

ЛЕКЦІЯ 3

Двійкова арифметика та машинне представлення чисел

1. Фіксована розрядність машинної арифметики

Математичне ціле число не має наперед заданої кількості цифр, а машинний код зберігається у фіксованій кількості бітів. Розрядність визначає діапазон значень і те, які біти будуть втрачені, якщо точний результат не вміщується. У всіх прикладах потрібно спочатку зафіксувати розрядність і формат — unsigned або signed two's complement.

Для n бітів unsigned-діапазон становить $0 \dots 2^n - 1$, а signed two's complement — $-2^{n-1} \dots 2^{n-1} - 1$. Одна й та сама бітова операція виконується апаратурою однаково, але правильність результату оцінюється по-різному залежно від інтерпретації.

2. Двійкове додавання

Додавання виконують від LSB до MSB, тобто справа наліво. У розряді i додаються x_i , y_i та c_{in} (carry-in — перенесення на вході). Формуються s_i — біт суми та c_{out} (carry-out — перенесення на виході). Перенесення на виході поточного розряду стає перенесенням на вході наступного.

x_i	y_i	c_{in}	s_i	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Докладний приклад: $0101_2 + 0011_2$

Розряд i	x_i	y_i	c_{in}	s_i	c_{out}
0	1	1	0	0	1
1	0	1	1	0	1
2	1	0	1	0	1
3	0	0	1	1	0

Біти результату читаємо від старшого до молодшого: 1000_2 . У десятковій системі $0101_2=5$, $0011_2=3$, $1000_2=8$, тому перевірка $5+3=8$ виконується.

Докладний приклад: додавання з вихідним перенесенням

У 4 бітах $1111_2 + 0001_2 = 1\ 0000_2$. Усередині розрядної сітки залишається 0000_2 , а крайня ліва 1 є перенесенням за межі чотирьох бітів. Для unsigned-чисел це означає, що точний результат 16 не вмістився в діапазон $0 \dots 15$.

3. Двійкове віднімання з позикою

Віднімання також виконують справа наліво. Якщо з поточного біта неможливо відняти біт від'ємника та вхідну позику, позичають 1 зі старшого розряду. У двійковій системі позичена одиниця має вагу 2, тому 0 після позики тимчасово розглядають як 10_2 .

x_i	y_i	b_{in}	Різниця d_i	b_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Докладний приклад: $1101_2 - 0110_2$

- Розряд 0: $1 - 0 = 1$, позики немає.
- Розряд 1: $0 - 1$ неможливо; позичаємо 1. Маємо $10_2 - 1 = 1$, позика до розряду 2 дорівнює 1.
- Розряд 2: $1 - 1 - 1 = -1$; знову позичаємо. Результат у розряді 1, позика до розряду 3 дорівнює 1.
- Розряд 3: $1 - 0 - 1 = 0$.
- Отримано 0111_2 . Перевірка: $1101_2 = 13$, $0110_2 = 6$, $13 - 6 = 7$.

4. Віднімання через додатковий код

У two's complement віднімання $A - B$ замінюють додаванням $A + (-B)$. Щоб отримати $-B$ у фіксованій розрядності, інвертують усі біти B і додають 1. Кінцеве перенесення поза розрядною сіткою відкидають. Цей метод дає змогу використовувати той самий двійковий суматор для додавання і віднімання.

Докладний приклад: $+13 + (-5)$ у 8 бітах

- $+13 = 00001101_2$.
- $+5 = 00000101_2$.
- Інверсія коду $+5$: 11111010_2 .
- Додавання 1: $-5 = 11111011_2$.
- Додавання: $00001101_2 + 11111011_2 = 1\ 00001000_2$.
- Кінцеве перенесення відкидаємо: $00001000_2 = 8$. Перевірка: $13 + (-5) = 8$.

Докладний приклад: 29–45 у 8 бітах

$29=00011101_2$, $45=00101101_2$. Для -45 : інверсія 11010010_2 , після додавання 1 — 11010011_2 . Сума $00011101_2+11010011_2=11110000_2$. MSB=1, тому результат від'ємний. Декодування: $11110000_2 \rightarrow$ інверсія $00001111_2 \rightarrow +1 = 00010000_2=16$. Отже, результат -16 , що збігається з $29-45$.

5. Прапорці CF та OF

CF (Carry Flag — прапорець перенесення) зберігає перенесення за межі MSB. Під час unsigned-додавання CF=1 означає, що точний результат перевищив максимальне значення. OF (Overflow Flag — прапорець знакового переповнення) показує, що результат signed-операції не належить знаковому діапазону.

Для додавання у two's complement OF=1, якщо операнди мають однаковий знак, а результат — протилежний. Еквівалентне апаратне правило: OF дорівнює XOR між перенесенням на вхід MSB та перенесенням на вихід MSB. XOR (exclusive OR — виключне АБО) дорівнює 1, коли два біти різні.

4-бітова операція	Unsigned-тлумачення	Signed-тлумачення	CF	OF
$0111+0001=1000$	$7+1=8$ — коректно	$+7+1$ дало -8 — помилка	0	1
$1111+0001=0000$	$15+1$ не вмістилося	$-1+1=0$ — коректно	1	0
$1000+1111=0111$	$8+15$ не вмістилося	$-8+(-1)$ дало $+7$ — помилка	1	1
$0100+0010=0110$	$4+2=6$ — коректно	$+4+2=+6$ — коректно	0	0

Докладний приклад: знакове переповнення без CF

У 4 бітах 0111_2 представляє $+7$, а 0001_2 — $+1$. Сума бітів дорівнює 1000_2 . Кінцевого перенесення за межі біта 3 немає, тому CF=0. Але 1000_2 у four-bit two's complement означає -8 . Два додатні числа дали від'ємний результат, тому OF=1. Математичний результат $+8$ не входить до signed-діапазону $-8\dots+7$.

6. Логічні та арифметичні зсуви

Скорочення	Повна назва	Правило
LSL	Logical Shift Left — логічний зсув ліворуч	Зліва біти втрачаються, справа входять нулі.
LSR	Logical Shift Right — логічний зсув праворуч	Справа біти втрачаються, зліва входять нулі.
ASR	Arithmetic Shift Right — арифметичний зсув праворуч	Справа біти втрачаються, зліва копіюється знаковий біт.

За відсутності переповнення LSL на k позицій відповідає множенню беззнакового або signed-числа на 2^k . LSR для невід'ємного беззнакового числа відповідає цілочисельному діленню на 2^k . ASR зберігає знак two's complement-числа й наближено відповідає діленню від'ємного числа на степінь двійки з округленням у бік мінус нескінченності.

Докладний приклад: LSL

00110100_2 LSL 2: після першого зсуву 01101000_2 , після другого — 11010000_2 . Два старші біти, що вийшли за межі, втрачені. Початкове unsigned-значення 52, результат $208=52\cdot 4$; у цьому прикладі точний результат умістився.

Докладний приклад: LSR і ASR для одного коду

Для 11110000_2 LSR 2 отримуємо $00111100_2=60$ unsigned, оскільки зліва входять нулі. Для 11110000_2 ASR 2 отримуємо 11111100_2 . У 8-bit two's complement початковий код означає -16 , а результат — -4 . Копіювання знакової одиниці зберегло від'ємний знак.

7. Формат із фіксованою крапкою

Формат із фіксованою крапкою (fixed-point format) зберігає ціле число I , але інтерпретує його з наперед заданою кількістю дробових бітів f . Реальне значення дорівнює $I/2^f$. У позначенні $Qm.f$ число m зазвичай вказує кількість бітів цілої частини, а f — дробової; конкретну домовленість щодо біта знака потрібно зазначати окремо.

У беззнаковому $Q4.4$ 8 бітів поділено на 4 біти цілої та 4 біти дробової частини. Крок між сусідніми числами дорівнює $2^{-4}=1/16=0,0625$. Діапазон — від 0 до 15,9375.

Докладний приклад: кодування 3,25 у $Q4.4$

1. Множимо число на $2^4=16$: $3,25 \cdot 16=52$.
2. Переводимо 52 у двійкову систему: $52_{10}=00110100_2$.
3. Умовно ставимо крапку перед чотирма молодшими бітами: 0011.0100_2 .
4. Ціла частина $0011_2=3$; дробова $0100_2=0 \cdot 1/2 + 1 \cdot 1/4 + 0 \cdot 1/8 + 0 \cdot 1/16=0,25$.
5. Перевірка: $52/16=3,25$.

Докладний приклад: число, яке потребує округлення

Для 2,1 у $Q4.4$ маємо $2,1 \cdot 16=33,6$, але зберігати потрібно ціле I . За округленням до найближчого $I=34$, код 00100010_2 . Відновлене значення $34/16=2,125$. Похибка становить $2,125-2,1=0,025$. Правило округлення має бути визначене заздалегідь.

8. Подання з рухомою крапкою

Формат із рухомою крапкою (floating-point format) подає число в науковій формі: знак, значуща частина та порядок. У двійковій системі нормалізоване ненульове число записують як $\pm 1, f_2 \cdot 2^e$, де $1, f_2$ — значуща частина, а e — реальний порядок. Рухома крапка забезпечує великий діапазон значень, але більшість дробів подається наближено.

IEEE 754 — стандарт Інституту інженерів з електротехніки та електроніки (Institute of Electrical and Electronics Engineers), який визначає поширені формати й правила операцій із рухомою крапкою. Формат binary32, або single precision — одинарна точність, містить 32 біти.

Поле	Назва	Біти	Зміст
S	Sign — знак	1	0 для додатного, 1 для від'ємного числа
E	Exponent — зміщений порядок	8	Для нормалізованого числа $E=e+127$
F	Fraction — дробова частина	23	Біти після провідної одиниці значущої частини

Для нормалізованих значень формула має вигляд $(-1)^s \cdot (1+F) \cdot 2^{E-127}$, де F розуміють як двійковий дріб. Провідна одиниця перед крапкою не зберігається в полі F , оскільки для нормалізованого ненульового числа вона завжди дорівнює 1.

Докладний приклад: 5,75 у IEEE 754 binary32

1. Знак додатний, тому $S=0$.
2. Перетворення: $5_{10}=101_2$; $0,75_{10}=0,11_2$; отже, $5,75_{10}=101,11_2$.

3. Нормалізація: $101,11_2 = 1,0111_2 \cdot 2^2$. Реальний порядок $e=2$.
4. Зміщений порядок: $E=e+127=129=10000001_2$.
5. Провідну 1 не записуємо. Поле F починається з 0111 і доповнюється нулями:
011100000000000000000000.
6. Повний код: 0 | 10000001 | 011100000000000000000000.

Докладний приклад: декодування простого binary32

Нехай $S=1$, $E=10000000_2=128$, $F=010000000000000000000000$. Знак від'ємний. Реальний порядок $e=128-127=1$. Значуща частина $1,01_2=1+1/4=1,25$. Значення: $-1,25 \cdot 2^1 = -2,5$.

9. Спеціальні значення IEEE 754

Комбінації $E=00000000_2$ та $E=11111111_2$ мають спеціальне призначення. Нульове поле E використовується для нуля й субнормальних чисел. Поле E з усіма одиницями використовується для нескінченності та NaN (Not a Number — не число). NaN виникає, наприклад, у результаті операції, яка не має числового результату у звичайній дійсній арифметиці.

E	F	Значення
00000000	усі 0	+0 або -0 залежно від S
00000000	не всі 0	Субнормальне число
00000001...11111110	будь-яке	Скінченне нормалізоване число
11111111	усі 0	$+\infty$ або $-\infty$
11111111	не всі 0	NaN — не число

10. Терміни та скорочення

Термін	Пояснення
CF	Carry Flag — прапорець перенесення для контролю unsigned-операцій.
OF	Overflow Flag — прапорець знакового переповнення.
Carry-in / carry-out	Перенесення на вході / на виході розряду.
Borrow	Позика зі старшого розряду під час віднімання.
Q4.4	Формат із фіксованою крапкою: 4 цілі та 4 дробові біти.
binary32	32-бітовий формат IEEE 754 одинарної точності.
Нормалізація	Перенесення двійкової крапки до форми $1, f_2 \cdot 2^e$.