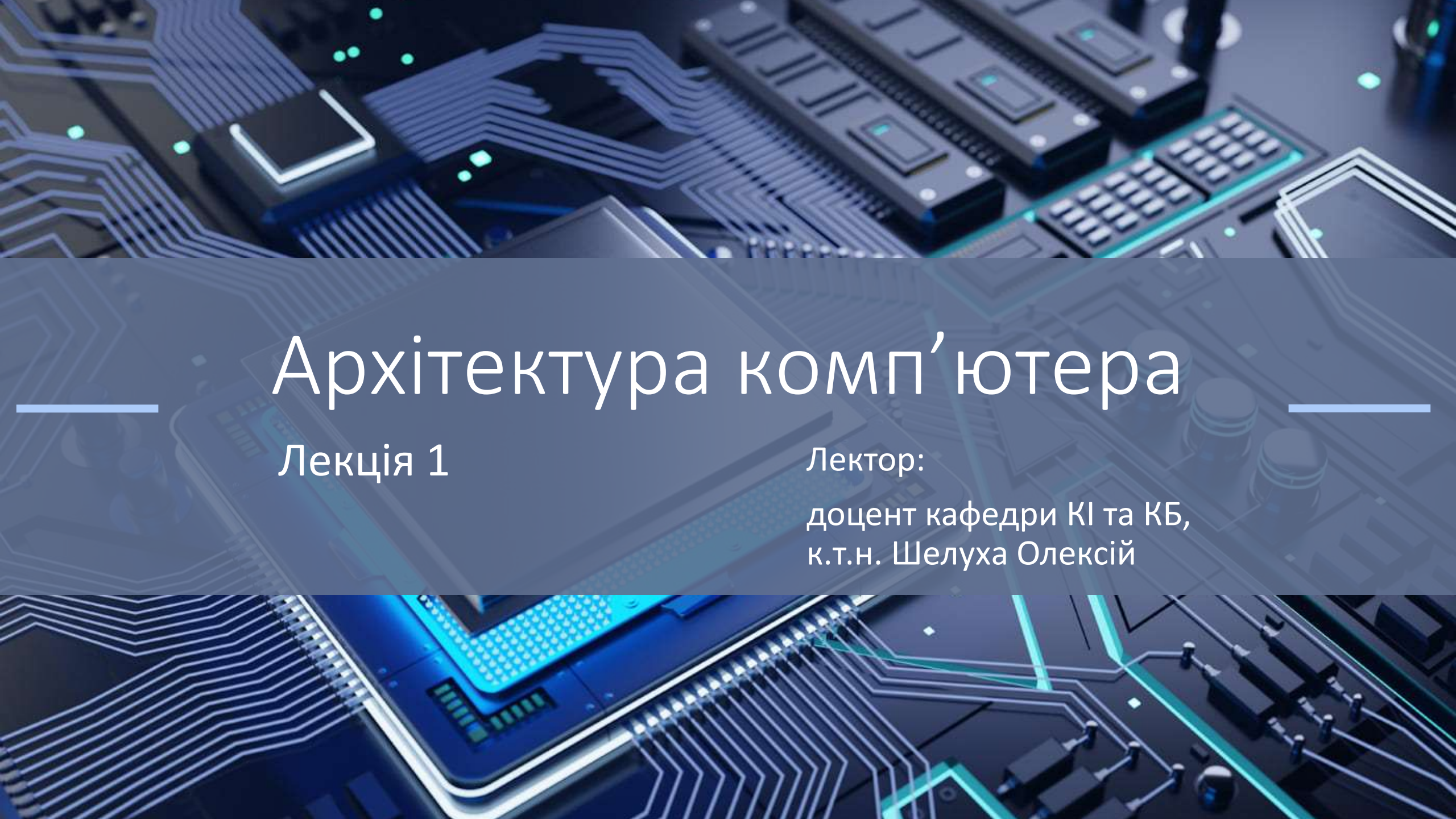




Архітектура комп'ютера

Лекція 1

Лектор:

доцент кафедри КІ та КБ,
к.т.н. Шелуха Олексій

Зміст лекції

- Представлення структури дисципліни
- Переведення дійсних чисел у двійкову систему числення
- Нормалізація чисел
- Загальне подання дійсних чисел в пам'яті комп'ютера згідно стандарту IEEE 754-2008 (ISO/IEC/IEEE 60559:2011)

Структура дисципліни

Системи
числення та
представлення
даних у пам'яті
комп'ютера



Структура
комп'ютера.
Архітектура
x86/IA-32 та
x86-64



Основи мови
Assembler.
Команди
обміну даними



Програмування
на мові
Assembler.
Використання
математичного
співпроцесора



Переведення дійсних чисел у двійкову форму системи числення

- Робота з прикладом дійсного числа: $12,125_{10}$
- Переводимо **цілу частину** в двійкову систему числення шляхом **ділення на основу** «2», доки не отримаємо число менше основи:
 $12 : 2 = 6$, остача 0
 $6 : 2 = 3$, остача 0
 $3 : 2 = 1$, остача 1
 $1 < 2$ залишаємо 1
- Відповідно записуємо результат з **кінця** $12_{10} = 1100_2$

Переведення дійсних чисел у двійкову форму системи числення

- Робота з прикладом дійсного числа: $12,125_{10}$
- Переводимо **дробну частину** в двійкову систему числення шляхом **множення на основу** «2» та виділення цілої частини добутку:
 $0,125 * 2 = 0,25$, ціла частина 0
 $0,25 * 2 = 0,5$, ціла частина 0
 $0,5 * 1 = 1$, ціла частина 1
- Відповідно записуємо результат по порядку $0,125_{10} = 0,001_2$

Переведення дійсних чисел у двійкову форму системи числення

- Робота з прикладом дійсного числа: $12,125_{10}$
- Об'єднуємо результати операцій переведення в двійкову систему числення і отримуємо $12,125_{10} = 1100,001_2$

- Відповідно ми можемо перевірити результат зворотнім переведенням в десяткову систему:

$\begin{matrix} 3 & 2 & 1 & 0, & -1 & -2 & -3 \\ 1 & 1 & 0 & 0, & 0 & 0 & 1 \end{matrix}$

$1100,001_2$

$$1100_2 = 0 * 2^0 + 0 * 2^1 + 1 * 2^2 + 1 * 2^3 = 0 + 0 + 4 + 8 = 12_{10}$$

$$,001_2 = 0 * 2^{-1} + 0 * 2^{-2} + 0 * 2^{-3} = 0 + 0 + 1 * 1/2^3 = 1/8 = 0,125_{10}$$

- Існує проблема, оскільки не кожне дробне число можна перевести в двійкову систему числення:
 $1/3 = 0,333333\dots$ або $0,(3)$ – це три в періоді.
- Спробуємо перевести спрощений варіант $0,3_{10}$ в двійкову систему X_2 :
 $0,3 * 2 = 0,6$ ціла частина 0
 $0,6 * 2 = 1,2$ ціла частина 1
 $0,2 * 2 = 0,4$ ціла частина 0
 $0,4 * 2 = 0,8$ ціла частина 0
 $0,8 * 2 = 1,6$ ціла частина 1
 $0,6 * 2 = 1,2$ ціла частина 1
 $0,2 * 2 = 0,4$ ціла частина 0
 $0,4 * 2 = 0,8$ ціла частина 0
 $0,8 * 2 = 1,6$ ціла частина 1
 $0,6 * 2 = 1,2$ ціла частина 1
- Бачимо, що $X_2 = 0,010011001(1001)$, тобто $0,3_{10} = 0,0(1001)_2$, де $(1001)_2$ в періоді, і не може бути представлене як скінчене число при обмежених ресурсах пам'яті комп'ютера.
- Відповідно представлення дійсних чисел у двійковій системі числення вимагають певних обмежень до точності запису цього числа.

Нормалізація десяткових чисел

- Маса $M = 2357 \text{ г} = 2,357 * 10^3 \text{ г}$
- Час $t = 0,000153 \text{ с} = 1,53 * 10^{-4} \text{ с}$
- Відстань $385000000 \text{ м} = 3,85 * 10^8 \text{ м}$
- Нормалізовані числа формуються так, щоб в цілому розряді було одне число і інші вже після коми із вказуванням на скільки розрядів вліво чи вправо було зміщено кому.

Нормалізація двійкового числа

- $\pm 1, m_2 * 2^p$
де $\pm$ - знак числа (обов'язково)
 m_2 – двійкова мантиса числа,
 2^p – порядок числа.
- Наприклад:
 $1100,001_2 = 1,100001_2 * 2^3$ (2^3 – в десятковій системі)
 $0,000001011_2 = 1,011_2 * 2^{-6}$ (2^{-6} – в десятковій системі)
- Виключення:
 $0,0_2$ – не нормалізується, оскільки немає 1.

Стандарт IEEE 754-2008 (ISO/IEC/IEEE 60559:2011)

- Дійсні числа – завжди знакові!
- Знак S: 0 – плюс, 1 – мінус.
- Характеристика = Зміщення + порядок
- Згідно стандарту є наступні представлення:
 - 32-бітні (float), зміщення = $7F_{16}$, 127_{10}
 - 64-бітні (double), зміщення = $7FF_{16}$, 2047_{10}
 - 80-бітні (long double), зміщення = $3FFF_{16}$, $16\,383_{10}$

1 – прихований розряд (не у всіх)

S	Характеристика			Нормалізована мантиса				
N-1	N-2					...	1	0
Біти								

Подання (формати) дійсних чисел

- Подання дійсного 32-бітного числа

1 – прихований розряд

S	Характеристика			Нормалізована мантика				
31	30	...	23	22	21	...	1	0
Біти								

- Знак – 1 розряд
- Характеристика – 8 розрядів, зміщення = $7F_{16}$, 127_{10}
- Нормалізована мантика – 23 розряди

Подання (формати) дійсних чисел

- Подання дійсного 64-бітного числа

1 – прихований розряд

S	Характеристика			Нормалізована мантика				
63	62	...	52	51	50	...	1	0
Біти								

- Знак – 1 розряд
- Характеристика – 11 розрядів, зміщення = $7FF_{16}$, 2047_{10}
- Нормалізована мантика – 52 розряди

Подання (формати) дійсних чисел

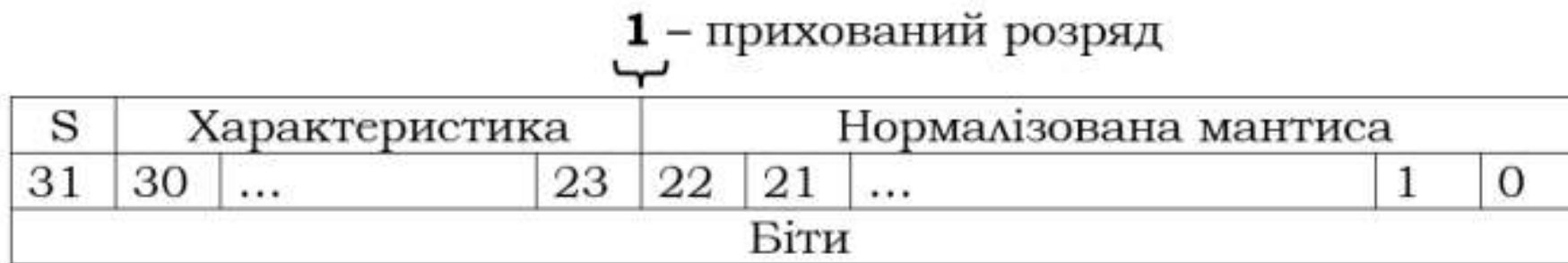
- Подання дійсного 80-бітного числа

S	Характеристика			Нормалізована мантика				
79	78	...	64	63	62	...	1	0
Біти								

- Знак – 1 розряд
- Характеристика – 15 розрядів, зміщення = $3FFF_{16}$, $16\,383_{10}$
- Нормалізована мантика – 64 розряди

Іншими словами про 32-бітне подання даних

- Значення дійсного числа (або числа з плаваючою комою) – це 32-бітний контейнер, що призначений для зберігання та виконання операцій над апроксимаціями (наближеннями) дійсних чисел.



- S (1 біт) – зберігання знаку
- E (8 біт) – характеристика (або експонента)
- M (23 біти) – мантиса (апроксимоване значення частини числа після коми)

$$(-1)^S * 1, M * 2^{E-127}$$

Подання (формати) дійсних чисел

- Розглянемо приклад подання дійсного 32-бітного числа **-1.0** в BIN та HEX-коді

1 – прихований розряд

S	Характеристика								Нормалізована мантиса						
31	30	29	28	27	26	25	24	23	22	21	20	19		1	0
Біти															
1	0	1	1	1	1	1	1	1	0	0	0	0		0	0
HEX-код															
B				F				8				0		0 0 0 0	

Приклад: $12,125_{10}$ записати (представити) в 32-бітному форматі (поданні).

- Переведення: $12,125_{10} = 1100,001_2$
- Нормалізація: $1100,001_2 = 1,100001_2 * 2^3$ звідки отримуємо
Порядок: $3_{10} = 3_{16}$
Мантиса: 100001_2
- Розрахунок характеристики:
Характеристика = Зміщення + Порядок $7F_{16} + 3_{16} = 82_{16} = 10000010_2$
- Формуємо представлення числа:
Знак (плюс) = 0
Характеристика = 10000010_2
Мантиса = 100001_2

Приклад: $12,125_{10}$ записати (представити) в 32-бітному форматі (поданні).

- Формуємо представлення числа:

Знак (плюс) = 0

Характеристика = 10000010_2

Мантиса = 100001_2

Знак		Характеристика							Мантиса																											
0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
4				1				4				2				0				0				0				0 ₁₆								

- Якщо не вистачає бітів для формування повного числа – проводиться операція доповнення нулями до правої частини мантиса.
- В пам'яті комп'ютера число $12,125_{10}$ буде подано, як 41420000_{16}

Приклад: $12,125_{10}$ записати (представити) в 32-бітному форматі (поданні).

- Використаємо форму представлення числа:

$$(-1)^S * 1, M * 2^{E-127}$$

- Де, число = $41420000_{16} = 0100\ 0001\ 0100\ 0010\ 0000\ 0000\ 0000\ 0000$

Знак (плюс) = 0,

Мантиса = 100001_2 ,

Характеристика = $10000010_2 = 130_{10}$

- Отримуємо $(-1)^0 * 1,100001_2 * 2^{130-127} =$
 $= 1 * 1,100001_2 * 2^3 = 1100,001_2 = 12,125_{10}$

Парадокси (проблеми)

- Проблема 1. Обмеження точності представлення:
 $0,1 + 0,2 \neq 0,3$
або $0.1 + 0.2 = 0.30000001192092895507812$
- Проблема 2. Переповнення мантиси
Для кожного другого цілого числа порядку 2^{24} , або більше 16777216 в 32-бітному представлені не дістанеться значення:
так $16777215 \neq 16777216$,
але $16777216 = 16777217$

Дякую за увагу!