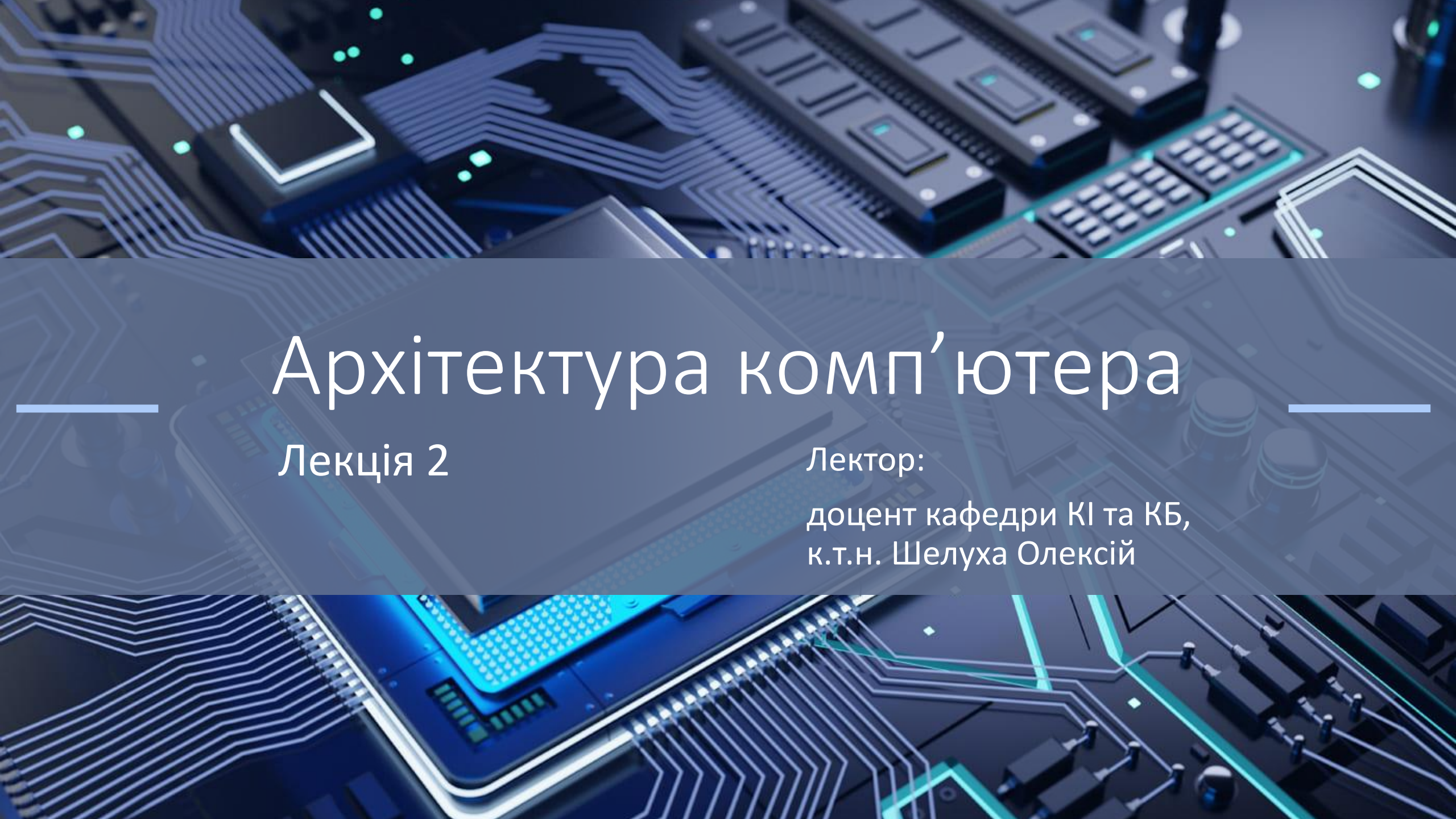




Архітектура комп'ютера

Лекція 2

Лектор:

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

Зміст лекції

- Типи даних
- Представлення цілочисельних типів даних в асемблері
- Псевдокоманди (директиви) DB, DW, DD, DQ, DT
- Внутрішнє представлення цілочисельних даних

Типи даних

- **Тип даних** – характеристика, яку явно чи неявно надано об'єкту (змінній, функції, полю запису, константі, масиву тощо).
- Тип даних визначає:
 - які види даних можуть бути використані,
 - множину їх припустимих значень,
 - формат їхнього збереження,
 - розмір виділеної пам'яті,
 - набір операцій, які можна робити над даними,
 - обмеження, які діють на них.
- Тип даних вказує, як інтерпретувати певну послідовність бітів або байтів як конкретну інформацію.

Машинні типи даних:

- Цілочисельні типи даних – **Integers**
- Тип даних з плаваючою комою – **Floating-Point Numbers**
- Логічний тип даних – **Booleans**
- Символьний тип даних – **Characters**
- Байтові типи даних – **Bytes**
- Перелічувальний тип даних – **Enumerated**
- Складні типи даних

Цілочисельні типи даних

- Цілочисельні типи даних використовуються для зберігання цілих чисел, тобто чисел без десяткової частини. Основні цілочисельні типи даних:
 - int
 - short int (short)
 - long int (long)
 - long long int (long long)
 - unsigned int
- Приклади цілих чисел:
 - 42
 - 10000
 - -233000
 - -100

Тип даних з плаваючою комою

- Типи даних з плаваючою комою використовуються для зберігання чисел, які можуть мати дробову частину. Основні типи даних з плаваючою комою:
 - float:
 - double
 - long double
- Приклади дійсних чисел:
 - 20.0005
 - 99.9
 - -5000.12
 - -9999.9991

Логічний тип даних

- Використовується для зберігання значень True або False, які представляють логічну правду або неправду. Основні типи:
 - bool (boolean)
 - bit
- Логічні типи даних є важливим елементом умовних конструкцій (наприклад, умовних операторів if-else), циклах, логічних операціях (наприклад, логічні "І", "АБО" і "НЕ") та інших аспектах програмування.

Символьний тип даних

- Символьні типи даних використовуються для зберігання символів, таких як літери, цифри, розділові знаки та інші видимі та недруковані символи.
Основні символьні типи даних:
 - `char`
 - `wchar_t`
 - `char16_t` та `char32_t`
 - `string`
 - `char[]` або `char*`
- Символьні типи даних дозволяють обробляти текстову інформацію, виконувати операції порівняння символів, шукати та замінювати символи у рядках і виконувати інші операції, пов'язані з роботою з текстом.

Байтові типи даних

- Байтові типи даних в програмуванні використовуються для зберігання і обробки даних на рівні байтів, які є основними одиницями інформації в комп'ютерних системах. Основні байтові типи:
 - `byte`
 - `sbyte`
 - `byte[]`
- Байтові типи даних зазвичай використовуються для оптимізації роботи з пам'яттю та для низькорівневого програмування, такого як робота з мережевими пакетами, зберігання бінарних даних і робота з пристроями вводу/виводу.

Перелічувальний тип даних

- Перелічувальний тип даних, або перелік - тип даних, що описується шляхом перелічення всіх можливих значень (кожне з яких позначається власним ідентифікатором), які можуть приймати об'єкти даного типу.
- Імена нумераторів зазвичай є ідентифікаторами, які поводяться як константи в мові. Змінній, яка була оголошена як така, що має перерахований тип, можна призначити будь-який з перелічувачів як значення.
- Використання перерахувань дозволяє зробити вихідні коди програм більш читаними, оскільки дозволяють замінити «магічні числа», що кодують певні значення, на імена, що читаються.

Складні типи даних

- Складні типи даних – це типи, які складаються з елементів, що відносяться до простих типів.
- Складні типи даних (або структури даних) використовуються для групування різних типів даних у один об'єкт, який може представляти складну структуру або об'єкт з багатьма атрибутами. Це дозволяє легше організовувати та обробляти дані, забезпечує модульність і полегшує взаємодію між об'єктами
- Складні типи даних є ключовими для роботи зі складними і структурованими даними в програмах. Вони дозволяють впорядковувати дані та робити їх більш легкодоступними та керованими.

Представлення цілочисельних даних в асемблері

- **Асемблер** — це низькорівнева мова програмування, яка взаємодіє безпосередньо з апаратними ресурсами комп'ютера. У багатьох архітектурах асемблера існують різні типи даних та робочі регістри, які використовуються для виконання операцій та зберігання даних.
- Основними типами даних для асемблера є наступні:
 - біт або bit (1 біт);
 - байт або byte (8 біт);
 - слово або word (16 біт);
 - подвійне слово або dword (32 біти);
 - зчетверене слово або qword (64 біти);
 - 10 байт ten byte (80 біт) і т.д.

Псевдокоманди (директиви) DB, DW, DD, DQ, DT

- Для розміщення певного значення в одному байті пам'яті асемблер має в своєму розпорядженні спеціальний механізм визначення байта (define byte) – псевдокоманду (директиву) DB формату.
- Отже, псевдокоманда DB:
 - формує одnobайтову константу - число;
 - формує у пам'яті заданий рядок символів (літерал);
 - просто резервує 1 байт пам'яті (за умови наявності).
- Псевдокоманди (директиви) визначають відповідно:
 - Слово визначається DW (define word);
 - Подвійне слово визначається DD (define doubleword);
 - Зчетверене слово визначається DQ (define quadword);
 - Десятибайтне слово визначається DT (define tenword).

Основні типи даних в Асемблері

Тип	Директива	Кількість байт
Байт (byte)	DB	1
Слово (word)	DW	2
Подвійне слово (dword)	DD	4
Зчетверене слово (qword)	DQ	8
10 байт (tbyte)	DT	10

Алгоритм внутрішнього представлення цілочисельних даних

- Внутрішнє представлення цілочисельних даних відбувається в два кроки:
- Спочатку задане число переводиться з десяткової системи числення в двійкову.
 - Для того, щоб змінити знак числа, потрібно інвертувати всі його біти і додати до нього одиницю – отримаємо представлення ВІД'ЄМНОГО числа в додатковому коді. Для цілих типів із знаком, під знак відводиться старший біт, причому для додатніх чисел він дорівнює 0, а для від'ємних – 1.
- Потім отриману двійкову послідовність необхідно перевести в шістнадцяткову систему числення.

Приклад, числа ± 23 та ± 1239

- $23:2 = 11$ остача 1
 $11:2 = 5$ остача 1
 $5:2 = 2$ остача 1
 $2:2 = 1$ остача 0
- $23_{10} =$ $0001\ 0111_{2}$ (byte)
- $-23_{10} =$ 1) $0001\ 0111_{2}$ - двійковий код числа $|-23|$
 2) $1110\ 1000_{2}$ - інвертуємо
 3) $\underline{\quad\quad\quad 1\quad\quad}$ - додаємо одиничку
 $1110\ 1001_{2}$ - додатковий код

Приклад, числа ± 23 та ± 1248

- $1239_{10} = 0000\ 0100\ 1110\ 0000_{10}$ (word)
- $-1239_{10} =$
 - 1) $0000\ 0100\ 1110\ 0000_{10}$ – двійковий код числа $|-1239|$
 - 2) $1111\ 1011\ 0001\ 1111_{10}$ – інверсія
 - 3) $+ \underline{\hspace{1.5cm}1\hspace{1.5cm}}$
 $1111\ 1011\ 0010\ 0000_{10}$ – додатковий код

Dec	Byte		Word	
	Bin	Hex	Bin	Hex
23	0001 0111	17	0000 0000 0001 0111	0017
-23	1110 1001	E9	1111 1111 1110 1001	FFE9
1248			0000 0100 1110 0000	04E0
-1248			1111 1011 0010 0000	FB20

Дякую за увагу!