2. ДВІЙКОВА АРИФМЕТИКА

Правила виконання арифметичних дій над двійковими числами задаються таблицями двійкового додавання, віднімання та множення див. табл. 2.1 [2].

Таблиця 2.1 – Таблиці арифметичних дій у двійковій системі числення

Таблиця двійкового додавання	**Таблиця двійкового віднімання**	**Таблиця двійкового множення**
0 + 0 = 0 **0 + 1 = 1** **1 + 0 = 1** **1 + 1 = 0 перенесення**	**0 - 0 = 0** **0 - 1 = 1 запозичення** **1 - 0 = 1** **1 - 1 = 0**	**0 × 0 = 0** **0 × 1 = 0** **1 × 0 = 0** **1 × 1 = 1**

2.1. Додавання цілих двійкових чисел.

При додаванні будь-яких, в тому числі й двійкових чисел, $A = a_n...a_1a_0$ та $B = b_m...b_1b_0$ в кожному розряді проводиться складання цифр доданків, причому якщо сума цифр a_i та b_i перевищує допустиме значення для цифри $a_i + b_i > p$-1, де p – 1-й елемент основи в даній ПСЧ, то відбувається перенесення 1 у наступний розряд. У двійковій ПСЧ це правило трансформується з урахуванням того, що тут $p=2^1=2$, а значить обмеження $a_i + b_i > p$-1 означає в $a_i + b_i > 1$. Крім того, в двійковій СПЧ 1 + 1 дає 0 в даному розряді та перенесення 1 в наступний розряд (див. таб. 2.1).

Приклади.

1) Додати два двійкові числа $x = 1101_2$ та $y = 101_2$:

```
  1 1
  1101   x
+  101   y
 10010   x + y
```

Результат $x + y = 1101_2 + 101_2 = 10010_2$.

2) Додати двійкові числа $x = 1101_2$, $y = 101_2$ та $z = 111$:

```
  1
  1 1 1
  1101   x
+  101   y
   111   z
 11001   x + y + z = 1101₂ + 101₂ + 111₂= 11001₂.
```

2.2. Віднімання цілих двійкових чисел

При відніманні двійкових чисел у кожному розряді проводиться віднімання цифр, причому, якщо в даному розряді значення цифри зменшуваного менше, ніж від'ємника $a_i < b_i$, то із старшого(их) розрядів запозичується (займається) 1. Ця запозичена 1 для даного розряду, очевидно, матиме кількісний еталон p в будь-якій системі числення (2 – у двйковій, 8 – у вісімковй, 10 – у десятковій, 16 – у шістнадцятковій ПСЧ).

Приклад.

Відняти від двійкового числа $x = 10010_2$ число $y = 101_2$:

```
  -1  -1
  10010   x
-   101   y
  01101   x − y
```

Результат $x - y = 10010_2 - 101_2 = 1101_2$.

2.3. Від'ємні числа та доповнювальний код

Поки мова йшла про числа без знаку. Це суттєво звужує можливості застосувань розглядуваних результатів, способів, методів. Адже т.з. цілі числа є основним базовим типом даних. Вихід є і базується він на тому, що множини цілих та натуральних чисел є зліченими, тобто існують бієкції $N \leftrightarrow N$ та $N \leftrightarrow Z$. Існування першої очевидне, кожне натуральне число одночасно є його номером. Друга легко може бути задана самими різними кодуваннями. Але враховуючи те, що для цифрової електроніки існує реальне обмеження в частині скінченності множин як натуральних, так і цілих, також раціональних, псевдодійсних та їм подібних чисел, то кодування може бути з урахуванням цього здійснене, наприклад, так: кодування додатніх чисел залишається без змін, а від'ємні числа кодуються як результат віднімання від 0 відповідного додатного числа, тобто, $-a=0-a$.

Цілі числа	Натуральні номери	Двійкові представлення
…	…	…
-8	16	1 000
-7	15	1 001
-6	14	1 010
-5	13	1 011
-4	12	1 100
-3	11	1 101
-2	10	1 110
-1	9	1 111
0	0	0 000
1	1	0 001
2	2	0 010
3	3	0 011
4	4	0 100
5	5	0 101
6	6	0 110
7	7	0 111
8	8	1 000
…	…	…

Легко помітити, що внесення знаку практично удвічі зменшує можливості представлення доступних абсолютних величин цілих чисел (див. таблицю). Виділяється т.з. знаковий розряд – знаходження там 0 означає, що число записане у менших регістрах (у таблиці їх 3) є додатнім, 1 – від'ємне. При цьому виникає казус. Наприклад, у даній таблиці маємо неоднозначне

трактування двійкового коду $1\ 000_2$ – як одночасно 8_{10} та -8_{10} . Але насправді ніяких казусів немає, якщо згадати, що 4-й регістр трактується як знаковий.

Важливо те, що при такому кодуванні зберігаються основні операції над числами. Наприклад, $1_{10}+(-1_{10}) = 0_{10}$, $1_{10}-(-1_{10}) = 2_{10}$ Теж саме у двійковій ПСЧ 0 001+1 111=0 000, а 0 001-1 111=0 010 і т.і.

Для того, щоб отримати від'ємне число, необхідно виконати віднімання від 0 його додатного аналогу, що не є досить зручним. Розглянемо спосіб, який дозволяє дещо спростити цю процедуру. Замість віднімання від 0 виконуватимемо віднімання від числа, яке представляє собою т.з. дуальний (від'ємний) нуль $1\ 000_2$, не забуваючи, що одиниця знаходиться за межами значущої розрядної сітки, тобто є «знаковою» ознакою. Очевидно, що така маніпуляція не впливатиме на результат віднімання, оскільки з цієї одиниці за необхідності братиметься запозичення, а якщо в цьому не буде потреби (тоді і тільки тоді, коли віднімається 0), вона просто ігноруватиметься у зв'язку з нашою попередньою домовленістю про конструктивну скінченність множини натуральних, цілих та ін. чисел. Це означає, що при потребі можна ігнорувати розряди за межами значущої розрядної сітки. Не складно переконатись, що якщо довжина розрядної сітки складає $n+1$, то наш «модифікований» нуль дорівнюватиме 2^n. Тоді від'ємне число $-A$ можна буде знайти як (див. також таб. вище): $E(2^n)_2 - A_2 = -A_2$. Далі, $2^n - A_2 = (2^n - 1) - A + 1$. Нескладно переконатись, що, наприклад, $E(2^4)_2 - 1_2 = 0\ 1111_2$ і узагалі, для будь-якого n: $2^n{}_{10} - 1_{10} \approx 1\ \underbrace{00...0_2}_{n} - 0\ \underbrace{00...01_2}_{n} = 0\ \underbrace{11...11_2}_{n}$. Також, нескладно переконатись, що $(E(2^n)_2 - 1_2) - A = \overline{A}$. Зокрема, нехай $A = 0\ 0011_2$. Тоді, враховуючи сказане про знаковий розряд і знаючи, що нулі у старших розрядах завжди можна відкинути через те, що вони не впливають на кількісний еталон двійкового числа, маємо:

$E((2^4 - 1))_2 - A = 0\ 1111_2 - 0\ 0011_2 = 1111_2 - 0011_2 = 1100_2 = 0\ 00...0100_2 = \overline{A},$
$E(\overline{A}) = E(1100_2) = E(0\ 00...01100_2) = 12$

. Тобто двійкові коди чисел A та $\overline{A}$ (у значущих регістрах! **Значущих, а не знакових**) є інверсними по відношенню один до одного. Таким чином, виходячи з того, що $2^n - A = -A$ приходимо до наступної простої процедури знаходження по додатному числу A відповідного йому від'ємного числа $-A$.

1. Інвертувати значущі розряди числа *A*.

2. До результату додати 1 у двійковому представленні, тобто виконати $\overline{A}+1$:

для $A = 0\ 011_2, E(A) = 3_{10}$ отримаємо, що $\overline{A} = 1\ 100_2$ та $\overline{A}+1 = 1100_2 + 0001_2 = 1101_2$.

$-A = 1\ 101_2, E(-A) = -3_{10}$ **(див. таб.)**

Приклад.

Знайти від'ємне значення числа 78_{10} у вигляді 8-розрядного двійкового числа:

$x = 78_{10} = 01001110_2$

1) Інвертування

$\overline{x} = 10110001_2$

2) Додавання одиниці

```
  10110001
+        1
  10110010 = -78₁₀ = −x
```

Перевірка:

$78_{10} + (-78_{10}) = 0$

```
  01001110
+ 10110010
1←00000000
```

Наведена вище процедура називається *доповненням до двох*, а результат, який при цьому утворюється – *доповнювальним кодом*. Пошук доповнювального коду числа еквівалентний обчисленню його від'ємного значення $A_{доп} = -A = \overline{a}_n \overline{a}_{n-1} ... \overline{a}_1 \overline{a}_0 + 1_2$.

Використання доповнювального коду дозволяє взагалі позбутися операції віднімання, так як останнє може бути замінене на додавання від'ємного числа.

Приклад. Відняти в двійковій системі числення два числа: $115_{10} - 47_{10}$ користуючись доповнювальним кодом.

$115_{10} - 47_{10} = 68_{10}$

$115_{10} = 01110011_2$

$47_{10} = 00101111_2$

Доповнювальний код:

```
¬ 00101111      інвертування
  11010000
+        1      додавання 1
  11010001 = -47₁₀
```

Пошук різниці шляхом додавання від'ємного числа $115_{10} - 47_{10}$:

```
  01110011
+ 11010001
1←01000100 = 68₁₀
```

Якщо необхідно змінити розрядність (збільшити довжину розрядної сітки) *беззнакових* та *додатних* двійкових чисел, то до них *зліва* можна дописати *необхідну кількість нулів*, а для від'ємних чисел – необхідно скопіювати

знаковий розряд, інакше кажучи дописати зліва необхідну кількість одиниць. Дана операція називається **знаковим розширенням**.

Приклад.

1) Представити число -7_{10} у 8-розрядному форматі, якщо відомо, що в 4-розрядному воно виглядає як 1001_2.

За правилом знакового розширення для 8-розрядної сітки число -7_{10} можна записати як: 11111001_2.

2) Перевести до шістнадцяткової системи числення число 10110_2, якщо відомо, що це від'ємне число:

Для запису шістнадцяткового числа необхідно утворити тетради – групи по 4 розряди, але оскільки число 10110_2 від'ємне, для утворення тетрад необхідно зліва до нього дописати потрібну кількість одиниць: $10110_2 = \underbrace{1111}_{F}\underbrace{0110}_{6} = F6_{16} = -10_{10}$

Для представлення від'ємних чисел застосовуються і інші формати. Наприклад, досить поширеним є т.з. прямий код. Його ідея полягає у тому, що двійкове число з n розрядами (бітами) використовує останній значущий розряд як знак, а інші $n - 1$ розряди представляють абсолютне значення даного числа. При цьому число вважається додатним, якщо знаковий розряд дорівнює 0 та є від'ємним, якщо він дорівнює 1.

Прямий код має деякі не завжди корисні особливості.

1. Додатній та від'ємний нулі - +0=00…0 та -0=10…0.
2. Виконання арифметичних операцій у ньому є більш складним, зокрема додавання додатного та від'ємного числа неможливо виконати природним чином.
   a. Наприклад, $5_{10} + (-3)_{10} = 2_{10}$, але $0101_2 + 1011_2 = 1 \leftarrow 0000_2 = 0_{10}$

Щоб операція виконувалась вірно необхідно в залежності від знакового розряду виконувати операцію віднімання або додавання. Це обумовлює необхідність мати пристрої додавання та віднімання та призводить до ускладнення апаратної частини, що є основним недоліком використання прямого коду.

2.4. Множення двійкових чисел

Множення двійкових чисел виконується за тими ж правилами, що й десяткове, за виключенням того, що дії здійснюються над нулями та одиницями за допомогою таблиць двійкового множення та додавання. Наприклад:

Множення двох додатних цілих 4-розрядних двійкових чисел.

```
   0101     множене
 × 1101     множник
   0101
+  0000
  0101     частинні добутки
 0101
-------
 1000001    результат
```

$0101_2 \times 1101_2 = 1000001_2$. В десятковій системі: $5_{10} \times 13_{10} = 65_{10}$.

В загальному випадку при множенні двох *n*-розрядних чисел формується 2*n*-розрядний результат.

Множення чисел зі знаком можна виконувати за наведеним вище правилом, але враховуючи розрядність результату.

Мається на увазі, що для отримання коректних частинних добутків, розрядність від'ємних множеного та/або множника *не повинна бути меншою* за розрядність результату.

Повернімося до наведеного вище прикладу: якщо числа в ньому вважати знаковими, то множник 1101_2 мав би бути від'ємним числом -3_{10} і при виконанні множення мав би бути отриманий результат -15_{10}, який у двійковій системі записувався як 10001_2 (7-розрядним числом 1110001_2), натомість було отримано 1000001_2, що є невірним. Якщо ж врахувати, що розрядність для представлення результату (числа -15_{10}) має складати щонайменше 5 розрядів, то множник слід було б записати як 11101_2 замість 1101_2. Тоді, обмежуючи розрядну сітку результату п'ятьма розрядами отримаємо:

```
  00101     множене
× 11101     множник
  00101
  0000
+ 101       частинні добутки
  01
  1
1←10001     результат
```

Тепер результат вірний.

Щоб не помилитися з вибором довжини розрядної сітки при обчисленні добутку *n*-розрядних знакових чисел у доповнювальному коді, можна прийняти розрядність результату рівною 2*n*.

2.5. Ділення двійкових чисел

Так само як і множення ділення двійкових чисел аналогічне діленню десяткових чисел. Найпростіший алгоритм ділення полягає у тому, що дільник на кожному кроці віднімається від діленого стільки разів (починаючи зі старших розрядів), скільки це можливо для отримання найменшого додатного залишку.

Кількість повторів дорівнює частці і показує, скільки разів дільник вкладається у ділене.

Наприклад, поділимо 35_{10} на 7_{10}: 1) 35 − 7 = 28; 2) 28 − 7 = 21; 3) 21 − 7 = 14; 4) 14 − 7 = 7; 5) 7 − 7 = 0. Отже відповідь є 5_{10}, оскільки число 7_{10} можна 5 разів відняти від числа 35_{10}. Очевидно, що така процедура є досить тривалою, тому її можна скоротити: визначити скільки разів дільник вкладається в старшу частину діленого, а потім послідовно переходити до його молодших

частин, враховуючи отримувані на кожному кроці остачі. Таку процедуру у двійковій системі числення реалізують два найбільш поширені алгоритми, які називаються діленням *з відновленням* і *без відновлення остачі*. Обидва ці алгоритми схожі, але алгоритм **без відновлення остачі** є більш коротким, а отже більш швидкодіючим, тому зупинимось на ньому.

Крок алгоритму ділення без відновлення остачі має наступний вигляд:

$$X - Y = a_0,$$

де X – ділене; Y – дільник, зсунутий до найбільшого значущого розряду діленого; a_0 – залишок.

Віднімання може бути замінене на операцію додавання доповнювального коду $Y_{\text{доп}}$.

Для дільника має бути використано найменшу кількість розрядів, необхідну для його представлення. Якщо $a_0 \geq 0$, то старший розряд частки $c_n = 1$, інакше – старший розряд частки $c_n = 0$.

Для визначення наступної цифри частки c_{n-1} необхідно знайти наступний залишок a_1, що визначається так:

$$a_1 = \begin{cases} 2a_0 - Y, & \text{якщо } a_0 \geq 0, \\ 2a_0 + Y, & \text{якщо } a_0 < 0. \end{cases}$$

Якщо $a_1 \geq 0$, то наступний, менший розряд частки $c_{n-1} = 1$, інакше – наступний розряд частки $c_{n-1} = 0$.

Загалом, для визначення наступної цифри частки $c_{n-к}$ $(k \leq n)$ необхідно знайти наступний залишок a_k, що визначається так:

$$a_k = \begin{cases} 2 \times a_{k-1} - Y, a_{k-1} \geq 0 \\ 2 \times a_{k-1} + Y, a_{k-1} < 0 \end{cases}$$

Необхідно здійснити стільки ітерацій (зсувів), на скільки розрядність діленого більше за розрядність дільника.

Знак залишку визначає не тільки чергову цифру частки, але і характер наступної процедури:

додавання дільника до збільшеного вдвічі (зсунутого вліво) діленого, якщо залишок менший 0, та віднімання дільника із зсунутого залишку, якщо він до зсуву був більшим або рівним 0.

Приклад.

1) Поділити 35_{10} на 5_{10}, тобто 100011_2 на 101_2.

Доповнимо числа знаковим розрядом (відділено знаком « | »), а дільник також представимо у доповнювальному коді для уникнення безпосередньої операції віднімання:

$35_{10} = 0|100011_2$
$+5_{10} = 0|101_2$
$-5_{10} = 1|011_2$

Процедура ділення:

0|100011	ділене
+ 1|011000	дільник у доп. коді (віднімання)
1|111011	$a_0 < 0$, $C = 0_2$
← 1|110110	$2a_0$ (зсув вліво)
+ 0|101000	дільник у прямому коді (додавання)
0|011110	$a_1 > 0$, $C = 01_2$
← 0|111100	$2a_1$ (зсув вліво)
+ 1|011000	дільник у доп. коді (віднімання)
0|010100	$a_2 > 0$, $C = 011_2$
← 0|101000	$2a_2$ (зсув вліво)
+ 1|011000	дільник у доп. коді (віднімання)
0|000000	$a_3 = 0$, $C = 0111_2$

Отже: $100011_2 / 101_2 = 111_2$ ($35_{10} / 5_{10} = 7_{10}$)

2) Поділити 204_{10} на 12_{10}, тобто 11001100_2 на 1100_2.

Доповнимо числа знаковим розрядом (відділено знаком « | »), а дільник також представимо у доповнювальному коді для уникнення безпосередньої операції віднімання:

$204_{10} = 0|11001100_2$
$+12_{10} = 0|1100_2$
$-12_{10} = 1|0100_2$

Процедура ділення:

0|11001100	ділене
+ 1|01000000	дільник (віднімання)
0|00001100	$a_0 > 0$, $C = 1_2$
← 0|00011000	$2a_0$
+ 1|01000000	дільник (віднімання)
1|01011000	$a_1 < 0$, $C = 10_2$
← 0|10110000	$2a_1$
+ 0|11000000	дільник (додавання)
1|01110000	$a_2 < 0$, $C = 100_2$
← 0|11100000	$2a_2$
+ 0|11000000	дільник (додавання)
1|10100000	$a_3 < 0$, $C = 1000_2$
← 1|01000000	$2a_3$
+ 0|11000000	дільник (додавання)
0|00000000	$a_4 = 0$, $C = 10001_2$

Отже: $11001100_2 / 1100_2 = 10001_2$ ($204_{10} / 12_{10} = 17_{10}$)

2.6. Числа з фіксованою точкою

Одним із способів представлення дробів у двійковій системі числення є числа з фіксованою точкою (комою). Цей спосіб є аналогічним представленню неправильних дробів і передбачає двійкову точку між бітами цілої та дробової частини числа, аналогічно десятковій точці між цілою і дробовою частинами звичайного десяткового числа.

Як і у всіх попередніх випадках представлення двійкових чисел (знакових/беззнакових), числа з фіксованою точкою є лише набором біт. Тому не існує способу дізнатися де саме знаходиться двійкова точка, окрім як про це заздалегідь домовитись і використовувати цю домовленість при інтерпретації чисел.

Наприклад, ми можемо домовитись, що наші дробові числа матимуть **по чотири біти** у цілій та дробовій частинах, тоді число $11{,}75_{10}$ в двійковій системі матиме вигляд:

$1011{,}1100_2 = 2^3 + 2^1 + 2^0 + 2^{-1} + 2^{-2} = 8 + 2 + 1 + 0{,}5 + 0{,}25 = 11{,}75_{10}$

Без точки, просто 10111100_2.

Знакові числа з фіксованою точкою можуть бути представлені як в прямому, так і доповнювальному коді. При цьому в прямому коді, як і раніше, найстарший біт виступає у ролі знакового, а доповнювальний код можна отримати за допомогою інверсії бітів абсолютного значення числа і додавання 1 до наймолодшого розряду.

Приклад.

Домовимось, що для представлення цілої частини числа будемо використовувати 8 біт, а для дробової – 4 біти, то число $-27{,}375_{10}$ можна представити так:

$00|011011{,}0110_2$ ← абсолютне значення

$10011011{,}0110_2$ ← прямий код

$11100100{,}1010_2$ ← доповнювальний код

Арифметичні дії з числами з фіксованою точкою

Операції додавання та віднімання чисел з фіксованою точкою виконуються аналогічно звичайним цілим числам.

Приклади.

Приймемо у прикладах розрядність цілої і дробової частин рівною 4.

1) Додати числа $1{,}625_{10}$ та $3{,}5_{10}$:

$$0001{,}1010_2 = 1{,}625_{10}$$
$$0011{,}1000_2 = 3{,}5_{10}$$

$$\begin{array}{r} 00011010 \\ +\ 00111000 \\ \hline 01010010 \end{array}$$

Відповідь: $0101{,}0010_2 = 5{,}125_{10}$.

2) Обчислити $1{,}625_{10} - 3{,}5_{10}$:

Виконаємо дану дію через додавання, представивши від'ємник у доповнювальному коді:

$$0001{,}1010_2 = 1{,}625_{10}$$
$$1100{,}1000_2 = -3{,}5_{10}$$

$$\begin{array}{r} 00011010 \\ +\ 11001000 \\ \hline 11100010 \end{array}$$

Відповідь: $1110{,}0010_2 = -1{,}875_{10}$.

3) Обчислити $-0{,}75_{10} - 2{,}125_{10}$:

Представимо обидва числа в доповнювальному коді:

$$1111{,}0100_2 = -0{,}75_{10}$$
$$1101{,}1110_2 = -2{,}125_{10}$$

$$\begin{array}{r} 11110100 \\ +\ 11011110 \\ \hline 1\leftarrow 11010010 \end{array}$$

Відповідь: $1101{,}0010_2 = -2{,}875_{10}$.

При виконання множення чисел з фіксованою точкою слід пам'ятати, що добуток, матиме вдвічі довшу розрядну сітку в порівнянні з довжиною розрядних сіток множеного і множника. При цьому збільшиться розрядність як цілої, так і дробової частин.

Приклад.

Знайти добуток чисел $10{,}25_{10}$ та $4{,}75_{10}$.

Виконаємо операцію множення і в десятковій і в двійковій системах.

```
  10,25          1010,01
× 04,75        × 0100,11
  ,5125          10,1001
 7,175          101,001
41,00           0000,00
000,0          00000,0
48,6875        101001
               000000
               0110000,1011
```

Число $110000{,}1011_2 = 48{,}6875_{10}$.

Числа з рухомою (плаваючою) комою

Структура числа

Число з рухомою комою складається з:

- Знаку мантиси (вказує на від'ємність чи додатність числа)
- Мантиси (виражає значення числа без урахування порядку)
- Знаку порядку
- Порядку (виражає степінь основи числа, на яке множиться мантиса)

Нормальна форма та нормалізована форма

Нормальною формою числа з рухомою комою називається така форма, в якій мантиса (без урахування знаку) міститься на напівінтервалі [0; 1).

Така форма запису має **недолік**: деякі числа записуються неоднозначно (наприклад, 0,0001 можна записати в 4 формах — $0{,}0001\times10^0$, $0{,}001\times10^{-1}$, $0{,}01\times10^{-2}$, $0{,}1\times10^{-3}$), тому поширеною (особливо в інформатиці) є й інша форма запису — *нормалізована*,

нормалізована форма, в якій мантиса десяткового числа набуває значень в [1; 10), а мантиса двійкового числа набуває значень в[1; 2). У такій формі будь-яке число (крім 0) записується єдиним чином.

Недолік полягає в тому, що в такому вигляді неможливо отримати 0, тому представлення чисел в інформатиці передбачає **спеціальну ознаку** (біт) для числа 0.

Оскільки старший розряд (ціла частина числа) мантиси двійкового числа (крім 0) в *нормалізованому* вигляді дорівнює «1», то при записі мантиси числа в ЕОМ старший розряд можна не записувати, що й використовується в стандарті IEEE 754.
В позиційних системах числення з основою, більшою, ніж 2 (в трійковій, четвірковій та інших), цієї властивості немає.

Резюме

Існує кілька способів того, як рядки з цифр можуть подавати числа:

- Найпоширеніший спосіб подання значення числа рядком з цифрами — у вигляді цілого числа — кома за замовчуванням розташована наприкінці рядка.
- Загалом, у математичному поданні рядок з цифр може мати яку завгодно довжину, а положення коми позначається явним записом символу коми в потрібному місці.

- У системах з поданням **чисел у форматі з фіксованою комою** існує певна умова щодо положення коми. Наприклад, у рядку з 8 цифр умова може вказувати положення коми в середині запису (між 4-ю і 5-ю цифрами). Таким чином, рядок «00012345» позначає число 1,2345 (нулі ліворуч завжди можна відкинути).
- В експоненційному записі використовують стандартний (*нормалізований*) вид подання чисел. Число вважається записаним у стандартному (*нормалізованому*) вигляді, якщо воно записане у вигляді де $1 \le a < q <$, ціле, називається показником степеня та — ціле, основа системи числення (на письмі це зазвичай 10).
- Тобто у мантиси кома поміщається одразу після першої значущої (не рівної нулю) цифри, рахуючи зліва направо, а подальший запис дає інформацію про дійсне значення числа. Наприклад, період обертання (на орбіті) супутника Юпітера Іо, який дорівнює 152853,5047 с, у стандартному вигляді можна записати як $1{,}528535047 \times 10^5$ с.
- Побічним ефектом обмеження на значення мантиси є те, що в такому записі неможливо зобразити число 0.
- Запис із рухомою комою схожий на запис чисел у стандартному вигляді, але мантиса та експонента записуються окремо. Повертаючись до прикладу з **Іо**, запис у формі з рухомою комою буде 1528535047 з показником 5.
- Це означає, що записане число в 10^5 разів більше числа 1,528535047, тобто для отримання потрібного числа кома зсувається на 5 розрядів вправо.
- Запис у формі з рухомою комою використовується переважно в електронному поданні чисел, для якого використовується основа системи числення **2**, а не **10**.
- Крім того, в двійковій системі запису мантиса зазвичай *денормалізована*, тобто вважається, що **кома стоїть перед першою цифрою**, а не після, і ціла частина взагалі не розглядається — так з'являється можливість значення 0 зберегти природним чином. Таким чином, десяткова 9 в двійковому поданні з рухомою комою буде записана як мантиса +1001000…0 і показник +0…0100. Звідси, наприклад, проблеми з двійковим поданням чисел на зразок однієї десятої (0,1), для якої двійкове подання мантиси виявляється періодичним двійковим дробом — за аналогією з дробом 1/3, який неможливо записати з скінченною кількістю цифр у десятковій системі числення.

Запис числа у формі з рухомою комою дозволяє виконувати обчислення над широким діапазоном величин, поєднуючи фіксовану кількість розрядів та точність. Наприклад, у десятковій системі подання чисел з рухомою комою (3 розряди) операція множення, яку ми записуємо так:

$0{,}12 \times 0{,}12 = 0{,}0144$

у нормальній формі подається у вигляді: $(1{,}20 \times 10^{-1}) \times (1{,}20 \times 10^{-1}) = (1{,}44 \times 10^{-2})$. У форматі з фіксованою комою ми б отримали вимушене округлення

$0{,}120 \times 0{,}120 = 0{,}014.$

При цьому втрачено крайній правий розряд числа, оскільки цей формат не дозволяє комі «рухатися» по запису числа.