Розділ 2

Представлення даних у комп'ютері

У сучасних комп'ютерах використовують різні форми та формати представлення даних, якими є числа та закодовані символи. Це дозволяє вибирати ті із них, що найбільшою мірою відповідають вимогам розв'язуваних задач. Тип використовуваних форм та форматів представлення даних суттєво впливає на характеристики комп'ютера.

У цьому розділі висвітлені основні питання представлення даних у комп'ютері, які є важливими для розуміння матеріалу наступних розділів. Описані позиційні системи числення та принципи подання даних у двійковому, вісімковому і шістнадцятковому кодах. Подані правила переведення чисел із системи числення з довільною основою до десяткової, а також переведення чисел із десяткової до системи числення з іншою основою. Розглянуті засади подання чисел зі знаком у прямому, оберненому та доповняльному кодах. Проведено аналіз особливостей подання даних у форматах із фіксованою та з рухомою комою, включаючи подання даних за стандартом IEEE-754. Зважаючи на важливість, розглянуто питання кодування алфавітно-цифрової інформації кодами ASCII, EBCDIC та Unicode.

2.1. Позиційні системи числення

Система числення – це спосіб подання довільного числа за допомогою алфавіту символів, які називають цифрами. Є різні системи числення. Від їх особливостей залежить наочність відображення чисел та складність виконання операцій над числами. Прикладом системи числення з дуже складним способом запису чисел і громіздкими правилами виконання арифметичних операцій є римська система числення.

Якщо в послідовності цифр, які зображають число, має значення позиція цифри, то систему числення називають позиційною. Такі системи числення характеризуються наочністю відображення чисел та простим виконанням арифметичних операцій. У позиційних системах числення при безпосередньому представленні цифр число записується у вигляді:

$$X = x_s x_{s-1} \ldots x_1 x_0 , x_{-1} \ldots x_{-m}.$$

Кома у цій послідовності відділяє цілу частину числа від дробової. Позиції цифр, які рахуються від коми, називають розрядами. Кількісний еквівалент, що виражається цим записом, визначається так:

$$X = k^s x_s + k^{s-1} x_{s-1} + \ldots + k^1 x_1 + k^0 x_0 + k^{-1} x_{-1} + \ldots + k^{-m} x_{-m},$$

де:

- k – основа системи числення, тобто кількість різних цифр, які використовуються в позиційній системі числення,

- $s+1$ – розрядність цілої частини числа,
- m – розрядність дробової частини числа,
- x_i – цифри i-го розряду запису числа ($x_i = 0, 1, ..., k-1$),
- k^i – вага i-го розряду.

У цьому випадку вага i-го розряду в k разів більша за вагу (i-1)-го розряду. Такі системи числення називають системами з природним порядком ваги. До них належать двійкова, вісімкова, десяткова і шістнадцяткова системи числення.

У звичній для нас десятковій системі числення довільне число подається цифрами від 0 до 9; при цьому має значення позиція цифри. Число в десятковій системі записується у вигляді:

$$D = D_{N-1} D_{N-2} \dots D_1 D_0, D'_1 D'_2 \dots D'_M,$$

а значення числа обчислюється за таким виразом:

$$D = D_{N-1} \cdot 10^{N-1} + D_{N-2} \cdot 10^{N-2} + \dots + D_1 \cdot 10^1 + D_0 \cdot 10^0 + D'_1 \cdot 10^{-1} + D'_2 \cdot 10^{-2} + \dots + D'_M \cdot 10^{-M},$$

де:

- N – кількість цифр (розрядів) у цілій частині числа (зліва від коми),
- M – кількість розрядів у дробовій частині числа (справа від коми),
- D_i – значення i-го розряду (розряди цілої частини),
- D'_i – значення i-го розряду (розряди дробової частини),
- D – значення числа.

Звичайно, що дробової або цілої частини числа може і не бути (N або M = 0).

2.2. Двійкові, вісімкові та шістнадцяткові числа

У зв'язку з тим, що елементи з двома станами використовуються як базові елементи комп'ютерної техніки, всі числа в комп'ютерах представляються у двійковій системі числення. Розглянемо особливості цієї системи.

Двійкова система числення будується за тим самим правилом, що і десяткова, але в ній використовуються лише дві цифри – 0 та 1. Число у двійковій системі числення записується у вигляді:

$$B = B_{N-1} B_{N-2} \dots B_1 B_0, B'_1 B'_2 \dots B'_M,$$

а значення числа обчислюється за таким виразом:

$$B = B_{N-1} \cdot 2^{N-1} + B_{N-2} \cdot 2^{N-2} + \dots + B_1 . 2^1 + B_0 \cdot 2^0 + B'_1 \cdot 2^{-1} + B'_2 \cdot 2^{-2} + \dots + B'_M \cdot 2^{-M},$$

де:

- N – кількість двійкових цифр (розрядів) у цілій частині числа,
- M – кількість двійкових розрядів у дробовій частині числа,
- B_i – значення i-го розряду цілої частини числа,
- B'_i – значення i-го розряду дробової частини числа,
- B – значення числа.

Приклади двійкових чисел:

$1011010{,}01_2 = 1.2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} =$
$= 64 + 16 + 8 + 2 + 0{,}25 = 90{,}25_{10}$

$101{,}01101_2 = 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 . 2^{-5} = 5{,}040625_{10}$

$1100110{,}11_2 = 102{,}75_{10}$

Часто у розробника, а то і в користувача комп'ютера, виникає потреба в перевірці коректності виконання операцій над двійковими числами комп'ютером або його вузлом.

А оскільки в комп'ютерах опрацьовуються багаторозрядні двійкові числа, і оперувати з такими довгими послідовностями нулів та одиниць (наприклад, рядок із 32 цифр) незручно, то набули поширення вісімкова та шістнадцяткова системи числення. У вісімковій системі числення використовують вісім цифр від 0 до 7, а у шістнадцятковій системі числення крім десяткових цифр від 0 до 9 використовують 6 літер латинського алфавіту (A, B, C, D, E, F) для позначення цифр від 10 до 15. Значення числа обчислюється за таким виразом:

$$H = H_{N-1} \cdot 16^{N-1} + H_{N-2} \cdot 16^{N-2} + ... + H_1 \cdot 16^1 + H_0 \cdot 16^0 + H'_1 \cdot 16^{-1} + H'_2 \cdot 16^{-2} + ... + H'_M \cdot 16^{-M};$$

де:

- N – кількість цифр (розрядів) у цілій частині числа (зліва від коми),
- M – кількість розрядів у дробовій частині числа (справа від коми),
- H_i – значення і-го розряду (розряди цілої частини),
- H'_i – значення і-го розряду (розряди дробової частини),
- H – значення числа.

Особливістю цих систем є зручний перехід до двійкової системи та навпаки. Три двійкових розряди переводяться в один вісімковий, а чотири двійкових розряди – в один шістнадцятковий, як показано в табл. 2.1.

Таблиця 2.1

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

Наприклад, двійкове число 01101101 у шістнадцятковій системі записуватиметься як 6D. Для переведення чисел із шістнадцяткової та вісімкової систем числення у двійкову необхідно кожну цифру числа, яке переводиться, замінити відповідно чотири- або трирозрядним двійковим еквівалентом – тетрадою або тріадою, а отримані двійкові цифри розташувати на місцях шістнадцяткових або вісімкових цифр.

У разі необхідності переведення чисел із десяткової системи числення у вісімкову, шістнадцяткову та двійкову переведення робиться лише в одну систему (вісімкову або шістнадцяткову). Подальше переведення виконується через двійкову систему, використовуючи тріади та тетради.

Приклад 1. Переведемо число 12345,67 з десяткової системи числення у двійкову, вісімкову, шістнадцяткову.

1. Переведення цілої частини числа у вісімкову систему:

12345 : 8 = 1543, залишок 1;

1543 : 8 = 192, залишок 7;

192 : 8 = 24, залишок 0;

24 : 8 = 3, залишок 0;

3 : 8 = 0, залишок 3.

Результат: 30071.

2. Переведення дробової частини числа у вісімкову систему:

0,67 x 8 = 5,36;

0,36 x 8 = 2,88;

0,88 x 8 = 7,04;

0,04 x 8 = 0,32.

Наближений результат: 0,5270... .

3. Отримання повного результату шляхом об'єднання результатів, отриманих в п. 1 та п. 2. Результат: 30071,5270... .

4. Переведення результату у двійкову та шістнадцяткову системи числення (табл. 2.2). Поділ двійкового числа на тріади та тетради починається від коми ліворуч і праворуч. Результат: 12345,6710=30071,52708=1100000111001,1010101112=3039,AB816.

Таблиця 2.2

3			0			0			7			1			,	5			2			7			0			8-кові цифри
0	1	1	0	0	0	0	0	0	1	1	1	0	0	1	,	1	0	1	0	1	0	1	1	1	0	0	0	2-кові цифри
3			0				3				9				,	A				B				8				16-кові цифри

2.3. Переведення чисел із системи числення з основою k у десяткову систему

Один із методів переведення чисел із системи числення з основою k у десяткову систему числення ґрунтується на використанні кількісного еквівалента числа. Для переведення необхідно записати число у його кількісному еквіваленті, замінивши цифри системи числення з основою k та основу k їхніми десятковими еквівалентами, а потім обчислити вираз за правилами десяткової арифметики.

Приклад 1. Переведемо двійкове число 1011,1001 у десяткову систему числення.

$1011{,}1001 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 + 1\cdot2^{-1} + 0\cdot2^{-2} + 0\cdot2^{-3} + 1\cdot2^{-4} =$

$= 8 + 0 + 2 + 1 + 0{,}5 + 0 + 0 + 0{,}0625 = 11{,}5625;$

Таким чином, $1011{,}1001_2 = 11{,}5625_{10}$.

Приклад 2. Переведемо вісімкове число 105,71 у десяткову систему числення.

$105{,}71 = 1\cdot8^2 + 0\cdot8^1 + 5\cdot8^0 + 7\cdot8^{-1} + 1\cdot8^{-2} = 64 + 0 + 5 + 0{,}875 + 0{,}015625 = 69{,}890625;$

Результат: $105{,}71_8 = 69{,}890625_{10}$.

Приклад 3. Переведемо шістнадцяткове число 2ED,0A до десяткової системи числення.

$2ED{,}0A = 2\cdot16^2 + 14\cdot16^1 + 13\cdot16^0 + 0\cdot16^{-1} + 10\cdot16^{-2} = 512 + 224 + 13 + 0 + 0{,}0390625 =$

$= 849{,}0390625;$

Результат: $2ED{,}0A_{16} = 849{,}0390625_{10}$.

2.4. Переведення чисел із десяткової системи у систему числення з основою k

Розглянемо переведення чисел із десяткової системи числення у іншу однорідну позиційну систему числення з основою *k*, коли дії виконуються в десятковій системі. У разі цього переведення окремо виконується переведення цілої частини числа й окремо – дробової; результати потім додаються.

Цілу частину десяткового числа X_{10} ділять на основу системи числення k за правилами десяткової арифметики до отримання залишку, який буде десятковим еквівалентом цифри молодшого розряду результату. Якщо частка від ділення не дорівнює 0, то вона стає діленим і процес ділення на k продовжується. Як тільки чергова частка стане рівною 0, процес ділення на k припиняється. Залишок, який отримали у результаті першого ділення на k, є цифрою розряду результату з вагою k^0, залишок у результаті другого ділення – цифрою з вагою k^1 і т. д. Останній залишок є цифрою старшого розряду результату.

Дробова частина десяткового числа X_{10} множиться на k за правилами десяткової арифметики. В отриманому добутку від'єднується ціла частина, яка може дорівнювати 0, а дробова частина знову множиться на k із наступним від'єднанням цілої частини. Ця операція повторюється або до отримання нульової дробової частини добутку, або до отримання необхідної кількості розрядів числа X_k. Цифра старшого розряду результату переведення (тобто, перша після коми) збігається з першою від'єднаною цілою частиною, цифра другого розряду результату переведення – із другою від'єднаною цілою частиною і т.д. При цьому від'єднані цілі частини необхідно представити в системі числення з основою k.

Приклад. Переведемо десяткове число 11,5625 у двійкову систему числення з точністю до п'яти розрядів після коми.

Переведення цілої частини:

11 : 2 = 5, залишок 1 (молодший розряд результату),

5 : 2 = 2, залишок 1,

2 : 2 = 1, залишок 0,

1 : 2 = 0, залишок 1 (старший розряд результату).

Результат: $11_{10} = 1011_2$.

Процедура переведення дробової частини наведена у табл. 2.3.

Таблиця 2.3

Крок	Дріб	Результат множення на $k = 2$	Ціла частина результату множення, яка вилучається	Вага двійкового розряду
1	0.5625	1.125	1	Старший (перший після коми)
2	0.125	0.25	0	
3	0.25	0.5	0	
4	0.5	1.0	1	Молодший
5	0.0	0.0	0	

Результат: $0.5625_{10} = 0{,}10010_2$.

Повний результат: $11{,}5625_{10} = 1011 + 0{,}10010 = 1011{,}10010_2$.

2.5. Представлення чисел зі знаком

Для позначення знаку числа в звичайній арифметиці використовують символи «–» та «+». Як зазначалося, у комп'ютерній техніці використовують елементи з двома станами, які можуть зберігати двійкову цифру (0 чи 1). Зрозуміло, що цю цифру доцільно використати і для позначення знаку числа, коли 0 відображає знак «+», а 1 – знак «–».

Для спрощення виконання арифметичних операцій додатні та від'ємні числа (тобто числа зі знаком) відображаються спеціальними кодами: прямим, оберненим та доповняльним.

2.5.1. Прямий код

У прямому коді лівий (його ще називають старшим) розряд позначає знак числа, а решта розрядів – саме число (рис. 2.1).

Рис. 2.1. Прямий код двійкового числа

Прямий код двійкового n-розрядного числа G визначається як

$$G_{ПР} = \begin{cases} G, \text{ при } G \geq 0; \\ A + |G|, \text{ при } G \leq 0; \end{cases}$$

де A – величина, рівна вазі старшого разряду розрядної сітки (для дробових чисел A = 1, а для цілих чисел $A = 2^n$). Діапазон представлення чисел в прямому коді $0 \leq |G| < A$. Додатні числа представляються кодами $0 \leq G_{ПР} < A$, а від'ємні $0 \leq G_{ПР} < 2A$

Ознакою представлення додатних або від'ємних чисел є наявність нуля або одиниці відповідно в старшому розряді, який називається знаковим. Цифрові розряди прямого коду представляють модуль числа, що забезпечує наочність представлення чисел в прямому коді

Наведемо кілька прикладів

$5_{10} = 00101_{\text{прямий код}}$ $\qquad$ $5_{10} = 10101_{\text{прямий код}}$

$25_{10} = 011001_{\text{прямий код}}$ $\qquad$ $-25_{10} = 111001_{\text{прямий код}}$

2.5.2. Обернений код

В оберненому коді, як і у прямому, старший розряд позначає знак числа (0 – додатне число, а 1 – від'ємне). Розряди додатного числа записуються у звичайному вигляді, а від'ємного – в інвертованому вигляді (замість 0 пишеться 1 і навпаки). На рис. 2.2 показано обернений код двійкового числа.

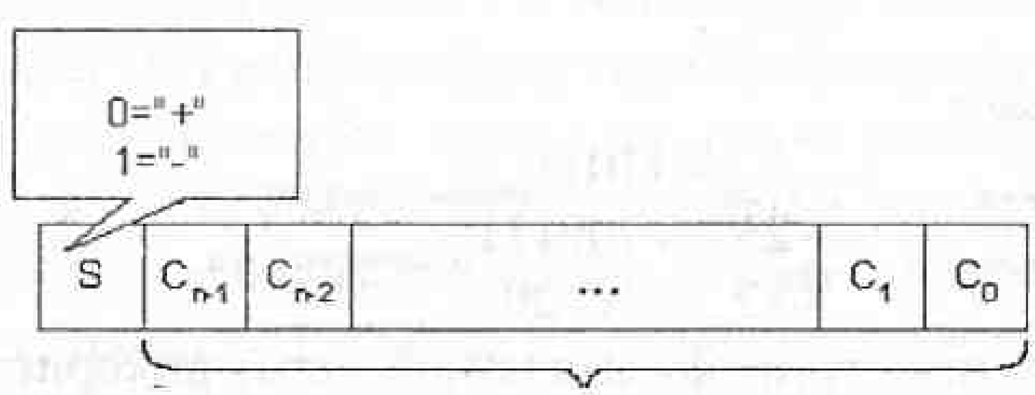

Рис. 2.2. Обернений код двійкового числа

Обернений код n-розрядного двійкового числа G визначається як

$$G_{ОБР}=\begin{cases} G,\ при\ G \ge 0; \\ B-|G|,\ при\ G \le 0; \end{cases}$$

де B – величина найбільшого числа без знаку, яке може бути розміщене в n- розрядній сітці (для дробових чисел $B = 2 - 2-(n-1)$, а для цілих чисел $B = 2n-1$). Діапазон зміни чисел в оберненому коді $0 \le |G| < A$. Додатні числа представляються кодами в діапазоні $0 \le G_{ПР.} < A$, а від'ємні – в діапазоні $A \le G_{ПР} < 2A$. За визначенням обернений код від'ємного числа є доповненням модуля вихідного числа до найбільшого числа без знаку, яке може бути розміщене в розрядній сітці. В зв'язку з цим отримання оберненого коду двійкового від'ємного числа зводиться до отримання інверсії n-розрядного коду модуля цього числа

Знову варто навести кілька прикладів

$5_{10} = 00101_{\text{обернений код}}$ $\qquad -5_{10} = 10101_{\text{обернений код}}$
$25_{10} = 011001_{\text{обернений код}}$ $\qquad -25_{10} = 111001_{\text{обернений код}}$

2.5.3. Доповняльний код

Доповняльний код будується на основі оберненого. Якщо число додатне, то не проводиться жодних дій, якщо від'ємне – після інвертування до молодшого розряду числа додається одиниця (рис. 2.3).

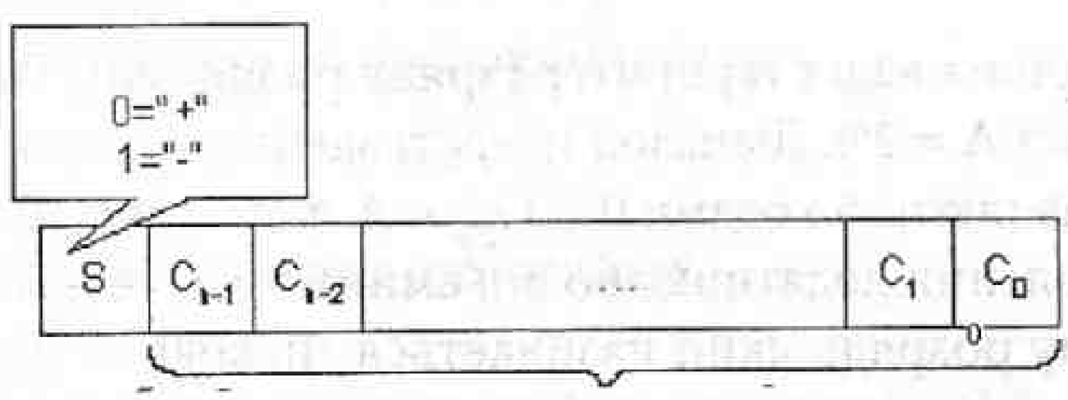

Рис. 2.3. Доповняльний код двійкового числа

Доповняльний код n-розрядного двійкового числа G визначається як

$$G_{ОБР}=\begin{cases} G,\ при\ G \ge 0; \\ C-|G|,\ при\ G \le 0; \end{cases}$$

де C – величина, рівна вазі розряду, який іде за старшим розрядом використаної розрядної сітки (для дробових чисел $C = 2$, а для цілих чисел $C = 2^{n+1}$). Діапазон зміни чисел у прямому коді $0 \le |G < A$. Цифровими розрядами доповняльного коду додатного числа виражається модуль цього числа. Як уже було зазначено, доповняльний код від'ємного числа зручно отримувати через обернений код шляхом додавання 1 до молодшого розряду оберненого коду

Розглянемо приклади

$5_{10} = 00101_{\text{доповняльний код}}$ $\qquad -5_{10} = 11011_{\text{доповняльний код}}$
$25_{10} = 011001_{\text{доповняльний код}}$ $\qquad -25_{10} = 100111_{\text{доповняльний код}}$
$12_{10} = 01100_{\text{доповняльний код}}$ $\qquad -12_{10} = 10100_{\text{доповняльний код}}$

Найчастіше серед розглянутих кодів в комп'ютерах використовується доповняльний код. Це зумовлено більшою зручністю проведення арифметичних операцій над числами, представленими в такому коді, оскільки при його застосуванні операція алгебраїчного додавання зводиться до додавання арифметичного.

2.6. Формати даних

2.6.1. Способи представлення чисел

Розряд двійкового числа представляється в комп'ютері деяким технічним пристроєм, наприклад тригером, двом різним станам якого приписують значення 0 та 1. Один двійковий розряд який може набувати ці два значення, є найменшою одиницею інформації, названої бітом. Набір відповідної кількості таких пристроїв слугує для представлення багаторозрядного двійкового числа (або в загальному випадку – двійкового коду слова). Розрядність слова може бути від 1 біта до довільної кількості n бітів. Слово із 8 бітів називають байтом. Як правило, коли йдеться про комп'ютерну техніку, всі виміри кількості розрядів наводяться в бітах або байтах Часто словом іще називають число із 32 бітів, а число із 16 бітів – півсловом.

Коли деяке число має 32 біти, то говорять, що воно представлене з одинарною точністю, якщо ж 64 біти – з подвійною точністю

Числові дані в комп'ютері зазвичай представляються трьома способами

- як цілі або дробові числа з фіксованою комою, які складаються із деякої кількості бітів;
- як числа з рухомою (ще деколи вживають "плаваючою") комою, кожне з яких має порядок та мантису
- як двійково кодовані десяткові, де байт (чи півбайта) представляє одну десяткову цифру, а послідовність байтів (чи півбайтів) представляє число.

Якщо певне число більше за максимальне, яке може бути представлене певною кількістю розрядів, то значення числа може втрачатися. Таку ситуацію називають переповненням Розробники комп'ютера або програми повинні передбачити, числа якої величини будуть використовуватись, і виділити для їх представлення таку кількість розрядів, щоб значення числа не втрачалось

Сучасні персональні комп'ютери використовують слова розрядністю від одного до 16 байтів. Спеціалізовані комп'ютери можуть використовувати слова й іншої розрядності, наприклад 128 байтів

Для кодування символів використовуються спеціальні коди, серед яких найпоширеніший у персональних комп'ютерах – американський стандартний код інформаційного обміну ASCII, а в мейнфреймах – розширений двійково-кодований десятковий код обміну EBCDIC Зазвичай для представлення одного символа використовується один байт

2.6.2. Числа з фіксованою комою

У разі використання чисел із фіксованою комою, представлення коми не виконується, але вважається, що вона є на певній наперед відомій позиції відносно розрядів числа. Найчастіше вважається, що кома стоїть після молодшого розряду числа (таким чином представляються цілі числа) або перед старшим розрядом числа (таким чином представляються дробові числа), хоча можливе застосування і змішаного варіанту. У такому форматі представляються числа з діапазону $-1 \le$ *число* < 1 (якщо є знаковий розряд) або $0 \le$ *число* < 1 (якщо знакового розряду немає).

На рис. 2.4 показано приклад розрядної сітки комп'ютера (формату даних) для представлення двійкових чисел із фіксованою комою в вигляді 32-розрядних слів для випадків закріплення коми перед старшим і після молодшого розряду. Розряди пронумеровані зліва направо.

	Знак	2^{-1}	2^{-2}	2^{-3}		2^{-30}	2^{-31}
a)	a_0	a_1	a_2	a_3		a_{30}	a_{31}

	Знак	2^{30}	2^{29}	2^{28}		2^{1}	2^{0}
b)	a_0	a_1	a_2	a_3		a_{30}	a_{31}

Рис. 2.4. Розрядна сітка при представленні двійкових чисел з фіксованою комою: а – кома фіксована перед старшим розрядом a_1, b – кома фіксована після молодшого розряду a_{31}

Для кодування знаку числа використовується знаковий розряд (a_0 на рис. 2.4). У цьому розряді 0 відповідає знаку «+», а 1 – знаку «–». На розрядній сітці вказано вагу кожного розряду. Найбільше додатне число, яке може бути представлене в розрядній сітці, показаній на рис. 2.4 а, рівне $0{,}11...1=1-2^{-31}$. Тут після коми розміщена 31 одиниця. А найменше додатне значуще число рівне $0{,}00...\ 01 =2^{-31}$. Тут після коми розміщено 30 нулів.

В розрядній сітці (рис. 2.4 а) можуть бути представлені числа в діапазоні від $-(1-2^{-31})$ до -2^{-31} і від $+2^{-31}$ до $+(1-2^{-31})$, що відповідає діапазону абсолютних десяткових чисел приблизно від $(1-10^{-9})$ до 10^{-9}. Числа $|x| < 2^{-31}$ не можуть бути представлені в розрядній сітці і приймаються рівними 0 (число виходить за розрядну сітку вправо). Всі числа $|x| >=1$ також не можуть бути представлені в прийнятій розрядній сітці.

Таке число виходить за межі сітки вліво (відбувається переповнення розрядної сітки), і його старші розряди (розряди зліва від коми) втрачаються, а результат обчислень виявляється неправильним. Тому, зазвичай, якщо при виконанні певної програми виникає переповнення, в арифметико-логічному пристрої формується сигнал, який фіксується в відповідному тригері та повідомляє операційну систему комп'ютера про наявність переповнення.

Якщо кома зафіксована праворуч від молодшого розряду, розрядна сітка (рис. 2.4 б) дозволяє представляти додатні та від'ємні цілі двійкові числа, модуль яких $1<=|x|<=2^{31}-1$, що відповідає діапазону абсолютних десяткових чисел приблизно від 1 до 10^9, а також 0.

Всі числа, модуль яких менший 1 або більший $(2^{31}-1)$, не можуть бути представлені в цій розрядній сітці (число виходить за межі розрядної сітки).

При виконанні на комп'ютері обчислень необхідно, щоб всі вихідні та отримувані в процесі обчислень проміжні і кінцеві дані не виходили за діапазон чисел, які можуть бути представлені в цій розрядній сітці. В іншому випадку в обчисленнях можуть виникнути помилки. Для цього під час написання програм дані, що задіяні в обчисленнях, беруться з відповідними масштабними коефіцієнтами.

При виконанні науково-технічних розрахунків масштабування є простішим, якщо всі числа по модулю не перевищують 1, тобто кома зафіксована перед старшим розрядом числа.

Комп'ютери, які опрацьовують числа в форматі з фіксованою комою, є простішими (меншими за габаритами) та швидшими порівняно з комп'ютерами, які опрацьовують числа в форматі з рухомою комою, але в них можливе виникнення проблем через потребу передбачення переповнення. Перші комп'ютери опрацьовували дані з фіксованою комою, причому кома, як правило, фіксувалась перед старшим розрядом числа. Зараз представлення чисел з фіксованою комою використовується як єдине лише в порівняно невеликих за своїми обчислювальними можливостями комп'ютерах, які використовуються для управління технологічними процесами та опрацювання вимірювальної інформації в реальному часі.

В комп'ютерах, призначених для вирішення широкого кола обчислювальних задач, основним є представлення чисел з рухомою комою, яке не вимагає масштабування даних.

Однак у таких комп'ютерах, крім представлення чисел в цьому форматі, часто використовується представлення з фіксованою комою, оскільки на виконання операцій з такими числами витрачається менше часу. При цьому в більшості випадків формат чисел із фіксованою комою слугує для представлення цілих двійкових чисел (кома ставиться праворуч від молодшого розряду числа) та виконання операцій над ними, що, зокрема, необхідно для операцій над кодами адрес (операцій індексної арифметики).

Розглянемо основні формати чисел із фіксованою комою, що використовуються у сучасних комп'ютерах, та діапазони представлення в них чисел (табл. 2.4).

Таблиця 2.4

Формати без знакового розряду		
довжина	**Min**	**Max**
16 байт = 128 біт	0	3.40282366920938e+38
8 байт = 64 біт	0	18446744073709551615
4 байти = 32 біт	0	4294967295
2 байти = 16 біт	0	65535
1 байт = 8 біт	0	255
Формати зі знаковим розрядом (доповняльний код)		
довжина	**Min**	**Max**
16 байт = 128 біт	–1.70141183460469e+38	1.70141183460469e+38
8 байт = 64 біт	–9223372036854775808	9223372036854775807
4 байти = 32 біт	–2147483648	2147483647
2 байти = 16 біт	–32768	32767
1 байт = 8 біт	–128	127

Інколи під час створення спеціалізованих комп'ютерів зручно ставити кому в іншу позицію (не лише праворуч від молодшого розряду числа або ліворуч від старшого розряду числа), і таким чином виокремлювати певну кількість розрядів для подання цілої частини числа, а також певну кількість розрядів для подання дробової частини числа.

2.6.3. Числа із рухомою комою

Не завжди діапазон представлення чисел у форматі з фіксованою комою є достатнім для проведення обчислень. В такому випадку використовується формат представлення чисел із рухомою комою.

У загальному випадку в форматі з рухомою комою число подається у вигляді $A = \pm m \cdot q^{\pm p}$, де m – мантиса числа, q – основа порядку, ± p – порядок числа. Попередній вираз можна записати як $A = \pm m_A \cdot \pm p_A$, де упущено основу порядку, оскільки в комп'ютерах вона незмінна. В більшості випадків основа порядку дорівнює основі системи числення, тобто 2.

Для однозначного і максимально точного відображення чисел число з рухомою комою представляють у нормалізованому вигляді. Якщо виконується нерівність $q - 1 <= |m| < 1$, а у випадку двійкової системи числення $0.5 <= |m| < 1$ (старший двійковий розряд мантиси дорівнює 1), то вважається, що число представлене в нормалізованому вигляді.

Таким чином, у двійкового нормалізованого числа у форматі з рухомою комою мантиса є правильним дробом і у старшому розряді мантиси завжди стоїть 1. Операцію приведення числа до нормалізованого вигляду називають нормалізацією. Нормалізація чисел у комп'ютері виконується або апаратно, або ж спеціальною програмою.

Для представлення двійкового числа у форматі з рухомою комою у розрядній сітці, наданій для цієї мети, виділяється:

- по одному розряду для представлення знаку числа Sm (поле знаку числа) і знаку порядку Sp;
- певне число розрядів для представлення значення порядку p;
- розряди для представлення значення модуля мантиси m (поле мантиси).

Наприклад, можливий такий варіант, коли формат числа складається з чотирьох полів (рис. 2.5) тобто, $[A] = S_p p_A S_m m_A$.

S_p	p_A	S_m	m_A

Рис. 2.5. Число з рухомою комою

Зазвичай у форматі з рухомою комою замість порядку p використовують так звану характеристику ("зміщений порядок") $r = \pm p + l$, де l – надлишок (зсув), значення якого підбирається таким чином, щоб у разі зміни значення показника від деякого мінімального значення $-|p_{min}|$ до максимального $+|p_{max}|$, характеристика r змінювалася від 1 до r_{max}. Отже, характеристика не змінює свого знаку. У цьому випадку відпадає необхідність у відображенні знаку порядку S_p. Для цього приймається, що $l = 2^{k-1}$, де k – число розрядів, виділених для представлення порядку числа у форматі з рухомою комою.

Тоді формат числа з рухомою комою можна подати так, як показано на рис. 2.6 (з використанням трьох полів), тобто, $[A] = S_m p_A m_A$.

S_m	p_A	m_A

Рис. 2.6. Число з рухомою комою із зміщеним порядком

Одиниця старшого розряду нормалізованої мантиси зазвичай не відображається у форматі числа, тобто є уявною. Розряд слова, в якому повинна була бути відображена ця одиниця, використовується як молодший розряд характеристики, або старший розряд мантиси, що дозволяє збільшити діапазон представлення чисел у форматі з рухомою комою, або точність обчислень.

Таким чином, мантиса в такому варіанті відображається, починаючи з розряду, що йде після старшого. Це слід враховувати під час виконання будь-якої операції з мантисою числа, і перед початком операцій відновлювати старший розряд мантиси. Після завершення операцій формування нормалізованого результату у відведеній для нього розрядній сітці, старша одиниця мантиси знову відкидається. Порядок із k-розрядним полем може змінюватися в межах від $-2^{k-1} + 1$ до $+ 2^{k-1} - 1$ (табл. 2.5, k = 3).

Таблиця 2.5

Показник порядку	Прямий код показника	Характеристика (показник + 4)	Примітки
+3	011	111	3 + 4 =7
+2	010	110	2 + 4 = 6
+1	001	101	1 + 4 = 5
0	000	100	0 + 4 = 4
–1	101	011	–1 + 4 = 3
–2	110	010	–2 + 4 = 2
–3	111	001	–3 + 4 = 1
		000	Ознаку нуля

Як зазначалося, характеристика r – це порядок p з надлишком $l = 2^{k-1}$. Вона не змінює свого знаку і змінюється від 1 (при $p = -2^{k-1}$) до 2^{k-1} (при $p = +2^{k-1} - 1$). Винятком є число 0, яке виражається нульовою характеристикою і нульовою мантисою (не обов'язково).

Основною перевагою представлення чисел у форматі з рухомою комою є великий діапазон машинних чисел і висока точність їхнього подання. Діапазон визначається довжиною розрядної сітки, виділеної для характеристики, а точність визначається довжиною розрядної сітки, виділеної для мантиси.

Особливості виконання операцій над числами з рухомою комою:

- збільшення мантиси у 2 рази здійснюється зсувом двійкового значення мантиси ліворуч (у бік старших розрядів);
- зменшення мантиси у 2 рази здійснюється зсувом двійкового значення мантиси праворуч (у бік молодших розрядів);
- величина числа не зміниться, якщо збільшити мантису в 2 рази і одночасно зменшити порядок на 1;
- величина числа не зміниться, якщо зменшити мантису в 2 рази і одночасно збільшити порядок на 1.

Тобто формат з рухомою комою має недолік, який полягає у відсутності унікального представлення для кожного числа. Усі числа, що наводяться на рис. 2.7, є еквівалентними. Слід зауважити, що цього недоліку не мають нормалізовані числа.

0	1 0 1 0 1	1 0 0 0 1 0 0 0

0	1 0 1 1 0	0 1 0 0 0 1 0 0

0	1 0 1 1 1	0 0 1 0 0 0 1 0

0	1 1 0 0 0	0 0 0 1 0 0 0 1

Рис. 2.7. Еквівалентні двійкові числа в форматі з рухомою комою

Під час арифметичних операцій над числами з рухомою комою виконуються дії як над порядком, так і над мантисою.

У деяких моделях комп'ютерів одержало поширення відображення чисел із рухомою комою з основою порядку, рівною цілому ступеню числа 2 ($s = 2^r$). При цьому порядок p відображається двійковим цілим числом, а мантиса m – числом, в якому групи по r двійкових розрядів зображають цифри мантиси з основою системи числення s.

Прикладами вживаних основ порядку є числа 8 та 16.

Використання для чисел з рухомою комою недвійкової основи порядку дещо зменшує точність обчислень (при заданому числі розрядів мантиси), але дозволяє збільшити діапазон чисел, що представляються в машині, і прискорити виконання деяких операцій, зокрема нормалізації, за рахунок того, що зсув проводиться відразу на кілька двійкових розрядів. Крім того, зменшується вірогідність появи ненормалізованих чисел в ході обчислень.

Наприклад, у разі використання шістнадцяткових чисел з рухомою комою число X вважається нормалізованим, якщо старша шістнадцяткова цифра X_1 відмінна від 0. Тобто у нормалізованому числі три старші двійкові цифри можуть дорівнювати 0. Це дещо зменшує точність представлення чисел при фіксованому числі розрядів мантиси. Якщо r старших шістнадцяткових розрядів мантиси рівні 0, то нормалізація в цьому випадку полягає в зсуві вліво мантиси на r шістнадцяткових розрядів і відповідному зменшенні показника порядку на r одиниць. Зсув на один шістнадцятковий розряд виконується як зсув мантиси відразу на чотири двійкові розряди.

Розглянемо кілька прикладів.

Припустимо, що потрібно подати у форматі з рухомою комою число 17. Для десяткової системи $17 = 17.0 \times 10^0 = 1.7 \times 10^1 = 0.17 \times 10^2$. Аналогічно в двійковій системі $17_{10} = 10001_2 \times 2^0 = 1000.1_2 \times 2^1 = 100.01_2 \times 2^2 = 10.001_2 \times 2^3 = 1.0001_2 \times 2^4 = 0.10001_2 \times 2^5$. Якщо використати останній запис, то 8-розрядна мантиса числа буде рівною 10001000, а 5-розрядний порядок буде рівним 00101. Тоді число 17 в форматі з рухомою комою в двійковій системі має вигляд, показаний на рис. 2.8 а. Використовуючи формат з рухомою комою можна представляти числа в значно ширшому діапазоні, ніж використовуючи формат з фіксованою комою, при тих самих 14 розрядах. Так, на рис. 2.8 b) показано число 65536 $= 0.1_2 \times 2^{17}$ у форматі з рухомою комою, для представлення якого у форматі з фіксованою комою потрібно було б 16 розрядів.

a)	0	0 0 1 0 1	1 0 0 0 1 0 0 0
b)	0	1 0 0 0 1	1 0 0 0 0 0 0 0

Рис. 2.8. Числа 17 a) та 65536 b) в двійковому форматі з рухомою комою

Як вже зазначалося, ідея зміщення порядку полягає в перетворенні його значень лише в додатні числа. Зміщення здійснюється шляхом додавання до кожного значення порядку фіксованого числа, рівного середньому значенню величини діапазону можливих чисел, яке вибирається для представлення нуля. В приведених вище прикладах як зміщення потрібно взяти число 16, тому що воно є середнім між 0 і 31 (порядок має 5 бітів, тому дозволяє представити $2^5 = 32$ значень). Будь-яке число, більше ніж 16, в полі порядку буде представляти додатне значення, а менше – від'ємне. Зауважимо іще раз, що значення порядку з усіма нулями та одиницями зазвичай резервується для спеціальних випадків (таких як нуль та нескінченість).

Повернемося до попереднього прикладу. Ми обчислили $17_{10} = 0.10001_2 \times 2^5$. Зміщення порядку рівне $16 + 5 = 21$, і число має вигляд, показаний на рис. 2.9 а. Аналогічно для числа $0.25 = 1.0 \times 2^{-2}$ будемо мати представлення, показане на рис. 2.9 b.

a)	0	1 0 1 0 1	1 0 0 0 1 0 0 0
b)	0	0 1 1 1 0	1 0 0 0 0 0 0 0

Рис. 2.9. Числа 17 a) та 0.25 b) в двійковому форматі з рухомою комою із зміщеним порядком

Розглянемо іще один приклад числа з рухомою комою, в даному випадку нормалізованого. Виразимо 0.03125_{10} в форматі з рухомою комою із зміщенням порядку на 16. Тоді $0.03125_{10} = 0.00001_2 \times 2^0 = 0.0001 \times 2^{-1} = 0.001 \times 2^{-2} = 0.01 \times 2^{-3} = 0.1 \times 2^{-4}$. Додавши до порядку зміщення отримаємо $16 - 4 = 12$. Повний вигляд числа показано на рис. 2.10.

0	0 1 1 0 0	1 0 0 0 0 0 0 0

Рис. 2.10. Число 0.03125_{10} в нормалізованому двійковому форматі з рухомою комою із зміщеним порядком

На рис. 2.11 наведено два приклади використовуваного в комп'ютерах формату представлення чисел з рухомою комою. Зверху наведено формат, який був використаний в комп'ютерах CDC 6600, CDC 7000, та CYBER 170 фірми СКС, а знизу – формат, який був використаний в комп'ютерах системи IBM/370 фірми IBM, причому тут основою порядку є число 16, тому мантиса вважається нормалізованою, якщо є хоча б одна одиниця в перших її чотирьох розрядах.

Рис. 2.11. Формат з рухомою комою комп'ютерів фірм CDC та IBM

Існує велика кількість задач, коли обробці підлягають масиви чисел, які змінюються в вузькому діапазоні значень. В цьому випадку з метою більш ефективного використання розрядної сітки для представлення чисел використовують так звану поблоково-рухому кому, коли для всього масиву чисел є лише один порядок. В спеціалізованих комп'ютерах це дозволяє суттєво зменшити витрати обладнання на побудову арифметико-логічного пристрою.

При потребі іще більшого розширення діапазону представлення даних використовується так званий формат з рухомою-рухомою комою. Тут використовується два поля порядку, як це показано на рис. 2.12.

Sm	P1	P2	M

Рис. 2.12. Формат з рухомою-рухомою комою

Тут значення числа визначається з виразу $A = 2^{e}$, де $e = P_1 2^{P2}$.

2.6.4. Стандарт IEEE-754

Для представлення чисел з рухомою комою у більшості сучасних комп'ютерів використовується стандарт IEEE-754. В попередньому пункті ми розглянули можливі варіанти представлення даних в форматі з рухомою комою. До середини 80-х років в різних комп'ютерах використовувались різні варіанти цього представлення, що суттєво ускладнювало виконання на них тих самих програм. У 1985 році Інститут інженерів електротехніків і радіоелектроніків (IEEE) розробив стандарт для чисел з рухомою комою, який офіційно відомий як IEEE-754 (1985).

Стандарт IEEE-754 для чисел з одинарною точністю використовує зміщення 8-розрядного порядку на 127. Це ще один спосіб представлення чисел із знаком без використання знаку мінус. Мантиса має 23 біти. Із знаковим розрядом включно повна довжина слова складає 32 біти (рис. 2.13).

Рис. 2.13. IEEE-754 стандарт для чисел з одинарною точністю

Значення числа обчислюється за формулою:

$$число = (-1)^{S} \cdot 2^{E-127} \cdot (1, M).$$

Мантиса представляється в прямому коді без знаку, знак мантиси представляється окремо. Суть нормалізації полягає в тому, що мантиса приводиться до вигляду 1.xxxxx, тобто вона знаходиться в межах від 1,000...0 до 1,111...1. Слід зауважити, що оскільки кожна мантиса після нормалізації починається з 1, то нема сенсу зберігати цей розряд, тому він не зберігається разом з числом. Його необхідно просто враховувати під час операцій над числами.

Числа з подвійною точністю в стандарті IEEE-754 подаються 64-розрядним словом, яке має знаковий розряд, 11-розрядний порядок і 52-розрядну мантису (рис. 2.14). Зміщення порядку дорівнює 1023.

Рис. 2.14. IEEE-754 стандарт для чисел з подвійною точністю

Значення числа обчислюється за формулою:

$$число = (-1)^S \cdot 2^{E-1023} \cdot (1, M).$$

Діапазон чисел, які можуть бути представлені в цьому форматі, показаний на рис. 2.15.

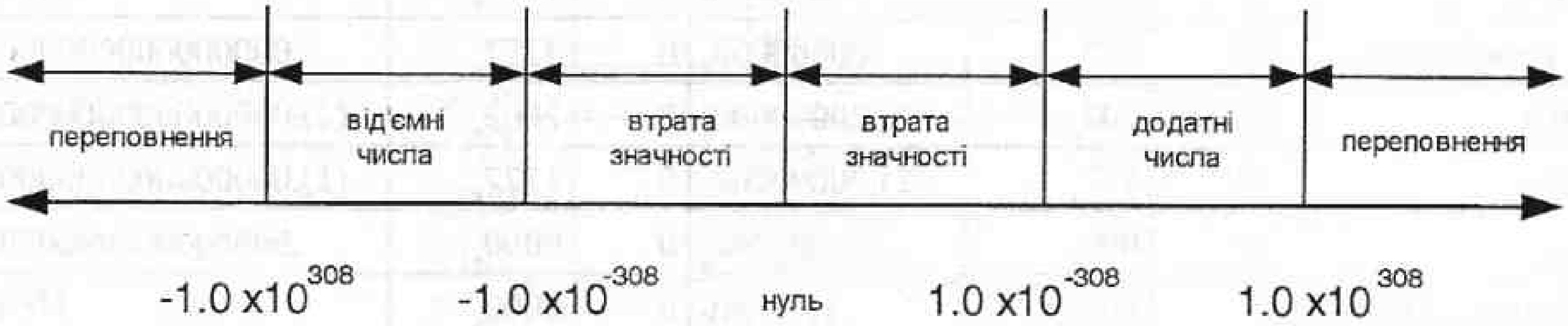

Рис. 2.15. Діапазон чисел, які відображаються у форматі за стандартом IEEE-754 з подвійною точністю

У табл. 2.6 наведено характеристики форматів подання двійкових чисел в стандарті IEEE-754 з одинарною та подвійною точністю.

Таблиця 2.6

Характеристика	Формат з одинарною точністю	Формат з подвійною точністю
Довжина слова	32 біти	64 біти
Мантиса (зі знаком)	24 біти	53 біти
Порядок	8 бітів	11 бітів
Зміщення	127	1023
Наближений діапазон	$2^{128} = 3.8\ 10^{38}$	$2^{1024} = 9\ 10^{307}$
Найменше нормалізоване число	$2^{-126} = 10^{-38}$	$2^{-1022} = 10^{-308}$
Наближена точність представлення чисел	$2^{-23} = 10^{-7}$	$2^{-52} = 10^{-15}$

Як числа з одинарною точністю, так і числа з подвійною точністю в стандарті IEEE-754 мають для нуля два варіанти представлення. Коли порядок і мантиса рівні нулю – число є нулем. При цьому значення знаку є несуттєвим. На цю обставину потрібно звертати увагу при проведенні операції порівняння числа з рухомою комою на збіжність з нулем.

Стандарт IEEE-754 передбачає використання певної кількості значень мантиси та порядку для представлення нескінчених, невизначених та малих значень. Так мінус та плюс нескінченість подаються максимальним значенням порядку (377_8 для числа з одинарною точністю та 3777_8 для числа з подвійною точністю) та нульовою мантисою. Для представлення невизначеного значення також використовується максимальне значення порядку та ненульова мантиса. Невизначене значення називають "не числом" – Not a Number (NaN). "Не число" використовується, щоб представити значення, яке не є дійсним числом і часто використовується як індикатор помилки, наприклад, коли відбулося ділення 0 на 0. Якщо число є дуже малим, то воно представляється нульовим порядком та ненульовою мантисою. У табл. 2.7 наведено приклади представлення різних величин в форматі за стандартом IEEE-754 для чисел з одинарною та подвійною точністю.

Таблиця 2.7

Приклади	32-розрядні числа			64-розрядні числа		
	Знак	Порядок	Мантиса	Знак	Порядок	Мантиса
NaN	?	377_8	Не нуль	0	3777_8	Не нуль
+ нескінченість	0	377_8	$.00000000_8$	0	3777_8	$.000000000000000000_8$
10,0	0	202_8	$(1).20000000_8$	0	2002_8	$(1).040000000000000000_8$
1,0	0	177_8	$(1).00000000_8$	0	1777_8	$(1).000000000000000000_8$
0,0	0	000_8	$.00000000_8$	0	0000_8	$.000000000000000000_8$
Ненормалізоване	0	000_8	Не нуль	0	0000_8	Не нуль
-0,0	1	000_8	$.00000000_8$	0	0000_8	$.000000000000000000_8$
-1,0	1	177_8	$(1).00000000_8$	1	1777_8	$(1).000000000000000000_8$
-10,0	1	202_8	$(1).20000000_8$	1	2002_8	$(1).040000000000000000_8$
- нескінченість	1	377_8	$.00000000_8$	1	3777_8	$.000000000000000000_8$

Тут знаком ? позначено несуттєве значення, а (1) – значення, яке не зафіксовується елементами пам'яті комп'ютера.

2.6.5. Кодування алфавітно-цифрової інформації

2.6.5.1. Двійково-кодовані десяткові числа

Вище було показано представлення в комп'ютері даних у двійковій системі числення. Далі розглянемо, як ці внутрішні дані можуть бути перетворені у форму, яка піддається інтерпретації людиною.

Двійково-кодоване десяткове число – це десяткове число, кожна цифра якого представлена в двійковій формі. Одна з перших числова система кодування десяткових чисел двійковим кодом (Binary-coded decimal – BCD) була використана в великих і середнього розміру комп'ютерних системах фірми IBM. Система BCD кодує кожну цифру десяткового числа 4-розрядним двійковим кодом. Коли використовується 8-розрядне число, тобто байт, то старші 4 біти називають зоною, а молодші – цифрою. Ця домовленість прийшла з часів перфокарт, де кожна колонка карти могла мати "зональний отвір" в одній з двох верхніх стрічок і "цифровий отвір" в одній з десяти нижніх стрічок. Старші чотири розряди в байті BCD використовуються для представлення знаку, який може мати одне з трьох значень: число без знаку представляється кодом 1111; додатне число представляється кодом 1100; від'ємне число представляється кодом 1101. Кодування для двійково-кодованих десяткових чисел показане в табл. 2.8.

Таблиця 2.8

Цифра	Код BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
Зони	
1111	Без знаку
1100	Додатне
1101	Від'ємне

Як видно з таблиці, шість можливих двійкових значень (від 1010 до 1111) не використовуються. Хоча втрачається приблизно 40 % можливих значень, але набуваються значні переваги в точності. Наприклад, десяткове число 0.3, перетворене в двійковий код та обмежене 8-розрядною сіткою, при зворотному перетворенні має значення 0.296875, тобто похибка складає приблизно 1.05 %. В коді BCD число запам'ятається безпосередньо як 1111 0011, не допускаючи жодної помилки.

Цифри в коді BCD займають лише чотири розряди, тому можна зберегти місце і спростити обчислення, розмістивши поряд числа з одним знаком. Цей процес називається пакуванням, а сформовані числа – пакованими десятковими числами.

Приклад:

Подамо число –1265 трьома байтами, використовуючи паковані цифри коду BCD.

Зонне десяткове кодування для числа 1265 є наступним:

1111 0001 1111 0010 1111 0110 1111 0101

Після пакування отримаємо:

0001 0010 0110 0101

Додавши знак після цифри молодшого розряду і заповнивши цифру старшого розряду одиницями до 3 байтів, отримаємо:

1111 0001 0010 0110 0101 1101.

Код BCD (або його іще називають кодом 8421) знайшов найбільше поширення в обчислювальній техніці. Цей код зручний для виконання перетворення з десяткової системи у двійкову і навпаки. Цей код адитивний, тобто сума представлення двох цифр є кодом їх суми.

Разом з тим, використання цього коду пов'язане з труднощами пошуку переносу в наступний десятковий розряд і важкістю переходу до оберненого та доповняльного коду для десяткових чисел. Це пояснюється тим, що код 8421 не є самодоповнюючим, тобто інверсія його двійкових цифр не дає коду доповнення десяткової цифри до 9. В табл. 2.9 наведено інші широко вживані двійково-десяткові коди, а саме код з надлишком 3 та код 2 з 5. Можна побудувати й інші двійково-десяткові коди, наприклад 2421, 5121 тощо.

Таблиця 2.9

Десяткові цифри	Код з надлишком 3	Код 2 з 5
0	0011	11 000
1	0100	00 011
2	0101	00 101
3	0110	00 110
4	0111	01 001
5	1000	01 010
6	1001	01 100
7	1010	10 001
8	1011	01 001
9	1100	10 100

Код з надлишком 3 зручний при виконанні арифметичних операцій над десятковими цифрами, оскільки він є самодоповнюючим. Крім того, легко визначається перенос, так як сума двох доданків, кожне з яких береться з надлишком 3, вийде з надлишком 6, що виключає лишні кодові комбінації. Для отримання правильного коду суми з отриманого результату відкидається 3. У деяких випадках для використання суттєво, що код нуля містить 1 і тому легко відрізнити наявність коду нуля від пропадання коду цифри. Код з надлишком 3 не дуже зручний для перетворення чисел з однієї системи числення в іншу. В коді 2 з 5 десяткові цифри зображаються п'ятьма розрядами, причому кожне значення містить дві 1. Ця надлишковість використовується для контролю правильності передачі цифри. Будь-яка помилка в одному розряді перетворює 0 в 1 або 1 в 0, в результаті вийде більше або менше двох 1, що вкаже на помилку. При одночасній появі двох помилок можливі випадки, коли їх можна не знайти (якщо 0 в одному розряді перетворюється в 1, а в іншому розряді 1 в 0).

2.6.4.2. Розширений двійково-кодований десятковий код обміну EBCDIC

Перед розробкою комп'ютерної системи IBM System/360, фірма IBM використала 6-розрядну версію двійково-кодованого десяткового числа для представлення символів і знаків. Цей код мав суттєві обмеження, пов'язані з малою розрядністю двійково-кодованого десяткового числа. Проектувальникам System/360 була потрібна більша інформаційна здатність коду так само як і узагальнений метод запам'ятовування і чисел, і символів. Для того, щоб підтримувати сумісність з попередніми комп'ютерами і периферійним устаткуванням, інженери IBM вирішили, що буде краще просто розширити код BCD від 6 бітів до 8 бітів. Відповідно, цей новий код було названо розширеним двійково-кодованим десятковим кодом обміну (EBCDIC). Фірма IBM і тепер продовжує використовувати код EBCDIC в мейнфреймах і обчислювальних системах середнього розміру. В табл. 2.10 код EBCDIC показано в зонально-цифровій формі.

Таблиця 2.10

Цифра

Зона	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	.F
0000	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI
0001	DLE	DC1	DC2	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
0010	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
0011			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
0100	SP										¢	.	<	(	+	|
0101	&										!	$	*	)	;	¬
0110	—	/										,	%	_	>	?
0111											:	#	@	'	=	"
1000		a	b	c	d	e	f	g	h	i						
1001		j	k	l	m	n	o	p	q	r						
1010			s	t	u	v	w	x	y	z						
1011										`						
1100		A	B	C	D	E	F	G	H	I						
1101		J	K	L	M	N	O	P	Q	R						
1110			S	T	U	V	W	X	Y	Z						
1111	0	1	2	3	4	5	6	7	8	9						

Знаки представлені шляхом додавання бітів до зонних бітів. Наприклад, знаку a відповідає код 1000 0001, а цифрі 3 – код 1111 0011. Зауважимо, що єдина різниця між верхніми і нижніми символами полягає в позиції розряду 2, що дозволяє зробити перетворення від верхніх до нижніх символів (або навпаки) шляхом переключення одного двійкового розряду. Зональні біти також роблять легшою для програміста перевірку правильності вхідних даних. Розшифрування абревіатур з табл. 2. 10 наведено в табл. 2.11.

Таблиця 2.11

NULL	Null	TM	Tape mark	ETB	End of transmission block
SOH	Start of heading	RES	Restore	ESC	Escape
STX	Start of text	NL	New line	SM	Set mode
ETX	End of test	BS	Backspace	CU2	Customer use 2
PF	Punch off	IL	Idle	ENQ	Enquiry
HT	Horizontal tab	CAN	Cancel	ACK	Acknowledge
LC	Lowercase	EM	End of medium	BEL	Ring the bell (beep)
DEL	Delete	CC	Cursor Control	SYN	Synchronous idle
RLF	Reverse if needed	CU1	Customer use 1	PN	Punch on
SMM	Start manual message	IFS	Interchange file separator	RS	Record separator
VT	Vertical tab	IGS	Interchange group separator	UC	Uppercase
FF	Form feed	IRS	Interchange record separator	EOT	End of transmission
CR	Carriage return	IGS	Interchange unit separator	CU3	Customer use 3
SO	Shift out	DS	Digit select	DC4	Device control 4
SI	Shift in	SOS	Start of significance	NAK	Negative acknowledgement
DLE	Data link escape	FS	Field separator	SUB	Substitute
DC1	Device control 1	BYP	Bypass	SP	Space
DC2	Device control 2	LF	Line feed		

2.6.4.3 Американський стандартний код інформаційного обміну ASCII

Американський стандартний код інформаційного обміну (American Standard Code for Information Interchange (ASCII)) з'явився завдяки зусиллям розробників покращити засоби передачі даних між системами. Міжнародна організація стандартизації ISO запропонувала цей 7-розрядний код взамін 5-розрядного коду, який використовувався в телетайпах.

Код ASCII визначає коди для 32 символів керування, 10 цифр, 52 букв, 32 спеціальних символів (таких як $ та #), а також для символу інтервалу (табл. 2.12). Старший восьмий біт було введено для забезпечення перевірки на парність. Цей біт дозволяє виявляти однократні помилки при передачі даних. Тут значення записані в десятковій системі числення.

Таблиця 2.12

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
010		!	"	#	$	%	&	'	(	)	*	+	,	—	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Розшифрування абревіатур з табл. 2.12 наведено в табл. 2.13.

Таблиця 2.13

Абревіатура	Розшифрування англійською	Розшифрування українською
NUL	NULL	Пусто
SOH	START OF HEADING	Початок заголовка
STX	START OF TEXT	Початок тексту
ETX	END OF TEXT	Кінець тексту
EOT	END OF TRANSMISSION	Кінець передачі
ENQ	ENQUIRY	Хто там?
ACK	ACKNOWLEDGE	Підтвердження
BEL	BELL	Дзвінок
BS	BACKSPACE	Повернення на крок
TAB	HORIZONTAL TABULATION	Горизонтальна табуляція
LF	LINE FEED	Подача нової стрічки
VT	VERTICAL TABULATION	Вертикальна табуляція
FF	FORM FEED	Подача нової форми
CR	CARRIAGE RETURN	Повернення каретки
SO	SHIFT OUT	Вихід
SI	SHIFT IN	Вхід
DLE	DATA LINK ESCAPE	Авторегістр 1
DC1	DEVICE CONTROL ONE	Контроль пристрою 1
DC2	DEVICE CONTROL TWO	Контроль пристрою 2
DC3	DEVICE CONTROL THREE	Контроль пристрою 3
DC4	DEVICE CONTROL FOUR	Контроль пристрою 4
NAK	NEGATIVE ACKNOWLEDGE	Ні
SYN	SYNCHRONOUS IDLE	Синхронізація
ETB	END OF TRANSMISSION BLOCK	Кінець блока
CAN	CANCEL	Анулювання
EM	END OF MEDIUM	Кінець носія
SUB	SUBSTITUTE	Заміна
ESC	ESCAPE	Авторегістр 2
FS	FILE SEPARATOR	Розділювач файлів
RS	RECORD SEPARATOR	Розділювач записів
US	UNIT SEPARATOR	Розділювач елементів
DEL	DELETE	Видалення

З підвищенням надійності комп'ютерної техніки важливість біта парності знизилась, тому на початку 80-х років його стали використовувати для розширення набору кодованих символів в межах від 128_{10} до 255_{10}. Це можуть бути, наприклад, математичні символи, або символи іноземних мов.

2.6.4.4. Стандарт кодування символів Unicode

Коди EBCDIC та ASCII забезпечили кодування букв латинського алфавіту. З метою забезпечення кодування букв інших алфавітів та підтримки мов народів світу в 1991 році було запропоновано код під назвою Unicode.

Unicode – це 16-розрядний алфавіт, сумісний з ASCII та погоджений з міжнародним алфавітом ISO/IEC 10646-1. Оскільки 16-ма розрядами можна закодувати 64K символів, цього достатньо для кодування всіх букв алфавітів народів світу.

Кодовий простір коду Unicode вміщує 5 частин, як це показано в табл. 2.14.

Таблиця 2.14

Тип символу	Опис набору символів	Кількість символів	Шістнадцяткові значення символів
Алфавіти	Латинський, кирилиця, грецький і т.д.	8192	Від 0000 до 1FFF
Символи	Графічні мітки, математичні символи і т.д.	4096	Від 2000 до 2FFF
CJK	Китайські, японські і корейські фонетичні символи і пунктуації	4096	Від 3000 до 3FFF
Han	Уніфіковані китайські, японські і корейські	40960	Від 4000 до DFFF
	Розширення чи надлишок від Han	4096	Від E000 до EFFF
Вказані користувачем		4096	Від F000 до FFFF

2.7. Короткий зміст розділу

У цьому розділі було показано, що в зв'язку з використанням у комп'ютерах елементів з двома станами, всі числа в них представляються у двійковій системі числення. Тому були висвітлені основні питання відображення чисел і символів у цій системі, а також її зв'язок з вісімковою, шістнадцятковою, та десятковою системами.

Було розглянуто особливості представлення чисел зі знаком спеціальними кодами: прямим, оберненим та доповнявльним, які використовуються для спрощення виконання арифметичних операцій.

. Оскільки числові дані в комп'ютері представляються у три способи: як цілі або дробові числа з фіксованою комою, як числа з рухомою комою та як двійково-кодовані десяткові числа, в цьому розділі було описано формати даних з фіксованою та з рухомою комою, включаючи стандарт IEEE-754, та було детально проаналізовано характеристики цих форматів.

2.8. Література для подальшого читання

Еволюція сисем числення показана в роботі [1]. У роботах [2-5] описано позиційні системи числення та представлення даних у двійковому, вісімковому та шістнадцятковому кодах, показано переведення чисел із системи числення з основою k до десяткової, та переведення чисел із десяткової до системи числення з основою k, а також описано представлення чисел із знаком в прямому, оберненому та доповнявльному кодах. Особливості представлення чисел в форматі з рухомою комою, включаючи і стандарт IEEE-754, подано в роботі [6]. Детальна інформація про Unicode може бути знайдена на сторінці Unicode Consortium www.unicode.org, так само як в описі стандарту *The Unicode Standard, Version 3.0* (2000). Сторінка International Standards Organization ISO може бути знайдена за адресою www. iso.ch. Багато інформації про стандарти є також на сторінці American National Standards Institute www.ansi.prg.

2.9. Література до розділу 2

1. Knuth, Donald E. The Art of Computer Programming,3rd ed. Reading, MA: Addison-Wesley, 1998.

2. Карцев М. А. Арифметика цифровых машин. – М.: Наука, 1969.

3. Каган Б. М. Электронные вычислительные машины и системы. – М.: Энергия, 1979.

4. Рабинович З. Л., Раманаускас В. А. Типовые операции в вычислительных машинах. – К.: Техніка, 1980. – 308 с.

5. Корнейчук В. И., Тарасенко В. П. Основы компьютерной арифметики. – К. Корнейчук, 2002. – 176 с.

6. Goldberg, David. "What Every Computer Scientist Should Know About Floating-Point Arithmetic." ACM Computing Surveys 23:1 March 1991. pp. 5–47.

2.10. Питання до розділу 2

1. Що таке система числення?
2. Що таке позиційна система числення?
3. Що таке основа системи числення?
4. Чому двійкові і десяткові системи числення названі позиційними?
5. Запишіть довільне число в позиційній системі числення та його кількісний еквівалент.
6. Поясніть зв'язок вісімкової та шістнадцяткової систем числення з двійковою системою числення.
7. Поясніть правило переведення чисел із системи числення з основою k до десяткової.
8. Поясніть правило переведення чисел із десяткової системи числення до системи числення з основою k.
9. Що означають слова біт, байт, слово?
10. Назвіть три способи представлення двійкових чисел із знаком в комп'ютерах, і поясніть їх відмінності.
11. Поясніть суть оберненого коду представлення двійкових чисел із знаком.
12. Поясніть суть доповняльного коду представлення двійкових чисел із знаком.
13. Поясніть суть прямого коду представлення двійкових чисел із знаком.
14. Яке саме з трьох цілочисельних представлень використовується частіше всього в комп'ютері?
15. Який діапазон представлення двійкових чисел із знаком в прямому, оберненому та доповняльному кодах?
16. Що таке переповнення, і як воно може бути виявлене?
17. Як переповнення для чисел без знаку відрізняється від переповнення для чисел із знаком?
18. Назвіть три способи представлення числових даних в комп'ютері.
19. Що означає представлення числа з одинарною та з подвійною точністю?
20. Де розміщується кома при представленні чисел з фіксованою комою?
21. На якій позиції розміщується кома при представленні цілих чисел?

22. На якій позиції розміщується кома при представленні дробових чисел?
23. Яке найбільше ціле додатне число може бути представлене в n-розрядній сітці?
24. Яке найбільше дробове додатне число може бути представлене в n-розрядній сітці?
25. Яке найменше ціле додатне число може бути представлене в n-розрядній сітці?
26. Яке найменше дробове додатне число може бути представлене в n-розрядній сітці?
27. Яке найбільше ціле від'ємне число може бути представлене в n-розрядній сітці?
28. Яке найбільше дробове від'ємне число може бути представлене в n-розрядній сітці?
29. Яке найменше ціле від'ємне число може бути представлене в n-розрядній сітці?
30. Яке найменше дробове від'ємне число може бути представлене в n-розрядній сітці?
31. Які є три складові частини чисел з рухомою комою?
32. Що таке зміщений порядок і яка мета його застосування?
33. Яка перевага використання зміщення взамін знакового біта в порядку?
34. Які найбільші та найменші додатні і від'ємні числа можуть бути представлені в форматі IEEE-754?
35. Чому використовується представлення чисел з рухомою комою в нормалізованій формі?
36. Що таке нормалізація?
37. Чому упускається одиниця старшого розряду нормалізованої мантиси при зберіганні числа з рухомою комою?
38. Скільки розрядів має число з рухомою комою в форматі IEEE-754 з одинарною точністю?
39. Скільки розрядів має число з рухомою комою в форматі IEEE-754 з подвійною точністю?
40. Які є переваги та недоліки використання відмінної від 2 основи порядку для чисел з рухомою комою?
41. Назвіть особливості виконання операцій над числами з рухомою комою.
42. Коли використовується представлення чисел з поблоково-рухомою комою?
43. В яких випадках використовується формат представлення чисел з рухомою-рухомою комою?
44. Поясніть суть кодування чисел кодом BCD.
45. Приведіть двійково-десятковий код з 2 з 5 та назвіть вигоди від його використання.
46. Приведіть двійково-десятковий код з надлишком 3 та назвіть вигоди від його використання.
47. Поясніть суть кодування чисел кодом EBCDIC.
48. Що таке код ASCII і чим він відрізняється від коду BCD?
49. Скільки розрядів використовується в коді Unicode для представлення символу?
50. Чому був запроваджений стандарт кодування символів Unicode?

2.11. Задачі до розділу 2

1. Запишіть довільне число в двійковій системі числення та його кількісний еквівалент.
2. Запишіть довільне число в вісімковій системі числення та його кількісний еквівалент.
3. Запишіть довільне число в шістнадцятковій системі числення та його кількісний еквівалент.
4. Виконати наступні перетворення, використовуючи віднімання або ділення з остачею:

a. $658_{10} = \_\_\_\_\_\_\_\__3$

b. $477_{10} = \_\_\_\_\_\_\_\__5$

c. $518_{10} = \_\_\_\_\_\_\_\__7$

d. $5401_{10} = \_\_\_\_\_\_\_\__9$

5. Виконати наступні перетворення, використовуючи віднімання або ділення з остачею:

a. $488_{10} = \_\_\_\_\_\_\_\__3$

b. $5254_{10} = \_\_\_\_\_\_\_\__5$
c. $752_{10} = \_\_\_\_\_\_\_\__7$
d. $6104_{10} = \_\_\_\_\_\_\_\__9$

6. Перевести наступні десяткові дробові числа в двійкові, обмежившись шістьма розрядами справа після коми:
   a. 36.78125
   b. 294.03125
   c. 498.796875
   d. 26.1240234375
7. Перевести наступні десяткові дробові числа в двійкові, обмежившись шістьма розрядами справа після коми:
   a. 35.84375
   b. 47.55
   c. 50.90625
   d. 74.874023
8. Представити наступні десяткові числа 8-розрядними двійковими із знаком в оберненому і доповняльному кодах:
   a. 67
   b. 52
   c. 219
   d. 207
9. Використовуючи 3-розрядні слова, назвати всі можливі двійкові числа із знаком та їх десяткові еквіваленти при їх представленні в:
   a. Прямому коді
   b. Оберненому коді
   c. Доповняльному коді
10. Використовуючи 4-розрядні слова, назвати всі можливі двійкові числа із знаком та їх десяткові еквіваленти при їх представленні в:
    a. Прямому коді
    b. Оберненому коді
    c. Доповняльному коді
11. Визначити значення двійкового числа із знаком 1001, представленого в оберненому коді, в десятковій системі.
12. Для числа з рухомою комою, яке має 3-бітовий порядок і 5-бітову мантису із знаком знайти:
    a. Найбільше додатне та найменше від'ємне нормалізовані числа.
    b. Зміщення порядку, при якому всі значення порядку є від'ємними.
13. Використовуючи 14-розрядний формат числа, причому 5 розрядів – для представлення порядку, 8 розрядів – для представлення нормалізованої мантиси та один для її знаку:
    a. Показати, як в цьому форматі будуть представлені числа 100.0 та 0.25.
    b. Додати ці два числа в названому форматі.
14. Які найбільші та найменші додатні і від'ємні числа можуть бути представлені в форматі IEEE-754?
15. Записати в десятковій системі 32-розрядне число F0ABCD78, представлене в форматі IEEE з рухомою комою. Тут для запису числа використана шістнадцяткова система.
16. Показати, як в форматі IEEE-754 з рухомою комою можуть бути представлені наступні числа: +1; -1; $356*2^{33}$.
17. Записати число N = -1 в форматі IEEE-754.
18. Записати число N = -1.5 в форматі IEEE-754.

19. Описати діапазони представлення чисел при використанні форматів з фіксованою і рухомою комою

20. Вибрати формат представлення даних який забезпечує точність до 8 десяткової цифри та динамічний діапазон від 10^{-25} до 10^{25}. Тут число в дужках показник степеня

21. Приведіть діапазони представлення чисел при використанні відмінної від 2 основи порядку для формату з рухомою комою

22. Показати як в форматі IEEE-754 з рухомою комою можуть бути представлені наступні числа: +1; -1; $356*2^{33}$

23. Вибрати формат представлення даних, який забезпечує точність до 12 десяткової цифри та динамічний діапазон від 10^{-22} до 10^{22}. Тут число в дужках показник степеня

24. Декодувати наступний лист в 7-розрядному ASCII коді без біта парності: 1001010 1001111 1001000 1001110 0100000 1000100 1000101