UNIVERSITATEA DE STAT DIN MOLDOVA

FACULTATEA MATEMATICĂ ŞI INFORMATICĂ
DEPARTAMENTUL INFORMATICĂ

CEMÎRTAN CRISTIAN

Lucrarea individuală nr. 1

la disciplina ***Arhitectura Calculatoarelor și Limbaje de Asamblare***

Coordonator: Sturza Greta, lector universitar

Chişinău, 2021

Cuprins

Interpretarea numerelor în diferite baze numerice.....3
Sarcina 1.....4
Sarcina 2.....5
Sarcina 3.....6
Sarcina 4.....7
Sarcina 5.....8
Concluzie.....9
Bibliografie.....10

Interpretarea numerelor în diferite baze numerice

Zecimal	Hexazecimal	Binar
0	00h	00000000b
1	01h	00000001b
2	02h	00000010b
3	03h	00000011b
4	04h	00000100b
5	05h	00000101b
6	06h	00000110b
7	07h	00000111b
8	08h	00001000b
9	09h	00001001b
10	0Ah	00001010b
11	0Bh	00001011b
12	0Ch	00001100b
13	0Dh	00001101b
14	0Eh	00001110b
15	0Fh	00001111b

Figura 1.

Sarcina 1

Sarcină: Transformați în sistemele binar și hexazecimal numerele zecimale: ***49; 30; 103***.

Rezolvare:

a. În hexazecimal:

1. 49 / 16 = 3 rest 1
3 / 16 = 0 rest 3
Rezultat: 31h

2. 30 / 16 = 1 rest 14 (0Eh în hexazecimal)
1 / 16 = 0 rest 1
Rezultat: 1Eh

3. 103 / 16 = 6 rest 7
6 / 16 = 0 rest 6
Rezultat: 67h

b. În binar:

- Descompunem cifrele hexazecimale în 4 biți.

1. 31h = 0011-0001b
Rezultat: 00110001b

2. 1Eh = 0001-1110b
Rezultat: 00011110b

3. 67h = 0110-0111b
Rezultat: 01100111b

Sarcina 2

Sarcină: Transformați în sistemele binar numerele hexazecimale: ***4Ah; 6ABh; 9ADh***.

Rezolvare:

1. 4Ah = 0100-1010b
   Rezultat: 01001010b

2. 6ABh = 0110-1010-1011b
   Rezultat: 011010101011b

3. 9ADh = 1001-1010-1101b
   Rezultat: 100110101101b

Sarcina 3

Sarcină: Transformați în sistemele zecimal și hexazecimal numerele binare: ***10000101b; 11100010b; 11001011b***.

Rezolvare:

a. În hexazecimal:

1. 1000-0101b = 85h
Rezultat: 85h

2. 1110-0010b = 0E2h
Rezultat: 0E2h

3. 1100-1011b = 0CBh
Rezultat: 0CBh

b. În zecimal:

1. $8 * 16^1 + 5 * 16^0 = 8 * 16 + 5 = 128 + 5 = 133$
Rezultat: 133

2. $14 * 16^1 + 2 * 16^0 = 14 * 16 + 2 = 224 + 2 = 226$
Rezultat: 226

3. $12 * 16^1 + 11 * 16^0 = 12 * 16 + 11 = 192 + 11 = 203$
Rezultat: 203

Sarcina 4

Sarcină: Reprezentați în cod complementar pe 8 poziții binare următoarele numere: ***-49; -30; -103***.

Rezolvare:

a. Transformare în sistem binar:

- Ducem numerele negative în pozitive, și apoi le inversăm bit-urile în următoarea etapă.

1. 49 / 16 = 3 rest 1
3 / 16 = 0 rest 3
31h = 0011-0001b

2. 30 / 16 = 1 rest 14
1 / 16 = 0 rest 1
1Eh = 0001-1110b

3. 103 / 16 = 6 rest 7
6 / 16 = 0 rest 6
67h = 0110-0111b

b. Reprezentare în cod complementar:

1. În cod invers: **11001110b**
În cod complementar: **11001111b**

2. În cod invers: **11100001b**
În cod complementar: **11100010b**

3. În cod invers: **10011000b**
În cod complementar: **10011001b**

Sarcina 5

Sarcină: Fie dată următoarea consecutivitate de definiții de date:

- **a** db "Salut", 10, 13
  b db -16, -20, 13h, 2 dup(0)
  c dw 62, 34, -15
  d dd 456C9h, 4567

a) determinați câți octeți va rezerva această consecutivitate de definiții de date;
b) determinați adresa relativă a fiecărei locații de memorie.

Rezolvare:

Caracterele ASCII [1]

0x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	53h	41h	4Ch	55h	54h	0Ah	0Dh	0F0h	0ECh	13h	00h	00h	3Eh	00h	22h	00h
001	0F1h	0FFh	0C9h	56h	04h	00h	0D7h	11h	00h	00h	Total rezervat: 26 octeți					

Figura 2. Reprezentarea segmentului de date pe memorie.

a)

a. "Salut" – 5 octeți;
10 – 1 octet;
13 – 1 octet;
Total – 7 octeți,

b. -16 – 1 octet;
-20 – 1 octet;
13h – 1 octet;
2 dup(0) – 2 octeți;
Total – 5 octeți.

c. 62 – 2 octeți;
34 – 2 octeți;
-15 – 2 octeți;
Total – 6 octeți.

d. 456C9h – 4 octeți;
4567 – 4 octeți;
Total – 8 octeți.

b)

a. 0x0000
b. 0x0007

c. 0x000C
d. 0x0012

Concluzie

În concluzie, pot spune că sistemele de numerație binare și hexazecimale reprezintă nativ modul de stocare a numerelor în calculator. Sistemul hexazecimal, fiind o interpretare compactă a sistemului binar, permite la o conversie eficientă (datorită alfabetului său extins) de a transforma numerele zecimale în sistemul binar, aplicând divizarea întreagă [2]. De exemplu, pentru numărul zecimal 1457, se necesită 3 pași de a fi transformat în sistemul hexazecimal, și apoi de divizat cifrele sale în perechi de 4 biți, transformându-l în sistemul binar. Dar, transformarea simplă (împărțire cu 2) în sistemul binar, necesită un număr mai mare de pași, care este 11.

Reprezentarea numerelor mai largi decât un octet, în memoria calculatorului, depinde de arhitectura microprocesorului. De exemplu, în arhitectura x86, octeții sunt memorați în ordine inversă. Acest tip de aranjare a secvențelor de octeți, se numește little endian [3], demonstrat în figura 2, la rezolvarea sarcinii 5.

Bibliografie

1. Tabel ASCII, https://moodle.usm.md/pluginfile.php/293614/mod_resource/content/1/Tabela%20codurilor%20ASCII.pdf.

2. R. Goncearenco, G. Sturza, M. Butnaru, Gh. Latul, Programare în limbaj de asamblare, Reprezentări interne ale datelor, Definirea și inițializarea datelor în limbaj de asamblare, Chișinău, https://moodle.usm.md/pluginfile.php/127735/mod_resource/content/7/Reprezentari%20interne%20ale%20datelor.pdf

3. Chris Lomont, Introduction to x64 Assembly, Intel, 2012, https://software.intel.com/content/www/us/en/develop/articles/introduction-to-x64-assembly.html.