

## ЛЕКЦІЯ 2 Арифметичні основи ЕОМ

### 1. Арифметичні операції в двійковій системі числення

Всі арифметичні операції в двійковій системі числення проводяться відповідно до відомих правил виконання арифметичних операцій в загальноприйнятій десятковій системі числення, але при цьому використовуються таблиці додавання і множення, складені для двійкової системи числення.

При складанні таблиці додавання в будь-якій системі числення можна користуватися таким правилом: якщо при підсумовуванні двох цифр  $a_i$  і  $a_j$

$$a_i + a_j < P, \quad \text{то} \quad a_i + a_j = a_k,$$

де  $a_k$  – цифра даної системи числення;  $P$  – основа системи числення, якщо  $a_i + a_j \geq P$ , то  $a_k = a_i + a_j - P$  і з'являється одиниця перенесення в наступний старший розряд. Правило складання таблиці додавання можна записати в наступному вигляді:

$$a_i + a_j = \begin{cases} a_k, & \text{якщо} \quad a_i + a_j < P; \\ 1a_k, & \text{якщо} \quad a_i + a_j \geq P, \end{cases}$$

де  $a_k = a_i + a_j - P$ .

Слід пам'ятати, що в двійковій системі числення кожен старший розряд містить дві одиниці сусіднього молодшого розряду.

Таблиця двійкового додавання	Таблиця двійкового віднімання	Таблиця двійкового множення
$0 + 0 = 0$	$0 - 0 = 0$	$0 \times 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 \times 1 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$1 \times 0 = 0$
$1 + 1 = 10$	$10 - 1 = 1$	$1 \times 1 = 1$
↑	↑	
одиниця перенесення в сусідній старший розряд	одиниця позички з сусіднього старшого розряду	

Додавання чисел в двійковій системі можна виконувати стовпчиком, починаючи з молодшого розряду. У кожному розряді у відповідності з правилами, зазначеними таблицею двійкового додавання, проводиться складання двох чисел доданків і одиниці перенесення з сусіднього молодшого розряду (якщо вона є). В результаті виходить цифра відповідного розряду суми, і, можливо, одиниця перенесення в сусідній старший розряд.

Приклад. Скласти число  $A_{(2)} = 1100101$  і число  $B_{(2)} = 1010011$  в двійковій системі числення.

Розв'язання:

одиниці перенесення

$$\begin{array}{r} \phantom{A+B=} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\ A = \overset{1}{\downarrow} 1 \ 1 \ 0 \ 0 \ \overset{1}{\uparrow} \ \overset{1}{\uparrow} \ \overset{1}{\uparrow} \\ + \phantom{A+B=} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\ B = 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline A+B = 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \end{array}$$

Відповідь:  $A + B = 10111000_{(2)}$ .

При відніманні чисел в двійковій системі числення здійснюється зайняття одиниці з сусіднього старшого розряду кожен раз, коли цифра в розряді від'ємника більше цифри в тому ж розряді зменшуваного. Ця займана одиниця дорівнює двом одиницям даного розряду.

Приклад. Провести операцію віднімання числа  $B_{(2)} = 1010101$  з числа  $A_{(2)} = 1101011$  в двійковій системі числення.

Розв'язання:

одиниці позички

$$\begin{array}{r} \phantom{A-B=} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\ A = 1 \ 1 \ \overset{1}{\downarrow} \ 0 \ \overset{1}{\downarrow} \ 1 \ 0 \ 1 \ 1 \\ - \phantom{A-B=} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\ B = 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \hline A - B = 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \end{array}$$

Відповідь:  $A - B = 10110_{(2)}$ .

При множенні двійкових чисел так само, як і при множенні десяткових, спочатку отримують часткові добутки, після чого знаходять їх суму. Відповідно до таблиці двійкового множення кожний частковий добуток дорівнює нулю, якщо у відповідному розряді множника стоїть нуль, або одне множене, зсунуте на відповідне число розрядів вліво, якщо в розряді множника стоїть одиниця.

Таким чином, операція множення багаторозрядних двійкових чисел зводиться до операцій зсуву і додавання. Комою відділяється кількість розрядів добутку, яке дорівнює сумі розрядів дробових частин співмножників.

Ділення чисел в двійковій системі числення проводиться аналогічно діленню десяткових чисел.

Операції множення та ділення виконуються в комп'ютері за спеціальними алгоритмами, які будуть розглянуті далі.

Завдяки простоті правил двійковій арифметики застосування в комп'ютерах двійкової системи числення дозволяє істотно спростити схеми арифметичних пристроїв.

## **2. Способи кодування двійкових чисел**

Під час виконання арифметичних операцій в арифметико-логічних пристроях необхідно враховувати можливість обробки як додатних, так і від'ємних чисел, а також виникнення переповнення розрядної сітки. Ця задача розв'язується застосуванням спеціальних кодів, за допомогою яких операція віднімання зводиться до арифметичного додавання, що призводить до спрощення арифметичних пристроїв комп'ютера. Взагалі для подання

двійкових чисел в комп'ютерах застосовують прямий, зворотний і доповняльний коди.

В усіх цих кодах додатні числа мають один і той же вигляд, а від'ємні – різний. Двійкові коди чисел будемо записувати в квадратних дужках. Розглянемо ці коди.

### 2.1. Прямий код двійкових чисел

В прямому коді число подається у вигляді абсолютного значення числа з кодом відповідного знаку.

Правило формування прямого коду може бути записане у вигляді

$$[A]_{\text{п}} = \begin{cases} A, & \text{якщо } A \geq 0; \\ 1 + |A|, & \text{якщо } A \leq 0. \end{cases} \quad (1)$$

При переході від двійкового зображення числа до зображення його в прямому коді зміст мантиси збігається з дробовою частиною двійкового числа.

Знак додатного числа в прямому коді зображується цифрою **0**, а від'ємного – **1**. Знаки відділяються від мантиси крапкою.

Приклад. Подати число  $A_{(2)} = 0,101101$  і  $B_{(2)} = -0,110111$  в прямому коді.

$$\begin{aligned} A_{(2)} = 0,101101 &\rightarrow [A]_{\text{п}} = 0.101101; \\ B_{(2)} = -0,110111 &\rightarrow [B]_{\text{п}} = 1.110111. \end{aligned}$$

З формули (2.12) видно, що нуль в прямому коді має два зображення:

$$\begin{aligned} A_{(2)} = +0,00\dots0 &\rightarrow [A]_{\text{п}} = 0.00\dots0; \\ A_{(2)} = -0,00\dots0 &\rightarrow [A]_{\text{п}} = 1.00\dots0. \end{aligned}$$

Це треба мати на увазі при порівнюванні чисел.

Додавання чисел, які подані в прямому коді і мають однакові знаки, виконується досить просто. Мантиси чисел додаються і сумі привласнюється код знаку доданків.

Приклад. Подати два числа  $A_{(2)} = -0,10010$  і  $B_{(2)} = -0,01011$  в прямому коді з перевіркою додавання в двійковій системі числення.

Розв'язання:

1 Спочатку проведемо додавання в двійковій системі числення.

$$\begin{array}{r} A_{(2)} = -0,10010 \\ + B_{(2)} = -0,01011 \\ \hline (A + B)_{(2)} = -0,11101 \end{array}$$

2. Проведемо додавання чисел  $A$  і  $B$  в прямому коді

$$\begin{array}{r} [A]_{\text{п}} = 1.10010 \\ + [B]_{\text{п}} = 1.01011 \\ \hline [A + B]_{\text{п}} = 1.11101 \end{array}$$

3. Перейдемо від прямого коду до двійкової системи числення, отримаємо

$$(A + B)_{(2)} = -0,11101$$

що підтверджує правильність виконання обчислень.

Операція додавання чисел з різними знаками (операція алгебраїчного додавання) є більш складною. В цьому випадку приходится визначати більше за модулем число, з мантиси більшого числа виконати віднімання мантиси меншого числа, а потім різниці привласнити знак більшого (за модулем) числа.

Таким чином, для реалізації операції віднімання (алгебраїчного додавання) чисел в прямому коді необхідне додаткове обладнання, що конструктивно ускладнює арифметико-логічний пристрій комп'ютера. Це є недоліком прямого коду.

В комп'ютерах прямий код взагалі застосовується для запису і зберігання чисел в пам'ятовуючому пристрої, при вводі інформації та виводі результатів, а також для виконання операцій множення і ділення, в ході яких можуть використовуватись абсолютні значення чисел [11].

Операцію віднімання в комп'ютерах часто замінюють операцією додавання чисел в спеціальних кодах. Такими кодами є зворотний і доповняльний.

## 2.2. Зворотний код двійкових чисел

В зворотному коді операція віднімання зводиться до операції простого арифметичного додавання. Зворотний код утворюється відповідно до формули

$$[A]_3 = \begin{cases} A, & \text{якщо } A \geq 0; \\ 10^{n+1} - 10^{-m} + A, & \text{якщо } A \leq 0, \end{cases} \quad (2)$$

де  $n$  – кількість розрядів у цілій частині числа;  
 $m$  – кількість розрядів дробової частини числа;  
 $10^{-m}$  – одиниця молодшого розряду числа  $A$ ;  
 $10$  – число 2 в двійковій системі числення.

Для чисел менших одиниці

$$[A]_3 = \begin{cases} A, & \text{якщо } A \geq 0; \\ 10 - 10^{-m} + A, & \text{якщо } A \leq 0. \end{cases} \quad (3)$$

В зворотному коді знаковий розряд розглядається як цифровий розряд цілої частини числа і при проведенні арифметичних операцій ним оперують як звичайним цифровим.

Зворотний код додатного числа збігається з його прямим кодом.

Зворотний код від'ємного числа утворюється заміною (інвертуванням) значущих цифр початкового числа на зворотні і встановленням в знаковий розряд одиниці.

Приклад. Подати числа  $A_{(2)} = 0,1011$  і  $B_{(2)} = -0,1011$  в зворотному коді.

Розв'язання:

$$[A]_3 = 0.1011, \quad [B]_3 = 10 - 0.1011 = 1.0100.$$

З формули (3) видно, що нуль в зворотному коді має два зображення:

$$A_{(2)} = +0,00\dots0 \rightarrow [A]_3 = 0.00\dots0;$$

$$A_{(2)} = -0,00\dots0 \rightarrow [A]_3 = 1.11\dots1.$$

Для переходу від зворотного коду до двійкового зображення числа необхідно замість знакового розряду записати мінус (якщо в знаковому розряді стоїть одиниця), а мантису числа інвертувати.

Приклад. Знайти двійкове зображення числа  $[A]_3 = 1.0100$ .

Відповідь:  $A_{(2)} = -0,1011$ .

Для від'ємних чисел перехід від прямого коду до зворотного здійснюється за таким правилом.

**ПРАВИЛО.** Для отримання зворотного коду від'ємного числа з прямого коду необхідно всі розряди мантиси про інвертувати, а в знаковому розряді залишити одиницю.

Це ж правило справедливе і при переведенні від'ємних чисел із зворотного коду в прямий.

При алгебраїчному додаванні чисел, поданих в зворотному коді, виконується арифметичне додавання цих кодів, включаючи розряди знаків, які при цьому розглядаються як звичайні цифрові розряди. При виникненні одиниці, що вийшла за знаковий розряд, вона додається до молодшого розряду суми кодів. Таке перенесення називається циклічним [11].

*Пояснимо це.* Згідно з (3) від'ємні числа в зворотному коді можуть бути подані таким чином:

$$[A]_3 = 10 - 10^{-m} - |A| \quad \text{і} \quad [B]_3 = 10 - 10^{-m} - |B|.$$

Знайдемо суму чисел і  $(-A)$   $(-B)$ .

$$[A]_3 + [B]_3 = (10 - 10^{-m} - |A|) + (10 - 10^{-m} - |B|) = \underbrace{10}_{\text{одиниця переносу із знакового розряду}} + (10 - 10^{-m} - 10^{-m} - |A + B|).$$

Вираз  $(10 - 10^{-m} - 10^{-m} - |A + B|)$  являє собою зменшену на одиницю молодшого розряду суму чисел  $A$  і  $B$ , яка подана в зворотному коді.

Тому, щоб отримати правильний результат суми, необхідно одиницю переносу із знакового розряду додати до молодшого розряду суми. Тоді сума чисел  $A$  і  $B$  буде виражена згідно з (3) і матиме вигляд:

$$[A + B]_3 = 10 - 10^{-m} - |A + B|.$$

Приклад. Додати два числа  $A_{(2)} = -0,011010$  і  $B_{(2)} = -0,100011$  в зворотному коді.

Розв'язання:

$$\begin{array}{r} [A]_3 = 1.100101 \\ + [B]_3 = 1.011100 \\ \hline \phantom{[A]_3 + [B]_3} \downarrow 1.000001 \\ \phantom{[A]_3 + [B]_3} \longleftarrow 1 \\ \hline [A + B]_3 = 1.000010 \end{array}$$

Відповідь:  $[A + B]_3 = 1.000010$

Приклад. Додати два числа  $A_{(2)} = -0,110110$  і  $B_{(2)} = 0,100101$  в зворотному коді. Результат подати в прямому коді.

Розв'язання:

$$\begin{array}{r} [A]_3 = 1.001001 \\ + [B]_3 = 1.100101 \\ \hline [A + B]_3 = 1.101110 \end{array}$$

Для отримання прямого коду необхідно інвертувати мантису.

Відповідь:  $[A + B]_n = 1.010001$ .

**Перевагою зворотного коду** є можливість звести операцію віднімання до операції додавання.

**Недоліки зворотного коду:**

– виникнення переповнення розрядної сітки в тому випадку, коли абсолютне значення суми більше одиниці, що призводить до зміни результатів обчислень. Це явище буде розглянуте нижче;

– виникнення циклічного переносу збільшує час додавання чисел і тим самим зменшує швидкість комп'ютерів. Цей недолік відсутній при додаванні чисел в доповняльному коді.

### 2.3. Доповняльний код двійкових чисел

Доповняльний код двійкових чисел утворюється відповідно до формули

$$[A]_d = \begin{cases} A, & \text{якщо } A \geq 0; \\ 10^{n+1} + A, & \text{якщо } A < 0, \end{cases}$$

де  $n$  – кількість розрядів у цілій частині числа;

$10$  – число 2 в двійковій системі числення.

Для чисел, менших одиниці,

$$[A]_d = \begin{cases} A, & \text{якщо } A \geq 0; \\ 10 + A, & \text{якщо } A < 0, \end{cases} \quad (4)$$

Доповняльний код додатного числа збігається з його прямим кодом.

Для отримання доповняльного коду від'ємного числа перетворимо вираз (4) таким чином:

$$[A]_d = 10 - |A| = 10 - 10^{-m} - |A| + 10^{-m} = [A]_3 + 10^{-m}. \quad (5)$$

Із виразу виходить що для отримання доповняльного коду від'ємного числа необхідно отримати його зворотний код, а потім до молодшого розряду числа, яке подане в зворотному коді, додати одиницю.

Приклад. Подати число  $A_{(2)} = -0,11010$  в доповняльному коді.

Розв'язання:

$$\begin{array}{l} [A]_3 = 1.00101 \\ [A]_d = 1.00110 \end{array}$$

Відповідь:  $[A]_d = 1.00110$

Якщо при виконанні арифметичних операцій в доповняльному коді виникає одиниця, яка вийшла за знаковий розряд, то вона відкидається.

*Пояснимо це.* Нехай треба додати два від'ємних числа  $A$  і  $B$  в доповняльному коді. Згідно з (4) доповняльний код від'ємних чисел буде мати наступний вигляд:

$$[A]_д = 10 - |A| \quad \text{і} \quad [B]_д = 10 - |B|.$$

Тоді

$$[A]_д + [B]_д = (10 - |A| + 10 - |B|) = \underbrace{10}_1 + (10 - |A + B|).$$

одиниця переносу відкидається

Вираз в дужках  $(10 - |A + B|)$  є сумою від'ємних чисел  $A$  і  $B$ , яка подана в доповняльному коді. Отже щоб отримати правильний результат суми в доповняльному коді, необхідно відкинути одиницю переносу із знакового розряду.

Відповідно до виразу (4) нуль в доповняльному коді має одне зображення:

$$\begin{aligned} [0]_д &= 0.00\dots 0, \\ [0]_д &= [0]_з + 10^{-m} = 1.11\dots 1 + 0.00\dots 1 = 10.00\dots 0 = 0.00\dots 0. \end{aligned}$$

$\downarrow$   
одиниця відкидається

Сформулюємо правило додавання чисел в доповняльному коді.

**ПРАВИЛО.** При алгебраїчному додаванні двійкових чисел, які подані в доповняльному коді, виконується додавання цих кодів, включаючи розряди знаків, які при цьому розглядаються як звичайні цифрові розряди. При виникненні переносу із знакового розряду одиниця переносу відкидається.

Сума отримується в прямому коді, якщо вона додатна, і в доповняльному коді, якщо вона від'ємна. Якщо сума від'ємна, то для отримання прямого коду необхідно взяти двійковий додаток від доповняльного коду [11].

*Приклад.* Додати два числа  $A_{(2)} = -0,10011$  і  $B_{(2)} = -0,01001$  в доповняльному коді. Результат подати в прямому коді.

*Розв'язання:*

$$\begin{array}{r} [A]_д = 1.01101 \\ + [B]_д = 1.10111 \\ \hline [A + B]_д = \underbrace{11.00100}_1 \end{array}$$

одиниця відкидається

$$[A + B]_п = [[A + B]_д]_д = 1.11100.$$

*Відповідь:*  $[A + B]_п = 1.11100.$

Таким чином, використання доповняльного коду дозволяє позбавитись від циклічної передачі одиниці переносу, що сприяє підвищенню швидкодії комп'ютера, але отримати доповняльні коди доданків важче, тому, що спочатку необхідно отримати зворотний код, а потім додати до нього одиницю молодшого розряду.



## 2.4. Модифіковані коди

В процесі виконання арифметичних операцій можливий варіант, коли модуль отриманого результату перевищує одиницю, тобто перевищує максимально допустиме число, яке може бути записане в розрядну сітку комп'ютера. Це явище, яке називається переповненням розрядної сітки, приводить до спотворення результатів. Покажемо це.

Приклад. Додати два числа  $A_{(2)} = -0,10101$  і  $B_{(2)} = -0,11001$  в зворотному і додатковому кодах.

Розв'язання:

Модуль суми цих чисел більший за одиницю, тобто  $|A + B| > 1$ .

Виконаємо додавання цих чисел:

$$\begin{array}{r}
 + [A]_3 = 1.01010 \\
 [B]_3 = 1.00110 \\
 \hline
 10.10000 \\
 \begin{array}{c} \downarrow \\ \rightarrow 1 \end{array} \\
 \hline
 [A + B]_3 = 0.10001
 \end{array}
 \qquad
 \begin{array}{r}
 [A]_4 = 1.01011 \\
 [B]_4 = 1.00111 \\
 \hline
 + \begin{array}{c} \downarrow \\ \rightarrow 1 \end{array} 10.10010 \\
 \hline
 \text{відкидається} \\
 [A + B]_4 = 0.10010
 \end{array}$$

Отриманий результат є невірним з двох позицій.

По-перше, в знаковому розряді стоїть нуль, який показує, що отримана сума додатна, а насправді результат повинен бути від'ємним.

По-друге, в цифрових розрядах числа також отримане невірне значення.

Це виходить через те, що результат додавання не вкладається в розрядну сітку комп'ютера, тобто спостерігається переповнення розрядної сітки. Таке явище відбувається кожного разу, коли абсолютне значення суми більше одиниці, тобто при  $|A + B| > 1$ .

Щоб результат обчислень вийшов правильним, необхідно виключити переповнення розрядної сітки.

Для цього використовують модифіковані коди.

Модифіковані коди відрізняються від розглянутих вище тим, що для зображення знака числа відводиться два розряди.

Якщо число додатне, то в знаковому розряді записується два нулі, якщо від'ємне – дві одиниці [11].

Приклад. Подати число  $A_{(2)} = -0,101011$  в прямому, зворотному і додатковому модифікованих кодах.

Розв'язання:

$$[A]_{\text{пр}}^M = 11.101011; \quad [A]_3^M = 11.010100; \quad [A]_4^M = 11.010101.$$

Алгебраїчне додавання двійкових чисел в модифікованих кодах виконується за такими ж правилами, як і в звичайних кодах, при цьому знакові розряди розглядаються як розряди цілої частини числа.

Приклад. Виконати додавання чисел  $A_{(2)} = -0,00101$  і  $B_{(2)} = -0,10011$  в зворотному і додатковому модифікованих кодах. Результат подати в прямому коді.



Розв'язання:

$$\begin{array}{r}
 [A]_3^M = 11.11010 \\
 [B]_3^M = 11.01100 \\
 \hline
 111.00110 \\
 \quad \underbrace{\quad\quad}_v \rightarrow 1 \\
 \hline
 [A + B]_3^M = 11.00111 \\
 [A + B]_п = 1.11000
 \end{array}
 \qquad
 \begin{array}{r}
 [A]_д^M = 11.11011 \\
 [B]_д^M = 11.01101 \\
 \hline
 111.01000 \\
 \quad \underbrace{\quad\quad}_v \text{ відкидається} \\
 \hline
 [A + B]_д^M = 11.01000 \\
 [A + B]_п = 1.11000
 \end{array}$$

Відповідь:  $[A + B]_п = 1.11000$

При використанні модифікованих кодів ознакою переповнення розрядної сітки є наявність в знакових розрядах різних цифр: 01 або 10. Якщо в старшому знаковому розряді суми стоїть 0 (комбінація 01), то це означає, що отриманий результат додатний. Якщо в старшому знаковому розряді суми стоїть 1 (комбінація 10), то це означає, що отриманий результат від'ємний.

У випадку переповнення розрядної сітки в комп'ютерах з фіксованою комою він зупиняється. Це свідчить про те, що масштабування виконане невірне. Для усунення переповнення розрядної сітки необхідно знову виконати масштабування.

Якщо здійснюється переповнення розрядної сітки в комп'ютері з плаваючою комою, то в цьому випадку комп'ютер здійснює так звану операцію нормалізації вправо на один розряд. В результаті виконання цієї операції зсувається вправо на один розряд код результату алгебраїчного додавання і додається одиниця до порядку числа. При цьому модуль числа залишається без змін.

Приклад. Додати два числа, записані в формі з плаваючою комою з використанням доповняльного модифікованого коду:

$$A_{(2)} = -0,110110 \cdot 10^{011} \text{ і } B_{(2)} = -0,001101 \cdot 10^{011} .$$

Розв'язання:

$$\begin{array}{r}
 [A]_д^M = 11.001010 \quad 0.011 \\
 + [B]_д^M = 11.110011 \quad 0.011 \\
 \hline
 110.111101 \quad 0.011 \\
 \hline
 [A + B]_д^M = 10.111101 \quad 0.011
 \end{array}$$

Видно, що відбулося переповнення розрядної сітки. Виконуємо зсув мантиси і її знаку на один розряд вправо і збільшення порядку на одиницю

$$[A + B]_д^M = 11.0111101 \quad 0.100.$$

Таким чином, модифіковані коди застосовуються для скорочення часу на визначення переповнення розрядної сітки.