 137357|_8$.
$BEEF|_{16} = (11 \cdot 16^3 + 14 \cdot 16^2 + 14 \cdot 16 + 15 \cdot 16^0) = 48879|_{10}$.

j) $0101\_1010\_1101|_2 = 5AD|_{16}$.
$5AD|_{16} = (5 \cdot 16^2 + 10 \cdot 16^1 + 13 \cdot 16^0)|_{10} = 1453|_{10}$.

k) $201|_{10} = (128+64+8+1)|_{10} = (1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^0)|_{10} = 1100\_1001|_2$.
$11\_001\_001|_2 = 311|_8$.
$1100\_1001|_2 = C9|_{16}$.

l) $ABCD|_{16} = 1010\_1011\_1100\_1101|_2$.
$1\_010\_101\_111\_001\_101|_2 = 125715|_8$.
$ABCD|_{16} = (10 \cdot 16^3 + 11 \cdot 16^2 + 12 \cdot 16^1 + 13 \cdot 16^0)|_{10} = 43981|_{10}$.

m) $1000\_1000|_2 = (1 \cdot 2^7 + 1 \cdot 2^3)|_{10} = 136|_{10}$.

o) $369|_{10} = (256+64+32+16+1)|_{10} = (1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^0)|_{10} = 1\_0111\_0001|_2$.
$101\_110\_001|_2 = 561|_8$
$1\_0111\_0001|_2 = 171|_{16}$

p) $1101\_0110|_2 = D6|_{16} = (13 \cdot 16^1 + 6 \cdot 16^0)|_{10} = 214|_{10}$
$11\_010\_110|_2 = 326|_8$

7. Rezolvaţi următoarele operaţii, după conversia tuturor operanzilor în baza rezultatului:
a) $100|_{10} + 100|_2 + 100|_8 + 100|_{16} = ?|_{10}$
b) $100|_{10} + 100|_2 + 100|_8 + 100|_{16} = ?|_{16}$
c) $100|_{10} + 100|_2 + 100|_8 + 100|_{16} = ?|_2$
d) $11|_{10} + 11|_8 + 11|_{16} = ?|_{16}$
e) $2012|_{16} + ABCD|_{16} = ?|_{16}$
f) $2012|_{16} + BEEF|_{16} = ?|_{16}$
g) $2012|_{10} + 2012|_8 + 2012|_{16} = ?|_{16}$

Soluţie

a) $100|_2 = 1 \cdot 2^2 = 4|_{10}$;
$100|_8 = 1 \cdot 8^2 = 64|_{10}$;
$100|_{16} = 1 \cdot 16^2 = 256|_{10}$.
În baza 10, calculul devine: $100 + 4 + 64 + 256 = 424|_{10}$.

b) $100|_{10} = (6 \cdot 16^1 + 4)|_{10} = 64|_{16}$;
$100|_2 = 4|_{16}$;
$100|_8 = 1\_000\_000|_2 = 0100\_0000|_2 = 40|_{16}$;
În baza 16, calculul devine $64|_{16} + 4|_{16} + 40|_{16} + 100|_{16} = 1A8|_{16}$.

c) $100|_{10} = (64 + 32 + 4)|_{10} = (1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^2)|_{10} = 0110\_0100|_2$;
$100|_8 = 0100\_0000|_2$;
$100|_{16} = 1\_0000\_0000|_2$;
În baza 2, calculul devine:

```
1_0000_0000+
0_0100_0000
0_0110_0100
0_0000_0100
-----------
1_1010_1000
```

d) $11|_{10} = B|_{16}$
$11|_8 = 001\_001|_2 = 1001|_2 = 9|_{16}$
În baza 16, calculul devine: $B + 11 + 9 = 25|_{16}$.

e), f) Numerele se pot aduna direct în baza 16 prin dispunerea lor unul sub altul şi adunarea de la dreapta la stânga a câte unei cifre, cu considerarea transportului.

```
000   <- transport
2012 +
ABCD
----
CBDF
```

```
011   <- transport
2012 +
BEEF
----
DF01
```

g) $2012|_{10} = (7 \cdot 16^1 + 13 \cdot 16^0 + 12)|_{10} = 7DC|_{16}$;
$2012|_8 = 010\_000\_001\_010|_2 = 0100\_0000\_1010|_2 = 40A|_{16}$;

În baza 16, calculul devine:

```
001   <- transport
2012 +
 7DC
 40A
----
2BF8
```

8. Convertiţi următoarele numere din baza 2 în baza 10 şi prezentaţi modul în care aţi obţinut rezultatul:
11010111, 1010_1010, 11001001001, 11000011010011111, 111_1101_1100, 0011_1000_0111_0000_0111.

Soluţie

O soluţie rapidă este să se convertească numerele din baza 2 în baza 16, prin gruparea a câte 4 biţi, şi apoi din baza 16 în baza 10.

$1101\_0111|_2 = D7|_{16} = (13 \cdot 16^1 + 7 \cdot 16^0)|_{10} = 215|_{10}$
$1010\_1010|_2 = AA|_{16} = (10 \cdot 16^1 + 10 \cdot 16^0)|_{10} = 170|_{10}$
$110\_0100\_1001|_2 = 649|_{16} = (6 \cdot 16^2 + 4 \cdot 16^1 + 9 \cdot 16^0)|_{10} = 1609|_{10}$

9. Convertiţi următoarele numere din baza 10 în baza 2 şi prezentaţi modul în care aţi obţinut rezultatul. Realizaţi aceleaşi conversii prin conversia intermediară a numerelor în baza 16.
104, 1024, 1907, 1921, 1989, 2012, 2020, 4095, 4097, 7777, 8192.

Soluţie

Conversiile în binar din cadrul acestui exerciţiu sunt realizate prin descompunerea numărului reprezentat în baza 10, ca sume ponderate ale puterilor lui 2.
$104|_{10} = (64 + 32 + 8)|_{10} = (1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3) = 110\_1000|_2$
$1024|_{10} = 1 \cdot 2^{10}|_{10} = 100\_0000\_0000|_2$
$1907|_{10} = (1024+512+256+64+32+16+2+1)|_{10} = (1 \cdot 2^{10}+1 \cdot 2^9+1 \cdot 2^8+1 \cdot 2^6+1 \cdot 2^5+1 \cdot 2^4+1 \cdot 2^1+1 \cdot 2^0)|_{10} = 111\_0111\_0011|_2$
$1989|_{10} = (1024+512+256+128+64+4+1)|_{10} = (1 \cdot 2^{10}+1 \cdot 2^9+1 \cdot 2^8+1 \cdot 2^7+1 \cdot 2^6+1 \cdot 2^2+1 \cdot 2^0)|_{10} = 111\_1100\_0101|_2$

În cazul conversiei intermediare în baza 16, se reprezintă numărul în baza 16, iar apoi se reprezintă fiecare cifră în binar pe 4 biţi. Conversia din baza 10 în baza 16 se poate face prin împărţiri succesive la 16 şi considerarea câturilor intermediare şi a restului final.
$104|_{10} = (6 \cdot 16^1 + 8 \cdot 16^0)|_{10} = 68|_{16} = 110\_1000|_2$.
$1024 = (4 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0)|_{10} = 400|_{16} = 100\_0000\_0000|_2$.
$1907|_{10} = (7 \cdot 16^2 + 7 \cdot 16^1 + 3 \cdot 16^0)|_{10} = 773|_{16} = 111\_0111\_0011|_2$.
$1989|_{16} = (7 \cdot 16^2 + 12 \cdot 16^1 + 5 \cdot 16^0)|_{10} = 111\_1100\_0101|_2$.

10. Determinaţi reprezentarea numărului $2012|_{10}$ în binar pe 16 biţi.

Soluţie

$2012|_{10} = (1024+512+256+128+64+16+8+4)|_{10} = (1 \cdot 2^{10}+1 \cdot 2^9+1 \cdot 2^8+1 \cdot 2^7+1 \cdot 2^6+1 \cdot 2^4+1 \cdot 2^3+1 \cdot 2^2)|_{10} = 111\_1101\_1100|_2$.
$2012|_{10} = 111\_1101\_1100|_2 = 0000\_0111\_1101\_1100|_2$.

11. Convertiţi numărul hexazecimal $6ABC$ în baza 2 şi apoi în baza 8. Repetaţi problema pentru numerele $ABCD$, $BEBE$, 2012, $DEAD$, $BEEF$.

Soluţie

Fiecare cifră hexazecimală va genera un grup de 4 cifre binare.
$6ABC|_{16} = 0110\_1010\_1011\_1100|_2 = 0\_110\_101\_010\_111\_100|_2 = 65274|_8$
$ABCD|_{16} = 1010\_1011\_1100\_1101|_2 = 1\_010\_101\_111\_001\_101|_2 = 125715|_8$
$BEBE|_{16} = 1011\_1110\_1011\_1110|_2 = 1\_011\_111\_010\_111\_110|_2 = 137276|_8$

12. Convertiţi numărul zecimal 387 în binar, în două moduri:
a) convertit direct din zecimal în binar,
b) convertit întâi din zecimal în hexazecimal şi apoi din hexazecimal în binar.
Care metodă este mai rapidă? Dar mai sigură? Repetaţi operaţia cu numerele: 57, 101, 202, 400.

Soluţie

a) Conversie direct din zecimal în binar:
$387|_{10} = (256 + 128 + 2 + 1)|_{10} = (1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^1 + 1 \cdot 2^0)|_{10} = 1\_1000\_0011|_2$;
b) Conversie din zecimal în hexazecimal şi din hexazecimal în binar:
$387|_{10} = (1 \cdot 16^2 + 8 \cdot 16^1 + 3 \cdot 16^0)|_{10} = 183|_{16} = 0001\_1000\_0011|_2$.
După realizarea calculelor, se poate concluziona că metoda conversiei prin baza 16 este mai rapidă şi mai sigură, întrucât sunt mai puţine operaţii.

a) $57|_{10} = (32 + 16 + 8 + 1)|_{10} = (1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^0) = 11\_1001|_2$;
b) $57|_{10} = (3 \cdot 16^1 + 9)|_{10} = 39|_{16} = 11\_1001|_2$.

a) $101|_{10} = (64 + 32 + 4 + 1)|_{10} = (1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^2 + 1 \cdot 2^0) = 110\_0101|_2$;
b) $101|_{10} = (6 \cdot 16^1 + 5 \cdot 16^0) = 65|_{16} = 110\_0101|_2$.

13. Convertiţi următoarele numere din binar în hexazecimal grupând câte 4 biţi pentru a forma o cifră în baza 16: 010101010011, 111111010010, 100101101, 11100011100, 1000001, 100, 10111101, 10100111, 1101111010101101101111011101111.

Soluţie

$010101010011|_2 = 0101\_0101\_0011|_2 = 553|_{16}$
$111111010010|_2 = 1111\_1101\_0010|_2 = FD2|_{16}$
$100101101|_2 = 1\_0010\_1101|_2 = 12D|_{16}$
$11100011100|_2 = 111\_0001\_1100|_2 = 71C|_{16}$
$1000001|_2 = 100\_0001 = 41|_{16}$
$100|_2 = 4|_{16}$
$10111101|_2 = 1011\_1101|_2 = BD|_{16}$
$10100111|_2 = 1010\_0111|_2 = A7|_{16}$

14. Convertiţi următoarele numere din hexazecimal în zecimal: EA2, ABC, 777, BABA, ABC5, ABACE, E, 14, 234, FF01. Convertiţi aceleaşi numere în binar prin asocierea a 4 cifre binare fiecărei cifre hexazecimale.

Soluţie

Conversia numerelor din hexazecimal direct în zecimal:
$EA2|_{16} = (14 \cdot 16^2 + 10 \cdot 16^1 + 2 \cdot 16^0)|_{10} = (3584 + 160 + 2)|_{10} = 3746|_{10}$;
$ABC|_{16} = (10 \cdot 16^2 + 11 \cdot 16^1 + 12 \cdot 16^0)|_{10} = (2560 + 176 + 12)|_{10} = 2748|_{10}$;
$777|_{16} = (7 \cdot 16^2 + 7 \cdot 16^1 + 7 \cdot 16^0)|_{10} = (1792 + 112 + 7)|_{10} = 1911|_{10}$;
$BABA|_{16} = (11 \cdot 16^3 + 10 \cdot 16^2 + 11 \cdot 16^1 + 10 \cdot 16^0)|_{10} = (45056 + 2560 + 176 + 10)|_{10} = 47802|_{10}$.

Conversia numerelor din hexazecimal în zecimal, cu conversie intermediară în binar prin gruparea a câte 4 biţi:
$EA2|_{16} = 1110\_1010\_0010|_2 = 3746|_{10}$;
$ABC|_{16} = 1010\_1011\_1100|_2 = 2748|_{10}$;
$777|_{16} = 0111\_0111\_0111|_2 = 1911|_{10}$;
$BABA|_{16} = 1011\_1010\_1011\_1010|_2 = 47802|_{10}$.

15. Scrieţi următoarele numere reprezentate pe 8 biţi, ca mărime şi semn, în complement faţă de 1 şi în complement faţă de 2: 33, -27, 14, -17, 100, -98, 125, -123.
Determinaţi numerele negate în aceleaşi tipuri de codificări: -33, 27, -14, 17, -100, 98, -125, 123.

Soluţie

- $33|_{10} = 10\_0001|_2$. Fiind un număr pozitiv, în toate cele trei reprezentări (MS, C1 şi C2) numărul are aceeaşi reprezentare de cifre binare (7 biţi mărime şi un bit de semn): $33|_{MS,C1,C2} = 0\_0100001$;
- $27|_{10} = 1\_1011|_2 = 0001\_1011|_2$. -27 fiind un număr negativ, cele trei reprezentări (MS, C1 şi C2) sunt diferite, dar au acelaşi bit de semn asociat numerelor negative: 1.
  $-27|_{10} = 1\_0011011|_{MS}$, obţinut prin negarea bitului de semn a reprezentării numărului pozitiv asociat;
  $-27|_{10} = 1\_1100100|_{C1}$, obţinut prin negarea bit cu bit a reprezentării numărului pozitiv asociat;
  $-27|_{10} = 1\_1100101|_{C2}$, obţinut prin adăugarea unei unităţi la reprezentarea $C1$ a numărului.
- $14|_{10} = 1110|_2 = 0\_000\_1110|_{MS,C1,C2}$.
- $17|_{10} = 1\_0001|_2 = 0001\_0001|_2$.
  $-17|_{10} = 1\_001\_0001|_{MS}$
  $-17|_{10} = 1\_110\_1110|_{C1}$
  $-17|_{10} = 1\_110\_1111|_{C2}$
- $100|_{10} = 110\_0100|_2 = 0\_110\_0100|_{MS,C1,C2}$.

16. Determinaţi numerele întregi reprezentate în complement faţă de 2 după cum urmează (bitul cel mai semnificativ este bitul de semn): 0110110, 1101111010101101, 01011010011, 1100000001, 001111110, 110001010.

Soluţie

- $0\_11\_0111|_{C2} = +37|_{16} = +54|_{10}$
- $1\_101\_1110\_1010\_1101|_{C2}$ are bitul de semn egal cu 1, deci este negativ.
  $\overline{1\_101\_1110\_1010\_1101}|_{C2} + 1 = 0\_010\_0001\_0101\_0011|_2 = 2153|_{16} = 2 \cdot 16^3 + 1 \cdot 16^2 + 5 \cdot 16^1 + 3 \cdot 16^0 = 8531_{10}$.
  Deci, numărul în baza 10 este $-8531|_{10}$
- $0\_10\_1101\_0011|_{C2} = +2D3|_{16} = +723|_{10}$

17. Demonstraţi următorul algoritm de aflare a complementului faţă de 2 al unui număr reprezentat în binar: *"se copiază biţii de la dreapta la stânga până la primul bit egal cu 1, inclusiv, iar ceilalţi biţi se neagă"*.
Exemplificaţi modul de aplicare a algoritmului în comparaţie cu algoritmul clasic:

"se neagă toţi biţii şi se adaugă 1".
Exemplificaţi cu numerele 53 şi 116.

Soluţie

a) $53|_{10} = 0011\_0101|_2$
$-53|_{10} = \overline{0011\_0101} + 1 = 1100\_1011|_{C2}$, se adaugă 1 la reprezentarea complementată a numărului pozitiv.
$-53|_{10} = 1100\_1011|_{C2}$, din reprezentarea numărului pozitiv, de la dreapta la stânga, se copiază primul bit (este 1) şi apoi se neagă ceilalţi.

b) $116|_{10} = 0111\_0100|_2$.
$-116|_{10} = \overline{0111\_0100} + 1 = 1000\_1100|_{C2}$, se adaugă 1 la reprezentarea complementată a numărului pozitiv.
$-116|_{10} = 1000\_1100|_{C2}$, din reprezentarea numărului pozitiv, de la dreapta la stânga, se copiază primii 3 biţi (doi de 0 şi un 1) şi apoi se neagă ceilalţi.

18. Precizaţi mulţimea numerelor ce pot fi reprezentate astfel:

a) numere pozitive, pe 7 biţi;

b) numere întregi, codificate ca mărime şi semn (MS), pe 7 biţi;

c) numere întregi, codificate în complement faţă de 1 (C1), pe 7 biţi;

d) numere întregi, codificate în complement faţă de 2 (C2), pe 7 biţi;

e) numere pozitive/întregi codificate MS/C1/C2, pe 8 biţi;

f) numere pozitive/întregi codificate MS/C1/C2, pe 4 bytes.

Determinaţi domeniile de reprezentare a numerelor codificate în MS/C1/C2 pentru un număr N de biţi. Utilizaţi formulele determinate pentru 8/16/32/64 biţi.

Soluţie

a) Pe N biţi, pot fi reprezentate 2^N numere pozitive, între 0 şi $2^N - 1$. Pentru $N = 7$ biţi, pot fi reprezentate numerele pozitive între 0 şi $2^7 - 1$, adică între 0 (000_0000) şi 127 (111_1111).

b) Numerele întregi reprezentate ca "mărime şi semn" au un bit (cel mai semnificativ) rezervat pentru semn. Rămân $N - 1$ biţi pentru codificarea mărimii. Însă, pentru cazul când mărimea este zero nu are semnificaţie combinaţia "număr negativ nul". Domeniul numerelor reprezentabile este între $-(2^{N-1} - 1)$ şi $(2^{N-1} - 1)$. Pentru $N = 7$ biţi, domeniul numerelor ce pot fi reprezentate este între $-(2^{(7-1)} - 1)$ şi $(2^{(7-1)} - 1)$, adică între -63 şi 63. Numărul binar 000_0000 reprezintă numărul 0 (zero). Combinaţia binară 100_0000 nu este legală (poate fi considerată asociată numărului "negativ" zero).

c) Reprezentarea numerelor pozitive în complement faţă de 1 este identică cu reprezentarea numerelor pozitive în domeniul 0 şi $2^{N-1} - 1$. Reprezentarea numerelor negative în complement faţă de 1 este obţinută din reprezentarea numerelor pozitive asociate şi negarea bit cu bit. Rezultă domeniu între $-(2^{N-1} - 1)$ şi 0. În acest mod, numărul zero are asociate două combinaţii (toţi biţii 0 sau toţi biţii 1). Domeniul de reprezentare este între $-(2^{N-1} - 1)$ şi $(2^{N-1} - 1)$. Pentru N=7 biţi, domeniul de reprezentare al numerelor în complement faţă de 1 este între $-(2^{7-1} - 1)$ şi $2^{7-1} - 1$, adică între -63 şi 63. Combinaţiile 000_0000 şi 111_1111 sunt asociate numărului 0 (zero).

d) În cazul numerelor întregi reprezentate în complement faţă de 2, domeniul este între -2^{N-1} şi $2^{N-1} - 1$. Pentru N=7 biţi, domeniul de reprezentare este între -64 şi 63. Numărul zero are o singură reprezentare 000_0000. Numărul pozitiv maxim este 63, reprezentat ca 011_1111. Numărul negativ -63 este reprezentat ca 100_0001. Numărul negativ minim este -64, reprezentat ca 100_0000.

19. Să se convertească numerele din zecimal în binar utilizând metoda scăderii ponderate. Să se verifice rezultatul prin efectuarea conversiilor inverse, din binar în zecimal, utilizând metoda adunării ponderate. Numere considerate: 23, 55, 100, 178.

Soluţie

```
23- 7- 3- 1-         55- 23- 7- 3- 1-          100- 36- 4-          178- 50- 18- 2-
16  4  2  1          32  16  4  2  1            64  32  4           128  32  16  2
==  =  =  =          ==  ==  =  =  =           ===  ==  =           ===  ==  ==  =
 7  3  1  0          23   7  3  1  0            36   4  0            50  18   2  0

16  8  4  2  1     32 16  8  4  2  1     64 32 16  8  4  2  1     128 64 32 16  8  4  2  1
 1  0  1  1  1      1  1  0  1  1  1      1  1  0  0  1  0  0       1  0  1  1  0  0  1  0
```

Verificarea rezultatului prin metoda adunării ponderilor presupune însumarea puterilor lui 2 ale căror indecşi apar în reprezentarea binară:
$1\_0111|_2 = (16 + 4 + 2 + 1)|_{10} = 23|_{10}$
$11\_0111|_2 = (32 + 16 + 4 + 2 + 1)|_{10} = 55|_{10}$
$110\_0100|_2 = (64 + 32 + 4)|_{10} = 100|_{10}$
$1011\_0010|_2 = (128 + 32 + 16 + 2)|_{10} = 178|_{10}$

20. Aflaţi reprezentarea binară a numărului $0010100100010111|_{BCD}$.

Soluţie

Se grupează numărul reprezentat în format BCD în grupuri de câte 4 biţi, de la dreapta la stânga, obţinându-se: $0010\_1001\_0001\_0111$. Pentru fiecare grup de 4 biţi ce reprezintă o cifră în format BCD, se află corespondentul în baza 10: $0010 = 2, 1001 = 9, 0001 = 1, 0111 = 7$. Rezultă că $0010\_1001\_0001\_0111|_{BCD} = 2917|_{10}$. Prin transformarea numărului din baza 10 în baza 2, rezultă: $2917|_{10} = 1011\_0110\_0101|_2$.
În concluzie, $0010\_1001\_0001\_0111|_{BCD} = 1011\_0110\_0101|_2$.

21. Determinaţi reprezentările următoarelor numere întregi în formatele pe 16 biţi: mărime şi semn, complement faţă de 1 şi complement faţă de 2.
a) -5 **b)** -10 **c)** -100 **d)** -10000 **e)** -29876.

Soluţie

a) $5|_{10} = 101|_2 = 0000\_0000\_0000\_0101|_2$
Prin schimbarea în 1 a bitului cel mai semnificativ: $-5|_{10} = 1000\_0000\_0000\_0101|_{MS}$
Prin complementarea bit cu bit a reprezentării binare a numărului pozitiv: $-5|_{10} = 1111\_1111\_1111\_1010|_{C1}$
Prin adăugarea unei unităţi la reprezentarea în complement faţă de 1: $-5|_{10} = 1111\_1111\_1111\_1011|_{C2}$

b) $10|_{10} = 1010|_2 = 0000\_0000\_0000\_1010|_2$
$-10|_{10} = 1000\_0000\_0000\_1010|_{MS} = 1111\_1111\_1111\_0101|_{C1} = 1111\_1111\_1111\_0110|_{C2}$

c) $100|_{10} = 110\_0100|_2 = 0000\_0000\_0110\_0100|_2$
$-100|_{10} = 1000\_0000\_0110\_0100|_{MS} = 1111\_1111\_1001\_1011|_{C1} = 1111\_1111\_1001\_1100|_{C2}$

d) $1000|_{10} = 11\_1110\_1000|_2 = 0000\_0011\_1110\_1000|_2$
$-1000|_{10} = 1000\_0011\_1110\_1000|_{MS} = 1111\_1100\_0001\_0111|_{C1} = 1111\_1100\_0001\_1000|_{C2}$

e) $29876|_{10} = 111\_0100\_1011\_0100|_2 = 0111\_0100\_1011\_0100|_2$
$-29876|_{10} = 1111\_0100\_1011\_0100|_{MS} = 1000\_1011\_0100\_1011|_{C1} = 1000\_1011\_0100\_1100|_{C2}$

22. Reprezentaţi numerele 123 şi 789 în format BCD şi apoi însumaţi-le. Repetaţi operaţia cu numerele (777 şi 456), (839 şi 352), (784 şi 609).

Soluţie

Se plasează cele două numere unul sub altul, ca la o adunare în baza 10:

```
123+      0001_0010_0011+
789       0111_1000_1001
---       --------------
912       1001_0001_0010
```

Adunarea cifrelor BCD nu trebuie făcută ca a numerelor în binar deoarece, în cazul numerelor BCD, transportul este generat la valoarea 10. Astfel $0011 + 1001 = 0010, transport = 1$, adică $3 + 9 = 2, transport = 1$. La fel $0010 + 1000 + 1 = 0001, transport = 1$ $(2 + 8 + 1 = 1, transport = 1)$ şi $0001 + 0111 + 1 = 1001, transport = 0$ $(1 + 7 + 1 = 9)$.

```
 839+       1000_0011_1001+         784+        0111_1000_0100+
 352        0011_0101_0010          609         0110_0000_1001
----    -------------------        ----    -------------------
1191    0001_0001_1001_0001        1393    0001_0011_1001_0011
```

23. Scrieţi-vă numele cu caractere binare în cod ASCII.

    Soluţie

    Să presupunem următorul nume: **Ion Stelescu**. Fiecare literă poate fi codată în ASCII şi exprimată în hexazecimal, după cum urmează:
    $I = 49|_{16}$, $o = 6F|_{16}$, $n = 6E|_{16}$, $"space" = 20|_{16}$, $S = 53|_{16}$, $t = 74|_{16}$, $e = 65|_{16}$, $l = 6C|_{16}$, $e = 65|_{16}$, $s = 73|_{16}$, $c = 63|_{16}$, $u = 75|_{16}$.

    Convertind codurile din baza 16 în baza 2 rezultă scrierea în binar a numelui **Ion Stelescu**:

```
49 6F 6E 20                                 Ion_
53 74 65 6C 65 73 63 75                     Stelescu

0100_1001 0110_1111 0110_1110 0010_0000
0101_0011 0111_0100 0110_0101 0110_1100 0110_0101 0111_0011 0110_0011 0111_0101
```

24. Decodaţi următorul text scris în binar cu coduri ASCII: 01000101 01001100 01000101 01000011 01010100 01010010 01001111 01001110 01001001 01000011 01000001 00100000 01000100 01001001 01000111 01001001 01010100 01000001 01001100 01000001.

    Soluţie

```
0100_0101 0100_1100 0100_0101 0100_0011 0101_0100 0101_0010 0100_1111
   4 5       4 C       4 5       4 3       5 4       5 2       4 7
    E         L         E         C         T         R         O

0100_1110 0100_1001 0100_0011 0100_0001 0010_0000
   4 E       4 9       4 3       4 1       2 0
    N         I         C         A

0100_0100 0100_1001 0100_0111 0100_1001 0101_0100 0100_0001 0100_1100 0100_0001
   4 4       4 9       4  7      4 9       5 4       4 1       4 C       4 1
    D         I         G         I         T         A         L         A
```

25. Decodaţi următorul text scris în hexazecimal cu coduri ASCII: 41 63 65 61 73 74 61 20 63 61 72 74 65 20 61 20 61 70 61 72 75 74 20 69 6E 20 61 6E 75 6C 20 32 30 31 32 2E.
    Transcrieţi mesajul în binar.

    Soluţie

    Textul decodat este (ASCII, binar, caracter):

```
   41       63       65       61       73       74       61       20
01000001 01100011 01100101 01100001 01110011 01110100 01100001 00100000
   A        c        e        a        s        t        a

   63       61       72       74       65       20       61       20
01100011 01100001 01110010 01110100 01100101 00100000 01100001 00100000
   c        a        r        t        e                 a

   61       70       61       72       75       74       20       69       6E       20
01100001 01110000 01100001 01110010 01110101 01110100 00100000 01101001 01101110 00100000
   a        p        a        r        u        t                 i        n

   61       6E       75       6C       20       32       30       31       32       2E
01100001 01101110 01110101 01101100 00100000 00110010 00110000 00110001 00110010 00101110
   a        n        u        l                 2        0        1        2        .
```

26. Cu ce operaţie aritmetică se pot determina codurile ASCII ale caracterelor ce reprezintă cifre, pe baza cifrei respective?

Soluţie

În codul ASCII, caracterele asociate cifrelor sunt codificate cu primele 4 cifre binare 0011. Ultimele 4 cifre binare reprezintă cifra scrisă în binar. Astfel, codul ASCII al caracterului "0" este $0011\_0000|_2 = 30|_{16}$. Se poate considera că pentru a obţine codul ASCII al unei cifre, trebuie adunată acea cifră la codul ASCII asociat caracterului "0". Astfel: "1"="0"+1, "7"="0"+7, "9"="0"+9. Această observaţie este utilizată pentru conversia numerelor reprezentate în binar în codul ASCII corespunzător cifrei, în vederea afişării acestuia în sisteme de afişaj.

27. Ce bit trebuie complementat în codul ASCII pentru a converti un şir de litere minuscule în corespondentele lor majuscule?

Soluţie

Între codurile ASCII asociate unei litere majuscule şi aceeaşi literă minusculă este o diferenţă de $20|_{16} = 32|_{10} = 2^5$. Cu alte cuvinte, minusculele şi majuscule au acelaşi cod ASCII cu excepţia bitului 5 care este 0 la litera majusculă şi 1 la litera minusculă. Pentru conversia unei litere din minusculă în majusculă trebuie complementat bitul 5.

28. Ce valoare în baza 10 reprezintă şirul binar 1000_1001_0111 dacă valoarea este codificată în:

a) cod BCD
b) cod complement faţă de 1
c) cod complement faţă de 2
d) număr pozitiv.

Soluţie

a) $1000\_1001\_0111|_{BCD} = 897|_{10}$

b) $1000\_1001\_0111|_{C1} = -(0111\_0110\_1000|_2) = -(2^{10}+2^9+2^8+2^6+2^5+2^3) = -(1024+512+256+64+32+8) = -1896|_{10}$

c) $1000\_1001\_0111|_{C2} = -(0111\_0110\_1000|_2+1) = -(0111\_0110\_1001|_2) = -(1024+512+256+64+32+8+1) = -1897|_{10}$

d) $1000\_1001\_0111|_2 = 2^{11} + 2^7 + 2^4 + 2^2 + 2^1 + 2^0 = 2048 + 128 + 16 + 4 + 2 + 1 = 2199|_{10}$

29. Convertiţi următoarele numere reprezentate în BCD în baza 10 şi prezentaţi modul în care aţi obţinut rezultatul: 10010111, 10000010, 11001001001, 11000011010011000, 11101011000, 001000110001000101110101.
Convertiţi în binar numerele prezentate.

Soluţie

Pentru a converti un număr reprezentat în format BCD în format zecimal, se grupează câte 4 biţi, de la dreapta la stânga. Fiecare grup de 4 cifre binare va genera o cifră a numărului exprimat în baza 10. Conversia în binar se va face din reprezentarea în baza 10.
$1001\_0111|_{BCD} = 97|_{10} = 110\_0001|_2$,
$1000\_0010|_{BCD} = 82|_{10} = 101\_0010|_2$,
$110\_0100\_1001|_{BCD} = 649|_{10} = 10\_1000\_1001|_2$,
$1\_1000\_0110\_1001\_1000|_{BCD} = 18698|_{10} = 100\_1001\_0000\_1010|_2$,
$111\_0101\_1000|_{BCD} = 758|_{10} = 10\_1111\_0110|_2$,
$0010\_0011\_0001\_0001\_0111\_0101|_{BCD} = 23117|_{10} = 101\_1010\_0100\_1101|_2$.

30. Un sistem de calcul operează cu numere reprezentate pe 48 de biţi. Câte numere pot fi reprezentate cu acest număr de biţi, dacă se utilizează următoarele coduri?
a) binar **b)** BCD **c)** ASCII

Soluţie

a) $2^{48} = 281.474.976.710.656$

b) $10^{(48/4)} = 10^{12} = 1.000.000.000.000$

c) $10^{(48/8)} = 10^6 = 1.000.000$

31. Determinaţi numărul minim de biţi utilizaţi pentru a codifica fiecare caracter dintr-o tastatură cu următoarele numere de taste: 9, 16, 22, 36, 104.

Soluţie

Numărul de biţi necesar se obţine rezolvând inecuaţia $2^{\,biţi} \geq \textit{număr taste}$.

Rezultă numărul de biţi necesar: 4, 4, 5, 6, 7.

32. Dacă următoarele numere de biţi sunt utilizate pentru a codifica fiecare caracter dintr-o tastatură, determinaţi numărul maxim de taste: 4, 5, 6, 7, 8.

    Soluţie

    Numărul de taste care se pot codifica se obţine rezolvând ecuaţia $număr\ taste = 2^{\,biţi}$. Rezultă numărul maxim de taste codificate: 16, 32, 64, 128, 256.

33. Determinaţi codul pentru comanda unui afişaj cu 7 segmente care să prezinte următoarele caractere: E, b, C, d, h, A, 7, 9. Pentru a lumina un segment acesta trebuie comandat cu valoarea logică 1.

    Soluţie

    Se consideră denumirea segmentelor ca în figura 1.1. Rezultă codurile, în ordinea $S_0S_1S_2S_3S_4S_5S_6$:
    $E = 1001111$, $b = 0011111$, $C = 1001110$, $d = 0111101$, $h = 0010111$, $A = 1110111$, $7 = 1110000$, $9 = 1111011$.

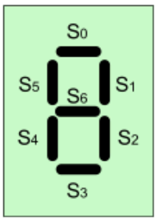


Figura 1.1 Afişaj cu 7 segmente referit la problema 33.

34. Asociaţi coduri binare celor 52 cărţi de joc (folosiţi un număr minim de biţi). Precizaţi patternul binar pentru selecţia următoarelor:
    a) toate cărţile de treflă;
    b) toate cărţile de roşu;
    c) toate cărţile de 9 (indiferent de culoare);
    d) cărţile de 10, roşu;
    e) cărţile de negru mai mari decât 7;
    f) cărţile mai mici decât 4.

    Soluţie

    Cărţile de joc sunt de 4 feluri: două de culoare roşie (inimă roşie şi romb) şi două de culoare neagră (inimă neagră şi treflă). Sunt necesari 2 biţi pentru a codifica cele 4 tipuri de cărţi. Dacă se codifică într-un anumit mod, se poate face ca un bit din cei doi să codifice culoarea cărţii. De exemplu, se codifică:

    00 = inimă roşie, 01 = romb, 10 = inimă neagră, 11 = treflă

    Din fiecare fel sunt 13 cărţi de joc, cu simbolurile: $2, 3, 4, 5, 6, 7, 8, 9, 10, A, J, Q, K$. Pentru codificarea acestora este nevoie de minim 4 biţi ($2^4 \geq 13$).
    Se codifică: $A = 1011, J = 1100, Q = 1101, K = 1110$.
    În total, sunt folosiţi 6 biţi ($2^6 = 64$ combinaţii) pentru a codifica 52 de obiecte. Deci, vor fi 12 coduri nealocate (codurile celor 4 culori, terminate cu $0000, 0001, 1111$. O parte din acestea pot fi alocate pentru cărţi speciale (joker). În cazul codificării propuse, patternul de selecţie a grupelor precizate este:

a) toate cărţile de treflă:	`11XXXX`
b) toate cărţile de roşu:	`0XXXXX`
c) toate cărţile de 9 (indiferent de culoare):	`XX1001`
d) cărţile de 10 roşu:	`0X1010`
e) cărţile de negru mai mari decât 7:	`1X1XXX`
f) cărţile mai mici decât 4:	`XX00XX`

35. Să se propună codificarea disciplinelor din planul de învăţământ, cu un număr minim de biţi şi care să conţină următoarele informaţii:

    - *specializarea*;
    - *disciplina*;
    - *tipul disciplinei* (O = obligatorie sau F = facultativă);

- *semestrul* în care se desfăşoară (4 ani, câte două semestre pe an, o disciplină se poate desfăşura în unul sau două semestre).

Să se propună codarea corelată atât pentru operatorul uman (uşor de interpretat de către oameni) cât şi pentru gestionarea de către calculator (codificare binară).

Propuneţi un cod unic pe ţară care să includă informaţii despre:

- *facultate*;
- *universitate*;
- *oraş*.

Includeţi şi caracteristici ale disciplinelor:

- *conţinut* (DF = disciplină fundamentală, DD = disciplină în domeniu, DS = disciplină de specialitate, DC = disciplină complementară)
- *opţionalitate* (DI = disciplină obligatorie, impusă, DO = disciplină opţională, la alegerea studenţilor).

Calculaţi numărul de biţi necesari pentru codificarea unei discipline.

Pe baza codificării binare, propuneţi patternuri de selecţie a disciplinelor pe anumite criterii.

- toate disciplinele unei specializări;
- toate disciplinele fundamentale din anul 2, ale unei specializări;
- toate disciplinele din semestrul 3 (semestrul 1, anul 2);
- toate disciplinele de specialitate, opţionale, din anul 2, ale unei specializări.

36. Justificaţi valoarea de adevăr a fiecărei afirmaţii:
a) Numărul binar 1111_0101 reprezintă numărul -10 exprimat în complement faţă de 2.
b) $101|_{10} + 100|_{16} = 1\_0110\_0101|_2$
c) $101|_{10} + 101|_{16} = 1\_0110\_0110|_2$

Soluţie

a) $10|_{10} = 0000\_1010|_2$, rezultă $-10 = 1111\_0101 + 1 = 1111\_0110|_{C2}$, deci afirmaţia este falsă.
$1111\_0101|_2$ reprezintă un număr negativ (deoarece are cel mai semnificativ bit egal cu 1).
$0000\_1010+1 = 0000\_1011$ Rezultă că numărul binar considerat reprezintă numărul $-11|_{10}$ în format complement faţă de 2 pe 8 biţi.
b) $101|_{10} = 64 + 32 + 4 + 1 = 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0110\_0101|_2$.
$100|_{16} = 1\_0000\_0000|2$, deci afirmaţia este adevărată.

```
  0110_0101+
1_0000_0000
-----------
1_0110_0101
```

d) Se poate observa că faţă de afirmaţia de la punctul **b)**, primul operand este acelaşi, iar cel de al doilea este mai mare cu 1, decât operanzii sumei de mai sus. Astfel, rezultatul acestei sume se poate afla prin adunarea la rezultatul anterior cu 1. Deci, afirmaţia este adevărată.

```
1_0000_0101+
          1
-----------
1_0110_0110
```

37. Reprezentarea în complement faţă de 2 a operaţiei $-4 + 7 = 3$ este:
a) $0000\_0100 + 1111\_1001 = 0000\_0011$;
b) $0100 + 0111 = 0011$;
c) $11\_1100 + 00\_0111 = 10\_0011$;
d) $1111\_1100 + 0000\_0111 = 0000\_0011$.

Soluţie

$4|_{10} = 0000\_0100|_2$, $-4|_{10} = 1111\_1100|_2$
$7|_{10} = 0000\_0111|_2$

$3|_{10} = 0000\_0011|_2$
Răspunsul corect este **d)**.

Răspunsul se poate determina şi fără a face conversia în reprezentarea binară a operanzilor dacă se observă următoarele:
- Răspunsurile **a)** şi **b)** sunt greşite, deoarece primul operand (-4) trebuie să aibă cel mai semnificativ bit egal cu 1, fiind negativ.
- Răspunsul **c)** este greşit, deoarece rezultatul 3 este pozitiv, deci ar trebui să aibă bitul de semn egal cu 0.

38. Realizaţi următoarele operaţii în baza de numeraţie precizată:
a) $371|_{16} + 4DAC|_{16} =?|_{16}$
b) $1101\_0001|_2 + 1101\_0111|_2 =?|_2$

Soluţie

Operaţiile se pot realiza prin dispunerea operanzilor unul sub altul şi adunarea cifrelor hexazecimale sau binare una câte una, începând de la cea mai puţin semnificativă (din partea dreaptă), cu considerarea transportului (menţionat pe primul rând, lângă simbolul <=).

```
1100 <=        110101110 <= transport
 371+           11010001 +
4DAC            11010111
====           ==========
511D           110101000
```

39. Aflaţi numărul negativ reprezentat pe 8 biţi în complement faţă de 2 asociat numărului binar pozitiv 0010_1110. Prezentaţi modul în care aţi obţinut rezultatul.

Soluţie

Numărul negativ asociat se obţine din reprezentarea binară a numărului pozitiv prin complementarea tuturor biţilor şi adăugarea unei unităţi:

$\overline{0010\_1110} = 1101\_0001$, $1101\_0001 + 1 = 1101\_0010$.

40. Adunaţi şi înmulţiţi următoarele numere binare fără a le converti în baza 10: (1101, 101), (1010, 1010), (101, 111).

Soluţie

Numerele binare pot fi înmulţite după acelaşi algoritm ca şi numerele zecimale prin plasarea operanzilor unul sub altul şi înmulţirea fiecărei cifre a celui de-al doilea operand cu fiecare cifră a primului operand. Operaţiile de înmulţire binare au rezultate naive: $0 \times 0 = 1 \times 0 = 0 \times 1 = 0$ şi $1 \times 1 = 1$. Ca o consecinţă, înmulţirea unui operand cu 1 este egală cu acel operand, iar înmulţirea unui operand cu 0 este egală cu 0. Însumarea rezultatelor parţiale trebuie să ţină cont de un eventual transport.

```
   1101x             1010x             101x
    101              1010              111
   ====             =====              ===
   1101              0000              101
  0000              1010              101
 1101              0000              101
=======           1010              ======
1000001           =======           100011
                  1100100
```

41. Realizaţi următoarele scăderi ale numerelor pozitive precizate prin determinarea numărului negativ asociat scăzătorului şi realizarea unei operaţii de adunare, conform formulei $A - B = A + (-B)$. Extindeţi bitul de semn pentru a putea reprezenta numerele negative.
a) 11011 – 11001   **b)** 110110 – 10101   **c)** 1100 – 110111   **d)** 1001010 – 1001011

Soluţie

Extinderea bitului de semn are ca scop obţinerea unei forme de reprezentare echivalente pe un număr mai mare de biţi, necesari pentru efectuarea unor operaţii aritmetice care generează un rezultat ce nu poate fi reprezentat pe numărul iniţial de biţi. Extinderea bitului de semn constă în adăugarea unor biţi egali ca valoare cu bitul de semn, în partea stângă, ca biţi mai semnificativi.

a) Numerele pozitive au 5 biţi. Pentru a reprezenta numerele negative asociate este necesar extinderea operanzilor cu un bit suplimentar $101011 - 11001 = 0\_11011 - 0\_11001$.
Operaţia binară corespunde operaţiei cu numere zecimale $27 - 25 = 27 + (-25) = 2$.
$-25 = -(0\_11001|_{C_2}) = 1\_00110 + 1 = 1\_00111$

```
011011+
100111
======
000010
```

b) Cel mai mare număr pozitiv are 6 biţi. Pentru a reprezenta numerele negative asociate este necesar extinderea operanzilor la 7 biţi: $110110 - 10101 = 0\_110110 - 0\_010101$
Operaţia binară corespunde operaţiei cu numere zecimale $54 - 21 = 54 + (-21) = 33$.
$-21 = -(0\_010101|_{C_2}) == 1\_101010 + 1 = 1\_101011$

```
0110110+
1101011
=======
0100001
```

c) Cel mai mare număr pozitiv are 6 biţi. Pentru a reprezenta numerele negative asociate este necesar extinderea operanzilor la 7 biţi: $1100 - 110111 = 0\_001100 - 0\_110111$
Operaţia binară corespunde operaţiei cu numere zecimale $12 - 55 = 12 + (-55) = -43$.
$-43 = -(0\_110111|_{C_2}) = 1\_001000 + 1 = 1\_001001$

```
0001100+
1001001
=======
1010101
```

$-(1\_010101|_{C_2}) = 0\_101011 = 43|_{10}$

d) Operaţia binară corespunde operaţiei cu numere zecimale $74 - 75 = 74 + (-75) = -1$.

```
01001010+
10110101
========
11111111
```

42. Reprezentaţi următoarele operaţii cu numere codificate în complement faţă de 2 pe 16 biţi. Verificaţi rezultatele prin conversia numerelor în baza 10.

a) (+87) + (+22) **c)** (-87) + (+22) **e)** (+8765) + (+502) **g)** (+8765) + (-502)
b) (+87) + (-22) **d)** (-87) + (-22) **f)** (-8765) + (+502) **h)** (-8765) + (-502)

Soluţie

a) $0000\_0000\_0101\_0111|_2 + 0000\_0000\_0001\_0110|_2 = 0000\_0000\_0110\_1101|_2 = 109|_{10}$;
b) $0000\_0000\_0101\_0111|_2 + 1111\_1111\_1110\_1010|_2 = 0000\_0000\_0100\_0001|_2 = 65|_{10}$;
c) $1111\_1111\_1010\_1001|_2 + 0000\_0000\_0001\_0110|_2 = 1111\_1111\_1011\_1111|_2 = -65|_{10}$;

43. Adunaţi, scădeţi şi înmulţiţi următoarele numere fără a le converti în zecimal.
a) $451|_8$ şi $175|_8$ **b)** $451|_{16}$ şi $175|_{16}$ **c)** $2E|_{16}$ şi $ABC|_{16}$ **d)** $1100|_2$ şi $10\_1011|_2$
Soluţie

a) $451|_8 + 175|_8 = 646|_8$
$451|_8 - 175|_8 = 254|_8$
$451|_8 \times 175|_8 = 110405|_8$

b) $451|_{16} + 175|_{16} = 5C6|_{16}$
$451|_{16} - 175|_{16} = 2DC|_{16}$
$451|_{16} \times 175|_{16} = 64A05|_{16}$

44. Realizaţi următoarele operaţii cu numere reprezentate în complement faţă de 2:

$7 + (-3)$ $97 + 28$ $100 + (-7)$ $38 + (-23)$ $40 + (-40)$ $58 + (-42)$
$7 - (-3)$ $97 - 28$ $100 - (-7)$ $-38 - (-23)$ $40 - (-40)$ $-58 - (-42)$.

1.4 Pentru cei ce vor să devină profesionişti

1. Câte cifre sunt egale cu 1 în reprezentarea binară a următoarelor numere zecimale: 4, 16, 33, 63, 126, 257, 1022? Găsiţi o modalitate mai rapidă de determinare, fără a face conversia numerelor în baza 2.

   Soluţie

   - 4 este o putere a numărului 2, $4 = 2^2$. Deci, în reprezentarea binară, are un singur 1.
   - Acelaşi raţionament se face pentru $16 = 2^4$.
   - $33 = 32 + 1 = 1 \cdot 2^5 + 1 \cdot 2^0$. Deci, are două cifre egale cu 1 în reprezentarea binară.
   - Acelaşi raţionament se face pentru $257 = 256 + 1 = 1 \cdot 2^8 + 1 \cdot 2^0$. Deci, două cifre egale cu 1.
   - 63 poate fi scris ca $64 - 1$. $64 = 2^6$, deci are reprezentarea 1000000 (un 1 urmat de 6 cifre 0). Dacă se scade 1, rezultă un număr binar cu 6 cifre de 1 (0111111).
   - Acelaşi raţionament se poate face pentru $1022 = 1023 - 1$, $1023 = 2^{10} - 1$ are 10 cifre 1, rezultă că 1022 are 9 cifre egale cu 1.

2. Un convertor analog-digital (Engl. *"ADC = Analog-Digital Converter"*) are o intrare analogică V_{in} şi n ieşiri digitale $D_{n-1}...D_0$. Circuitul prezintă la ieşire tensiunea analogică de la intrare, codificată binar. Un circuit ADC este caracterizat de mărimile: rezoluţia de bit (R_{bit}), domeniul maxim al tensiunii de intrare (V_{max}) şi numărul de biţi (n). Cele trei mărimi sunt în relaţia: $V_{max} = 2^n \times R_{bit}$. Determinaţi numărul minim de biţi de ieşire pentru fiecare ADC:
   a) Rezoluţie 0.25V, tensiune maximă 5V,
   b) Rezoluţie 70mV, tensiune maximă 10V,
   c) Rezoluţie 8mV, tensiune maximă 12V,
   d) Rezoluţie 0.75mV, tensiune maximă 12V.

   Soluţie

   Utilizând relaţia $V_{max} = 2^n \times R_{bit}$ rezultă numărul minim de biţi:

   a) $R_{bit} = 0.25V$, $V_{max} = 5V$, $n = 5$;

   b) $R_{bit} = 70mV$, $V_{max} = 10V$, $n = 8$;

   c) $R_{bit} = 8mV$, $V_{max} = 12V$, $n = 11$;

   d) $R_{bit} = 0.75mV$, $V_{max} = 12V$, $n = 14$.

3. Un convertor digital-analog (Engl. *"DAC = Digital-Analog Converter"*) are o intrare digitală pe n biţi $D_{n-1}...D_0$ şi produce la ieşire o tensiune analogică V_{out} în 2^n trepte de tensiune. Determinaţi numărul minim de biţi de intrare pentru ca circuitul DAC să producă o tensiune de ieşire cu următoarele numere de trepte: 32, 128, 425, 1024.

   Soluţie

   Relaţia dintre numărul de biţi ai convertorului digital-analog şi numărul maxim de trepte este:
   $număr\ trepte \leq 2^{\,biţi}$.

   Rezultă numerele de biţi: 5, 7, 9, 10.

4. Explicaţi funcţionarea tastaturii de telefon prezentată în figura 1.2. Care este codul binar $O_3O_2O_1O_0I_2I_1I_0$ care identifică apăsarea tastelor: 1, *, 5, 7, 6?

   Soluţie

   Circuitul prezintă o matrice de butoane prin apăsare, normal deschise. Porturile de intrare ale circuitului de comandă pentru tastatură, $I_2I_1I_0$, sunt conectate prin rezistenţe la tensiunea de alimentare. Asta înseamnă că, în mod normal, când butonul este neapăsat, portul de intrare primeşte valoarea logică 1. Prin apăsarea unui buton, intrarea corespunzătoare primeşte valoarea liniei de ieşire corespunzătoare $O_3O_2O_1O_0$. Dacă pe linia de ieşire (a rândului) se aplică valoarea logică 0, se va putea determina dacă butonul a fost apăsat sau nu.
   Circuitul de comandă activează secvenţial câte o linie în portul de ieşire $O_3O_2O_1O_0$ plasând succesiunea de coduri: 0111, 1011, 1101, 1110. La aplicarea codului 0111 se selectează primul rând de taste (tastele 1, 2, 3). Apăsarea uneia sau mai multor taste din acest grup este identificată de către circuitul de comandă prin scanarea portului de intrare $I_2I_1I_0$. De exemplu, dacă a fost apăsată tasta 1, $I_2I_1I_0 = 011$. Prin modificarea codului de ieşire $O_3O_2O_1O_0$, se modifică rândul de taste care se scanează la intrarea $I_2I_1I_0$. Tasta 1 se află la intersecţia rândului selectat de O_3 cu coloana scanată pe intrarea I_2.

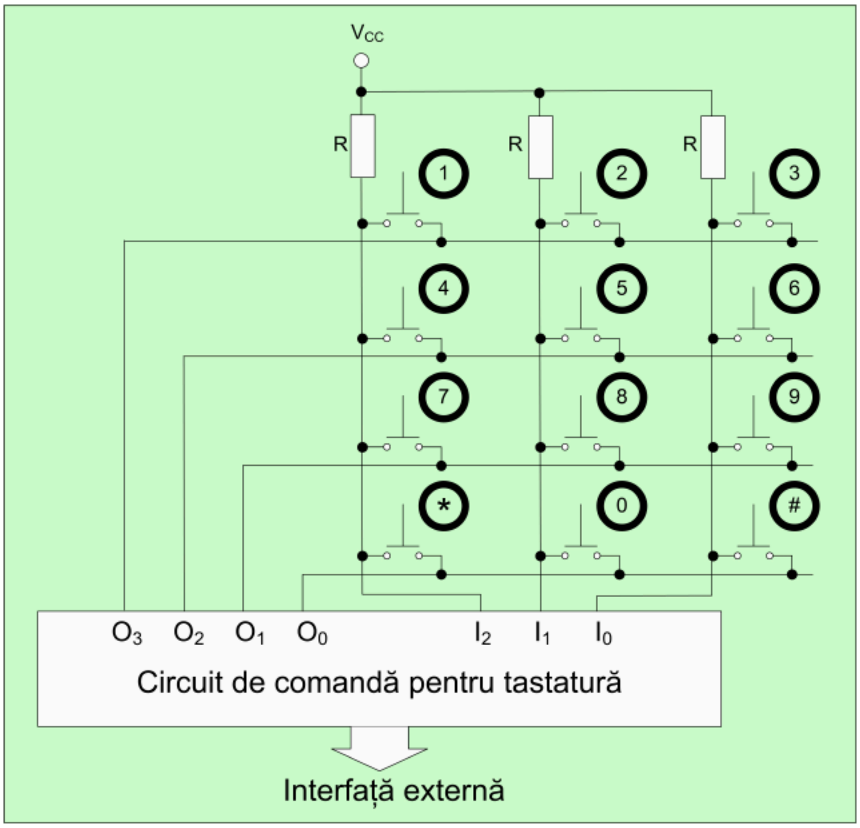


Figura 1.2 Tastatură de telefon referită la problema 4.

Codul tastei 1 este $O_3O_2O_1O_0I_2I_1I_0 = 0111\_011$.
Codul tastei * este $O_3O_2O_1O_0I_2I_1I_0 = 1110\_011$.
Codul tastei 5 este $O_3O_2O_1O_0I_2I_1I_0 = 1011\_101$.
Codul tastei 7 este $O_3O_2O_1O_0I_2I_1I_0 = 1101\_011$.
Codul tastei 6 este $O_3O_2O_1O_0I_2I_1I_0 = 1011\_110$.

5. Algoritmul uzual de împărţire se bazează pe înmulţiri şi scăderi repetate. Realizaţi următoarele împărţiri în binar (obţineţi câtul şi restul). Verificaţi operaţia ”deîmpărţit = împărţitor × cât + rest” cu numerele convertite în zecimal.
   a) 101_1110 : 101 **b)** 1010_1110 : 1010 **c)** 1011_0111 : 110 **d)** 1011_1110 : 1011

   Soluţie

   a) Algoritmul de împărţire este:

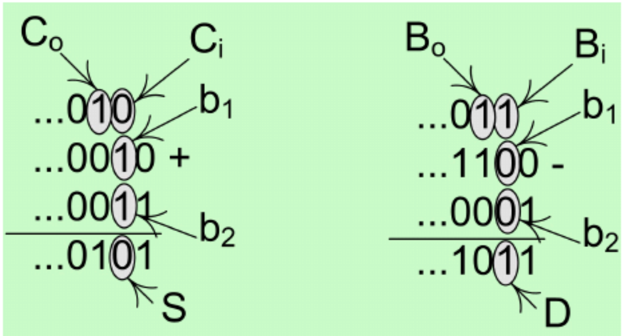
```
Deimpartit -> 1011110 | 101   <- Impartitor
              101     | 10010 <- Cat
              ===
              ---1
                 0
                 =
                 11
                 00
                 ===
                 111
                 101
                 ===
                 -100 <- Rest
```

Rezultatul este: Cât = $10010|_2$, Rest = $100|_2$.
Verificare: $101\_1110|_2 = 101|_2 \times 1\_0010|_2 + 100|_2$, $5E|_{16} = 5|_{16} \times 12|_{16} + 4|_{16}$, $94|_{10} = 5 \times 18|_{10} + 4|_{10}$, adevărat.

6. Realizaţi suma şi diferenţa dintre cuplurile de numere propuse, în complement faţă de 2: (00110010, 11110011), (10101100, 00010001), (0001111, 00011101), (11001100, 00011111). Verificaţi răspunsurile prin conversia numerelor şi efectuarea operaţiilor în baza 10.

Soluţie

Operaţile de adunare şi scădere se pot face direct asupra reprezentărilor binare.
În cazul adunării, există relaţia: $\{C_o, S\} = b_1 + b_2 + C_i$. S-a notat C_i transportul de intrare (Engl. *"Carry"*), C_o transportul de ieşire/generat de operaţia pe bit pentru bitul de ordin superior şi S suma biţilor.
În cazul scăderii, există relaţia: $\{B_o, D\} = b_1 - b_2 - B_i$. S-a notat B_i împrumutul de intrare (Engl. *"Borrow"*), B_o împrumutul de ieşire/preluat de operaţia pe bit de la bitul de ordin superior şi D diferenţa biţilor.
Aceste operaţii sunt exemplificate în figura 1.3.



Adunare

$\{C_o, S\} = b_1 + b_2 + C_i$

b_1	b_2	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Scădere

$\{B_o, D\} = b_1 - b_2 - B_i$

b_1	b_2	B_i	B_o	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Figura 1.3 Exemplificarea formulelor de adunare ($\{C_o, S\} = b_1 + b_2 + C_i$) şi scădere ($\{B_o, D\} = b_1 - b_2 - B_i$) a biţilor.

```
1110 010   <- transport          1111 111   <- imprumut
0011_0010 +                      0011_0010 -                    50 +           50 -
1111_0011                        1111_0011                     -13            -13
---------                        ---------                     ---            ---
0010_0101                        0011 1111                      37             63


0000 000   <- transport          0010 011   <- imprumut
1010_1100 +                      1010_1100 -                   -84+           -84-
0001_0001                        0001_0001                      17             17
---------                        ---------                     ---            ----
1011_1101                        1001_1011                     -67           -101
```