

A. Reprezentarea datelor

A.1. Reprezentarea datelor în sistemul de numerație binar

Notația binară este o metodă de reprezentare a valorilor numerice folosindu-se numai cifrele zero și unu.

Într-o reprezentare binară a unei valori numerice, cifra cea mai din dreapta are asociată ponderea 1 (2^0), următoarea are ponderea 2 (2^1), a treia are pondere 4 (2^2) și a patra are ponderea $2^3=8$, etc.

De exemplu:

$$1011=1*2^3+0*2^2+1*2^1+1*2^0$$

Pentru trecerea unui număr întreg pozitiv din sistemul de numerație zecimal în cel binar se folosește următorul algoritm:

1. se împarte valoarea la 2 și se memorează restul împărțirii;
2. atât timp cât rezultatul împărțirii este diferit de zero, se continuă împărțirea lui la 2 memorându-se restul;
3. când s-a obținut rezultatul egal cu zero, reprezentarea în binar a valorii inițiale constă din resturile împărțirilor afișate în ordinea inversă obținerii.

Exemple:

$$\begin{array}{r}
 172 \overline{) 2} \\
 \underline{16} \quad 86 \overline{) 2} \\
 \underline{12} \quad 8 \quad 43 \overline{) 2} \\
 \underline{12} \quad 6 \quad 4 \quad 21 \overline{) 2} \\
 \underline{\quad 0} \quad 6 \quad 3 \quad 2 \quad 10 \overline{) 2} \\
 \quad \underline{\quad 0} \quad 2 \quad 1 \quad 10 \quad 5 \overline{) 2} \\
 \quad \quad \underline{\quad 1} \quad \quad 0 \quad 4 \quad 2 \overline{) 2} \\
 \quad \quad \quad \underline{\quad 1} \quad 2 \quad 1 \\
 \quad \quad \quad \quad \underline{\quad 0}
 \end{array}$$

$$(172)_{10}=(10101100)_2$$

$$\begin{array}{r}
211 \overline{) 2} \\
\underline{210} \quad 105 \overline{) 2} \\
1 \quad 104 \quad 52 \overline{) 2} \\
\underline{1} \quad 52 \quad 26 \overline{) 2} \\
0 \quad 26 \quad 13 \overline{) 2} \\
0 \quad 12 \quad 6 \overline{) 2} \\
1 \quad 6 \quad 3 \overline{) 2} \\
0 \quad 2 \quad 1 \\
\underline{1}
\end{array}$$

$$(211)_{10} = (1101\ 0011)_2$$

A.2. Reprezentarea numerelor în sistemul de numerație hexazecimal

Este o metodă de codificare a valorilor numerice folosind 16 simboluri:

- cifrele 0,1,2,3,4,5,6,7,8,9 având aceeași semnificație ca la sistemul zecimal;
- simbolurile A,B,C,D,E,F echivalente cu valorile 10, 11, 12, 13, 14, 15 din sistemul de numerație zecimal.

Mai jos sunt prezentate reprezentările în hexazecimal și binar a numerelor de la 1 la 16 din sistemul de numerație zecimal:

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

Similar cu notația binară, în reprezentarea hexazecimală un număr este reprezentat tot de o sumă de puteri, de această dată de puteri ale lui 16.

Exemplu:

$$\begin{aligned}
3b0f &= 3 \times 16^3 + b \times 16^2 + 0 \times 16^1 + f \times 16^0 \\
&= 3 \times 4096 + 11 \times 256 + 0 \times 16 + 15 \times 1 \\
&= 12288 + 2816 + 0 + 15
\end{aligned}$$

$$= 15119$$

Pentru reprezentarea în hexazecimal a unui număr întreg zecimal se respectă același algoritm cu deosebirea că împărțirea se face la 16, nu la 2. Pentru conversiile din hexazecimal în binar se efectuează prin gruparea a câte patru cifre binare, începând din dreapta numărului și înlocuirea lor cu echivalentul hexazecimal corespunzător.

Exemplu de conversie din binar în hexazecimal, respectiv hexazecimal în binar:

$$\begin{array}{cccc}
 (1011 & | & 1010 & | & 0011 & | & 0010)_2 \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 (B & & A & & 3 & & 2)_{16}
 \end{array}
 \quad \text{respectiv} \quad
 \begin{array}{cccc}
 (4 & F & C & 2)_{16} \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 (0100 & 1111 & 1100 & 0010)_2
 \end{array}$$

Exemplu de conversie din zecimal în hexazecimal:

$$\begin{array}{r}
 2359 \overline{)16} \\
 \underline{16} \\
 75 \\
 \underline{64} \\
 119 \\
 \underline{112} \\
 7
 \end{array}$$

$$(2359)_{10} = (937)_{16}$$

În interiorul mașinilor numerice, numerele sunt reprezentate numai prin grupe de cifre binare. Aceste grupe de cifre binare sunt numite **cuvinte**. Sunt tipice **cuvinte** cu o lungime de 4, 8, 16 sau 32 de biți. **Un cuvânt de 8 biți** se mai numește și **octet** sau, în terminologia tehnică consacrată, **byte**. Cuvintele sunt memorate în registrele microprocesoarelor sau în memoria sistemului de calcul. Un registru poate fi implementat cu un șir de circuite bistabile, fiecare dintre acestea putând fi considerat un bit de memorie. Din punct de vedere funcțional, o mașină numerică poate fi privită ca un sistem format din registre, iar datele sunt transferate dintr-un registru în altul în cursul efectuării operațiilor aritmetice și logice. **Notația hexazecimală oferă doar o modalitate facilă de interpretare a informațiilor binare vehiculate într-un calculator.**

Cuvintele de date de 8 biți vor putea conține valori numerice pozitive între 0 și 255 ($2^8 - 1$). Cuvintele de date pe 16 biți vor putea conține valori între 0 și 65535 ($2^{16} - 1$).

B. Operații cu numere în reprezentare binară

B.1. Operații logice cu numerele în reprezentare binară

Logica binară este cea mai simplă și cea mai comună formă de structurare a operațiilor logice. Operațiile pur binare permit combinarea în mod predeterminat a unor numere binare. Astfel, operația binară AND aplicată asupra a două cuvinte de date produce ca rezultat tot un cuvânt, în care fiecare bit este evaluat ca fiind 1 dacă ambii biți corespunzători ai celor doi operanzi sunt 1, sau 0 în celelalte cazuri. Operațiile logice binare cele mai folosite sunt AND, OR, XOR.

Exemplu:

$$\begin{array}{r} 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ \text{AND} \\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1 \\ \hline 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1 \end{array}$$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ \text{OR} \\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1 \end{array}$$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ \text{XOR} \\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1 \\ \hline 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0 \end{array}$$

Operația de mascare binară este cel mai frecvent caz în care se folosesc operațiile logice simple între două cuvinte de date. Mascarea binară este folosită pentru a afla informații specifice din cadrul unei structuri mai mari de date.

Într-o operație de mascare binară unul dintre operanzi este o variabilă iar celălalt este o valoare predeterminată, folosită ca "sită" pentru a "cerne" biții primului operand. Acest al doilea operand este denumit convențional **mască**.

Această denumire provine de la faptul că, așa cum se vede în exemplul de mai jos, biții cu valoare nulă din mască corespund în mod predeterminat cu valori nule în rezultat, în timp ce biții cu valori nenule ai măștii corespund în rezultat cu valoarea originală a celuilalt operand. Astfel, octetul denumit convențional "mască" are într-adevăr funcția de a masca în rezultat anumiți biți din operand.

Exemplu:

$$\begin{array}{r} \text{Variabilă} \rightarrow 1\ 0\ 1\ 1\ \mathbf{0}\ 1\ 1\ 1\ \text{AND} \\ \text{Mască} \rightarrow 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ \hline \text{Rezultat} \rightarrow 0\ 0\ 0\ 0\ \mathbf{0}\ 1\ 1\ 1 \end{array}$$

B.2. Adunarea în binar

Algoritm:

- se aliniaza numerele de la dreapta la stanga incepand cu bitul cel mai puțin semnificativ. Pentru fiecare rang binar se realizează operația de adunare după regula:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0*

*cu transport 1 la rangul următor

- Se adună pentru fiecare rang binar valorile biților de la dreapta la stanga, adunându-se și eventualul transport de la rangul inferior.

Exemple:

$$\begin{array}{r}
 1\ 0\ 0\ 0\ + \\
 0\ 1\ 1\ 0 \\
 \hline
 1\ 1\ 1\ 0
 \end{array}
 \qquad
 \begin{array}{r}
 8\ + \\
 6 \\
 \hline
 14
 \end{array}$$

$$\begin{array}{r}
 \leftarrow\ \leftarrow\ \leftarrow\ \leftarrow\ \leftarrow\ \leftarrow \\
 1\ 0\ 1\ 1\ 0\ 1\ + \\
 0\ 1\ 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1\ 0\ 0\ 0
 \end{array}
 \qquad
 \begin{array}{r}
 45\ + \\
 27 \\
 \hline
 72
 \end{array}$$

\leftarrow : transport la rangul următor

B.3. Scăderea în binar

Algoritm:

- se aliniaza de la dreapta la stanga numerele incepand cu cel mai puțin semnificativ bit.
- Pe fiecare rang binar se realizează operația de scădere după regula:

A	B	Y
0	0	0
0	1	1*
1	0	1
1	1	0

*cu împrumut 1 de la rangul superior

- se realizează secvența de la dreapta la stânga bit cu bit luând în calcul eventualul împrumut către rangul inferior.

Exemplu:

$$\begin{array}{r}
 \leftarrow \qquad \qquad \leftarrow \\
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ - \quad 45 \ - \\
 0 \ 1 \ 1 \ 0 \ 1 \ 1 \quad \underline{27} \\
 \hline
 0 \ 1 \ 0 \ 0 \ 1 \ 0 \quad 18
 \end{array}$$

←: împrumut de la rangul superior

În cazul operațiilor de adunare sau scădere , trebuie remarcată problema apariției depășirilor (overflow), sau a împrumuturilor (borrow), dacă suma sau diferența nu este reprezentabilă într-un cuvânt de date (de exemplu, suma a două numere reprezentate pe 8 biți, care adunate dau un rezultat peste 256, va necesita 9 biți pentru reprezentare corectă). Dată fiind lungimea finită a cuvintelor de date în microprocesor, operația de adunare sau de scădere va trebui să semnalizeze suplimentar efectuării calculului, apariția acestui bit de depășire; altfel, dacă rezultatul este considerat pe 8 biți, se pierde chiar bitul cel mai semnificativ, alterându-se rezultatele și obținându-se cu totul alte valori decât cele corecte. De exemplu, o asemenea situație apare în cazul următor:

$$\begin{array}{r}
 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\
 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ + \\
 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \\
 \hline
 1 \ | \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \\
 \uparrow \\
 \text{Depășire}
 \end{array}$$

O situație similară poate apărea în cazul scăderii, dacă descăzutul este mai mare decât scăzătorul (apare un bit de împrumut).

B.4. Înmulțirea în binar

Algoritm:

- se aliniază numerele unul sub altul de la dreapta la stânga, începând cu cel mai puțin semnificativ bit;
- pentru fiecare bit 1 de la înmulțitor se scrie sub acesta primul operand începând din dreptul locației respective de la dreapta spre stânga;
- se efectuează operația de adunare prezentată anterior asupra operanzilor astfel rezultați.

Exemple:

$$\begin{array}{r}
 1\ 1\ 0\ 1\ X \\
 0\ 1\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 1 \\
 \hline
 1\ 0\ 0\ 0\ 0\ 0\ 1
 \end{array}
 \qquad
 \begin{array}{r}
 13\ X \\
 5 \\
 \hline
 65
 \end{array}$$

$$\begin{array}{r}
 0\ 1\ 0\ 1\ x \\
 1\ 1\ 0\ 1 \\
 \hline
 0\ 1\ 0\ 1 \\
 \hline
 0\ 1\ 0\ 1 \\
 \hline
 0\ 1\ 0\ 1 \\
 \hline
 1\ 0\ 0\ 0\ 0\ 0\ 1
 \end{array}
 \qquad
 \begin{array}{r}
 5\ x \\
 13 \\
 \hline
 65
 \end{array}$$

În ceea ce privește problema depășirii la operația de înmulțire, trebuie remarcat că rezultatul înmulțirii a două numere reprezentate pe câte un cuvânt de date se va reprezenta pe două cuvinte de date. Înmulțirea unui număr reprezentat pe m biți cu un număr reprezentat pe n biți (numere întregi, fără bit de semn), conduce la obținerea unui rezultat reprezentat pe (m+n) biți.

Toate microprocesoarele au implementate în setul de instrucțiuni operațiile de adunare și de scădere. Doar unele microprocesoare au în setul de instrucțiuni operațiile de înmulțire. În cazul absenței acestora din setul de instrucțiuni al microprocesorului respectiv, implementarea lor se face prin program, pe baza operațiilor elementare de adunare, a operațiilor logice și a deplasărilor de biți în interiorul cuvintelor.

Exerciții 1:

1. Converteți fiecare dintre următoarele reprezentări binare în formă zecimală echivalentă :

a. 10011 b. 1001100 c. 1110101 d. 110110 e. 100001
f. 10010101 g. 10110011 h. 111011 i. 10011 j. 10011001

2. Converteți fiecare dintre următoarele reprezentări zecimale în formă binară echivalentă:

a. 6 b. 13 c. 168 d. 192 e. 227 f. 424

3. Utilizați notația hexazecimală pentru a reprezenta următoarele șiruri de biți :

a. 0110.1010.1111.0010 b. 1110.1000.0101.0101.0001.0111
c. 0100.1000 d. 1110.1001.1101.0100.0110 e. 1010.1111.0010
f. 1001.1011.0011 g. 1001.1101.0011.1110 h. 1001.0011.0111.1101

4. Care sunt șirurile de biți reprezentate de următoarele numere hexazecimale ?

a. FD97 b. 610A c. ABCD d. 0100 e. 5C01 f. 10AA
g. 1234 h. 3bF8 i. EF00 j. EE10 k. 45C9 l. 12CD

5. Care valori au reprezentări binare ce conțin un singur bit cu valoarea 1? Scrieți reprezentările binare ale celor mai mici șase valori care au această proprietate.

6. Să se precizeze care din următoarele șiruri de biți sunt reprezentări binare ale unor numere zecimale impare:

a. 10011001 b. 101110010 c. 11011001 d. 10011010110110
e. 010110110011 f. 101010101000 g. 1010101111001101 h. 100010111

Ce mască și ce operație logică ar trebui să folosim pentru a afla dacă aceste numere sunt pare sau impare.

7. Efectuați operațiile următoare:

a: 11001011 b: 10000011 c: 11111111
 AND 10101011 AND 11101100 AND 00101101

d: 01001011 e: 10000011 f: 11111111
 OR 10101011 OR 11101100 OR 00101101

g: 01001011 h: 10000011 i: 11111111
 XOR 10101011 XOR 11101100 XOR 00101101

8. Să presupunem că vreți să izolați cei mai semnificativi trei biți dintr-un șir de șapte biți plasând 0 pe pozițiile celorlalți patru biți, fără a afecta valorile celor trei biți specificați. Ce mască și ce operație va trebui să utilizați pentru a face acest lucru? Dați exemplu.

9. Să presupunem că vrei să negați cei trei biți din centrul unui șir de șapte biți lăsând ceilalți patru biți nemodificați. Ce mască și ce operație va trebui să utilizați pentru aceasta?

10. Presupunem că realizați un XOR între primii doi biți dintr-un șir de biți și apoi continuați parcurgerea șirului efectuând succesiv XOR între rezultatul operației precedente și următorul bit din șir. Ce legătură este între rezultatul final și numărul de apariții ale valorii 1 în șirul de biți?

11. Adesea este preferabil să se utilizeze o operație logică în locul uneia aritmetice. De exemplu, operația logică AND combină doi biți în aceeași manieră ca și înmulțirea. Care operație logică este aproape similară adunării a doi biți și care este diferența?

12. Să se efectueze următoarele operații cu numere binare, specificând cazurile în care apare depășire:

a. $1011101_2 + 0011010_2$ b. $11010001_2 + 01010_2$ c. $11011001_2 + 11110_2$ d. $11001001_2 + 11111101_2$

e. $1011001_2 + 1110110_2$ f. $1011101_2 + 100101_2$ g. $11100110_2 + 00011101_2$ h. $11001101_2 + 11110010_2$

13. Să se efectueze următoarele operații de scădere a numerelor binare, specificând cazurile în care apare depășire:

a. $1111001_2 - 1010010_2$ b. $1010001_2 - 1111010_2$ c. $10001001_2 - 1110110_2$ d. $11001101_2 - 10011101_2$

e. $1011001_2 - 110110_2$ f. $101101_2 - 100101_2$ g. $1100110_2 - 11101_2$ h. $11001101_2 - 11110010_2$

14. Să se efectueze operațiile de înmulțire de mai jos:

a. $11011_2 * 1100_2$ b. $11001_2 * 10110_2$ c. $110011_2 * 10110_2$ d. $11001_2 * 1010_2$ e. $10011_2 * 10110_2$ f. $1001101_2 * 101101_2$

15. Determinați operația logică și masca respectivă care, atunci când sunt aplicate asupra unui octet, produc ca rezultat un octet cu toți biții 0, dacă și numai dacă octetul de intrare este 10000001.

16. Descrieți o secvență de operații logice (prezentând și măștile corespunzătoare operațiilor) care, atunci când sunt aplicate unui șir de intrare de opt biți, produc la ieșire un octet cu toți biții 0 dacă șirul aplicat la intrare începe și se termină cu 1. Altfel, octetul de la ieșire trebuie să conțină cel puțin un bit cu valoarea 1.

Exerciții 2:

1. Converteți fiecare dintre următoarele reprezentări binare în formă zecimală echivalentă :

- a. 01011 b. 100001 c. 101011101 d. 1010110
e. 100101 f. 1010101 g. 1010011 h. 1011011
i. 100111 j. 10010011

2. Converteți fiecare dintre următoarele reprezentări zecimale în formă binară echivalentă:

- a. 9 b. 16 c. 148 d. 124 e. 277 f. 542

3. Utilizați notația hexazecimală pentru a reprezenta următoarele șiruri de biți :

- a. 0110.1010.1001.1010 b. 1010.1100.1101.0101.0101.1111
c. 0101.1010 d. 1010.1011.1101.0110.0111 e. 1010.1101.1010
f. 1001.1001.0111 g. 1011.1111.1011.1100h. 0001.0011.0111.1001

4. Care sunt șirurile de biți reprezentate de următoarele numere hexazecimale ?

- a. 1997 b. 61AA c. DCDA d. 0100 e. 5C01 f. 10DA
g. 1234 h. 3bF8 i. 0100 j. FE10 k. F5A9 l. 12AD

5. Care valori au reprezentări binare ce conțin un singur bit cu valoarea 1? Scrieți reprezentările binare ale celor mai mici șase valori care au această proprietate.

6. Să se precizeze care din următoarele șiruri de biți sunt reprezentări binare ale unor numere zecimale impare:

- a. 10011000 b. 101110011 c. 11011000 d. 10011010110111
e. 010110110011 f. 101010101000 g. 1010101111001100 h. 100010110

Ce mască și ce operație logică ar trebui să folosim pentru a afla dacă aceste numere sunt pare sau impare.

7. Efectuați operațiile următoare:

- a: 01111110 b: 10111011 c: 11111111
AND 10001011 AND 11101001 AND 00100101

- d: 01101011 e: 10110111 f: 11101111
OR 10101000 OR 11101100 OR 10111101

- g: 01110101 h: 11100011 i: 11101011
XOR 10111011 XOR 10101100 XOR 00001111

8. Să presupunem că vreți să izolați cei mai puțin semnificativi trei biți din centrul unui șir de șapte biți plasând 0 pe pozițiile celorlalți patru biți, fără a afecta valorile

celor trei biți specificați. Ce mască și ce operație va trebui să utilizați pentru a face acest lucru? Dați exemplu.

9. Să presupunem că vreți să negați cei mai puțin semnificativi 4 biți dintr-un șir de 8 biți lăsând ceilalți patru biți nemodificați. Ce mască și ce operație va trebui să utilizați pentru aceasta? Exemplificați.

10. Presupunem că realizați un XOR între primii doi biți dintr-un șir de biți și apoi continuați parcurgerea șirului efectuând succesiv XOR între rezultatul operației precedente și următorul bit din șir. Ce legătură este între rezultatul final și numărul de apariții ale valorii 1 în șirul de biți?

11. Adesea este preferabil să se utilizeze o operație logică în locul uneia aritmetice. De exemplu, operația logică AND combină doi biți în aceeași manieră ca și înmulțirea. Care operație logică este aproape similară adunării a doi biți și care este diferența?

12. Să se efectueze următoarele operații cu numere binare specificând cazurile în care apare depășire:

a. $1011101_+ + 1011010$ b. $10010001_+ + 1001010$ c. $11010001_+ + 10110$ d. $101101001_+ + 11001101$

e. $1000011_+ + 110110$ f. $1000101_+ + 100101$ g. $11100110_+ + 10101$ h. $11001101_+ + 1000110$

13. Să se efectueze următoarele operații de scădere a numerelor binare specificând cazurile în care apare depășire:

a. $1101001_- - 1110010$ b. $1010001_- - 101010$ c. $10001001_- - 110110$ d. $11001101_- - 10000101$

e. $1011101_- - 100110$ f. $101101_- - 101101$ g. $1100110_- - 10101$ h. $11001101_- - 1000010$

14. Să se efectueze operațiile de înmulțire de mai jos:

a. $11011_* \cdot 11100$ b. $110101_* \cdot 10111$ c. $110011_* \cdot 11110$ d. $111001_* \cdot 111010$ e. $10101_* \cdot 11010$ f. $1000001_* \cdot 101001$

15. Determinați operația logică și masca respectivă care, atunci când sunt aplicate asupra unui octet, produc ca rezultat un octet cu toți biții 0, dacă și numai dacă octetul de intrare este 10000001.

16. Descrieți o secvență de operații logice (prezentând și măștile corespunzătoare operațiilor) care, atunci când sunt aplicate unui șir de intrare de opt biți, produc la ieșire un octet cu toți biții 0 dacă șirul aplicat la intrare începe și se termină cu 1. Altfel, octetul de la ieșire trebuie să conțină cel puțin un bit cu valoarea 1.

Exerciții 3:

1. Converteți fiecare dintre următoarele reprezentări binare în formă zecimală echivalentă :

- a. 11011 b. 1101100 c. 10100101 d. 100110 e. 1010001
f. 10011101 g. 10100011 h. 101101 i. 10111 j. 10110001

2. Converteți fiecare dintre următoarele reprezentări zecimale în formă binară echivalentă:

- a. 4 b. 17 c. 187 d. 121 e. 219 f. 329

3. Utilizați notația hexazecimală pentru a reprezenta următoarele șiruri de biți :

- a. 0110.0010.0111.1010 b. 1111.0000.0111.0111.0101.0011
c. 0110.1001 d. 1100.1001.1101.0100.0100 e. 1010.1011.0110
f. 1011.1001.0011 g. 1011.1111.0011.0110 h. 1001.1011.0111.1001

4. Care sunt șirurile de biți reprezentate de următoarele numere hexazecimale ?

- a. 97FD b. 10AD c. ACDE d. 0100 e. FC01 f. 1A0A
g. 12B0 h. 3bF8 i. 6C60 j. BE10 k. 65C9 l. 128D

5. Care valori au reprezentări binare ce conțin un singur bit cu valoarea 1? Scrieți reprezentările binare ale celor mai mici șase valori care au această proprietate.

6. Să se precizeze care din următoarele șiruri de biți sunt reprezentări binare ale unor numere zecimale impare:

- a. 10001001 b. 100110011 c. 11011100 d. 10001010110111
e. 010110110010 f. 101010101001 g. 1010101111001100 h. 101010110

Ce mască și ce operație logică ar trebui să folosim pentru a afla dacă aceste numere sunt pare sau impare.

7. Efectuați operațiile următoare:

- a: 01010001 b: 11001011 c: 11101111
AND 10101001 AND 11101110 AND 10001001

- d: 01001101 e: 10011011 f: 10010111
OR 10100111 OR 10001100 OR 11111101

- g: 01011010 h: 11010011 i: 11111011
XOR 10001011 XOR 11000100 XOR 10010010

8. Să presupunem că vreți să izolați cei trei biți din centrul unui șir de șapte biți plasând 0 pe pozițiile celorlalți patru biți, fără a afecta valorile celor trei biți de la mijlocul șirului. Ce mască și ce operație va trebui să utilizați pentru a face acest lucru? Exemplificați.

9. Să presupunem că vreți să negați cei mai puțin semnificativi trei biți dintr-un șir de șapte biți lăsând ceilalți patru biți nemodificați. Ce mască și ce operație va trebui să utilizați pentru aceasta?

10. Presupunem că realizați un XOR între primii doi biți dintr-un șir de biți și apoi continuați parcurgerea șirului efectuând succesiv XOR între rezultatul operației precedente și următorul bit din șir. Ce legătură este între rezultatul final și numărul de apariții ale valorii 1 în șirul de biți?

11. Adesea este preferabil să se utilizeze o operație logică în locul uneia aritmetice. De exemplu, operația logică AND combină doi biți în aceeași manieră ca și înmulțirea. Care operație logică este aproape similară adunării a doi biți și care este diferența?

12. Să se efectueze următoarele operații cu numere binare specificând cazurile în care apare depășire:

a. $1010001+$ b. $11010001+$ c. $11011001+$ d. $11011001+$
 1011110 101010 10110 11101101

e. $1000001+$ f. $1011101+$ g. $11001110+$ h. $11000101+$
 110110 1100101 10101 1000010

13. Să se efectueze următoarele operații de scădere a numerelor binare specificând cazurile în care apare depășire:

a. $1111001-$ b. $1011111-$ c. $11101001-$ d. $11001101-$
 1011010 1101010 10110 10011101

e. $1011101-$ f. $101101-$ g. $1100110-$ h. $11101101-$
 110110 110101 101101 1110110

14. Să se efectueze operațiile de înmulțire de mai jos:

a. $11011*$ b. $11101*$ c. $111011*$ d. $11001*$ e. $10001*$ f. $1101101*$
 10111 10010 11110 11010 10110 1011010

15. Determinați operația logică și masca respectivă care, atunci când sunt aplicate asupra unui octet, produc ca rezultat un octet cu toți biții 0, dacă și numai dacă octetul de intrare este 10000001.

16. Descrieți o secvență de operații logice (prezentând și măștile corespunzătoare operațiilor) care, atunci când sunt aplicate unui șir de intrare de opt biți, produc la ieșire un octet cu toți biții 0 dacă șirul aplicat la intrare începe și se termină cu 1. Altfel, octetul de la ieșire trebuie să conțină cel puțin un bit cu valoarea 1.