1. ПРЕДСТАВЛЕННЯ ЧИСЛОВОЇ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

1.1. СИСТЕМИ ЧИСЛЕННЯ

Системою числення називається сукупність правил запису чисел. Системи числення поділяються на позиційні та непозиційні. Як позиційні, так і непозиційні системи числення використовують певний набір символів – цифр, послідовний запис яких утворює число. Непозиційні системи числення з'явилися раніше. Вони характеризуються тим, що символи, які в них використовуються, не змінюють свого значення залежно від місцеположення в складі числа. Прикладом непозиційної системи числення є римська (I-1, X-10, L-50, C-100, D-500).

У позиційній системі числення кількість символів, які використовуються для запису числа, дорівнює основі системи числення. Місце кожної цифри в числі називається *позицією*. Номер позиції символу (за винятком одиниці) в числі називається *розрядом*. Розряд 0 називається молодшим розрядом. Нумерація розрядів починається з нуля.

У загальному випадку кількісний еквівалент деякого позитивного числа А в позиційній системі числення можна представити виразом:

$$A_{(p)} = a_{n-1}p^{n-1} + a_{n-2}p^{n-2} + ... + a_1p^1 + a_0p^0, \qquad (1.1)$$

де p – основа системи числення (деяке позитивне число);

a – цифра даної системи числення;

n – номер старшого розряду числа.

Для отримання кількісного еквівалента числа в деякій позиційній системі числення необхідно скласти добутки кількісних значень цифр на степені основи, показники яких дорівнюють номерам розрядів.

Десяткова система числення. Найбільш відома система числення, оскільки постійно використовується у повсякденному житті. Десяткова система числення використовує такий набір цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Основа системи числення $p = 10$. Кількісний еквівалент деякого цілого n-значного десяткового числа розраховується з виразу

$$A_{(10)} = a_{n-1} * 10^{n-1} + a_{n-2} * 10^{n-2} + ... + a_1 * 10^1 + a_0 * 10^0.$$

Наприклад значення числа $A_{(10)} = 4523$ дорівнює

$$4*10^3 + 5*10^2 + 2*10^1 + 3*10^0.$$

Двійкова система числення. Набір цифр для двійкової системи числення: 0 та 1. Основа системи числення $p = 10$. Кількісний еквівалент деякого *n*-значного двійкового числа розраховується за формулою:

$$A_{(2)} = a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + ... + a_1 * 2^1 + a_0 * 2^0.$$

Наприклад, кількісний еквівалент двійкового числа $A_{(2)} = 11010$ буде таким:

$$1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0.$$

Вісімкова система числення. Набір цифр для вісімкової системи числення: 0, 1, 2, 3, 4, 5, 6, 7. Основа системи числення $p = 8$. Кількісний еквівалент деякого *n*-значного вісімкового числа розраховується за формулою:

$$A_{(8)} = a_{n-1}8^{n-1} + a_{n-2}8^{n-2} + ... + a_1 8^1 + a_0 8^0.$$

Наприклад, кількісний еквівалент вісімкового числа $A_{(8)} = 2103$ буде таким:

$$2*8^3 + 1*8^2 + 0*8^1 + 3*8^0.$$

Шістнадцяткова система числення. Набір цифр для шістнадцяткової системи числення: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *A*, *B*, *C*, *D*, *E*, *F*. Основа системи числення $p = 16$. Кількісний еквівалент деякого *n*-значного шістнадцяткового числа розраховується за формулою:

$$A_{(16)} = a_{n-1}16^{n-1} + a_{n-2}16^{n-2} + ... + a_1 16^1 + a_0 16^0.$$

Наприклад, кількісний еквівалент шістнадцяткового числа *ed*23*f* дорівнює

$$14*16^4 + 13*16^3 + 2*16^2 + 3*16^1 + 15*16^0.$$

Таблиця 1.1

Десяткове число	Двійкова тетрада	Шістнадцяткове число
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5

Закінчення табл. 1.1

Десяткове число	Двійкова тетрада	Шістнадцяткове число
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	*A*, *a*
11	1011	*B*, *b*
12	1100	*C*, *c*
13	1101	*D*, *d*
14	1110	*E*, *e*
15	1111	*F*, *f*

1.2. ПРЕДСТАВЛЕННЯ ЦІЛИХ ЧИСЕЛ У РІЗНИХ СИСТЕМАХ ЧИСЛЕННЯ

Переведення в десяткову систему числення. Цей тип переводу найпростіший. Його виконують за допомогою так званого алгоритму заміщення, суть якого полягає в тому, що спочатку в десяткову систему переводиться основа степеня *p*, а потім – цифри вихідного числа. Результати підставляють до виразу (1.1). Отримана сума і буде результатом.

Переведення у двійкову систему числення.

Переведення з десяткової системи числення.

1. Поділити десяткове число *A* на 2. Запам'ятати частку q та остачу r.

2. Якщо в результаті першого кроку частка $q \neq 0$, то прийняти її за нове ділене й виділити остачу r, яка буде наступною значущою цифрою числа, повернутися до першого кроку, на якому як ділене використовується частка, яка отримана на другому кроці.

3. Якщо в результаті першого кроку частка $q \neq 0$, то алгоритм зупиняється. Записати остачі в порядку, зворотному їх отриманню.

Отже, загальний алгоритм переведення полягає у послідовному діленні числа на основу та визначенні остачі.

Наприклад, десяткове число $A_{10} = 247$ у двійковій системі числення буде представлене так: $A_2 = 11110111$.

Переведення із шістнадцяткової системи числення. Суть алгоритму полягає в послідовній заміні шістнадцяткових цифр відповідними двійковими тетрадами відповідно до табл. 1.1. Наприклад, шістнадцяткове число $A_{16} = e4d5$ у двійковій системі числення буде представлене так: $A_2 = 1110\ 0100\ 1101\ 0101$.

Переведення в шістнадцяткову систему числення.

Переведення з десяткової системи числення. Загальний алгоритм переведення аналогічний до розглянутого вище. Наприклад, порядок переведення десяткового числа $A_{10} = 32767$ у шістнадцяткову систему числення буде таким:

$32767 : 16 = 2047 \rightarrow$ îñòà÷à $= 15(f)$

$2047 : 16 = 127 \rightarrow$ îñòà÷à $= 15(f)$

$127 : 16 = 7 \rightarrow$ îñòà÷à $= 15(f)$

Записується результат $A_{16} = 7fff$.

Переведення з двійкової системи числення. Алгоритм переведення полягає в тому, що двійкове число розбивається на тетради, починаючи з молодшого розряду. Потім кожна тетрада переводиться до відповідного шістнадцяткового числа (табл. 1.1).

Приклад 1.1. Представити двійкове число $A_2 = 11101010011101$ в шістнадцятковій системі числення.

Розв'язання. Число розбивається на тетради:

$$0011\ 1010\ 1001\ 1101.$$

Кожна тетрада згідно з табл. 1.1 представлена в шістнадцятковій системі числення. Результат: $A_{16} = 3a9d$.

Переведення з вісімкової системи числення. Застосовується проміжна система числення, наприклад $A_8 \rightarrow A_2 \rightarrow A_{16}$.

Приклад 1.2. Представити число $A_8 = 214$ в шістнадцятковій системі числення.

Розв'язання. Число представлене у двійковій системі числення

$$A_2 = 010\ \ 001\ 100.$$

Двійкове число розбивається на тетради:

$$0\ \ 1000\ 1100.$$

Кожна тетрада представлена в шістнадцятковій системі числення: $A_{16} = 8C$.

Переведення у вісімкову систему числення.

Переведення з десяткової системи числення. Загальний алгоритм переведення аналогічний до розглянутого вище й полягає у послідовному діленні числа на основу та визначенні остачі. Наприклад, десяткове число $A_{10} = 135$ у вісімковій системі числення буде представлене так: $A_8 = 207$.

Переведення з двійкової системи числення. Алгоритм переведення полягає в тому, що двійкове число розбивається на триади, починаючи з молодшого розряду. Потім кожна триада переводиться у відповідне вісімкове число.

Приклад 1.3. Представити двійкове число $A_2 = 010000111$ у вісімковій системі числення.

Розв'язання. Двійкове число розбивається на триади:

$$010\ 000\ 111.$$

Кожна триада замінюється цифрою вісімкової системи числення. Записується результат: $A_8 = 207$.

Переведення з шістнадцяткової системи числення. Суть алгоритму полягає в переведенні числа, яке представлене у шістнадцятковій системі числення, у проміжну систему числення, а потім у вісімкову систему числення. Наприклад: $A_{(16)} \to A_{(2)} \to A_{(8)}$ або $A_{(16)} \to A_{(10)} \to A_{(8)}$.

1.3. ПРЕДСТАВЛЕННЯ ДІЙСНИХ ЧИСЕЛ У РІЗНИХ СИСТЕМАХ ЧИСЛЕННЯ

Для представлення дійсних чисел у десятковій системі числення використовують формулу

$$A_{(p)} = a_{n-1} * p^{n-1} + a_{n-2} * p^{n-2} + \ldots + a_1 * p^1 + a_0 * p^0 + \\ + a_{-1} * p^{-1} + a_{-2} * p^{-2} + \ldots + a_{-m} * p^{-m}. \qquad (1.2)$$

Процедуру представлення чисел розглянуто на прикладах.

Приклад. Представити в десятковій системі числення дріб, який представлений у двійковій системі числення $A_{(2)} = 100{,}01001$.

Для переведення використовується формула (1.2):

$$100{,}01001_2 = 1*2^2 + 0*2^1 + 0*2^0 + \\ + 0*2^{-1} + 1*2^{-2} + 0*2^{-3} + 0*2^{-4} + 1*2^{-5}$$

Для розрахунку дробової частини виразу зручно використовувати табл. 1.2.

Таблиця 1.2

Результати розрахунків

m	2^{-m}
1	0,5
2	0,25
3	0,125
4	0,0625
5	0,03125

Приклад 1.4. Перевести в десяткову систему числення дріб, що представлений у шістнадцятковій системі числення: $A_{(16)} = df\,2, a1e$.

Розв'язання. Застосовується вираз (1.2)

$$df\,2,a1e_{16} = 13*16^{2} + 15*16^{1} + 2*16^{0} + +10*16^{-1} + 1*16^{-2} + 14*16^{-3}.$$

Таблиця 1.3

Результати розрахунків

m	16^{-m}
1	0,0625
2	0,00390625
3	0,000244140625
4	0,0000152587890625
5	0,00000095367431640625

Для розрахунку дробової частини зручно використовувати табл. 1.3.

Переведення з двійкової системи числення в шістнадцяткову систему і назад виконується на основі тетрад.

Процедура *переведення десяткових дробів у двійкову та шістнадцяткову системи числення*:

1. Виділити цілу частину дробу та виконати її переведення в вибрану систему числення за алгоритмом, що розглянутий раніше.

2. Виділити дробову частину й помножити її на основу вибраної нової системи числення.

3. В отриманій після множення дробової частини десяткового дробу виділити цілу частину та прийняти її як значення першого після коми розряду в новій системі числення.

4. Якщо дробова частина значення, отримана після множення, дорівнює нулю, то припинити процес переводу. Процес переводу припиняється також у випадку, якщо досягнуто необхідну точність розрахунку. В іншому випадку необхідно перейти до третього кроку.

Наприклад: десяткове число 18,406 у двійковій системі числення буде представлене таким чином: 10010,011, а в шістнадцятковій – 12,67E.

Якщо виконати зворотне перетворення, то буде таке значення вихідного числа: 18,40234375. Наявність помилки є наслідком обмеженої кількості розрядів після коми в шістнадцятковому (двійковому) числі.

При переводі в шістнадцяткову систему числення доцільно спочатку перевести дріб у двійкову систему числення, а потім двійкове представлення розбити на тетради окремо до та після коми. Розбиття на тетради двійкових розрядів цілої частини виконують від коми в бік старших розрядів. Неповну старшу тетраду доповнюють зліва нулями. Розряди дробової частини, навпаки, розбивають на тетради вправо від коми в бік молодших розрядів. Якщо остання тетрада неповна, то її доповнюють справа нулями.

1.4. ПРЕДСТАВЛЕННЯ ЦІЛИХ ЧИСЕЛ У ПАМ'ЯТІ КОМП'ЮТЕРА

Цілочислові величини являють собою один з основних форматів даних і є базисом для формування та використання решти, більш складних типів даних таких, як дійсні та комплексні. На основі принципів цілочислових даних реалізовано всі апаратні розширення процесорів, такі як модуль обробки чисел із рухомою комою (FPU, *Float Point Unit*), більш відомий як "математичний сопроцесор".

Своєю чергою цілочислові значення формуються на основі іншого принципу – принципу двійкового представлення інформації, мінімальною одиницею якого є біт. Біт може набувати одного з двох значень – 0 або 1, та є надто малою одиницею, не завжди зручною для практичного застосування. Тому біт служить для будови іншої одиниці даних, що називається байтом. Один байт містить в собі 8 бітів і є найменшою одиницею інформації, до якої можна отримати доступ у пам'яті комп'ютера, при цьому старший біт має номер 7, а молодший – 0. У цьому випадку говорять про байтову організацію пам'яті, хоча пам'ять може організовуватись і у вигляді двобаштових блоків (так звана послівна організація). Платформа IA-32 використовує побайтову організацію пам'яті.

У сучасних комп'ютерах для представлення даних потрібно, як правило, більше одного байта. Якщо дані представлено двома байтами, то говорять, що вони представлені словом. Дані, що потребують 4 байти, позначаються терміном "подвійне слово" (рис. 1.1). Нарешті дані можуть бути представлені 8 байтами (почетверене слово) або 16 байтами (подвійне почетверене слово).

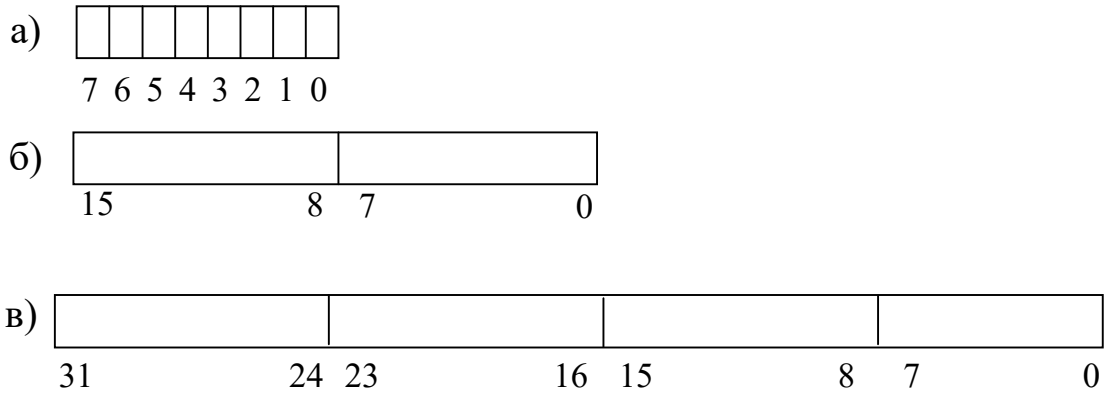

Рис. 1.1. Основні типи даних процесора:
а) нумерація бітів у байті; б) нумерація байтів у слові; в) подвійне слово

Усі сучасні процесори лінії IA-32 обробляють цілочислові дані у двох форматах: двійковому та двійково-десятковому (BCD). Своєю

чергою двійково-десяткові числа можна представити в одному з форматів:

- ASCII (*American National Standard Code for Information Interchange* – Американський національний код для обміну інформацією);
- неупакований двійково-десятковий;
- упакований двійково-десятковий.

Цілі беззнакові числа. Цілі числа без знака в пам'яті комп'ютера представлені у двійковій системі числення. Наприклад, десяткове число $A_{10} = 40000$ у двійковій системі числення буде виглядати так:

$$A_2 = 1001\ 1100\ 0100\ 0000.$$

Для збереження такого числа необхідно мінімум 16 двійкових розрядів. У термінах мови *C++* це формат *unsigned int* для платформи *Win16*. За необхідності це число можна розмістити і в 32 бітах:

0000 0000 0000 0000 1001 1100 0100 000.

Для C++ це формати *unsigned long* та *long* – для платформи *Win16* або *unsigned int*, *int*, *unsigned long*, *long* – для платформи *Win32*.

Таблиця 1.4

Тип	Кількість бітів	Діапазон допустимих значень
unsigned char (байт без знака)	8	0…255
char (байт)	8	–128…+127
unsigned int (коротке ціле без знака)	16	0…65535
short int (коротке ціле число)	16	–32768…+32767
unsigned long (ціле число без знака)	32	0…4 294 967 295
int (ціле)	32	–2 147 483 648…2 147 483 647

Цілі числі зі знаком. Те саме двійкове число, залежно від того як трактується старший біт операнда, можна розглядати як значення без знака, так і значення зі знаком (як додатне та від'ємне). При цьому даний біт для чисел без знака є старшим значущим бітом операнда. Якщо розглядати число зі знаком, то нульове значення цього біта відповідає позитивному, а одиничне – від'ємному значенню, решта розрядів є значущими. Значущі розряди числа не завжди визначають його абсолютне значення (модуль) –

це справедливо тільки для додатних чисел. Для від'ємних чисел вони являють собою так званий *додатковий код* числа.

Додатковий код деякого від'ємного числа являє собою результат інвертування (заміни 1 на 0 та навпаки) кожного біта двійкового числа, яке дорівнює модулю вихідного від'ємного числа із додаванням одиниці. Додатковий код від'ємного числа знаходять за таким алгоритмом:

1) здійснюється переведення модуля числа з десяткової системи числення у двійкову (*dec* → *bin*);

2) здійснюється інверсія отриманого двійкового представлення;

3) до отриманого двійкового представлення додається 1.

Приклад 1.5. Представити число $A_{(10)} = -25$ в додатковому коді припускаючи, що в пам'яті комп'ютера числу виділено один байт.

Розв'язання.

1. Модуль даного числа переводиться у двійкову систему числення (знаковий розряд не враховується)

001 1001.

2. Здійснюється інверсія розрядів

110 0110.

3. Додається одиниця

110 0111.

4. Кодується знак числа. Через те, що число від'ємне, на позицію старшого розряду ставиться 1:

1110 0111.

Для перетворення числа з додаткового коду у двійкову систему числення виконуються такі дії:

1) виконується інвертування додаткового коду;

2) до отриманого двійкового числа додається одиниця.

Приклад 1.6. Представити результат перетворення попереднього прикладу (число 1110 0111) у двійковій системі числення.

Розв'язання.

1. Виконується інвертування розрядів (знаковий розряд не враховується)

001 1000.

2. Додається одиниця

001 1001.

У десятковій системі числення це число 25.

3. Через те, що у вихідному числі на позиції старшого розряду стоїть 1, то до отриманого результату дописується знак “–”:

$$A_{(10)} = -25.$$

Представлення чисел в ASCII-форматі застосовується тоді, коли необхідно ввести дані з консолі або вивести їх на екран дисплея або принтер. Після виконання арифметичних операцій із такими числами необхідно відкорегувати результат за допомогою спеціальних команд. Самі *ASCII-числа* формуються таким чином: старший (лівий) півбайт кожного байта містить значення 3h, а молодший (правий) півбайт – значення десяткового розряду. Наприклад, число 1095 у форматі *ASCII* представлено так: 31303935h, при цьому старший байт містить значення 31h, а наймолодший – 35h (рис. 1.2).

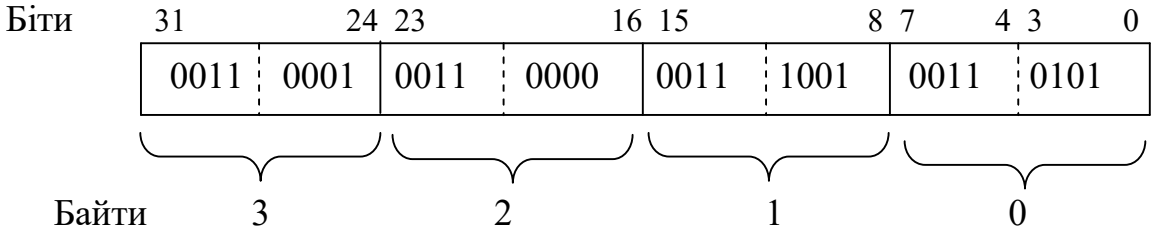

Рис. 1.2. ASCII-формат

Числа в неупакованому двійково-десятковому форматі відрізняються від їх представлення в *ASCII*-форматі тим, що ліві півбайти таких чисел містять 0. Це значно полегшує виконання операцій множення та ділення над ними. Наприклад, число 1095 в неупакованому форматі виглядає так: 01000905h. При цьому найстарший байт містить число 01h, а наймолодший – 05h (рис.1.3).

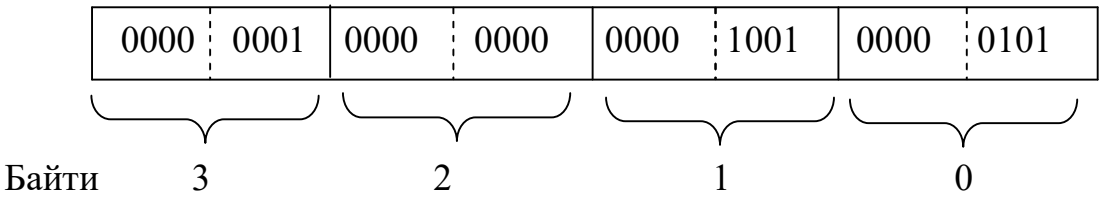

Рис. 1.3. Двійково-десятковий формат

Ще одна форма представлення цілих чисел – **упакований двійково-десятковий формат**, у якому кожен байт упакованого числа містить дві десяткові цифри. У цьому випадку кожне десяткове число буде представлене чотирма бітами. Наприклад, число 6591 в упакова-

ному форматі буде представлене так: 6591h. При цьому старший байт містить значення 65h, молодший – 91h. Упаковані двійкові числа можна тільки додавати та віднімати, а для виконання інших операцій необхідно додаткове перетворення в неупакований формат. На сьогодні застосування упакованих двійково-десяткових чисел досить обмежене.

1.5. ПРЕДСТАВЛЕННЯ ДІЙСНИХ ЧИСЕЛ У ПАМ'ЯТІ КОМП'ЮТЕРА

Основний тип даних, з яким працює процесор ЕОМ, – дійсний. Дійсні числа подаються у форматі з рухомою комою.

Числа з рухомою комою використовуються для представлення дробових величин, а сам термін "рухома кома" базується на формі представлення дробових десяткових чисел, які мають десяткову кому. Виходячи із цього число з рухомою комою представляють у вигляді трьох складових. Для прикладу розглянуто число $+12{,}345\times10^{2}$. Число характеризується такими компонентами:

- знаком ("+" або "–");
- значущою частиною (мантисою), яка являє собою дійсне десяткове число. У цьому прикладі мантиса дорівнює 12,345 і, своєю чергою, складається з цілої (12) та дробової (345) частин, відокремлених комою;
- порядком (або інакше характеристикою), яка визначає степінь числа 10, на який слід домножити мантису. У цьому прикладі порядок дорівнює 2.

Процесори *Intel* можуть оперувати трьома форматами чисел із рухомою комою:

- коротким (одинарна точність, *single precision*), 32 біти, діапазон таких чисел від 2^{-126} до 2^{+127} (рис. 1.4);
- довгим (подвійна точність, *double precision*), 64 біти, діапазон таких чисел від 2^{-1022} до 2^{+1023}, (рис. 1.5);
- розширеним (розширена подвійна точність, *double extended precision*), 80 бітів, діапазон таких чисел від 2^{-16382} до 2^{+16383} (рис. 1.6).

Для представлення дійсного числа використовується вираз

$$A = (\pm M) * N^{\pm p}, \qquad (1.3)$$

де M – мантиса числа, яка повинна задовольняти умову $|M| < 1$;

N – основа системи числення, яка є цілим додатним числом;

p – порядок числа, який показує справжнє положення коми в розрядах мантиси.

Саме через те, що положення коми визначається значенням порядку, дійсні числа отримали назву чисел із рухомою комою.

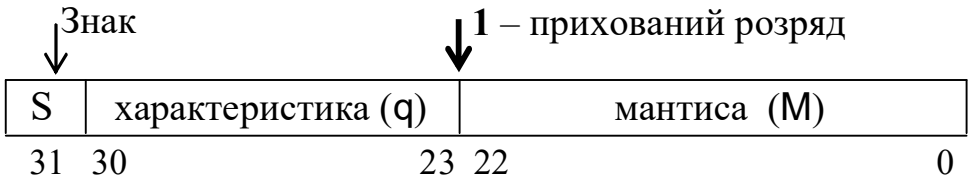

Рис. 1.4. Короткий формат дійсного числа

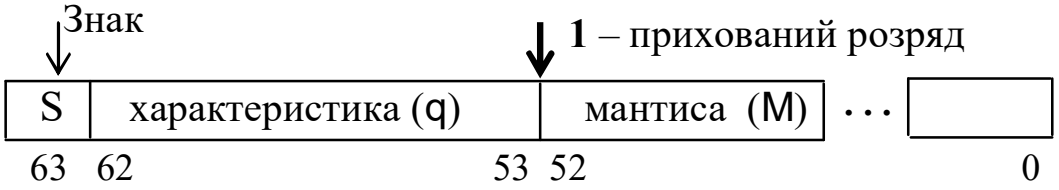

Рис. 1.5. Довгий формат дійсного числа

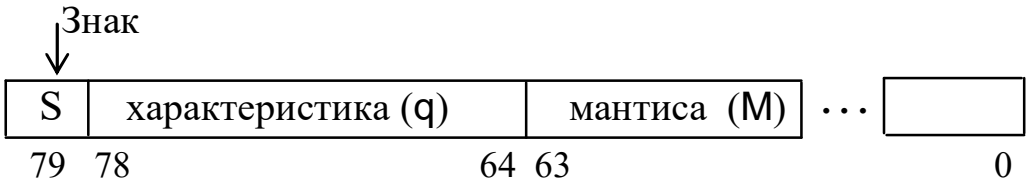

Рис. 1.6. Розширений формат дійсного числа

Для зручності обробки чисел із рухомою комою, архітектурою комп'ютера на складові виразу (1.3) накладаються деякі обмеження. Для сопроцесорів, які застосовуються в архітектурі *Intel*, ці умови полягають у тому, що:

1. Основа системи числення $N = 2$.

2. Мантиса M повинна бути представлена у *нормалізованому* вигляді.

Для різних процесорів нормалізація може розрізнятись. Для архітектури мікропроцесора *Intel* нормалізованим є число, яке представлене у вигляді

$$1{,}00000.... < |M| < 1{,}111111....$$

Тобто перша цифра до коми повинна бути одиницею, а порядок p відповідно таким, щоб ця умова виконувалась.

Для архітектури мікропроцесора *Intel* дійсні числа представлені у вигляді виразу:

$$A = (-1)^S * M * N^q. \qquad (1.4)$$

Тут S – значення знакового розряду (0 – число додатне, 1 – число від'ємне);

q – характеристика, значення якої аналогічне значенню p у формулі (1.3).

У цьому виразі знак мають як порядок числа, так і мантиса. Згідно з рис. 1.4–1.6 формат представлення дійсних чисел передбачає тільки наявність поля (біта) для збереження знака мантиси. Де зберігається знак порядку?

В архітектурі *Intel* на апаратному рівні встановлено, що знак порядку p в форматі дійсного числа визначається особливим значенням, яке називається *характеристикою q*. Величина q пов'язана з порядком p виразом

$$q = p + \textit{фіксоване\_зміщення}. \qquad (1.5)$$

Для кожного з трьох форматів фіксоване зміщення має своє значення (табл. 1.5). Для визначення місця зберігання знаку порядку використано табл. 1.5. Із наведених даних видно, що знак порядку зберігається в старшому біті характеристики.

Таблиця 1.5

Порядок дійсного числа	Значення характеристики	Двійкове представлення характеристики
0	127	0111 1111
–1	126	0111 1110
+1	128	1000 0000

Оскільки нормалізоване дійсне число завжди має цілу одиничну частину, то при його представленні в пам'яті є можливість за замовчуванням вважати перший (зліва) розряд числа одиничним і враховувати його тільки на апаратному рівні. Це дає можливість збільшити діапазон чисел, оскільки з'являється ще один розряд, який можна використати для представлення мантиси числа. Але це справедливо лише для короткого та довгого форматів. Розширений формат містить цілу одиничну складову дійсного числа в явному вигляді.

Методика визначення дійсного числа в пам'яті комп'ютера в короткому та довгому форматах:

1. Перевести число у двійкову систему числення.
2. Представити число в нормалізованому вигляді, визначивши при цьому порядок p, який вказує на скільки розрядів зсунуто кому.
3. Відкинути одиницю з найстаршого розряду (крайню ліву).
4. Розрахувати характеристику q.
5. Заповнити розрядну сітку: поля знакове, характеристики та мантиси.

Наприклад, десяткове число 45,56 у пам’яті комп’ютера в короткому форматі буде виглядати так: $42363d71_{(16)}$, або у двійковій системі числення так, як показано на рис. 1.7.

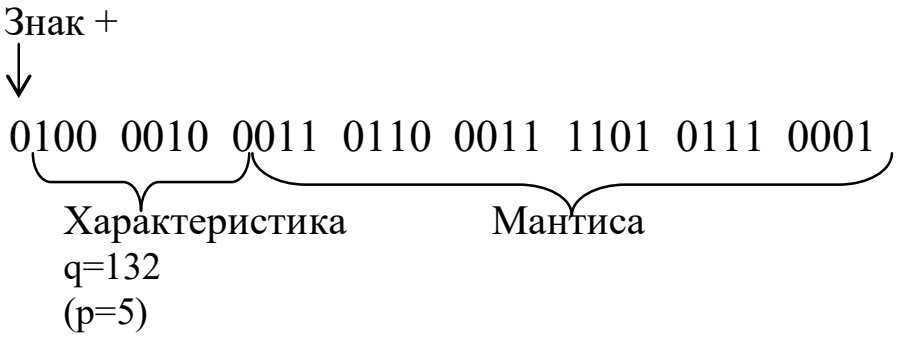

Рис. 1.7. Представлення числа у двійковій системі числення

Контрольні питання

1. Дати визначення поняттю система числення.
2. Назвати позиційні системи числення.
3. У яких форматах представлені цілі числа в пам’яті комп’ютера?
4. У яких форматах представлені дійсні числа в пам’яті комп’ютера?
5. Як в пам’яті комп’ютера представлені від’ємні цілі числа?
6. Які компоненти містить формат дійсних чисел?